

## Research Article

# A Decision-Making Method for Ship Collision Avoidance Based on Improved Cultural Particle Swarm

Yisong Zheng , Xiuguo Zhang , Zijing Shang , Siyu Guo , and Yiquan Du 

*School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China*

Correspondence should be addressed to Xiuguo Zhang; zhangxg@dlmu.edu.cn

Received 21 September 2020; Revised 30 November 2020; Accepted 15 December 2020; Published 15 January 2021

Academic Editor: Petr Dolezel

Copyright © 2021 Yisong Zheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the process of ship collision avoidance decision making, steering collision avoidance is the most frequently adopted collision avoidance method. In order to obtain an effective and reasonable steering angle, this paper proposes a decision-making method for ship collision avoidance based on improved cultural particle swarm. Firstly, the ship steering angle direction is to be determined. In this stage, the Kalman filter is used to predict the ship's trajectory. According to the prediction parameters, the collision risk index of the ship is calculated and the situation with the most dangerous ship is judged. Then, the steering angle direction of the ship is determined by considering the Convention on the International Regulations for Preventing Collisions at Sea (COLREGs). Secondly, the ship steering angle is to be calculated. In this stage, the cultural particle swarm optimization algorithm is improved by introducing the index of population premature convergence degree to adaptively adjust the inertia weight of the cultural particle swarm so as to avoid the algorithm falling into premature convergence state. The improved cultural particle swarm optimization algorithm is used to find the optimal steering angle within the range of the steering angle direction. Compared with other evolutionary algorithms, the improved cultural particle swarm optimization algorithm has better global convergence. The convergence speed and stability are also significantly improved. Thirdly, the ship steering angle direction decision method in the first stage and the ship steering angle decision method in the second stage are integrated into the electronic chart platform to verify the effectiveness of the decision-making method of ship collision avoidance presented in this paper. Results show that the proposed approach can automatically realize collision avoidance from all other ships and it has an important practical application value.

## 1. Introduction

The increasing density of ship traffic flow leads to the frequent occurrence of ship collision accidents. More than 90% of ship collision accidents are related to human factors, of which at least 60% are directly caused by human errors [1]. At present, a large number of advanced auxiliary systems such as Global Positioning System (GPS), Automatic Radar Mapping Assistant (ARPA), Automatic Identification System (AIS), and Electronic Chart Display and Information System (ECDIS) have brought great help to the acquisition of ships during navigation [2]. However, the progress of information acquisition has not reduced the collision probability of ships to the ideal level [3]. On the other hand, due to the increasingly complex collision avoidance environment faced by ships, it is difficult to describe all possible

conditions with rules, which requires ships to have the ability to autonomously analyze the surrounding environment so as to make accurate collision avoidance decisions.

Autonomous ship navigation can be divided into two main research areas: collision avoidance and track keeping [4]. Among them, collision avoidance is the core of autonomous ship navigation. The realization of ship collision avoidance first needs to divide the encounter situation of the target ship on the basis of the known environment, quantify the collision risk of the ship, determine the key avoiding ships, then establish the mathematical model of steering, and finally calculate the avoidance decision of own ship in the situation [5]. However, apart from the general rules outlined in the International Regulations for Preventing Collisions at Sea (COLREGs) and the traditional operating specifications, there are no special rules to guide the ship's navigation

system for avoidance manipulation in situations involving multiple ships, and there is no guidance. When multiple ships meet, the collision avoidance decision to avoid collision depends on many factors, such as ship speed, heading, relative position, and maneuverability of the ship [6]. Changing the course of a ship can achieve better collision avoidance than changing the speed. At the same time, it is easier for the surrounding ships to observe the change of the ship's course on vision and radar [7]. Therefore, finding the optimal steering angle of a ship is currently the most frequently adopted collision avoidance decision-making method.

Finding the optimal steering angle is an objective optimization problem. At present, objective optimization problems are divided into linear optimization and nonlinear optimization. Linear optimization problems include the simplex method, interior point method, and polynomial method. The Newton method and conjugate gradient method are commonly used in nonlinear optimization problems. In essence, these optimization problems can be summarized into two steps: the first step is to build a mathematical model based on the actual problem according to its description; The second step is to use a certain method to find the optimal solution within its limited range [8]. The optimal solution of a single-objective optimization problem is generally a solution that can be clearly defined, and the optimal solution of a multiobjective optimization problem is generally an optimal solution set containing multiple optimal solutions [9]. In the objective optimization problem, there are currently two optimization algorithms: deterministic and random algorithms. Deterministic algorithms are mainly based on gradient search methods [10]. The shortcomings of this method are as follows: on the one hand, it has high requirements for the initial value points and gradient information, and on the other hand, it is difficult to optimize the nonderivable and feasible region disconnection. Random algorithms mainly include simulated annealing algorithm (SA) [11], tabu search algorithm (TS) [12], and evolutionary algorithms [13]. The evolutionary algorithm is more robust than SA and TS [14] and also has strong search efficiency; compared with deterministic algorithms, evolutionary algorithms have fewer constraints and are easier to implement when solving constrained optimization problems. With the successful application of evolutionary algorithms in many fields, people's research efforts have also been strengthened, which has also promoted the rapid development of evolutionary algorithms.

Particle swarm optimization (PSO) was proposed by Kennedy and Eberhart in 1995 based on the predatory behavior of birds [15], which belongs to an evolutionary algorithm. This algorithm has the following advantages: simple calculation and good robustness. However, there are still some weaknesses in solving the problem of objective optimization, such as easy to fall into local optimum, poor global convergence, and so on. In order to overcome the defects and improve the local optimization ability of PSO algorithm, scholars at home and abroad have proposed many improvement methods for the inertia weight of particles. For example, in [16], a nonlinear decreasing

algorithm of inertia weight is proposed. Based on the basic idea of linearly decreasing inertial weight values, three nonlinear decreasing strategies are proposed. In [17], the problem of adaptive change of inertial weight is studied. In [18], the inertia weight change according to the exponential law is proposed. In [19], stochastic inertia weighting strategies are proposed. Compared with the standard PSO algorithm, these methods can improve the convergence speed and optimization accuracy.

However, in these previous improvement studies, the linearly decreasing inertial weight method is usually used, which cannot adapt to complex nonlinear optimization problems, such as finding the optimal steering angle of a ship. Besides, only the optimization and improvement of inertial weights are improved. The algorithm still has the possibility of falling into a local optimum.

In view of the above problems, this paper proposes an improved cultural particle swarm optimization algorithm, which can dynamically and nonlinearly change the inertia weights, and solves the problem of falling into a local optimum. The cultural particle swarm optimization algorithm (CPSO) [20, 21] integrates the PSO algorithm into the framework of cultural algorithms and composes the main group space (lower space) and the knowledge space (upper space) based on PSO. Both levels have their groups and evolve independently and in parallel. The lower space regularly provides excellent individuals to the upper space. After the upper space has evolved, it regularly guides the evolution of the lower space; therefore, a "double evolution and double promotion" mechanism is formed so as to increase the population diversity of PSO and avoid "prematurity" to improve the calculation accuracy and computational efficiency. This article introduces the index of the premature convergence degree of the particle swarm, which is calculated to determine the spatial state of the population, and the inertia weight is dynamically and nonlinearly changed, to improve the optimization efficiency of the cultural particle swarm algorithm. Simulation results prove that in terms of the steering angle of a multitarget ship, the algorithm can effectively avoid the defect of being easily trapped into a local optimum, and the convergence speed and optimization accuracy are far better than the single PSO and CPSO algorithms.

The rest of the paper is organized as follows. The related works are introduced in Section 2. Section 3 gives a ship collision avoidance decision framework based on improved cultural particle swarms. Section 4 presents the steering angle direction decision method based on ship collision risk. Section 5 presents the optimal steering angle decision algorithm based on improved cultural particle swarms. Section 6 integrates the method of this paper with the electronic chart platform and performs experimental analysis. Finally, the summary and conclusions are given in Section 7.

## 2. Related Research

At present, in the practice of ship collision avoidance, it is still inseparable from the subjective decision of the driver, that is, to manually complete the collision avoidance task

based on experience. In the early stage of the shipping industry, the density of traffic flow is small, and the ship speed is low. The method of subjective judgment and manual operation for collision avoidance decision is still competent. However, as the density of ships traffic increases, it becomes more and more difficult to manually complete the collision avoidance decision task of ships. So far, there has been no effective method to satisfy mariners in the area of ship automation collision avoidance decision, and it has become an urgent problem that needs to be solved. The problem of collision avoidance decision is essential to find the optimal steering angle of the ship without collision. At present, many solutions to this problem have been proposed. The mainstream methods include traditional methods such as ship optimal control theory [22], speed obstacle method [23], distributed tabu search algorithm [24], including evolutionary algorithms [25], and other related methods.

In terms of traditional methods, Johansen et al. [22] proposed a ship collision avoidance model based on the optimal control theory. By changing the offset to the autopilot's heading angle, a limited set of alternative control behaviors can be obtained, and then whether these alternative control behaviors meet the COLREGs or not is determined and the relevant collision risk is evaluated to find the best control behavior. But this method mainly considers the situation of one-to-one ships and does not include the treatment of priority avoidance of the most dangerous target ship in case of multiple target ships (two or more). Kuwata et al. [23] proposed a ship autonomous motion model based on the speed obstacle method. This model combines COLREGs to generate a cone-shaped obstacle in the ship's speed space. It dynamically jumps out of the obstacle range by changing the course of the ship to ensure that the ship's speed vector is not within the range of the cone obstacle. The advantage of this method is that it can check ship collisions along arbitrary trajectories. However, since collision checking needs to be performed on many time slices, it is computationally expensive and time-consuming. Kim et al. [24] used distributed taboo search algorithm (DTSA) to find the best course for the ship. When multiple ships meet, it is assumed that the highest priority of selecting the next course will be given to the ship, which can minimize the risk of collision by changing the course. This method calculates the collision risk from the information of the current heading received by neighboring ships. This process is repeated until the collision risk disappears. Although these works consider a distributed coordination structure involving multiple ships, they ignore the AIS (automatic identification system). The problem that the data cannot be updated on time causes the ship to be unable to change the course according to the prescribed time.

In recent years, the rapid development of intelligent technologies and methods has overcome the difficulties of abstract factors and quantification difficulties encountered in the early use of deterministic methods to solve collision avoidance decision problems [26]. For example, the impact of these parameters on ship collisions estimated only based on existing experiments and expert experience will make the collision avoidance decision unreliable. The fuzzy

distribution usually needs to find a function similar to the ship collision within a given fuzzy distribution function range. The actual parameters are determined through prior knowledge or experimental data to obtain the specific membership function and realize the quantification of the collision risk of ships. Therefore, the application of evolutionary algorithms in ship collision avoidance has gradually become a new research direction. For example, Lazarowska et al. [25] applied the ant colony (ACO) algorithm to the unmanned surface vehicle (USV) control system and realized the decision of USV optimal steering angle included in the high seas and restricted waters. The system converts the problem of finding the optimal rudder angle into an optimization problem, takes collision risk and voyage loss as the objective function, and uses the ACO algorithm to solve the optimal solution. However, the convergence speed of the ACO algorithm is slow and it is prone to appear stagnation, and the ship information acquisition error is not considered. Tam et al. [27] thought of all ships as moving points and used the genetic algorithm (GA) to analyze the relevant information of the ships that meet in close range to find the optimal steering angle of the ship. Although the GA has a good global search capability, the operation of the algorithm requires more training time.

In summary, the current mainstream ship collision avoidance methods still have various problems, which are prominently manifested in the following aspects:

- (1) Most of the methods of steering and collision avoidance do not consider the problems of delayed data update, equipment operation errors, and other problems leading to untimely data updates and inaccurate ship trajectory, which affects the decision of steering angle.
- (2) Most studies focus on the collision avoidance behavior of one-to-one ships and do not analyze the situation of the ship's encounter with the most dangerous target ships, in the case of encounters with multitarget ships (two and more), resulting in the ship's steering angle direction decisions not satisfying COLREGs.
- (3) Although the evolutionary algorithm has a good performance in finding the optimal steering angle of the ship without collision, the related evolutionary algorithm is liable to fall into a local optimum or the training time is too long.

Aiming at the above problems, this paper proposes a collision avoidance decision method based on improved cultural particle swarm optimization, which combines Kalman filtering algorithm, fuzzy distribution algorithm, CPSO algorithm, and COLREGs. The main contributions of the method in this paper are as follows:

- (1) This method firstly uses the advantages of the Kalman filter algorithm in target tracking and prediction and smoothing and prediction of ship motion trajectory through AIS ship observation node data; the error caused by the AIS data transmission is reduced, the accuracy of the data is improved, and

the accuracy of the collision avoidance decision is further improved. The problem that the AIS data are not updated in time is solved, the accuracy of the data is improved, and the accuracy of collision avoidance decision is further improved.

- (2) Using the advantage of fuzzy distribution algorithm in dealing with uncertain mathematical models, this paper establishes the corresponding membership function for several factors that mainly affect the ship's risk degree, solves the ship's collision risk degree, and finds out the most dangerous target ship. Combined with COLREGs, the problem of steering angle direction is solved by analyzing the encounter situation between the ship and the most dangerous target ship when meeting with two or more target ships.
- (3) In the calculation of the optimal steering angle, the index to evaluate the premature convergence degree of particle swarm is introduced to dynamically change the inertia weight of particles, and the advantage of cultural particle swarm algorithm in finding the global optimal solution is higher than that of GA, PSO, and other common evolutionary algorithms in calculation accuracy and efficiency, which improves the convergence speed and optimization accuracy of the algorithm and solves the problem that the evolutionary algorithm is easy to fall into local optimum and training time is too long.

### 3. Ship Collision Avoidance Decision Framework Based on Improved Cultural Particle Swarm

The ship collision avoidance decision framework based on the improved cultural particle swarm is shown in Figure 1. The framework is mainly composed of three stages of functions. In the first stage, the collision risk between the ship and the surrounding target ships is analyzed and the optimal steering angle direction is determined according to COLREGs. In the second stage, the ship's steering angle is calculated based on the steering angle direction determined in the first stage, and the obtained steering angle is input to the ship control module to realize the ship collision avoidance operation. In the third stage, the method of the ship steering angle direction decision in the first stage and the method of the ship steering angle decision in the second stage are integrated into the electronic chart platform, and the validity of the collision avoidance decision method studied in this paper is verified.

*3.1. Ship Steering Angle Direction Decision.* In this stage, the direction of ship steering angle is determined. Since the ship is on the high seas and the waters connected with it that can be navigable, in addition to the implementation of local rules in ports and rivers, COLREGs will be followed first, that is, before calculating the optimal steering angle of the ship, the direction of ship steering angle needs to be determined

according to COLREGs. This stage includes 4 steps: ship trajectory prediction, collision risk calculation, judgment of the situation with the most dangerous ship, and judgment of the steering angle direction.

First, the Kalman filter algorithm is used to smooth and predict the ship's motion trajectory to solve the problem of untimely updating of AIS data. Then, the fuzzy distribution algorithm is used to solve the collision risk of the ship and find the most dangerous target ship. Finally, according to the influence parameters of own ship and the most dangerous target ship, it is divided into 16 kinds of ship encounter situations according to literature [28], and the meeting situation of own ship and the most dangerous target ship is judged. By consulting the captain and crew having practical navigation experience and according to the collision avoidance rules and the encounter situation between the own ship and target ship, the rudder angle direction of the ship is determined.

*3.2. Optimal Steering Angle Decision.* At this stage, the optimal steering angle of the ship is calculated based on the improved cultural particle swarm algorithm.

First, an index that evaluates the degree of premature convergence of the particle swarm is introduced to determine the state of the particles in the population space to determine the degree of convergence of the population, and then the inertial weight is adaptively changed according to the result of the judgment to ensure the diversity of the particles in the population space. Then, the optimal steering angle is calculated based on the improved cultural particle swarm algorithm. The fitness function is constructed with the help of the collision risk of the ship, and the main group space (lower space) and knowledge space (upper space) of the algorithm are established. The interaction between them improves the evolution efficiency to find the optimal steering angle of the ship more effectively. Finally, the obtained optimal steering angle is input to the ship control module to achieve collision avoidance of the ship.

*3.3. Electronic Chart Platform Integration and Display.* This paper uses electronic charts as a display and verification platform and integrates the steering angle decision method based on ship collision risk and the optimal steering angle decision method based on improved cultural particle swarm algorithm into the electronic chart platform. The platform can dynamically display the status and trajectory of all ships in real time, which can easily show the experimental results of collision avoidance between own ship and target ship and verify the feasibility of the method.

### 4. Direction Decision Method of Steering Angle Based on Ship Collision Risk

*4.1. Optimal Steering Angle Decision.* The real-time acquisition of accurate ship AIS information is the database of ship collision avoidance decision making. Only the real-time acquisition of ship AIS data can effectively monitor the navigation status of the water area and timely find the ship

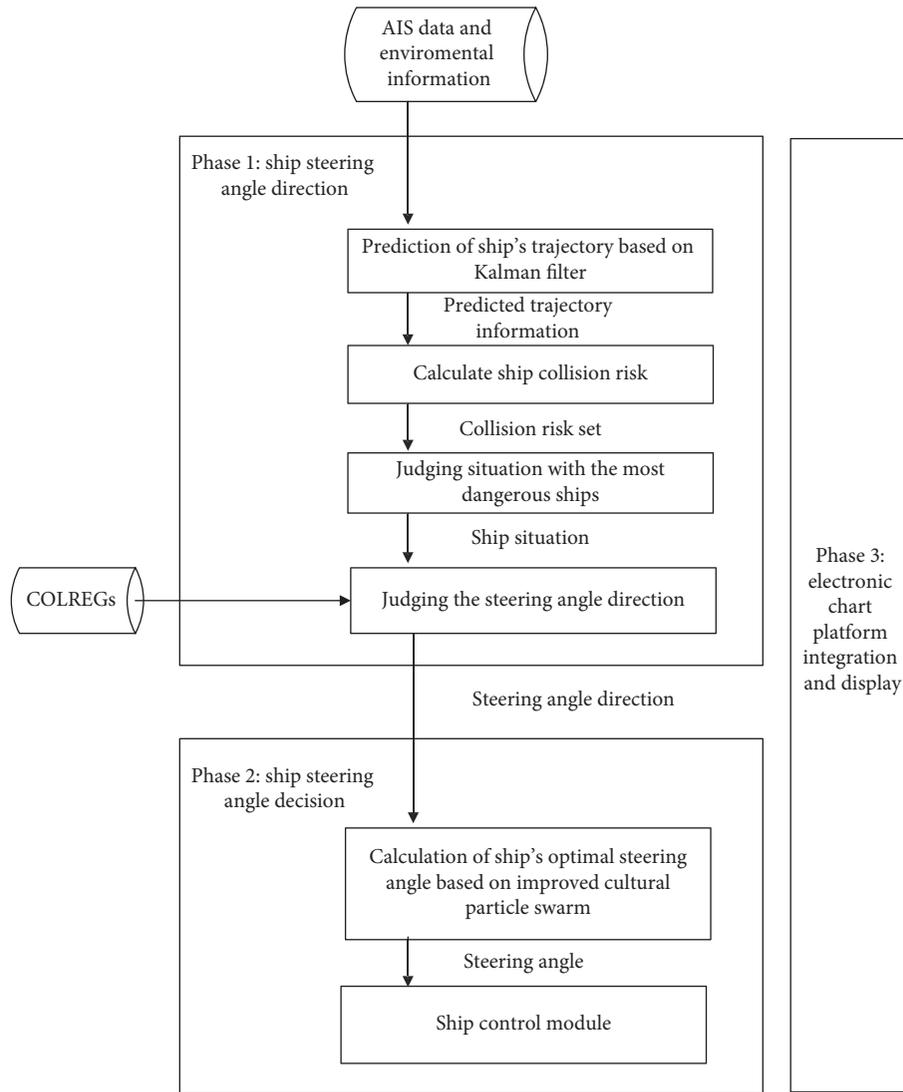


FIGURE 1: A ship collision avoidance decision framework based on cultural particle swarm.

information with collision risk in the navigation water area. At present, although a large number of AIS instruments have been installed in ship and shore-based management, due to various subjective and objective reasons, as well as congestion and network transmission of AIS ship stations and shore stations, such as improper AIS operation and data broadcast by AIS equipment due to ships, the distance from the AIS base station, the abnormal operation of the AIS device itself, or the artificial shutdown of the AIS device caused the AIS data received by the shore-based AIS to be incorrect and the data location report to be updated in a timely manner.

In order to compensate for the delayed data update caused by AIS data blockage and other reasons, resulting in inaccurate or large errors in the ship's trajectory, this paper uses the Kalman filter algorithm and considers the measurement noise caused by the AIS data transmission. Smoothing and predicting the ship's motion trajectory can reduce the error caused by the AIS data transmission, improve the accuracy of the data, and further improve the accuracy of collision avoidance decisions.

#### 4.1.1. Basic Concepts

- (1) *AIS Data*. AIS (automatic ship identification system) is a new navigation aid system for shore-to-ship, ship-to-shore, and ship-to-ship communication [29]. AIS equipment can send own ship information such as latitude and longitude, speed, heading, ship name, and other information to the target ship via VHF channel after processing by the processor and automatically obtain the position of other ships with similar equipment in real time, speed, heading, etc.
- (2) *Mercator Projection*. Mercator projection is a map projection technology that uses an equiangular projection method. The principle is to assume that the Earth is enclosed in a hollow cylinder, whose datum of latitude (equator) is tangent to the cylinder, and then imagine a lamp in the center of the Earth, project the figure on the sphere onto the cylinder, and expand the cylinder to get a map drawn by the "Mercator projection" on the selected datum line [30].

The experimental model needs to convert the WGS84 latitude and longitude coordinates in the AIS data to Mercator  $x$ ,  $y$  coordinates to predict the ship's trajectory. According to the principle of Mercator projection, the projection of the equator is  $X$  axis, the projection of the prime meridian is  $Y$  axis, and the projection of the intersection of the two is the origin of the coordinate, which is positive toward east and north and negative toward west and south, forming the Mercator plane rectangular coordinate system. Let the long axis of the Earth be  $a$  and the short axis be  $b$ . The longitude of a point on the Earth is  $\theta \in (-\pi, +\pi)$  with latitude  $\alpha \in (-\pi/2, +\pi/2)$ ; according to the equiangular principle, the coordinate conversion formula is as follows:

$$\begin{cases} x = a * \theta, \\ y = a * \ln\left(\tan\left(\frac{\pi}{4} + \frac{\alpha}{2}\right) + \frac{e}{2} \ln\left(\frac{1 - e * \sin \alpha}{1 + e * \sin \alpha}\right)\right), \end{cases} \quad (1)$$

where  $e = \sqrt{a^2 - b^2}/a^2$  is the first eccentricity of the Earth's ellipsoid, and formula (1)  $(\theta, \alpha)$  translates to plane coordinates  $(x, y)$ .

- (3) *Trajectory Vector Set.* Modeling the two-dimensional plane  $X$  axis and  $Y$  axis of Euclidean space, using vectors in the directions of the two coordinate axes to represent trajectory data, and let  $T$  be the trajectory vector set.  $T = \{Trj_1, Trj_2, \dots, Trj_n\} = \{(p_x^1, p_y^1), (p_x^2, p_y^2), \dots, (p_x^n, p_y^n)\} = \{(p_x^1, p_x^2, \dots, p_x^n)^T, (p_y^1, p_y^2, \dots, p_y^n)^T\}$ , where  $p_x^i = (x_{i1}, x_{i2}, \dots, x_{id})^T$  represents the projection vector set of the  $i$ th trajectory in the  $X$  axis direction,  $p_y^i = (y_{i1}, y_{i2}, \dots, y_{id})^T$  represents the projection vector set of the  $i$ th trajectory in the  $Y$  axis direction, and  $(p_x^i, p_y^i)$  is called the vector set of the trajectory  $Trj_i$ .
- (4) *Kalman Filtering.* The core is to use recursive algorithms to estimate the optimal system state and realize the prediction of target motion behavior [31]. The Kalman filtering system of state equations and observation equations is as follows:

$$\begin{cases} X(k+1) = A(k)X(k) + \Gamma(k)U(k) + T(k)W(k), \\ Z(k) = H(k)X(k) + V(k), \end{cases} \quad (2)$$

where  $X(k+1)$  represents  $k+1$  state value at the moment,  $A(k)$  is the state transition matrix,  $\Gamma(k)$  is the control matrix,  $U(k)$  is the control input vector,  $T(k)$  is the interference transfer matrix,  $W(k)$  is the system state noise of the motion model,  $Z(k)$  is the observation vector,  $H(k)$  is the observation matrix, and  $V(k)$  is the observation noise.

- (5) *Prediction Error.* For the geometric spatial error between the predicted trajectory point and the actual trajectory point, the root mean square error is shown as follows:

$$RMSE = \frac{\sum_{i=1}^k \sqrt{(x'_i - x_i)^2 + (y'_i - y_i)^2}}{k}, \quad (3)$$

where  $(x_i, y_i)$  represents the position of the actual trajectory point,  $(x'_i, y'_i)$  position information represents predicted trajectory points, and  $k$  represents the number of predicted trajectory points.

*4.1.2. Ship Trajectory Prediction.* In this paper, the AIS observation data transmitted by the VHF channel are used to optimally estimate the system state such as the ship position and speed to make real-time prediction of the ship trajectory.

(1) *Ship Trajectory Modeling.* The latitude and longitude of the ship are transformed into coordinates  $(x, y)$  in the Cartesian coordinate system through Mercator projection.  $W_x(k)$  and  $W_y(k)$  are the two orthogonal components of Gaussian white noise  $W(k)$  with zero mean and variance  $\sigma_W^2$ ,  $W_x(k)$  and  $W_y(k)$  are independent of each other at any time. The values are related to factors such as actual sea wind, waves, and weather.  $V_x(k)$  and  $V_y(k)$  are two orthogonal components of Gaussian white noise  $V(k)$  with zero mean and variance  $\sigma_V^2$ ,  $V_x(k)$  and  $V_y(k)$  are independent of each other at any time. The value is related to the error of the actual AIS information in the detection and transmission process.  $W(k)$  and  $V(k)$  are independent Gaussian white noises, which the corresponding covariances are  $Q$  and  $R$ .

The model obtains the optimal state estimate  $X'(k)$  of the system at time  $k$  based on the number of observations transmitted by AIS in the first  $kt$  times. There are two different update processes in the process of the stochastic linear discrete Kalman filtering cycle, which are the time update process and the observation update process. The time update process is to predict the state at the current moment according to the optimal state estimation at the previous moment (see equation (4)). The observation update process is that after predicting the ship's trajectory point at  $k+1$  time, the observed value of the ship's trajectory point at the actual  $k+1$  time is used for linear fitting to find out the optimal estimated position of the trajectory point. That is to say, according to the observed value and the predicted value, the optimal estimation point of the ship's trajectory at  $k+1$  time is derived through the observation update equation, as shown in equation (5).

$$\begin{cases} x(k+1, k) = x(k, k) + v(k)t \cos(\theta(k)), \\ y(k+1, k) = y(k, k) + v(k)t \sin(\theta(k)), \end{cases} \quad (4)$$

$$\begin{cases} x(k+1, k+1) = x(k+1, k) + K(k+1) * (x_z(k+1) - x(k+1, k)), \\ y(k+1, k+1) = y(k+1, k) + K(k+1) * (y_z(k+1) - y(k+1, k)), \end{cases} \quad (5)$$

where  $x(k+1, k)$  is the predicted x coordinate of the ship at time  $k+1$ .  $y(k+1, k)$  is the predicted y coordinate of the ship at time  $k+1$ .  $x(k, k)$  is the optimal estimate of the x coordinate of the ship at time  $k$ .  $y(k, k)$  is the optimal estimate of the y coordinate of the ship at time  $k$ .  $v(k)$  is the ship's speed at time  $k$ , the  $\theta(k)$  is the heading of the ship at time  $k$ , and  $t$  is the time interval.  $K(k+1)$  is the filter gain matrix at time  $k+1$ . For the value and the update formula of the covariance  $P(k+1, k+1)$  of the optimal state estimation at time, see formula (6).

$$\begin{cases} P(k+1, k) = A(k)P(k, k)A(k)^T + T(k)Q(k)T(k)^T, \\ S(k+1) = H(k+1)P(k+1, k)H(k+1)^T + R(k+1), \\ K(k+1) = P(k+1, k)H(k+1)^T S(k+1)^{-1}, \\ P(k+1, k+1) = P(k+1, k) - K(k+1)S(k+1)K(k+1)^T. \end{cases} \quad (6)$$

Where  $Q(k)$  is the system noise of the ship at node  $k$ ,  $R(k)$  is the observed noise,  $P(k, k)$  is the error variance matrix,  $A(k)$  is the ship state transition matrix,  $P(k+1, k)$  is the error variance matrix of the predictive state  $x(k+1, k)$  and  $y(k+1, k)$ , and  $K(k)$  is the filter gain matrix. Besides that,  $W_x(k)$  and  $W_y(k)$  have zero mean. Gaussian white noise with variance  $\sigma_w^2$  and the time update equation of equation (5), we can get  $T(k)$  and  $A(k)$  as the identity matrix. Furthermore, the coordinates do not undergo unit conversion. The observation matrix  $H(k)$  is also the identity matrix. According to the above formula, the optimal position prediction value of the ship at a single step is obtained. If the ship  $nt$  position is predicted, the ship trajectory prediction can be completed iteratively.

The ship trajectory prediction algorithm based on Kalman filtering is shown in Algorithm 1. The detailed steps are as follows.

In Algorithm 1, the function *trajectory prediction*( $T$ ) is used to collect and decode ship AIS data and perform  $(x, y)$  preprocessing operations such as coordinate transformation. The function *initParameters* is used to determine the parameters of the ship's motion model according to the state equation and observation equation of the system; besides, this function is also used to initialize the ship's  $A$ ,  $U$ , and  $Q$  with  $R$  and other parameters. The function *getCurrentState*( $D$ ) is based on the optimal state estimate at the initial moment  $X(0, 0)$  and estimates error variance matrix  $P(0, 0)$ , predicting the ship trajectory value at the next moment according to the system state equation  $X(1, 0)$  and obtaining the covariance matrix of the estimation error at the same time  $P(1, 0)$ . According to the observed value  $Z(1)$  of the ship trajectory at the next moment, the optimal state estimation value  $X(1, 1)$  and the covariance matrix  $P(1, 1)$  of the optimal estimation error are obtained to complete the single-step filtering process of the ship trajectory. Then iteratively obtain the optimal state estimate  $X(n-1, n-1)$  at the time of  $n-1$  to complete the filtering process. The function of Kalman Predict( $D$ ) is to estimate the position of the ship's trajectory point at time  $n+1$  based on the optimal state estimation  $X(n-1, n-1)$  at time  $n-1$  and the observed value of current time  $n$ . The function of

*getRMSE*( $p, p'$ ) is to compare the predicted point  $p'$  with real trajectory point  $p$  to calculate the filtering prediction error. Finally, repeat the operation  $k$  times to complete the prediction of the future  $k$  steps of ship trajectory points. Then, calculate and output the average prediction error.

(2) *Simulation Experiment Verification*. Experimental model using  $s_k$  indicates that the ship's real location is at the sampling time  $kt$ .  $y(k)$  means observations of AIS data transmission at the moment  $kt$ , where  $y(k) = s(k) + v(k)$ , and  $v(k)$  represents observation noise.  $s'(k)$  means speed of ship at the moment  $kt$ , acceleration  $a(k)$  is a combination of maneuvering acceleration  $u(k)$  and random acceleration  $w(k)$ . The uniform acceleration motion and acceleration formula of the ship is as follows:

$$\begin{cases} s(k+1) = s(k) + s'(k)t + 0.5t^2 a(k), \\ s'(k+1) = s'(k) + ta(k), \\ a(k) = u(k) + w(k), \end{cases} \quad (7)$$

where  $u(k)$  are the control signals of the ship's power system; define the system status  $x(k)$  at the sampling time  $kt$  for the position and speed of the ship; the equations of motion and observation of the ship are shown as follows:

$$\begin{cases} \begin{bmatrix} s(k+1) \\ s'(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s(k) \\ s'(k) \end{bmatrix} + \begin{bmatrix} 0.5T_0^2 \\ T_0 \end{bmatrix} u(k) + \begin{bmatrix} 0.5T_0^2 \\ T_0 \end{bmatrix} w(k), \\ y(k) = [1 \ 0] \begin{bmatrix} s(k) \\ s'(k) \end{bmatrix} + v(k). \end{cases} \quad (8)$$

This experiment does not consider the ship's own dynamic factors ( $u(k) = 0$ ) and state vector  $X_k$  generalized to four dimensions; its expression and system equations are shown in equation (9), and the initial position is set as  $(-100 \text{ m}, 200 \text{ m})$ ,  $X$  shaft speed is set as  $2 \text{ m/s}$ , and  $Y$  shaft speed is set as  $20 \text{ m/s}$ . The experimental record of the ship trajectory formed at 20 time points is shown in Figure 2, and the error analysis diagram is shown in Figure 3.

$$\begin{cases} \begin{bmatrix} x_k \\ x_k^- \\ y_k \\ y_k^- \end{bmatrix} = \begin{bmatrix} 1, T, 0, 0 \\ 0, 1, 0, 0 \\ 0, 0, 1, T \\ 0, 0, 0, 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ x_{k-1}^- \\ y_{k-1} \\ y_{k-1}^- \end{bmatrix} + \begin{bmatrix} 0.5T^2, 0 \\ T, 0 \\ 0, 0.5T^2 \\ 0, T \end{bmatrix} w_k, \\ Z_k = \begin{bmatrix} 1, 0, 0, 0 \\ 0, 0, 1, 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_k^- \\ y_k \\ y_k^- \end{bmatrix} + v_k, \\ X_k = [x_k, x_k^-, y_k, y_k^-]. \end{cases} \quad (9)$$

The observed ship trajectory obviously oscillates in Figure 2, indicating that the measurement noise has a great impact, and after Kalman filtering, the filtering estimation is relatively close to the true trajectory of the ship. Figure 3 shows that the observed noise of the ship displacement is

```

Input: trajectory dynamic AIS dataset of moving ships  $T = \{Trj_1, Trj_2, \dots, Trj_n\}$ 
Output: mean value of ship trajectory error RMSE
(1)  $D = \text{trajectPretreatment}(T)$ 
(2)  $\text{initParameters}()$ 
(3)  $\text{state} = \text{getCurrentState}(D)$ 
(4) for  $i = 1$  to  $k$ :
(5)    $p = \text{KalmanPredict}(D)$ 
(6)    $e[i] = \text{getRMSE}(p, p')$ 
(7) end for
(8)  $\text{RMSE} = (\sum_{i=1}^k e[i])/k$ 
(9) Output RMSE
    
```

ALGORITHM 1: Ship trajectory prediction algorithm based on Kalman filter.

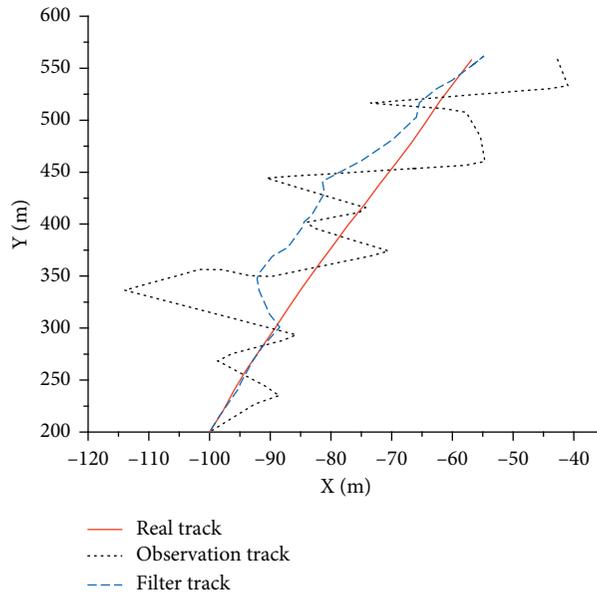


FIGURE 2: Ship tracking trajectory.

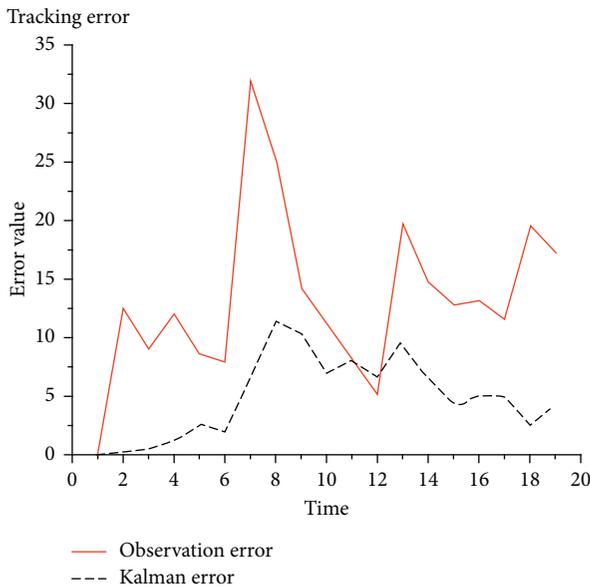


FIGURE 3: Ship tracking error.

close to 32 m, and the experiment is only simulation; the actual ship AIS measurement error will not be so large. After Kalman filter processing, the ship's displacement deviation is reduced to less than 12 m. Experiments show that Kalman filter can limit the impact of noise to the greatest extent and better predict the position after the ship.

4.2. Calculation of Ship Collision Risk. The membership function of different factors determines the value of ship collision risk. The collision risk of ships plays a key role in measuring the collision risk of ships and is an important basis for ship collision avoidance decision making [32]. The crew can decide the timing and order of avoidance according to the collision risk. There are many factors that affect the collision risk of ships. From the statistical research on the crew's collision avoidance behavior at sea by experts and scholars [33] and the research on the decision-making process, it can be seen that the main factors for judging the collision risk are intership distance, azimuth, DCPA, TCPA, and speed ratio. These factors can fully reflect the encounter

situation with the target, so the membership function of these factors is used as input to infer the collision risk of the ship.

**4.2.1. Calculation of Influence Parameters.** Kalman filtering prediction is used to obtain the ship  $X$  axis position coordinate  $x_0$ , the  $Y$  axis position coordinate  $y_0$ , the own ship's speed on the  $X$  axis component  $x_{v0}$  and the  $Y$  axis component  $y_{v0}$ , other ship  $X$  axis position coordinate  $x_1$ ,  $Y$  axis

position coordinate  $y_1$ , the component  $x_{v1}$  on the  $X$  axis, and the component  $y_{v1}$  on the  $y_{v1}$  axis. Then, the longitude  $A_j$ , latitude  $A_w$ , heading  $A_h$ , and speed  $A_s$  of the own ship and the longitude  $B_j^m$ , latitude  $B_w^m$ , heading  $B_h^m$ , and speed  $B_s^m$  of other ships are obtained through the reverse solution of the Mercator projection equation. By substituting the above information into equation (10), the relevant influencing factor values can be solved.

$$\left. \begin{aligned}
 D &= 2 * R * \arcsin \left( \sqrt{\sin^2 \left( \frac{B_w^m - A_w}{2} \right) + \cos(A_w) * \cos(B_w^m) * \sin \left( \frac{B_j^m - A_j}{2} \right)} \right), \\
 T_B &= \begin{cases} \arctan \left( \frac{(B_j^m - A_j) * \cos(B_w^m)}{B_w^m - A_w} \right); & \left( \arctan \left( \frac{(B_j^m - A_j) * \cos(B_w^m)}{B_w^m - A_w} \right) \in \text{first quadrant} \right), \\ 360 + \arctan \left( \frac{(B_j^m - A_j) * \cos(B_w^m)}{B_w^m - A_w} \right); & \left( \arctan \left( \frac{(B_j^m - A_j) * \cos(B_w^m)}{B_w^m - A_w} \right) \in \text{second quadrant} \right), \\ 180 - \arctan \left( \frac{(B_j^m - A_j) * \cos(B_w^m)}{B_w^m - A_w} \right); & \left( \arctan \left( \frac{(B_j^m - A_j) * \cos(B_w^m)}{B_w^m - A_w} \right) \in \text{third and fourth quadrants} \right), \end{cases} \\
 B &= A_h - TB, \\
 \text{RSH} &= \arctan \left( \frac{B_s^m * \cos(B_h^m) - A_s * \cos(A_h)}{B_s^m * \sin(B_h^m) - A_s * \sin(A_h)} \right), \\
 \text{RS} &= \sqrt{B_s^{m2} + A_s^2 - 2 * B_s^m * A_s * \cos(A_h - B_h^m)}, \\
 \text{DCPA} &= D * \sin(\text{RSH} - RA - 180^\circ), \\
 \text{TCPA} &= D * \cos(\text{RSH} - RA - 180^\circ),
 \end{aligned} \right\} \quad (10)$$

where  $D$ ,  $TB$ ,  $B$ ,  $\text{RSH}$ , and  $\text{RS}$  are the distance, azimuth, relative azimuth, relative speed course, and relative speed between ships.

**4.2.2. Establishment of Collision Risk Model.** The collision risk calculation method used in this paper is calculated through fuzzy sets. Fuzzy set is a set used to express the concept of fuzziness. It is defined as a given research range  $U$ ; then a mapping from  $U$  to the unit interval  $[0, 1]$  is called a fuzzy set on  $U$  or a set of  $U$  fuzzy subset. Fuzzy set is described by membership function. The membership function occupies an extremely important position in fuzzy set theory. It is defined as if any element  $x$  in the research range  $U$  has a number  $A(x) \in [0, 1]$  corresponding to it, then  $A$  is called the

fuzzy set on  $U$ , and  $A(x)$  is called the membership degree of  $x$  to  $A$ . When  $x$  changes in  $U$ ,  $A(x)$  is a function called  $A$ 's membership function. In the fuzzy set, the value range of its characteristic function is  $[0, 1]$  interval; this characteristic function is also called membership function. Membership function is an important basis of fuzzy set theory, so how to determine the membership function is a key issue. However, the object is "fuzzy" and empirical, so it is unrealistic to find a unified calculation method for membership. The fuzzy distribution method currently used is used to construct the membership function of each influencing factor to realize the calculation of ship collision risk. The membership function formula ((11)–(17)) used in this paper refers to the relevant formula in reference [33].

(1) *DCPA Membership Function*  $t_{DCPA}$ . DCPA is the distance that the ship will meet recently. Statistics show that when the distance between the two ships is less than the distance between the two ships, the possibility of collision between the ships is extremely large. According to the empirical data of collision avoidance in open waters, when  $DCPA \leq 0.6n$  mile, it is dangerous,  $t_{VD(DCPA)} = 1$ ; when  $0.6n \text{ mile} < DCPA \leq 1.8n$  mile, it is considered very

dangerous,  $t_{VD(DCPA)}$ ; when  $0.6n \text{ mile} < DCPA \leq 1.2n$  mile, it is considered dangerous,  $t_{D(DCPA)}$ ; when  $1.0n \text{ mile} < DCPA \leq 1.6n$  mile, it is considered fair,  $t_{N(DCPA)}$ ; when  $1.4n \text{ mile} < DCPA \leq 2.5n$  mile, it is considered safer,  $t_{S(DCPA)}$ ; when  $DCPA > 2.5n$  mile, it is considered very safe,  $t_{VS(DCPA)}$ . The membership function is shown in equation (11), and the corresponding function image is shown in Figure 4.

$$\begin{aligned}
 t_{VD(DCPA)} &= \begin{cases} 1, & DCPA \leq 0.6n \text{ mile}, \\ 0.5 - 0.5 \sin \frac{\pi}{0.2} (x - 0.7), & 0.6n \text{ mile} < DCPA < 0.8n \text{ mile}, \\ 0, & 0.8n \text{ mile} \leq DCPA, \end{cases} \\
 t_{D(DCPA)} &= \begin{cases} 0, & DCPA \leq 0.6n \text{ mile}, \\ 0.5 + 0.5 \sin \frac{\pi}{0.2} (x - 0.7), & 0.6n \text{ mile} < DCPA < 0.8n \text{ mile}, \\ 1, & 0.8n \text{ mile} \leq DCPA \leq 1n \text{ mile}, \\ 0.5 - 0.5 \sin \frac{\pi}{0.2} (x - 1.1), & 1n \text{ mile} < DCPA < 1.2n \text{ mile}, \\ 0, & 1.2n \text{ mile} \leq DCPA, \end{cases} \\
 t_{N(DCPA)} &= \begin{cases} 0, & DCPA \leq 1n \text{ mile}, \\ 0.5 + 0.5 \sin \frac{\pi}{0.2} (x - 1.1), & 1n \text{ mile} < DCPA < 1.2n \text{ mile}, \\ 1, & 1.2n \text{ mile} \leq DCPA \leq 1.4n \text{ mile}, \\ 0.5 - 0.5 \sin \frac{\pi}{0.2} (x - 1.5), & 1.4n \text{ mile} < DCPA < 1.6n \text{ mile}, \\ 0, & 1.6n \text{ mile} \leq DCPA, \end{cases} \\
 t_{S(DCPA)} &= \begin{cases} 0, & DCPA \leq 1.4n \text{ mile}, \\ 0.5 + 0.5 \sin \frac{\pi}{0.3} (x - 1.55), & 1.4n \text{ mile} < DCPA < 1.7n \text{ mile}, \\ 1, & 1.7n \text{ mile} \leq DCPA \leq 2n \text{ mile}, \\ 0.5 - 0.5 \sin \frac{\pi}{0.5} (x - 2.25), & 2n \text{ mile} < DCPA < 2.5n \text{ mile}, \\ 0, & 2.5n \text{ mile} \leq DCPA, \end{cases} \\
 t_{VS(DCPA)} &= \begin{cases} 0, & DCPA \leq 2.0n \text{ mile}, \\ 0.5 + 0.5 \sin \frac{\pi}{0.5} (x - 2.25), & 2.0n \text{ mile} < DCPA < 2.5n \text{ mile}, \\ 1, & 2.5n \text{ mile} \leq DCPA. \end{cases}
 \end{aligned} \tag{11}$$

(2) *TCPA Membership Function*  $t_{TCPA}$ . *TCPA* is the recent meeting time of the ship, which reflects the urgency of the meeting situation between the two ships. Statistics and empirical data show that  $TCPA \leq 6$  min Time,  $t_{VD(TCPA)} = 1$ ; when  $8 \text{ min} \leq T CPA \leq 9$  min Time,  $t_{D(TCPA)} = 1$ ; when

$11 \text{ min} \leq T CPA \leq 12$  min Time,  $t_{N(TCPA)} = 1$ ; when  $14 \text{ min} \leq T CPA \leq 15$  min Time,  $t_{S(TCPA)} = 1$ ; when  $T CPA \geq 17$  min Time,  $t_{VS(TCPA)} = 1$ . The membership function is shown in equation (12), and the corresponding function image is shown in Figure 5.

$$\begin{aligned}
 t_{VD(TCPA)} &= \begin{cases} 1, & T CAP \leq 6 \text{ min}, \\ 0.5 - 0.5 \sin \frac{\pi}{2} (x - 7), & 6 \text{ min} < T CPA < 8 \text{ min}, \\ 0, & 8 \text{ min} \leq T CPA, \end{cases} \\
 t_{D(TCPA)} &= \begin{cases} 0, & T CAP \leq 6 \text{ min}, \\ 0.5 + 0.5 \sin \frac{\pi}{2} (x - 7), & 6 \text{ min} < T CPA < 8 \text{ min}, \\ 1, & 8 \text{ min} \leq T CPA \leq 9 \text{ min}, \\ 0.5 - 0.5 \sin \frac{\pi}{2} (x - 10), & 9 \text{ min} < T CPA < 11 \text{ min}, \\ 0, & 11 \text{ min} \leq T CPA, \end{cases} \\
 t_{N(TCPA)} &= \begin{cases} 0, & T CAP \leq 9 \text{ min}, \\ 0.5 + 0.5 \sin \frac{\pi}{2} (x - 10), & 9 \text{ min} < T CPA < 11 \text{ min}, \\ 1, & 11 \text{ min} \leq T CPA \leq 12 \text{ min}, \\ 0.5 - 0.5 \sin \frac{\pi}{2} (x - 13), & 12 \text{ min} < T CPA < 14 \text{ min}, \\ 0, & 14 \text{ min} \leq T CPA, \end{cases} \\
 t_{S(TCPA)} &= \begin{cases} 0, & T CAP \leq 12 \text{ min}, \\ 0.5 + 0.5 \sin \frac{\pi}{2} (x - 13), & 12 \text{ min} < T CPA < 14 \text{ min}, \\ 1, & 14 \text{ min} \leq T CPA \leq 15 \text{ min}, \\ 0.5 - 0.5 \sin \frac{\pi}{2} (x - 16), & 15 \text{ min} < T CPA < 17 \text{ min}, \\ 0, & 17 \text{ min} \leq T CPA, \end{cases} \\
 t_{VS(TCPA)} &= \begin{cases} 0, & T CAP \leq 15 \text{ min} \\ 0.5 + 0.5 \sin \frac{\pi}{2} (x - 16), & 15 \text{ min} < T CPA < 17 \text{ min}, \\ 1, & 17 \text{ min} \leq T CPA. \end{cases}
 \end{aligned} \tag{12}$$

(3) *D Membership Function*  $t_D$ . *D* It is the distance between ships; the smaller the value is, the closer the target ship is and the greater the collision risk of the ship is. Statistical

empirical data show that when  $D \leq 1.5$  n mile,  $t_{VD(D)} = 1$ ; when  $1.7$  n mile  $\leq D \leq 2$  n mile,  $t_{D(D)} = 1$ ; when  $2.2$  n mile  $< D < 2.5$  n mile,  $t_{N(D)} = 1$ ; when

$2.7 n \text{ mile} < D < 3 n \text{ mile}$ ,  $t_{S(D)} = 1$ ; when  $D \geq 3.2 n \text{ mile}$ ,  $t_{VS(D)} = 1$ . The membership function is shown in equation

(13), and the corresponding function image is shown in Figure 6.

$$\begin{aligned}
 t_{VD(D)} &= \begin{cases} 1, & D \leq 1.5 n \text{ mile}, \\ 0.5 - 0.5 \sin \frac{\pi}{0.2} (x - 1.6), & 1.5 n \text{ mile} < D < 1.7 n \text{ mile}, \\ 0, & 1.7 n \text{ mile} \leq D, \end{cases} \\
 t_{D(D)} &= \begin{cases} 0, & D \leq 1.5 n \text{ mile}, \\ 0.5 + 0.5 \sin \frac{\pi}{0.2} (x - 1.6), & 1.5 n \text{ mile} < D < 1.7 n \text{ mile}, \\ 1, & 1.7 n \text{ mile} \leq D \leq 2 n \text{ mile}, \\ 0.5 - 0.5 \sin \frac{\pi}{0.2} (x - 2.1), & 2 n \text{ mile} < D < 2.2 n \text{ mile}, \\ 0, & 2.2 n \text{ mile} \leq D, \end{cases} \\
 t_{N(D)} &= \begin{cases} 0, & D \leq 2 n \text{ mile}, \\ 0.5 + 0.5 \sin \frac{\pi}{0.2} (x - 2.1), & 2 n \text{ mile} < D < 2.2 n \text{ mile}, \\ 1, & 2.2 n \text{ mile} \leq D \leq 2.5 n \text{ mile}, \\ 0.5 - 0.5 \sin \frac{\pi}{0.2} (x - 2.6), & 2.5 n \text{ mile} < D < 2.7 n \text{ mile}, \\ 0, & 2.7 n \text{ mile} \leq D, \end{cases} \\
 t_{S(D)} &= \begin{cases} 0, & D \leq 2.5 n \text{ mile}, \\ 0.5 + 0.5 \sin \frac{\pi}{0.2} (x - 2.6), & 2.5 n \text{ mile} < D < 2.7 n \text{ mile}, \\ 1, & 2.7 n \text{ mile} \leq D \leq 3 n \text{ mile}, \\ 0.5 - 0.5 \sin \frac{\pi}{0.2} (x - 3.1), & 3 n \text{ mile} < D < 3.2 n \text{ mile}, \\ 0, & 3.2 n \text{ mile} \leq D, \end{cases} \\
 t_{VS(D)} &= \begin{cases} 0, & D \leq 3 n \text{ mile}, \\ 0.5 + 0.5 \sin \frac{\pi}{0.2} (x - 3.1), & 3 n \text{ mile} < D < 3.2 n \text{ mile}, \\ 1, & 3.2 n \text{ mile} \leq D. \end{cases}
 \end{aligned} \tag{13}$$

(4) *B Membership Function*  $t_B$ .  $B$  is the relative position between ships, and the danger of coming ships with different relative positions is different from their own ships. Statistical empirical data show that the safety of each area is ranked as  $E, A, B, C, D$ .  $E$  area arrival range is  $(335^\circ \sim 5^\circ)$ , and the meeting situation between the coming ship and own ship in

this area is the most dangerous meeting situation.  $A$  area arrival range is  $(5^\circ \sim 67.5^\circ)$ , and the meeting situation between the coming ship and own ship in this area is a starboard small angle crossing meeting situation, which is more dangerous.  $B$  area arrival range is  $(67.5^\circ \sim 122.5^\circ)$ , and the meeting situation between the coming ship and own ship

in this area is a starboard high-angle crossing meeting situation, and the danger is average. *C* area arrival range is  $(122.5^\circ \sim 210.5^\circ)$ , and the meeting situation between the coming ship and own ship in this area is safer. *D* area arrival range is  $(247.5^\circ \sim 355^\circ)$ , and the meeting situation between the incoming ship and own ship in this area is safe. The membership function is shown in equation (14), and the corresponding function image is shown in Figure 7.

$$t_{VD(B)} = \begin{cases} 1 - \frac{B}{360^\circ} & (0^\circ \leq B \leq 5^\circ), \\ \frac{B}{360^\circ} & (355^\circ \leq B \leq 0^\circ), \end{cases}$$

$$t_{D(B)} = 1 - \frac{|B - 36.25^\circ|}{360^\circ} \quad (5^\circ < B < 67.5^\circ),$$

$$t_{N(B)} = 1 - \frac{|B - 90^\circ|}{360^\circ} \quad (67.5^\circ \leq B \leq 122.5^\circ),$$

$$t_{S(B)} = 1 - \frac{|B - 180^\circ|}{360^\circ} \quad (122.5^\circ < B < 247.5^\circ),$$

$$t_{VS(B)} = 1 - \frac{|B - 301.75^\circ|}{360^\circ} \quad (247.5^\circ \leq B < 355^\circ).$$
(14)

(5) *Ship Speed Ratio Membership Function*  $t_K$ . Define the speed ratio  $K$  of the own ship and the targetship  $m$  as  $A_s/B_s^m$ . The statistical research on collision avoidance shows that when the own ship is a low-speed ship, collision avoidance actions must be taken as soon as possible and it needs to turn a large angle. Take  $t_K = 0.5$  as the boundary point between safety and danger. The larger the value of  $t_K$ , the higher the risk of collision. The membership function is shown in equation (15), and the corresponding function image is shown in Figure 8.

$$t_K = \frac{1}{1 + (1/K)^2}. \quad (15)$$

(6) *Calculation of Ship Collision Risk*. Drawing on the research results at home and abroad and the experience of collision avoidance in open waters at sea, this paper establishes the evaluation criteria for the danger avoidance of each factor as shown in Table 1. The importance of the factor set  $\{DCPA, TCPA, D, B, K\}$  that affects the collision risk of ships is in order of  $DCPA > TCPA > D > B > K$ . According to the investigation in the literature [34], during the course of crew training bridge resource management, the corresponding weight of the factor set is  $W = [0.4, 0.367, 0.167, 0.033, 0.033]$ . The final risk vector  $R$  can be obtained through the evaluation matrix formed by the weight vector  $W$  and membership function of each factor (see equation (16)).

$$R = [W_{DCPA}, W_{TCPA}, W_D, W_B, W_K] \begin{bmatrix} t_{VD(DCPA)} & t_D(DCPA) & t_N(DCPA) & t_S(DCPA) & t_{VS(DCPA)} \\ t_{VD(TCPA)} & t_D(TCPA) & t_N(TCPA) & t_S(TCPA) & t_{VS(TCPA)} \\ t_{VD(D)} & t_D(D) & t_N(D) & t_S(D) & t_{VS(D)} \\ t_{VD(B)} & t_D(B) & t_N(B) & t_S(B) & t_{VS(B)} \\ t_{VD(K)} & t_D(K) & t_N(K) & t_S(K) & t_{VS(K)} \end{bmatrix} \quad (16)$$

$$= \begin{bmatrix} W_{DCPA}t_{VD(DCPA)} \oplus W_{TCPA}t_{VD(TCPA)} \oplus W_D t_{VD(D)} \oplus W_B t_{VD(B)} \oplus W_K t_{VD(K)} \\ W_{DCPA}t_D(DCPA) \oplus W_{TCPA}t_D(TCPA) \oplus W_D t_D(D) \oplus W_B t_D(B) \oplus W_K t_D(K) \\ W_{DCPA}t_N(DCPA) \oplus W_{TCPA}t_N(TCPA) \oplus W_D t_N(D) \oplus W_B t_N(B) \oplus W_K t_N(K) \\ W_{DCPA}t_S(DCPA) \oplus W_{TCPA}t_S(TCPA) \oplus W_D t_S(D) \oplus W_B t_S(B) \oplus W_K t_S(K) \\ W_{DCPA}t_{VS(DCPA)} \oplus W_{TCPA}t_{VS(TCPA)} \oplus W_D t_{VS(D)} \oplus W_B t_{VS(B)} \oplus W_K t_{VS(K)} \end{bmatrix}.$$

Here,  $\oplus$  is a hazard synthesis operator. According to the actual operation requirements of collision avoidance at sea, the synthesis operator is regulated as follows:

$$\alpha \oplus \beta = \min(1, \alpha + \beta). \quad (17)$$

4.2.3. *Calculation of Ship Collision Risk*. Literature [34] sets the size of the collision risk of the target ship relative to the own ship as a set  $[-2, -1, 0, 1, 2]$ . It is divided into five dangers: low danger, low danger, normal danger, high danger, and high danger, calculated by weighting the vector

$R$  by its weight value. Table 2 shows the relevant data and the influencing factors (the first is the own ship data) of the own ship and the 4 experimental ships at that moment. Figure 4 shows the hazard curves of the ship and the four experimental ships when the steering angle of the ship is limited within the range  $[-35, 35]$ . The horizontal axis is the number of steering angles of the ship. Negative values represent counterclockwise steering.

From Figure 9 and Table 2, it can be seen that when own ship is at the initial heading (the horizontal axis of the image is 0), the danger is first ship > second ship > third ship > fourth ship, and as the ship's clockwise steering angle

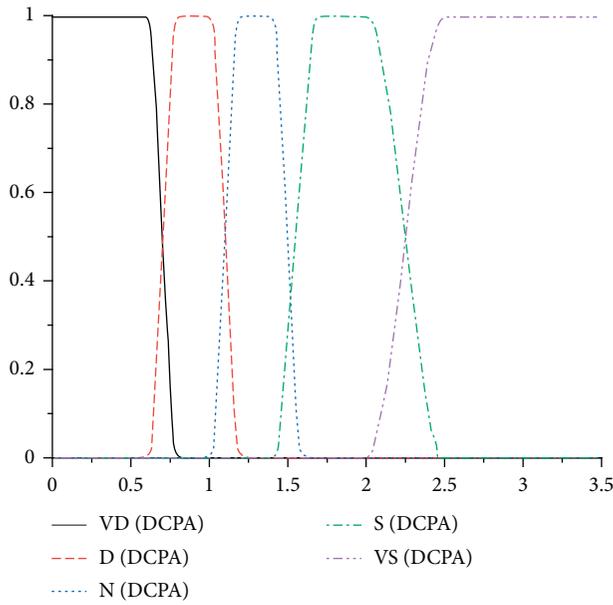


FIGURE 4: DCPA membership function image.

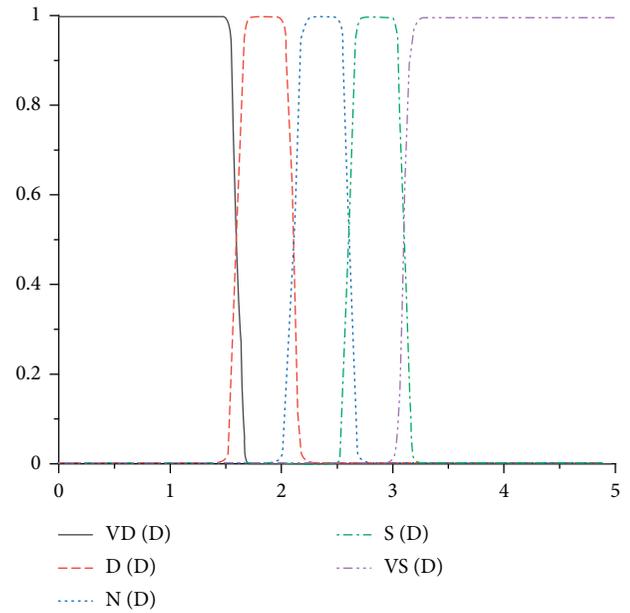


FIGURE 6: *D* membership function image.

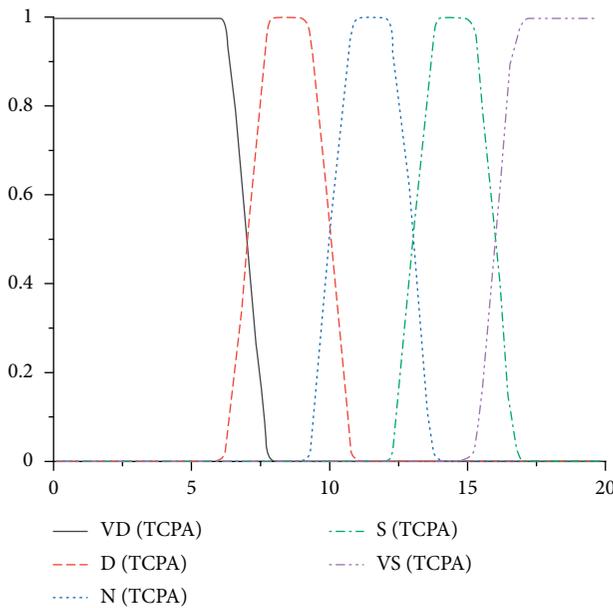


FIGURE 5: TCPA membership function image.

increases, the danger of the second ship becomes higher, making it more prone to collision risks.

**4.3. Ship Steering Angle Decision.** By analyzing the collision risk curve of the ship in Figure 4 and following the rule that the ship will always avoid the most dangerous ship each time when avoiding a collision, the lower the collision risk of the ship in the curve, the more likely it is to collide. We analyze the collision risk curve of the ship, take the lowest value of the ship's danger degree corresponding to each steering angle, and record the name (serial number) of the ship with

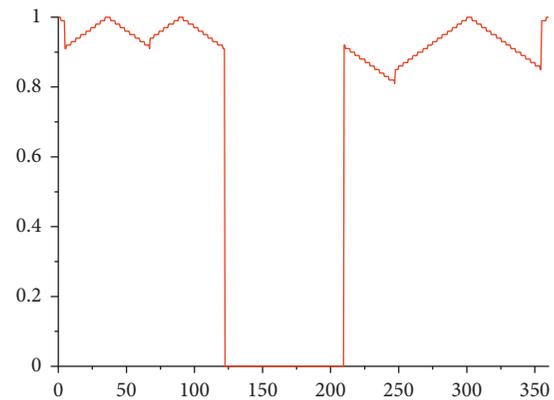


FIGURE 7: *B* membership function image.

the lowest collision danger when the steering angle is zero so as to analyze the current situation of the ship corresponding to the most dangerous target ship situation and determine the direction of the ship's steering angle.

The encounter situation of ships is determined in accordance with the applicable provisions of the rules for avoiding collisions. The rules for avoiding collisions are the basic guidelines for the safe navigation of ships at sea, and they are also an important legal basis for regulating ship avoidance actions. The rules clearly stipulate the methods of avoidance of ships in various situations and play an important role in guiding the safe navigation and avoidance of ships. The encounter situation is mainly divided into two types: first, under the provisions of the collision avoidance rules, the meeting situation is divided according to the possible collision avoidance actions of the crew and the degree of collision avoidance difficulty. The second is from the angle of collision avoidance action (left and right collision avoidance) and

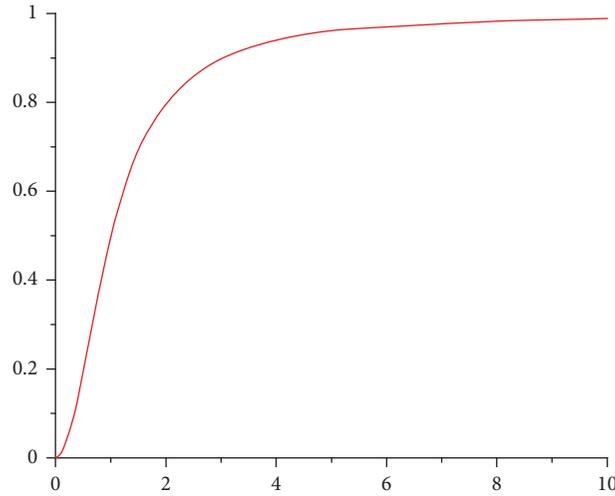


FIGURE 8: Ship speed ratio membership function image.

TABLE 1: Criteria for risk evaluation of various factors.

Factor	Low risk	Low risk	Average risk	High risk	High risk
DCPA (nmile)	2.5 ~ 4.0	1.2 ~ 2.5	1.0 ~ 1.6	0.6 ~ 1.2	0.6 ~ 0.8
TCPA (min)	18	15	12	9	6
D (nmile)	4.0	3.0	2.5	2.0	1.5
B (°)	301.75°	180°	90°	36.25°	0°
K	< 0.5	0.5 ~ 0.8	0.8 ~ 1.2	1.2 ~ 2	> 2

TABLE 2: Related data of own ship and other ships.

Longitude	Latitude	Course	Speed	Bearing	Relative bearing	Distance	Relative speed heading	Relative	DCPA	TCPA	Collision risk
23	30	0	50	—	—	—	—	—	—	—	—
23.07	30.02	220	60	71.72	71.72	4.33	201.89	103.42	-3.31	1.62	-0.077
22.88	30.01	310	60	275.52	275.52	7.23	256.03	47.36	2.41	8.63	0.057
23.05	29.98	260	60	114.78	114.78	3.18	224.36	84.51	-2.99	0.75	0.28
22.87	29.97	40	60	255.11	255.11	7.97	95.98	38.78	2.84	11.52	1.17

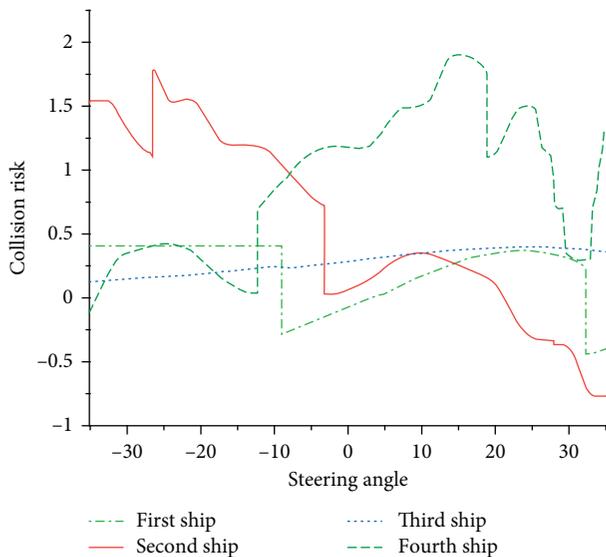


FIGURE 9: Curve of ship collision risk.

according to the relative azimuth (side angle) of the target ship [34]. Reference [28] divides three kinds of ship encounter situations, including 16 kinds of ship encounter situations, based on the relative position relationship between the own ship and the target ship, the avoidance responsibility, and the steering avoidance action method and good visibility. In this paper, by collecting the actual control experience of the master and crew and combining with the provisions of the International Regulations for Preventing Collisions at Sea in 1972, the specific directions of the ship's steering angles for the overtaking ship, the giving way ship, and the ship in the encounter are given.

Rule 13 of the Collision Avoidance Rule: “when a ship is catching up with other ships from a certain direction that is more than 22.5 degrees behind the other ship, that is, the position of the ship it is chasing, it can only see the taillight of the overtaken ship at night, but not the other. Any of its side lights should be considered to be overtaking.” According to the rules of collision avoidance and actual navigational experience, when the ship is an overtaking

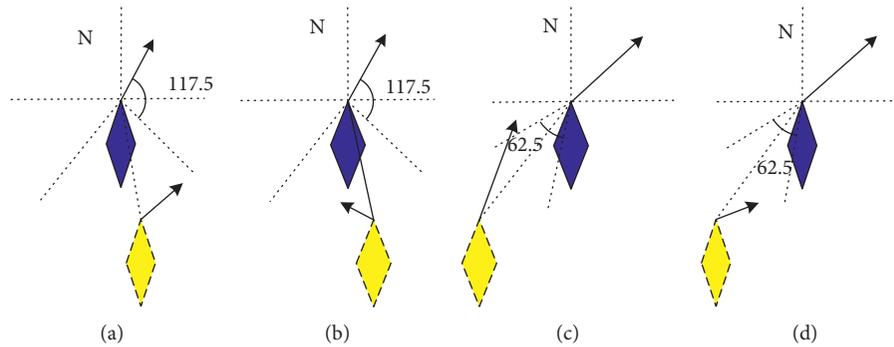


FIGURE 10: Overtaking situation (own ship is a direct ship).

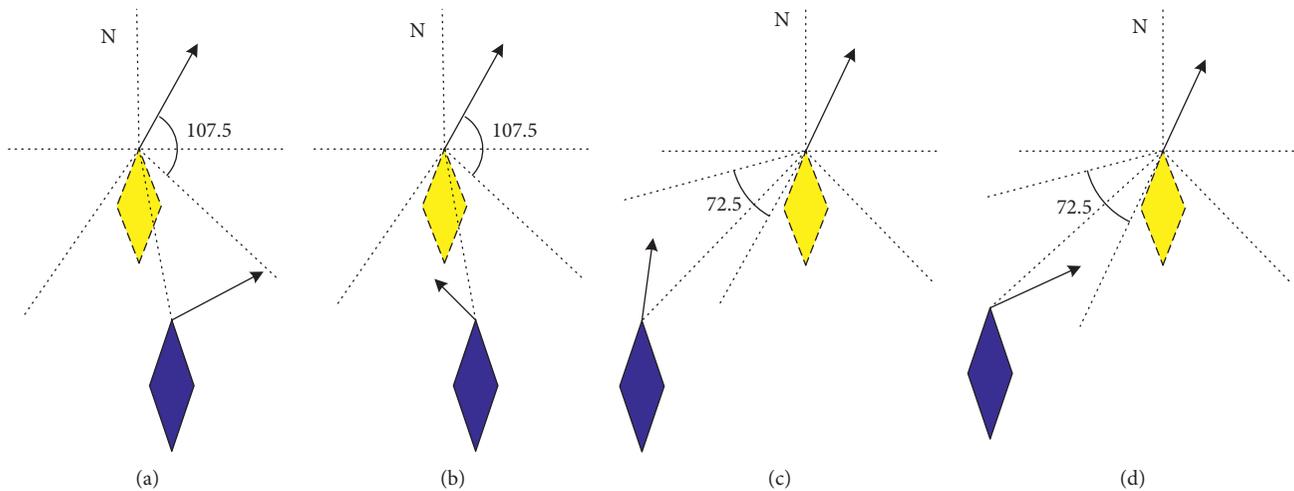


FIGURE 11: Overtaking situation (own ship is overtaking ship).

ship, the crew usually practices the rudder to the left. Regardless of any change in the orientation between the two ships, until the responsibility of the overtaken ship is relieved and the ship passes by to clear it, the situation is shown in Figures 10 and 11, where Figure 10 is a direct ship and Figure 11 is an overpass. In the following picture, the blue solid line is the main ship and the yellow dotted line is the target ship.

Rule 14 of the Collision Avoidance Rule: “when two motor ships meet in opposite or near opposite headings and there is a danger of collision, they each turn to the right so that they each pass through the port side of other ships”; according to collision avoidance, the crew members should normally steer the rudder to the right, and the situation is shown in Figure 12.

Rule 15 of the Collision Avoidance Rule: “when two motor boats cross each other and there is a danger of collision, a ship with other ships on the starboard side of the ship should make way for other ships. If the environment permits at the time, they should also avoid crossing the front of other ships.” According to the rules of collision avoidance and actual navigational experience, when the ship is a yielding ship, the crew usually practices the rudder to the right until the responsibility of the

yielding ship is relieved. The situation is shown in Figures 13 and 14. Among them, the first two small pictures in Figures 13 and 14 limit the speed ratio of own ship to the target ship.

According to the literature [35], the ship’s danger is divided into 5 levels. The minimum value of the collision danger corresponding to the ship at the current moment is obtained through the curve in Figure 5. When the value is less than  $-0.4$ , the collision risk of the ship is more likely to require a collision avoidance decision. At the moment, the meeting situation of the ship with the lowest collision risk is determined. According to the above conclusions, the determination of the ship rudder angle direction is completed.

## 5. Decision Method of Optimal Steering Angle Based on Improved Cultural Particle Swarm

After finding the direction of ship’s steering angle under the collision avoidance rules in Section 3, this section will find the optimal steering angle of the ship under this direction to achieve ship collision avoidance. In order to be more in line with the actual sailing environment of marine vessels and obtain more accurate results, this paper uses the CPSO (cultural particle swarm optimization) algorithm to establish

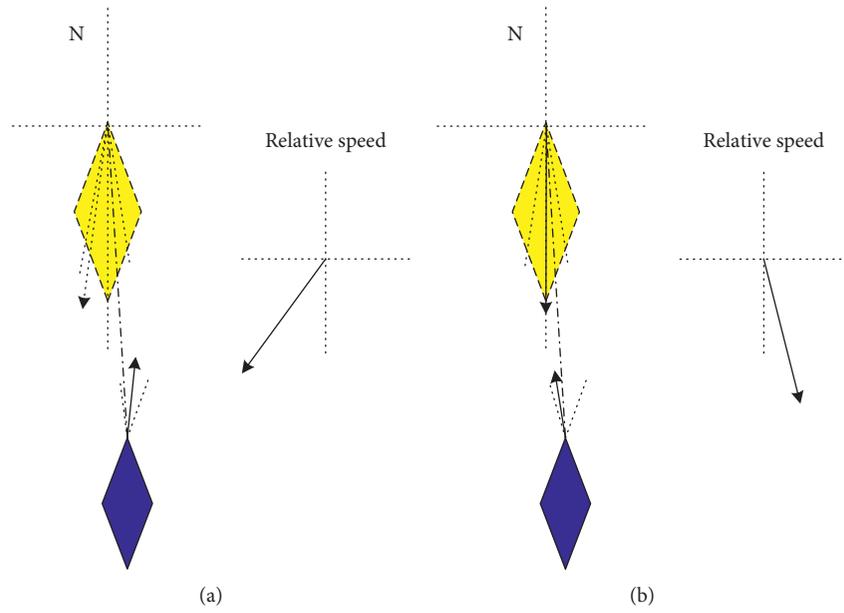


FIGURE 12: Meeting situation.

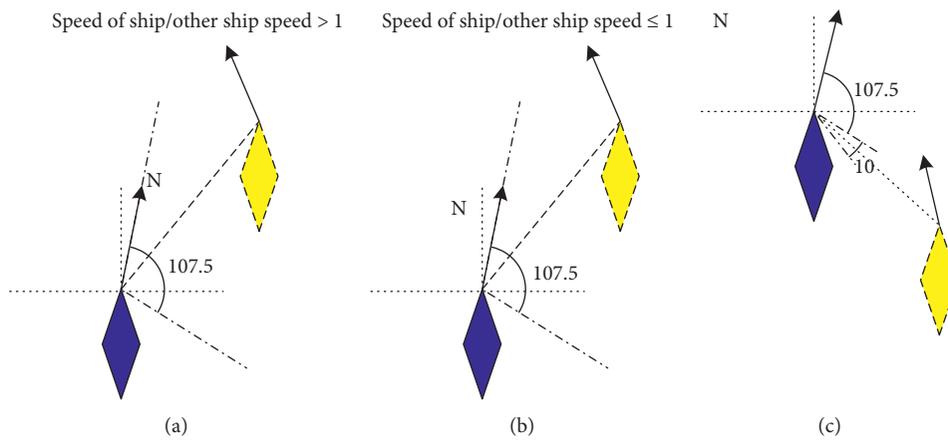


FIGURE 13: Crossing situation (own ship is a give way ship).

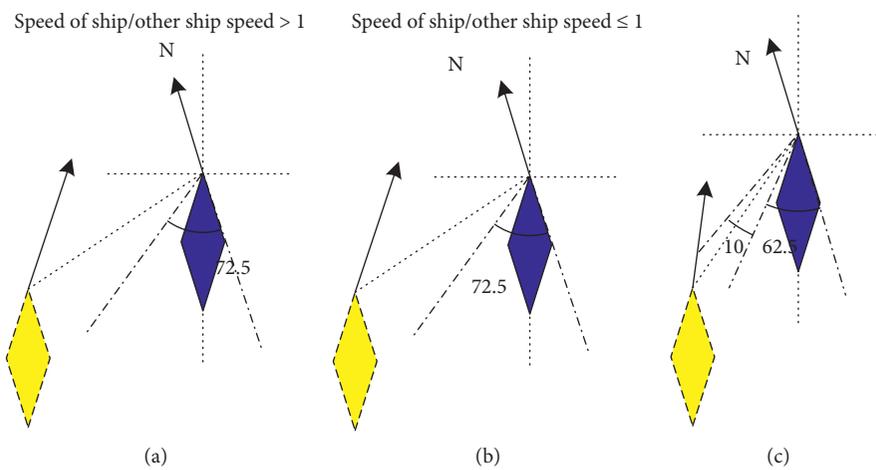


FIGURE 14: Cross situation (own ship is a direct ship).

the main swarm space (lower space) and knowledge space (upper space) of the particle swarm and improve the evolution efficiency through the interaction between the two levels so as to find the optimal steering angle of the ship more effectively.

### 5.1. Cultural Particle Swarm Algorithm

**5.1.1. Cultural Algorithm.** Cultural algorithm (CA) is a new global optimization search algorithm created by Reynolds et al. to solve complex calculations in 1994. Researchers [36] see culture as a carrier of information that potentially affects all members of society and is useful in guiding the evolutionary process of contemporaries and offspring. Cultural algorithm is a two-layer evolution system based on knowledge which contains two evolution spaces: one is a belief space composed of experience and knowledge acquired during the evolution process and the other one is the population space composed of specific individuals, which is solved iteratively through evolutionary operations and performance evaluation. The basic framework of the cultural algorithm is shown in Figure 15.

As shown in Figure 15, population space and belief space are two relatively independent evolutionary processes. The two spaces are linked by a set of communication protocols consisting of acceptor function Accept and influence function Influence. During the evolution process of individuals in the population space, individual experiences are formed, and the individual experiences are transferred to the belief space through Accept functions. Belief space compares and optimizes individual experience according to certain behavior rules to form population experience. Belief space is updated with Update functions based on existing population experience and new individual experience; the Influence function can use the empirical knowledge of the problem to be solved in the belief space to guide the evolution of the population space so that the population space can obtain higher evolution efficiency. In a cultural system, there are many kinds of knowledge, among which normative knowledge and situational knowledge are regarded as the most important knowledge. Normative knowledge provides behavioral norms and guiding principles for individuals, and situational knowledge provides examples for individuals. They can provide guidance information for the evolution of populations. objective function is the objective function (fitness function). Its function is to evaluate the fitness value of individuals in the population space. generate function generates the next generation of individuals based on the rules of individual behavior and the parameters of the parents. The select function selects a part of the newly generated individuals as the parents of the next generation of individuals according to the rules (see Algorithm 2).

**5.1.2. Particle Swarm Algorithm.** The term “group intelligence” was first coined by Beni et al. [37]. It was proposed in the research of cellular robot systems. Typical swarm intelligence consists of a group of simple agents with local influence and their environment. Although there is no

centralized control structure in the swarm intelligence system to guide the behavior of these agents, local influences can often generate global behavior. Particle swarm optimization, as a typical swarm intelligence optimization algorithm, originates from the research on the simplified social model of bird swarms and the simulation of behaviors [38].

Suppose each particle in the particle swarm algorithm is flying at a certain speed in the  $n$ -dimensional search space;  $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$  is particles  $i$ 's position,  $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$  is particles  $i$ 's speed, and  $pbest_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{in})$  is optimal position experienced by particles  $i$ . The current optimal position, global optimal position, particle moving speed, and position update formula are shown in the following equation:

$$\begin{cases} pbest_i(t+1) = \begin{cases} pbest_i(t), & \text{if } f(X_i(t+1)) \geq f(pbest_i(t)), \\ X_i(t+1), & \text{if } f(X_i(t+1)) < f(pbest_i(t)), \end{cases} \\ gbest(t) = \min\{f(pbest_1(t)), f(pbest_2(t)), \dots, f(pbest_N(t)), \\ v_{ij}(t+1) = wv_{ij}(t) + c_1r_1(pbest_{ij}(t) - x_{ij}(t)) + c_2r_2(gbest_j(t) - x_{ij}(t)), \\ x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \end{cases} \quad (18)$$

where  $gbest(t)$  is the global optimal position,  $w$  is the inertia weight,  $i$  is the first  $i$  particles,  $j$  is particle dimension,  $c_1, c_2$  are the acceleration factors, and  $r_1, r_2$  are random numbers among  $[0, 1]$ . Particle swarm algorithm speed update equation consists of three parts: the first part is the previous speed of the particle; the second part is called the “cognitive” part, which comes from the experience and thinking of the particle itself; the acceleration factor  $c_1$  can adjust the flight step size of the particle's best position; the third part is the “social” part of the particle, indicating information sharing and interaction between particles and acceleration factors  $c_2$ . You can adjust the particle's flight step to the optimal position of the group.

**5.1.3. Cultural Particle Swarm Algorithm.** The CA algorithm framework provides a computational model of multiple evolutionary processes, so from the perspective of the computational model, any evolutionary algorithm that meets the requirements of a cultural algorithm can be embedded in the cultural algorithm framework as an evolutionary process of population space. The current research studies mainly focus on the group intelligence algorithms such as GA, POS, and ACO. Unlike the existing evolutionary algorithms, CA provides a dual evolution mechanism. Based on the group space evolutionary algorithm, through the acquisition, preservation, analysis, and integration of the empirical knowledge generated by individual evolution, the evolution process of the population space is further guided so that the evolution speed of the population surpasses the evolution speed solely relying on ordinary evolution algorithms and has good global optimization performance.

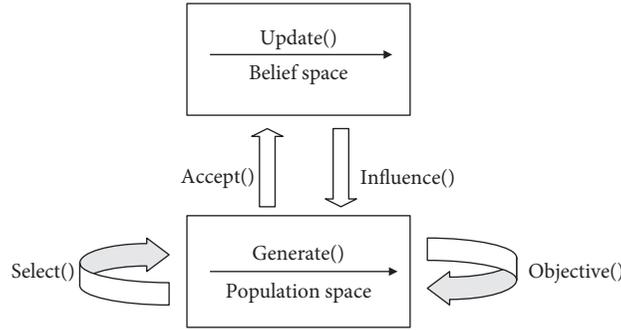


FIGURE 15: Graphical representation of the cultural algorithm.

```

(1) begin
(2)  t = 0;
(3)  POP(t); //Initialize the population space
(4)  BLF(t); //Initialize the belief space
(5)  repeat
(6)  POP(t); //Assessing population space
(7)  update (BLF(t), accept (POP(t)));
(8)  generate (POP(t), influence (BLF(t)));
(9)  t = t + 1;
(10) select POP(t) from POP(t-1);
(11) until Termination condition
(12) end

```

ALGORITHM 2: Cultural algorithm pseudocode.

The essence of cultural particle swarm optimization algorithm is a combination of cultural algorithm model and particle swarm optimization. During the PSO evolution process, particles track two “extreme values”: one is the individual historical optimal solution, the other is the global historical optimal solution, and the global historical optimal solution is passed from generation to generation, which naturally forms the trajectory of the global optimal solution. These optimal solutions are regarded as the global knowledge information of iterative search which is kept continuously, and the knowledge space is updated; the knowledge solution is evolved. At the same time, the further solution of PSO group is guided. Through the dual evolution and mutual influence of PSO space (main group space) and knowledge space, the solution has better global search ability.

## 5.2. Improvements to the Cultural Particle Swarm Algorithm

**5.2.1. Premature Convergence Judgment.** The diversity of the particle swarm population is a prerequisite for the effective operation of the algorithm, and the greater the population diversity, the more likely it is to produce better offspring individuals. However, due to the solution process of the particle swarm algorithm, the diversity will be gradually lost, and the individuals will converge with each other. The particle with the optimal current position will guide other particles to approach it quickly. If the position of the current

optimal particle is a local optimal solution, the entire group cannot search again and can fall into a precocious convergence state. Generally, the fitness of different distributed particles in the population space is different. The degree of diversity of particles can be expressed by the degree of dispersion of fitness distribution of particles. When the PSO algorithm is in the state of premature convergence or global convergence, particles will gather in one place or in several specific places. Therefore, the state of the particle swarm can be judged by studying the change of the particle fitness value in the population space, thereby determining whether the algorithm converges.

This paper introduces the index proposed in [34] to evaluate the premature convergence of the particle swarm, which is defined as: Let the size of the particle swarm be  $n$ . In the  $k$ -th iteration, the fitness value of the particle is  $f_i^k$  and the average fitness value of the particle swarm is  $f_{avg}^k$ ;  $\overline{f_{avg}^k}$  is obtained by averaging the fitness values of particles whose fitness value is better than  $f_{avg}^k$ , which is defined as follows:

$$\rho = \left| \frac{f_m^k - \overline{f_{avg}^k}}{f_m^{k-1} - \overline{f_{avg}^{k-1}}} \% \right|, \quad (19)$$

where  $\rho$  evaluates the degree of convergence of the particle swarm algorithm; the smaller  $\rho$  is, the more the particle swarm tends to converge.

**5.2.2. Dynamic Adjustment of Inertia Weight.** The standard particle swarm algorithm uses linearly decreasing inertial weights because the linearly decreasing inertial weight changes are too singular, and the process of searching the optimal steering angle of a ship by the particle swarm algorithm is very complicated and nonlinear; it is suitable for complex problems. Both ability and regulation ability are very limited. Aiming at the above problems, this paper proposes a strategy for dynamically adjusting the inertia weights, combining the changes of the inertia weights with indicators to evaluate the degree of premature convergence of the particle swarm:

$$w_i(t) = 1 - (1 + e^{-\rho})^{-1}. \quad (20)$$

It can be seen from the above formula that the particle's inertia weight value changes within  $(0, 1)$ ,  $w_i(t)$  describes the effect of the inertia of the  $i$ -th particle on the  $t$ -th velocity, and the value can adjust the global optimization ability of the PSO and local optimization capabilities. When the PSO is in a state of convergence, a large inertia weight is required to increase the search step size, thereby enhancing the global optimization capability of the PSO. When the PSO is in the global search state, a small inertia weight is required to reduce the search step size, thereby enhancing the local optimization capability of the PSO. When the index for evaluating the degree of premature convergence of the particle swarm decreases,  $w_i(t)$  increases accordingly, which enhances the global search ability; when  $\rho$  increases, the value of  $w_i(t)$  decreases accordingly, increasing the local search ability.

### 5.3. Optimal Steering Angle Decision Algorithm

#### 5.3.1. Particle Swarm Space Design and Iteration Rules

- (1) **Coding Scheme.** The particle swarm in the population space encodes the particle swarm according to the Michigan coding scheme, that is, each particle represents a ship collision avoidance strategy, and all the collision avoidance strategies correspond to a certain particle swarm, the particles are composed of different dimensions, and the dimensions of the particles are correspondingly related to the feature term; this article only searches for the ship's collision avoidance steering angle and does not involve other feature items such as ship speed. Therefore, the steering angle is expressed as coordinates  $x_i$ ,  $i = 1, 2, \dots, N$ ; then, this particle swarm individual is encoded as  $x_1 x_2 x_3 \dots x_N$ .
- (2) **Fitness Function.** Determine the fitness function of the individual particle swarm based on the collision risk of the ship and set the number of ships that need

to be avoided at present as  $M$ ; the ship collision risk function is CollisionRisk, and the value of the fitness function is mainly related to the collision risk of the currently most dangerous ships, so its fitness function is expressed as

$$\text{objective}(\theta) = \text{Min}(\text{CollisionRisk}(i, \theta)), \quad i \in M. \quad (21)$$

Among them,  $\theta$  represents the size of the steering angle, positive represents the clockwise direction, and negative represents the counterclockwise direction. According to the collision risk curve of the ship in Figure 9 of this paper, the fitness function curve at that point in time is obtained, as shown in Figure 16.

**5.3.2. Design and Update Rules of Belief Space.** The structure of the belief space uses the structure pair  $\langle S, N \rangle$  proposed in article [39], where  $S = \{s^t, s_2^t, \dots, s_m^t\}$  is situational knowledge, which represents the optimal individual set,  $s_i^t$  represents the  $i$ -th optimal individual in the  $t$  generation population, and  $m$  represents the scale of the optimal individual set. This article only takes the current best individual  $s^t$  to update the situation knowledge  $S$  in belief space, which is  $S = \{s^t\}$ , and searching for the optimal steering angle of the ship is to find the maximum value  $\text{Max}(\text{fitness}(\theta))$  of the fitness function, so the update formula is as follows:

$$s^{t+1} = \begin{cases} x_{\text{best}}^t, & \text{objective}(x_{\text{best}}^t) > \text{objective}(s^t), \\ s^t, & \text{otherwise.} \end{cases} \quad (22)$$

Among them,  $x_{\text{best}}^t$  represents the optimal particle in the  $t$  generation. In the current population, if the optimal particle  $x_{\text{best}}$  is better than the existing one  $s^t$ , then the knowledge of the belief space situation is adjusted and updated by replacing  $x_{\text{best}}$  and  $s^t$ .

$N = \langle X_1, X_2, \dots, X_n \rangle$  is standardized knowledge; it represents the value interval information of each variable.  $n$  is the number of variables, and  $X_i$  is  $\langle I, L, U \rangle$ , where  $I$  is defined as

$$I = [l, u] = \{x | l \leq x \leq u\}. \quad (23)$$

Among them,  $l$  and  $u$  represent the lower and upper boundaries of the particle steering angle decision, respectively, and  $L$  and  $U$  represent the fitness values corresponding to these two boundaries. In  $t$ -th iterations of the particle swarm, it is assumed that the steering angle parameter of  $i$ -th particle affects its lower boundary  $l_i^t$  and its fitness value  $L_i^t$ . Similarly, the steering angle parameter of the  $k$ -th particle affects its upper boundary  $u_k^t$  and its fitness value  $U_k^t$ . For the parameter update, see equation (24).

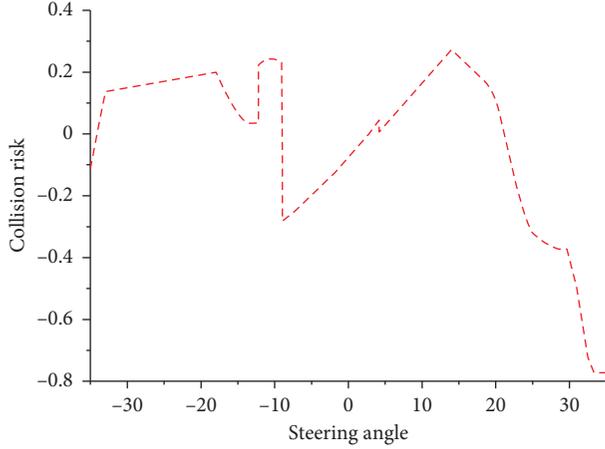


FIGURE 16: Fitness function curve.

$$\begin{aligned}
 l_i^{t+1} &= \begin{cases} x_i^t, & x_i^t \leq l_i^t \text{ or } \text{objective}(x_i^t) > L_i^t, \\ l_i^t, & \text{otherwise,} \end{cases} \\
 L_i^{t+1} &= \begin{cases} \text{objective}(x_i^t), & x_i^t \leq l_i^t \text{ or } \text{objective}(x_i^t) > L_i^t, \\ L_i^t, & \text{otherwise,} \end{cases} \\
 u_k^{t+1} &= \begin{cases} x_k^t, & x_k^t \geq u_k^t \text{ or } \text{objective}(x_k^t) > U_k^t, \\ u_k^t, & \text{otherwise,} \end{cases} \\
 U_k^{t+1} &= \begin{cases} \text{objective}(x_k^t), & x_k^t \geq u_k^t \text{ or } \text{objective}(x_k^t) > U_k^t, \\ U_k^t, & \text{otherwise.} \end{cases}
 \end{aligned} \quad (24)$$

5.3.3. *Acceptance Function and Influence Function.* This article uses a dynamic acceptance function, which introduces evolutionary algebra as a dynamic factor and adjusts the acceptance ratio. The expression is

$$\eta_{\text{accept}} = \left( p\% + \frac{p\%}{t} \right) * n. \quad (25)$$

Among them,  $n$  is the size of the population,  $t$  is the current evolutionary algebra, and  $p\%$  is a predetermined fixed ratio. The belief space can affect the mutation operator in the population space in two ways: firstly, changing the step size of the variable change, and secondly, changing the direction of the variable. In this section, normative knowledge is used to adjust the variable step size, and situational knowledge is used to adjust its forward direction. The specific expression is as follows:

$$x_i^{t+1} = \begin{cases} x_i^t + \left| \text{size}(I_i^t) * N(-1, 1) \right|, & x_i^t < s_i^t, \\ x_i^t - \left| \text{size}(I_i^t) * N(-1, 1) \right|, & x_i^t > s_i^t, \\ x_i^t + \text{size}(I_i^t) * N(-1, 1), & x_i^t = s_i^t, \end{cases} \quad (26)$$

where  $N(-1, 1)$  is a random number that follows a normal distribution and  $\text{size}(I_i^t)$  is the difference between the upper and lower bounds of the normative knowledge of the particle swarm at the  $t$  iteration. Through normative knowledge and

situational knowledge, offspring individuals move as far as possible to a given interval (see Algorithm 3).

5.4. *Experimental Verification.* The experiment uses the data of own ship and related ships in Table 2 of Section 4, taking the total number of particles  $n = 30$ , the maximum number of iterations  $I = 1000$ , acceleration factor  $c_1 = c_2 = 2$ , ship steering angle whose range is  $[-35, 35]$ , the fitness function which is objective  $(\theta) = \text{Min}(\text{CollisionRisk}(i, \theta))$ , where  $i$  is the number of vessels to avoid, and acceptance ratio = 1.5; the data of the first 40 iterations of the experiment are shown in Figure 17.

Figure 17 shows that in the cultural particle swarm curve, according to the situational knowledge and normative knowledge, the rate of particles toward the global optimal solution is gradually increased with the increase of the number of population iterations. At the 9-th iteration, the global optimal particle position is found, and the fitness value is 0.268394533746147 (see Figure 18). When the particle iterates to the 8-th time, the particle position exceeds the global optimal solution, but according to the knowledge and particle swarm velocity update formula, the particle quickly returns to the global optimal solution position again, which indicates that this algorithm has good global search capabilities.

In order to analyze the global search ability and convergence performance of the cultural particle swarm algorithm in this section, a comparison experiment is performed between the algorithm in this paper and the standard PSO, that is, the PSO algorithm, GA (genetic) algorithm, and AFSA (fish school) algorithm. The results are shown in Figure 19.

Figure 19 shows the display of the four algorithms for finding the optimal steering angle of the ship. It can be seen from the figure that after 40 iterations, the CPSO (cultural particle swarm optimization) algorithm adopted in this paper is better than the standard PSO, GA, and AFSA algorithms. This is because the belief space of the cultural particle swarm algorithm uses the influence function to guide the population space, enhances the diversity of particles in the population space, and uses a dynamic inertia weight adjustment strategy to adaptively adjust weight according to the degree of population convergence, making the algorithm have a strong global optimizing ability.

The standard PSO algorithm reached the optimal solution the 24-th iteration, but because the linearly decreasing inertial weight is used, the inertial weight change is too single, and the process of finding the optimal steering angle of the ship is nonlinear and does not consider the population iteration information. The adjustment ability of the standard PSO algorithm is limited, so the algorithm has decreased compared with the cultural particle swarm algorithm. The GA and AFSA algorithms reached the optimal solution in the 26-th and 16-th generations, respectively, but because there is no influence function to guide the population space and it has a certain degree of blindness to the population iteration, the convergence speed has decreased compared to the cultural particle swarm algorithm.

- (1) Initialize particle population size  $N$ , dimension 1, iterations item, steering angle search space  $[-x_{\max}, x_{\max}]$ , particle maximum velocity  $v_{\max}$ .
- (2) Initialize the population space, particle initial position  $x$ , initial velocity  $v$ , initial  $\langle S, N \rangle$ .
- (3) Calculate the objective value of population space particles, particle' current optimal position  $pbest_i$ , global optimal position  $gbest$ .
- (4) Update inertia weights according to formula (21), and equation (19) updates particle velocity and position, by  $x_i$  update  $pbest_i$ ,  $gbest$ .
- (5) Replace the least fit particles in the belief space with the best fit particles in the population space based on acceptance function and update the position of the population space particles by the influence function.
- (6) Determine whether the maximum number of iterations item has been reached; if not, go to step 4; otherwise, go to step 7.
- (7) Output  $gbest$ , algorithm ends.

ALGORITHM 3: Optimal steering angle decision algorithm based on cultural particle swarm.

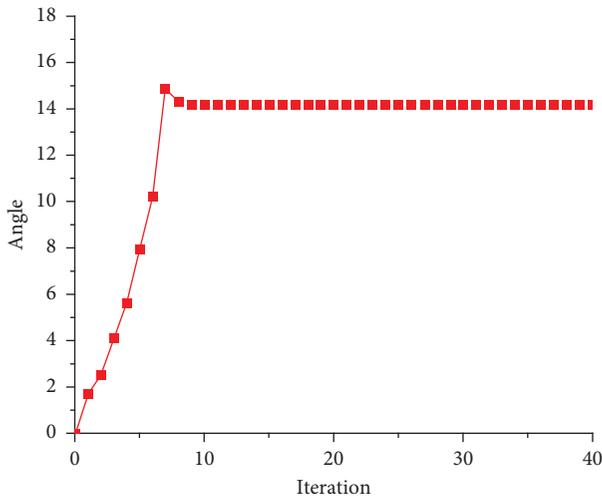


FIGURE 17: Cultural particle swarm curve.

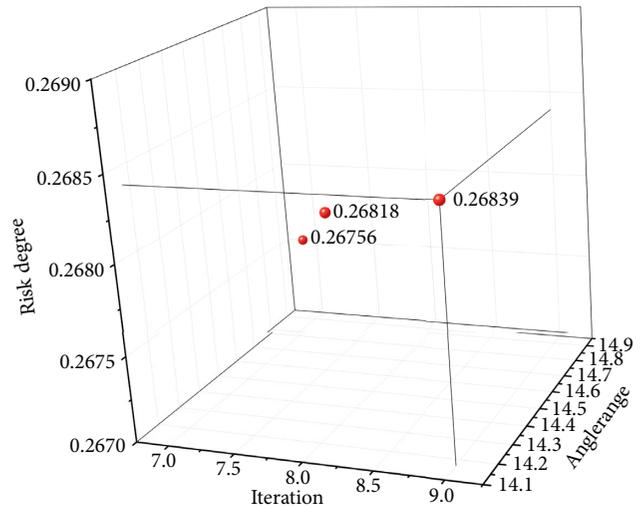


FIGURE 18: Algorithm's three-dimensional figure (7-9 generations).

## 6. Electronic Chart Platform Integration and Experimental Analysis

We integrate the method proposed in this paper into the electronic chart platform. The electronic chart platform is used as the verification environment of the method. At the same time, the AIS data of the ship are displayed on the electronic chart platform and provided to the related algorithm. Electronic chart platform is an important navigation tool chart for ships and other marine vehicles, which can provide true and complete environmental information required during navigation, mainly including land, ocean, water depth, obstacles, and islands. So, it is widely used in ocean path planning and other aspects.

In this paper, the international standard electronic chart platform developed in C++ language is used as the experimental verification environment, as shown in Figure 20. The system has the following characteristics:

- (1) The system uses international standards to display chart information, and we can zoom and observe the chart interface at any scale.
- (2) The system uses the MFC (Microsoft Foundation Class) framework to generate a dynamic link library

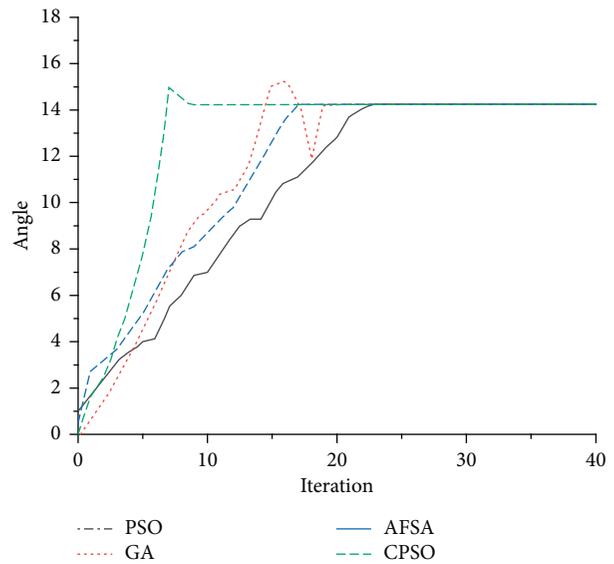


FIGURE 19: Comparative experiment chart.

of related algorithms, providing a flexible and convenient interface design.



FIGURE 20: Electronic chart simulation environment.

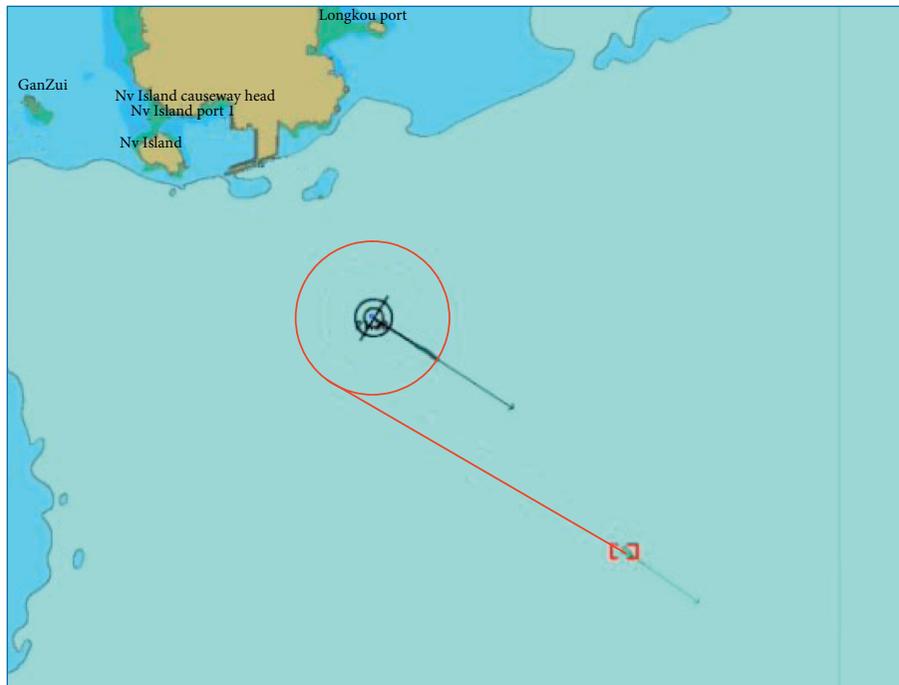


FIGURE 21: Own ship is overtaking ship.



FIGURE 22: Right-steering collision avoidance.

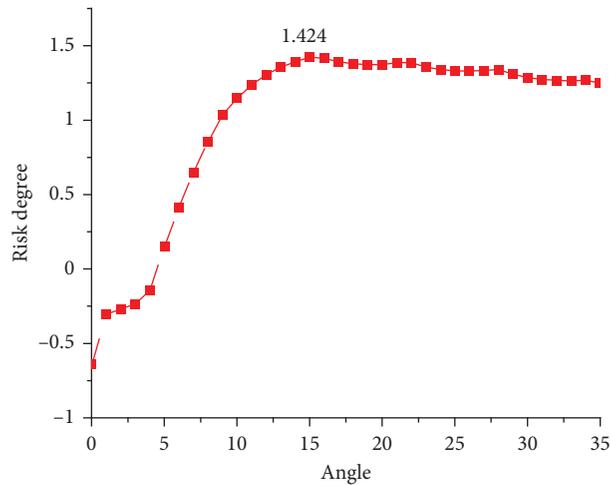


FIGURE 23: The hazard curve of ship collision avoidance described in this scene.

- (3) The system provides operations for autonomously setting the movement information of the current ship and other ships, including information such as longitude, latitude, heading, and speed, and all ship models in the system use bulk carriers.
- (4) It can dynamically display the movement tracks of all ships in the chart, which is convenient to observe the simulation results of real environment. The interface is shown in Figure 20.

6.1. Steering Angle Direction Decision Verification. Figures 21–29 show the ship encounter scenarios, verifying whether the model meets the rules of collision avoidance.

Figures 21 and 22 are the scenes of own ship as overtaking ship. The own ship speed is 15 knots and the target ship speed is 8 knots. Besides that, the specific parameters of the ship are shown in Table 3.

There is a danger of collision in the current scene. The analysis in Section 4.3 shows that the steering should be driven to the left. Figure 23 shows the collision risk curve of the ship depicted in the scene. As the ship steers to the left (the left direction is counterclockwise and the sign is negative), the ship’s collision risk curve gradually increases first, reaching the highest point of the curve at 1.424 at 15.2 degrees, and then it almost stabilized in the 1.42 to 1.20 range. Therefore, the ship steers 15.2 degrees to the left to



FIGURE 24: Own ship is facing the ship.

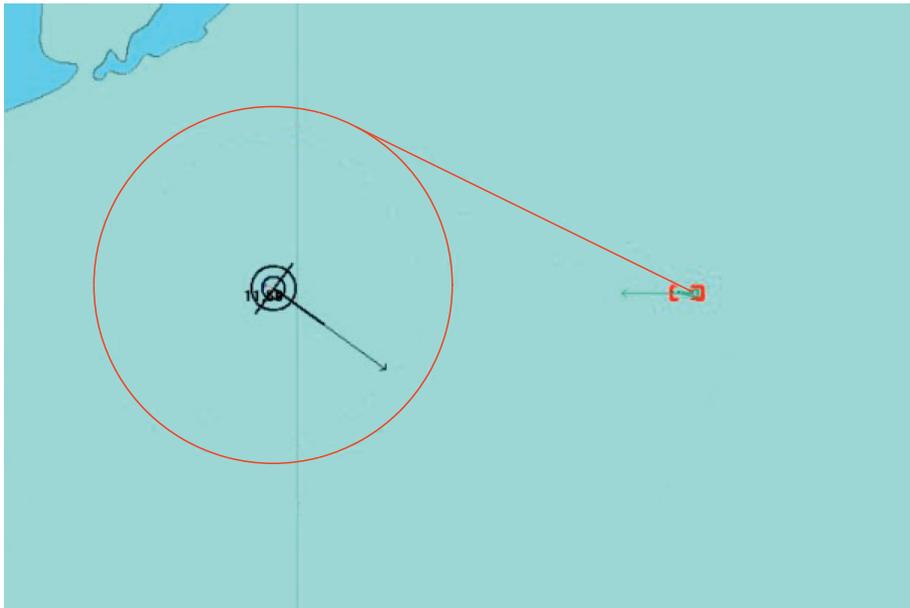


FIGURE 25: Right-steering collision avoidance.

avoid collisions, and the experimental results meet the requirements of the ship collision avoidance rules.

Figures 24 and 25 are the encounter scenes of the ship. The speed of the ship is 15 knots and the speed of the target ship is 8 knots. The specific parameters of the ship are shown in Table 4.

There is a danger of collision in the current scene. The analysis in Section 4.3 above shows that the steering should be driven to the right. Figure 26 shows the collision risk curve of the ship depicted in the scene. As the ship steers to the right (the clockwise direction is positive), the ship's collision risk curve gradually increases first, reaching the

highest point of the curve at 27.8 degrees at 1.126, and then slightly drops to around 1.0. Therefore, the ship steers 27.8 degrees to the right to avoid the collision, and the experimental results meet the requirements of the ship collision avoidance rules.

Figures 27 and 28 are the crossing situation of the ship, and the specific parameters of the ship are shown in Table 5.

There is a danger of collision in the current scene. The analysis in 4.3 above shows that the rudder should be driven to the right. The collision risk curve of own ship depicted in the scene is shown in Figure 29. As the ship steers to the right, the collision risk curve of the ship gradually increases,

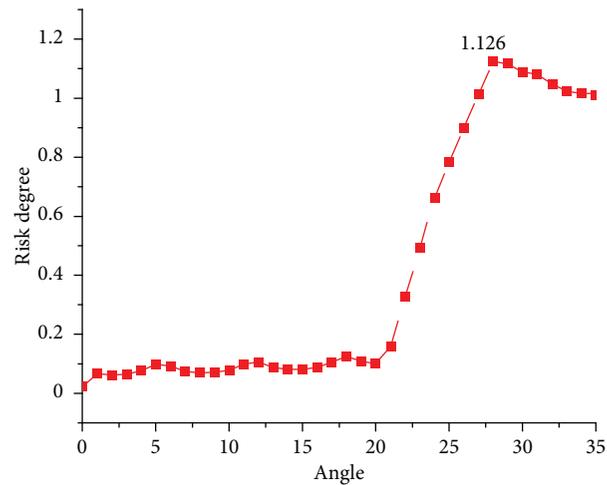


FIGURE 26: The hazard curve of ship collision avoidance described in this scene.



FIGURE 27: Own ship is a give way ship.

reaching the highest point 1.066 of the curve at 35 degrees. Therefore, own ship steers 35 degrees to the right to avoid the collision, and the experimental results meet the requirements of the ship collision avoidance rules.

**6.2. Steering Angle Decision Verification.** We use the ship steering collision avoidance model proposed in this paper to carry out the collision avoidance simulation experiment. The simulation environment diagram is shown in Figure 30. In the simulation initial conditions, the initial course of the ship is 92.0 degrees, the speed is 15 knots, the longitude is 120.87 degrees, and the latitude is 36.35 degrees; Ship-1 is the first target ship, its course is 270.0 degrees, its speed is 7 knots/hour, longitude is 120.87 degrees, and the latitude is 36.35

degrees; Ship-2 is the second target ship, its course is 40.0 degrees, the speed is 8 knots/hour, the longitude is 120.90 degrees, and the latitude is 36.29 degrees; Ship-3 is the third target ship, its course is 190.0 degrees, its speed is 9 knots, longitude is 120.87 degrees, and its latitude is 36.35 degrees.

Figures 31(a) and 31(b) are the situations where the ship and the three target ships will encounter the situation. By analyzing the system log of the collision avoidance module in the scenario in Figure 31, the collision risk change during the collision avoidance process is obtained. The drawn curve is shown in Figure 32.

In the initial scenario, own ship and Ship-1 are facing each other, and there is a danger of collision. The analysis in 4.3 above shows that own ship should steer to the right at this

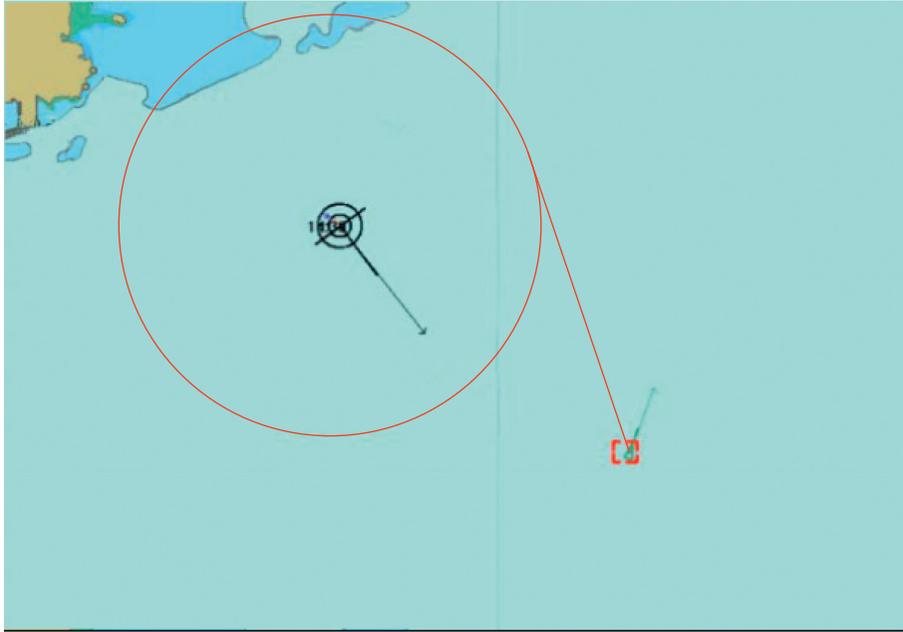


FIGURE 28: Right-steering collision avoidance.

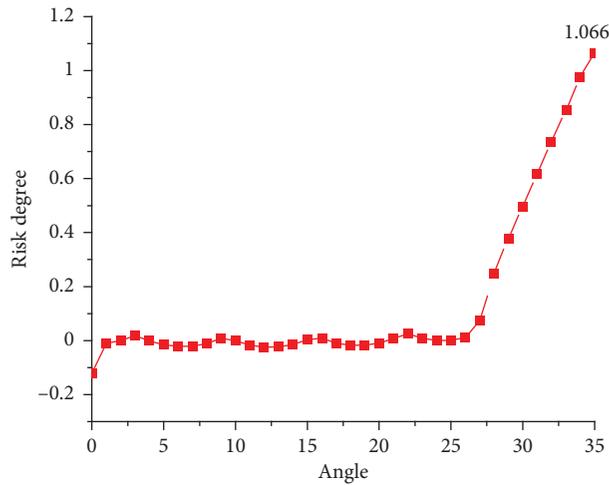


FIGURE 29: The hazard curve of ship collision avoidance described in this scene.

TABLE 3: Relevant data of chase scene experiment.

Related parameters	Value
Position	132.85°
Relative position	10.75°
Relative velocity	7.02 <i>n</i> mile/h
Relative speed heading	298.80°
DCPA	-0.77/ <i>n</i> mile
TCPA	26.32/min
Relative heading	2.90°
Ship spacing	3.48 <i>n</i> mile
Current risk	-0.62

time. Figure 32 shows the ship collision risk curve depicted in the scene. As the ship steers to the right, the collision hazard curve of own ship and Ship-1 gradually increases,

while the hazard curve with Ship-2 gradually decreases. This is because own ship steers to the right, causing the ship to move closer to Ship-2, which increases the risk of collision.

TABLE 4: Relevant data of encounter scene experiment.

Related parameters	Value
Position	91.0°
Relative position	0.70°
Relative velocity	23.0 <i>n</i> mile/h
Relative speed heading	270.20°
DCPA	-0.07/ <i>n</i> mile
TCPA	12.51/min
Ship spacing	2.83 <i>n</i> mile
Current risk	0.02
Relative heading	179.70°

TABLE 5: Cross-encounter scene experimental data.

Related parameters	Value
Position	129.28°
Relative position	38.48°
Relative velocity	14.50 <i>n</i> mile/h
Relative speed heading	302.21°
DCPA	-0.53/ <i>n</i> mile
TCPA	17.76/min
Relative heading	70.80°
Ship spacing	3.23 <i>n</i> mile
Current risk	-1.18

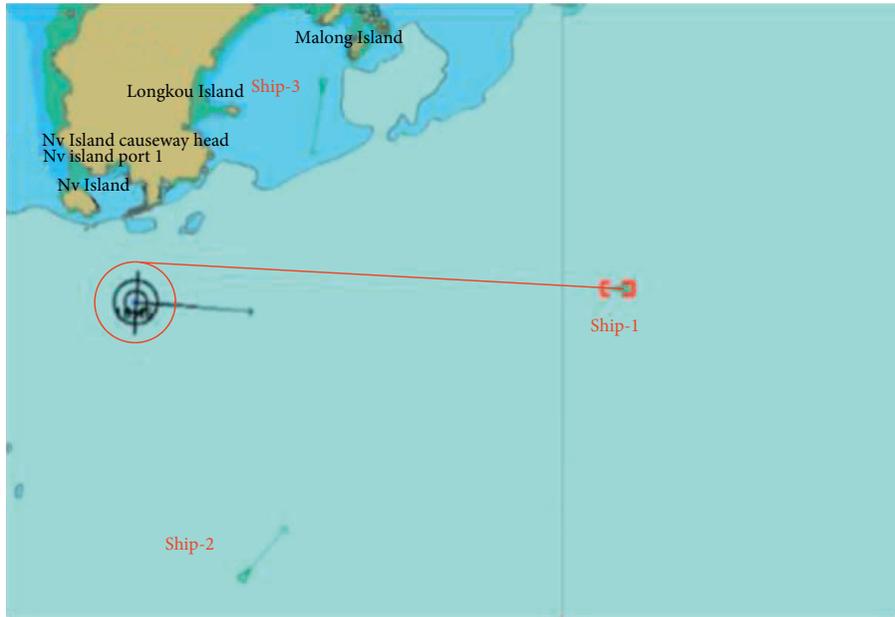


FIGURE 30: Initial simulation environment diagram.

The hazard curve of the own ship and Ship-3 is relatively stable above 1.4, and there is no danger of collision. The hazard curve of Ship-2 and Ship-1 intersects at 22.5 degrees, so own ship steers 22.5 degrees to the right to avoid the collision. The collision avoidance state is shown in Figure 31(a). After evading Ship-1 (the most dangerous ship at present), evade Ship-2. The collision avoidance state is shown in Figure 31(b).

As shown in Figure 33(a), the own ship is ready to take action to resume sailing after avoiding the dangerous target ship and keeping clear. As shown in Figure 33(b), the position of own ship finally returned to the original route. Hence, the results indicate that the own ship can take proper action to avoid other target ships and has a better performance. During the entire experiment, the ship can avoid other target ships reasonably and has a better avoidance effect.

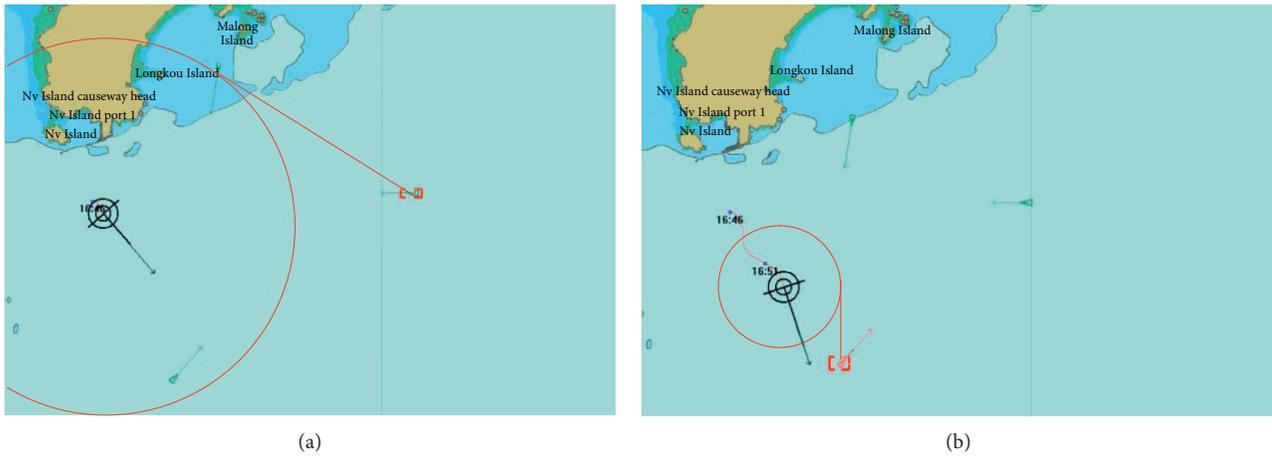


FIGURE 31: The encounter situation between the ship and the three target ships. (a) Avoid target Ship-1. (b). Avoid target Ship-2.

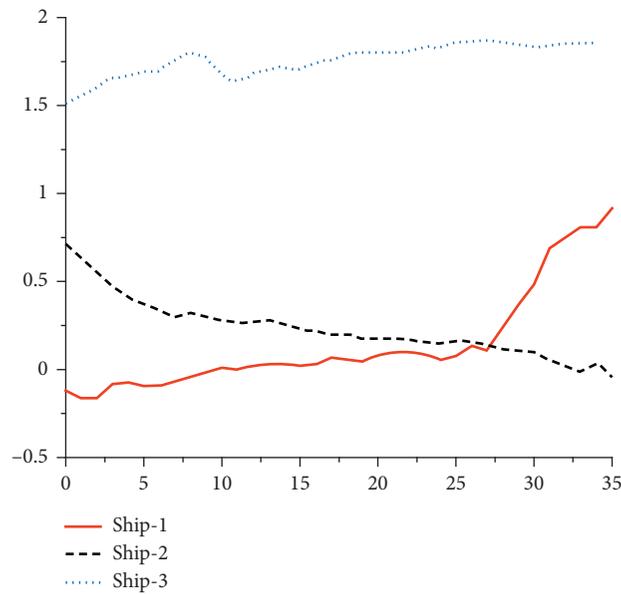


FIGURE 32: Change curve of collision risk during collision avoidance.



FIGURE 33: Picture of the end of own ship avoidance. (a) Location map of own ship after evading Ship-2. (b) End of own ship avoidance.

## 7. Conclusions

In this paper, a decision-making method for ship collision avoidance based on improved cultural particle swarm is proposed. Aiming at the fact that traditional evolutionary algorithms do not use the information of population iteration which leads to the slow adjustment ability and convergence speed of the algorithms, the cultural particle swarm is improved and applied to collision avoidance system in this paper. Firstly, AIS data of ship are acquired based on electronic chart platform. However, in the actual AIS ship data transmission, the update is not timely and the error is large. Hence, the Kalman filter is used to smooth and predict the ship trajectory in this paper. Secondly, based on the fuzzy distribution method, the danger degree of the ship is estimated and the steering angle direction is determined by combining the COLREGs and the encounter status of the ship. Finally, the optimal steering angle of the ship is found by cultural particle swarm optimization algorithm to realize the decision making of ship collision avoidance.

The complex situations of three ships are simulated in electronic chart. The results show that the ship adopts the optimal and reasonable steering angle decision in this encounter scene and realizes the autonomous collision avoidance. The cultural particle swarm optimization algorithm adopted in this paper is compared with other three evolutionary methods. Experiments demonstrate that the cultural particle swarm algorithm has faster convergence speed and higher accuracy and can achieve continuous job output, which further validates the effectiveness of the algorithm. However, the speed decision of ships is not considered in this paper, and there are a large number of collision avoidance scenarios for ships at sea. The test scenarios in this paper are not comprehensive enough. How to consider the decision of ship steering angle and ship speed, as well as to divide and test the scene in a complex encounter situation, is the focus of the next step of this thesis.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This study was supported by the National Key R&D Program of China (grant no. 2018YFB1601502), the National Natural Science Foundation of China (grant no. 51679025), the Liaoning Revitalization Talents Program (grant no. XLYC1902071), and the Fundamental Research Funds for the Central Universities (grant nos. 3132019313 and 3132019354).

## References

- [1] A. Lazarowska, "Decision support system for collision avoidance at sea," *Polish Maritime Research*, vol. 19, pp. 19–24, 2012.
- [2] S. Li, J. Liu, and R. R. Negenborn, "Distributed coordination for collision avoidance of multiple ships considering ship maneuverability," *Ocean Engineering*, vol. 181, pp. 212–226, 2019.
- [3] Y. Huang, L. Chen, P. Chen, R. R. Negenborn, and P. H. A. J. M. van Gelder, "Ship collision avoidance methods: state-of-the-art," *Safety Science*, vol. 121, pp. 451–473, 2020.
- [4] T. Statheros, G. Howells, and K. M. Maier, "Autonomous ship collision avoidance navigation concepts, technologies and techniques," *Journal of Navigation*, vol. 61, no. 1, pp. 129–142, 2008.
- [5] W. Shaobo, Z. Yingjun, and L. Lianbo, "A collision avoidance decision-making system for autonomous ship based on modified velocity obstacle method," *Ocean Engineering*, vol. 215, p. 107910, 2020.
- [6] C. Tam, R. Bucknall, and A. Greig, "Review of collision avoidance and path planning methods for ships in close range encounters," *Journal of Navigation*, vol. 62, no. 3, pp. 455–476, 2009.
- [7] X. Wang, Z. Liu, and Y. Cai, "The ship maneuverability based collision avoidance dynamic support system in close-quarters situation," *Ocean Engineering*, vol. 146, pp. 486–497, 2017.
- [8] H. Chen, M. Wang, and X. Zhao, "A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems," *Applied Mathematics and Computation*, vol. 369, p. 124872, 2020.
- [9] S. Wu, *Research on Improvement and Application of Multi-Objective Particle Swarm Optimization Algorithm*, Jiangnan University, Wuxi, China, 2013.
- [10] G. Yuan, Z. Wei, and Y. Yang, "The global convergence of the Polak-Ribière-Polyak conjugate gradient algorithm under inexact line search for nonconvex functions," *Journal of Computational and Applied Mathematics*, vol. 362, pp. 262–275, 2019.
- [11] J. Guo, W. Yuan, X. Dang, and M. S. Alam, "Cable force optimization of a curved cable-stayed bridge with combined simulated annealing method and cubic B-Spline interpolation curves," *Engineering Structures*, vol. 201, Article ID 109813, 2019.
- [12] A. M. Mohammed and S. O. Duffuaa, "A tabu search based algorithm for the optimal design of multi-objective multi-product supply chain networks," *Expert Systems with Applications*, vol. 140, Article ID 112808, 2020.
- [13] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Computing*, vol. 23, no. 9, pp. 3137–3166, 2019.
- [14] Y. Lu, C. A. Phillips, and M. A. Langston, "A robustness metric for biological data clustering algorithms," *BMC Bioinformatics*, vol. 20, no. 15, pp. 1–8, 2019.
- [15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the ICNN'95-International Conference on Neural Networks*, pp. 1942–1948, IEEE, Perth, Australia, November-December 1995.
- [16] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, pp. 4104–4108, IEEE, Orlando, FL, USA, October 1997.

- [17] G.-C. Luh, C.-Y. Lin, and Y.-S. Lin, "A binary particle swarm optimization for continuum structural topology optimization," *Applied Soft Computing*, vol. 11, no. 2, pp. 2833–2844, 2011.
- [18] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, pp. 1945–1950, IEEE, Washington, DC, USA, July 1999.
- [19] J. Riget and J. S. Vesterstrøm, "A diversity-guided particle swarm optimizer-the ARPSO," Dept. Comput. Sci., Univ. of Aarhus, Aarhus, Denmark, Tech. Rep, 2002.
- [20] M. Daneshyari and G. G. Yen, "Cultural-based multiobjective particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 2, p. 553, 2011.
- [21] X.M. Tao and L. Yang, "Cultural particle swarm optimization algorithm with adaptive guidance," *Computer Engineering and Application*, vol. 47, no. 14, pp. 37–41.
- [22] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3407–3422, 2016.
- [23] Y. Kuwata, M. T. Wolf, D. Zargitsky, and T. L. Huntsberger, "Safe maritime autonomous navigation with COLREGS, using velocity obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, 2013.
- [24] D. Kim, K. Hirayama, and T. Okimoto, "Distributed stochastic search algorithm for multi-ship encounter situations," *Journal of Navigation*, vol. 70, no. 4, pp. 699–718, 2017.
- [25] A. Lazarowska, "Ant colony optimization based navigational decision support system," *Procedia Computer Science*, vol. 35, pp. 1013–1022, 2014.
- [26] H. D. Nguyen, N. V. Do, V. T. Pham, A. Selamat, and E. Herrera-Viedma, "A method for knowledge representation to design intelligent problems solver in mathematics based on rela-ops model," *IEEE Access*, vol. 8, pp. 76991–77012, 2020.
- [27] C. Tam and R. Bucknall, "Path-planning algorithm for ships in close-range encounters," *Journal of Marine Science and Technology*, vol. 15, no. 4, pp. 395–407, 2010.
- [28] D. X. Liu, J. P. Zhang, and F. W. Wang, "Decision model of encountering situations in ship intelligent collision avoidance decision system," *Marine Technology*, no. s1, pp. 118–123, 2004.
- [29] A. Goudossis and S. K. Katsikas, "Towards a secure automatic identification system (AIS)," *Journal of Marine Science and Technology*, vol. 24, no. 2, pp. 410–423, 2019.
- [30] L. Yan, D. P. Roy, Z. Li, H. K. Zhang, and H. Huang, "Sentinel-2A multi-temporal misregistration characterization and an orbit-based sub-pixel registration methodology," *Remote Sensing of Environment*, vol. 215, pp. 495–506, 2018.
- [31] X. Nie, "Detection of grid voltage fundamental and harmonic components using Kalman filter based on dynamic tracking model," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 1191–1200, 2019.
- [32] Y. Hu, A. Zhang, W. Tian, J. Zhang, and Z. Hou, "Multi-ship collision avoidance decision-making based on collision risk index," *Journal of Marine Science and Engineering*, vol. 8, no. 9, p. 640, 2020.
- [33] L. Chen, *Research on Intelligent Collision Avoidance Assisted Decision-Making for Maritime Multi-Target Ships*, Wuhan University of Technology, Wuhan, China, 2011.
- [34] L. Li, Z. Xiong, Y. Gao, and Q. Ren, "Generation and optimization of intelligent decision-making for collision avoidance of single ship," *China Navigation*, vol. 2, pp. 49–52, 2002.
- [35] J. Zhou and C. Wu, *Construction of Ship Collision Risk Model*, vol. 2004, no. 1, pp. 61–65, 2004.
- [36] M. Jafari, E. Salajegheh, and J. Salajegheh, "An efficient hybrid of elephant herding optimization and cultural algorithm for optimal design of trusses," *Engineering with Computers*, vol. 35, no. 3, pp. 781–801, 2019.
- [37] G. Beni, "The concept of cellular robotic system," in *Proceedings IEEE International Symposium on Intelligent Control 1988*, pp. 57–62, IEEE, Arlington, VA, USA, August 1988.
- [38] N. Yang and Y. Jing, "Research on function extreme value optimization based on improved PSO algorithm," *Computer Simulation*, vol. 32, no. 9, pp. 263–266, 2015.
- [39] M. Z. Ali, N. H. Awad, R. G. Reynolds, and P. N. Suganthan, "A balanced fuzzy cultural algorithm with a modified levy flight search for real parameter optimization," *Information Sciences*, vol. 447, pp. 12–35, 2018.