Journal of
Marine Science and Engineering

MDPI

Article

Collision Avoidance for Unmanned Surface Vehicles in Multi-Ship Encounters Based on Analytic Hierarchy Process–Adaptive Differential Evolution Algorithm

Zhongming Xiao [1], Baoyi Hou [1], Jun Ning [1,*], Bin Lin [2] and Zhengjiang Liu [1]

[1] Navigation College, Dalian Maritime University, Dalian 116026, China; xiaozhongming@dlmu.edu.cn (Z.X.); houbaoyi@dlmu.edu.cn (B.H.); liuzhengjiang@dlmu.edu.cn (Z.L.)
[2] Information Science and Technology College, Dalian Maritime University, Dalian 116026, China; binlin@dlmu.edu.cn
* Correspondence: junning@dlmu.edu.cn; Tel.: +86-158-406-820-86

Abstract: Path planning and collision avoidance issues are key to the autonomous navigation of unmanned surface vehicles (USVs). This study proposes an adaptive differential evolution algorithm model integrated with the analytic hierarchy process (AHP-ADE). The traditional differential evolution algorithm is enhanced by introducing an elite archive strategy and adaptively adjusting the scale factor F and the crossover factor CR to balance global and local search capabilities, preventing premature convergence and improving the search accuracy. Additionally, the collision risk index (CRI) model is optimized and combined with the quaternion ship domain, enhancing the precision of CRI calculations and USV autonomous collision avoidance capabilities. The improved CRI model, the International Regulations for Preventing Collisions at Sea, and the optimal collision avoidance distance were incorporated as evaluation factors in a fitness function assessment, with weights determined through the AHP to enhance the rationality and accuracy of the fitness function. The proposed AHP-ADE algorithm was compared with the improved particle swarm algorithm, and the performance of the algorithm was comprehensively evaluated using safety, economy, and operational efficiency. Simulation experiments on the MATLAB platform demonstrated that the proposed AHP-ADE algorithm exhibited better performance in scenarios involving multiple ship encounters, thus proving its effectiveness.

Keywords: unmanned surface vehicle; path planning; collision avoidance; elite archive strategy; differential evolution algorithm

check for updates

Citation: Xiao, Z.; Hou, B.; Ning, J.; Lin, B.; Liu, Z. Collision Avoidance for Unmanned Surface Vehicles in Multi-Ship Encounters Based on Analytic Hierarchy Process–Adaptive Differential Evolution Algorithm. *J. Mar. Sci. Eng.* **2024**, *12*, 2123. https://doi.org/10.3390/jmse12122123

Academic Editor: Marco Cococcioni

Received: 13 October 2024
Revised: 13 November 2024
Accepted: 19 November 2024
Published: 21 November 2024

1. Introduction

Unmanned surface vehicles (USVs) are intelligent control systems that integrate advanced technologies such as path planning, communication, autonomous decision-making, and automatic target recognition [1]. With the continuous development in the level of technology, the application of USVs is increasingly expanding across various fields, such as environmental protection [2] and monitoring [3,4], maritime rescue [5] and disaster response [6], scientific research [7], and marine life conservation. Additionally, the role of USVs in the military sector is also growing. In future conflicts, the strategic application of unmanned ships and drones may become a critical factor in determining the outcome of wars.

Path planning is a critical component of autonomous navigation for USVs, and how to find the optimal path is the focus of attention in the shipping industry. The path planning process is not only limited by complex sea conditions and a ship's maneuvering performance, but also must follow the requirements of the International Regulations for Preventing Collisions at Sea (COLREGs). Therefore, planning a path that is safe, economical, and consistent with navigation practices faces great challenges.

As intelligent algorithms continue to evolve, scholars have applied various heuristic algorithms to the path planning of USVs, such as the A* algorithm [8] and Dijkstra [9]. At the same time, numerous intelligent optimization algorithms have emerged, such as ant colony algorithms (ACO) [10,11], particle swarm algorithms (PSO) [12], genetic algorithms (GA) [13,14], rapidly exploring random trees (RRT) [15], velocity obstacle methods, artificial potential field methods (APF) [16], dynamic window approaches (DWA) [17], and deep learning [18–20]. These algorithms employ specific operational strategies to obtain feasible paths, thereby facilitating safe maritime navigation for USVs. However, traditional algorithms also have some limitations in the process of path planning. For instance, algorithms like A* and Dijkstra exhibit low computational efficiency in dynamic environments. The ACO and PSO algorithms are prone to becoming trapped in local optima. The GA may converge slowly in complex environments. The paths generated by RRT algorithms are often not smooth enough. Velocity obstacle and APF methods may lead to oscillations or fail to handle mixed environments with static and dynamic obstacles. The DWA underperforms in environments with high-density obstacles, and deep learning algorithms require substantial training data and significant computational resources.

Consequently, many scholars have made improvements to various algorithms. For example, an improved A* algorithm [21] adopted a bi-directional search strategy and an enhanced heuristic function to reduce node traversal and applied path smoothing to eliminate inflections and address path folding. The modified RRT algorithm [22] incorporates the dynamic constraints of USVs into the RRT framework, utilizing calculated movement state information to enhance path search rules. An improved velocity obstacle method [23] optimized the heading angle, reducing the risk of capsizing due to large heading angles at high speeds. The enhanced genetic algorithm [24] adaptively adjusts the direction of mutation and the magnitude of mutation of individuals by introducing several genetic factors, increasing the convergence speed. Considering the limitations of single algorithms, scholars have combined various algorithms to fully leverage their strengths. For instance, the integration of the A* algorithm with the DWA [25] enables ships to generate adaptive, collision-free routes, while adhering to the COLREGs. The combination of the APF method with the A* algorithm [26] takes different ocean currents into account, ensuring that a ship consistently maintains a safe position. The integration of deep reinforcement learning (DRL) with the APF method [27] leverages the APF to enhance the motion space and objective function within the DRL algorithm, effectively achieving autonomous collision prevention and path planning. However, in complex marine environments, current algorithms often do not manage to maintain both the safety and economic efficiency of paths, resulting in planned routes that do not align with nautical practices.

Differential evolution (DE) is an evolutionary algorithm for solving multi-dimensional real-number optimization problems. Differential evolution exhibits strong robustness and adaptability, presenting substantial advantages in handling multi-dimensional, complex optimization problems, and can find optimal solutions in a variety of complex environments, as well as uncertain environments, which provides a better solution for ships navigating in complex marine environments. Due to its effectiveness, the DE algorithm has increasingly gained favor among scholars, and various improvement strategies for the algorithm have been proposed. For instance, the SAF-DE algorithm [28] utilizes a perturbation formula to disturb individuals, increasing the diversity of the population. To improve its differential strategy, Tien [29] introduced new mutation mechanisms to enhance the performance of the algorithm. A combination of the grey wolf optimization algorithm and differential evolution algorithm [30] improved the effectiveness of solving continuous global optimization problems. However, applications of the differential evolution algorithm integrated with collision avoidance decision models remain limited in the field of USV path planning.

In light of the numerous challenges faced by existing algorithms, such as the planned path not taking into account path safety and economy, being too close to obstacles or target ships in the process of collision avoidance, and failing to return to the original route in a timely manner at the end of avoidance actions, and so on, this paper innovatively improves upon the traditional differential evolution algorithm by proposing an adaptive differential evolution algorithm that integrates the analytic hierarchy process (AHP-ADE), applied to the path planning and collision avoidance of USVs. The ability of the algorithm to find the optimal solution is enhanced by introducing an elite archiving strategy and a parameter adaptive adjustment mechanism. Meanwhile, combining the quaternion ship domain with an optimized collision risk model makes the calculation of the collision risk index (CRI) more in line with nautical practice. In addition, a fitness function is constructed based on factors such as the COLREGs and the optimal collision avoidance distance, and the hierarchical analysis method is utilized to determine the weights of the factors, so as to improve the quality of the path points. The main contributions of this paper are as follows:

(1) Unlike the improved DE algorithms proposed by HuChunAn [28], Tien [29], Jitkongchuen [30], an elite archiving strategy is introduced for the mutation operation in the differential evolution algorithm, so that the individual mutation process has both the accuracy of local search and the extensiveness of global search. In addition, the control factors F and CR are adaptively adjusted, so as to dynamically optimize the search capability of the algorithm and finely tune the quality of the solution.

(2) With respect to the collision avoidance decision models used by Seo [31] and Hu [32], this paper integrates a quaternion ship domain model [33] with the CRI [34]. Improvements are made to the CRI model by adding constraint factors when determining the membership function for the distance of the closest point of approach (DCPA), thereby improving the ship's collision avoidance capabilities.

(3) Differently from the fitness function constructed by Kang [35] and Tsou [34], this paper incorporated the CRI, COLREGs, and optimal collision avoidance distance as evaluation factors in the assessment of the fitness function. The weights of each evaluation factor were determined by the analytic hierarchy process, thus enhancing the rationality and accuracy of the fitness function.

The structure of the remainder of this paper is as follows: Section 2 provides an overview of the theoretical background of the relevant algorithms, Section 3 details the adaptive differential evolutionary algorithm model combined with the analytic hierarchy process proposed in this paper, Section 4 conducts comparative experiments through scenarios involving multiple ships encounters, and Section 5 summarizes the research findings of this paper and discusses potential directions for future research.

2. Algorithm Background and Collision Avoidance Decision Model

2.1. Differential Evolution Algorithm

Differential evolution (DE) [36] is a population-difference-based algorithm for solving continuous optimization problems and searches for optimal solutions by modeling mechanisms such as natural selection and cross-breeding during biological evolution. A differential evolution algorithm primarily comprises five steps: population initialization, fitness evaluation, differential mutation, crossover operation, and selection of new individuals.

2.1.1. Population Initialization

First, an initial population of size M is randomly generated within the solution space, where each individual consists of an n-dimensional vector. The population size M is typically chosen between $5n$ and $10n$, but it should not be less than $4n$.

$$X_i(0) = (X_{i,1}(0), X_{i,2}(0), X_{i,3}(0), \ldots, X_{i,n}(0)) \tag{1}$$

$$X_{i,j}(0) = X_{i\,\min} + rand(0,1)(X_{i\,\max} - X_{i\,\min}) \tag{2}$$

Here, $X_i(0)$ denotes a randomly generated individual, and $X_{i,j}(0)$ represents the j-th dimensional vector of the individual.

2.1.2. Fitness Evaluation

The fitness function typically needs to be defined according to the specific optimization problem. In this study, the fitness function is used to evaluate the quality of path points, with lower fitness values indicating better individual fitness. Although the fitness value only reflects the quality of the individual, it contains rich population information, which reflects the evolutionary state of the population and information about the optimization problem to a certain extent, and thus guides the search process of the algorithm to a certain extent.

2.1.3. Differential Mutation

The following are some common mutation strategies:

DE/rand/2

$$V_i(G) = X_{r1}(G) + F \times (X_{r2}(G) - X_{r3}(G)) + F \times (X_{r4}(G) - X_{r5}(G)) \tag{3}$$

DE/best/2

$$V_i(G) = X_{best}(G) + F \times (X_{r1}(G) - X_{r2}(G)) + F \times (X_{r3}(G) - X_{r4}(G)) \tag{4}$$

DE/current-to-rand/1

$$V_i(G) = X_i(G) + rand[0,1) \times (X_{r1}(G) - X_i(G)) + F \times (X_{r2}(G) - X_{r3}(G)) \tag{5}$$

Taking *DE/rand/2* as an example, $X_{r1}(G)$, $X_{r2}(G)$, and $X_{r3}(G)$ are three different vectors randomly selected from the parent generation, where $r_1 \neq r_2 \neq r_3 \neq i \in \{1, 2, 3 \ldots, M\}$, M is the population size, and F is the scaling factor with a range of [0–2], typically set to 0.5. $V_i(G)$ is the new vector generated through differential mutation. Different mutation strategies exhibit varying optimization capabilities for the population. To better understand the common properties of the various mutation strategies, Feoktistov summarized them in a general form: $V_i = \beta_i + F \times \delta_i$, where β_i is the base vector and δ_i is the differential vector.

2.1.4. Crossover Operation

There are two main types of crossover operations: binomial crossover and exponential crossover. Binomial crossover can independently make decisions for each dimension, providing better diversity. Therefore, this paper chose binomial crossover.

$$U_{i,j}(G) = \begin{cases} V_{i,j}(G), rand[0,1) < CR \text{ } or \text{ } j = jrand \\ X_{i,j}(G), otherwise \end{cases} \tag{6}$$

CR is the crossover factor, with a value range of [0–1]. For each dimension of the current individual, if $rand[0,1)$ is less than the crossover factor CR, the corresponding dimension of the new individual comes from the mutant individual; otherwise, it comes from the initial individual. j is the current dimension of the vector, and $jrand$ is a randomly generated dimension within a range of 1 to n. Adding the condition $j = jrand$ prevents the new individual from being identical to the initial individual. The crossover process is illustrated in Figure 1.

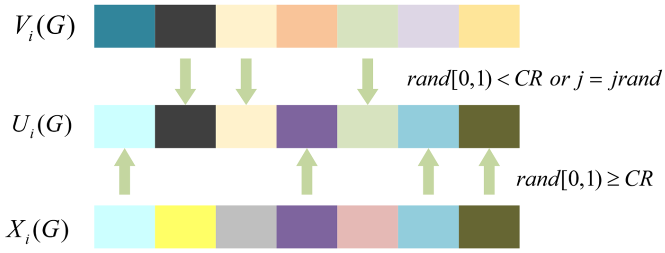

Figure 1. Crossover operation.

2.1.5. Selection of New Individuals

The differential evolution algorithm employs a greedy strategy for selection. This selection process evaluates the fitness values of individuals, choosing those with better fitness to be the initial individuals for the next generation, thereby guiding the population in a better direction. After the initial individuals undergo mutation and crossover operations to generate new individuals, the direction of the population evolution is determined using the following formula:

$$X_i(G+1) = \begin{cases} U_i(G), f(U_i(G)) \le f(X_i(G)) \\ X_i(G), \text{otherwise} \end{cases} \tag{7}$$

2.2. Quaternion Ship Domain

In 1971, the concept of the ship domain was first introduced by Japanese scholars Fujii and Tanaka [37], who developed an elliptical ship domain model and corresponding dimensions suitable for narrow waterways through traffic surveys and probabilistic statistical methods. Since then, various scholars have proposed numerous ship domain models. For instance, in 1973, Goodwin [38] proposed an approach for ascertaining the ship domain based on radar simulator performance and North Sea traffic surveys. In 2010, Wang [33] introduced a fuzzy quaternion ship domain model that comprehensively considers various factors, such as ship length, fundamental maneuvering performance, and real-time speed. This ship domain model consists of four ellipses with varying major and minor axes, and it calculates the scale of the ship domain based on maneuverability parameters such as the ship's advance distance and tactical diameter. Therefore, the model can adaptively adjust under the control of the ship's motion state. Compared to traditional circular or elliptical models, the quaternion ship domain can more accurately assess collision risks. Therefore, this study adopted the quaternion ship domain proposed by Wang, as illustrated in Figure 2. R_{fore} and R_{aft} represent the fore and aft radii of the ship domain, respectively, while R_{port} and R_{starb} represent the port and starboard radii, respectively. Equation (8) is the boundary equation for the ship domain.

$$\begin{aligned} f(x,y) = &\left(\frac{2x}{(1+sngx)R_{fore} - (1-sngx)R_{aft}} \right)^2 \\ &+ \left(\frac{2y}{(1+sngy)R_{starb} - (1-sngy)R_{port}} \right)^2 \end{aligned} \tag{8}$$

The radius equation is

$$\begin{cases} R_{fore} = (1 + 1.34\sqrt{k_{AD}^2 + (k_{DT}/2)^2})L \\ R_{aft} = (1 + 0.67\sqrt{k_{AD}^2 + (k_{DT}/2)^2})L \\ R_{starb} = (0.2 + k_{DT})L \\ R_{port} = (0.2 + 0.75k_{DT})L \end{cases} \tag{9}$$

$$sngx = \begin{cases} -1, x < 0 \\ 1, x \ge 0 \end{cases} \quad sngy = \begin{cases} -1, \text{y} < 0 \\ 1, y \ge 0 \end{cases} \tag{10}$$

$$\begin{cases} k_{AD} = 10^{0.3591 \lg v + 0.0952} \\ k_{DT} = 10^{0.5441 \lg v - 0.0795} \end{cases} \tag{11}$$

where v is the ship's speed and L is the length of the ship.

Due to the lack of explicit standards for the minimum safe encounter distance in the COLREGs, this study defines the minimum safe encounter distance based on the quaternion ship domain model.

$$r_1 = \begin{cases} \sqrt{R_{fore}^2 R_{starb}^2 / (R_{fore}^2 \sin \alpha_1^2 + R_{starb}^2 \cos \alpha_1^2)} & y > 0, x \geq 0 \\ \sqrt{R_{starb}^2 R_{aft}^2 / (R_{starb}^2 \sin \alpha_2^2 + R_{aft}^2 \cos \alpha_2^2)} & y \geq 0, x < 0 \\ \sqrt{R_{port}^2 R_{aft}^2 / (R_{port}^2 \sin \alpha_3^2 + R_{aft}^2 \cos \alpha_3^2)} & y < 0, x \geq 0 \\ \sqrt{R_{fore}^2 R_{port}^2 / (R_{fore}^2 \sin \alpha_4^2 + R_{port}^2 \cos \alpha_4^2)} & y \geq 0, x > 0 \end{cases} \tag{12}$$

where $\alpha_1, \alpha_2, \alpha_3$, and α_4 represent the angles between the line connecting the ship to the nearest encounter and R_{fore}, R_{starb}, R_{port}, and R_{fore}, respectively.

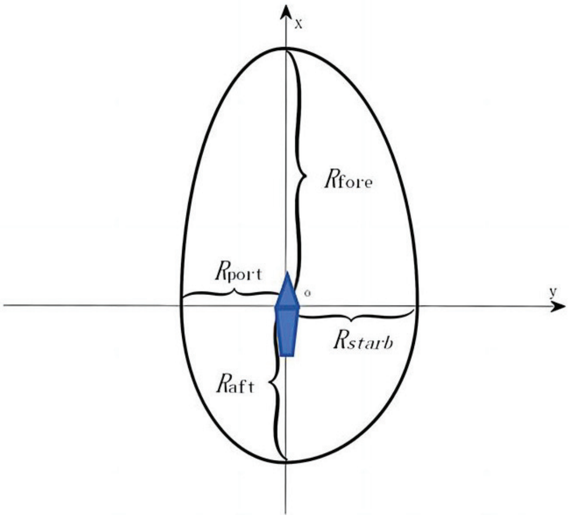

Figure 2. Quaternion ship domain model.

2.3. Ship Encounter Situations and Allocation of Responsibilities

The evaluation of encounter situations is crucial for determining the responsibilities of ships in giving way and the evasive actions to be taken. It is also a key component of automatic collision avoidance decision-making in intelligent ships. Numerous ship accidents have shown that one of the main causes of uncoordinated maritime traffic, and even collisions, is the failure to correctly evaluate an encounter situation between ships. When ships are in sight of one another, the ships' collision avoidance behavior should strictly adhere to rules 8, 13, 14, and 15 of the COLREGs. These rules specify the evasive actions ships should take in different encounter situations. Rule 8 stipulates that ships must take appropriate action to avoid collisions when encountering potential collision risks. This includes reducing speed, altering course, and maintaining vigilance to ensure effective communication and observation. Rule 13 addresses overtaking situations, stating that the overtaking ship must prioritize avoiding collisions with the ship being overtaken, ensuring safety during the maneuver. Rule 14 defines head-on encounters, emphasizing that when ships are on a direct course toward each other, both should take appropriate evasive actions according to the rules to ensure safe passage. Rule 15 pertains to crossing encounters, requiring ships to take evasive measures based on their respective courses and speeds to prevent collisions. Therefore, this paper takes the COLREGs into account, fully considering the impact of ship encounter situations on collision avoidance behavior. Ship encounter situations and responsibility allocations are shown in Table 1.

Table 1. Ship encounter situations and responsibility allocations.

True Bearing of TS to OS/°	Course Difference/°	Encounter Situation	OS	TS
$354 \leq \theta_r \leq 6$	$174 \leq \Delta C \leq 186$	Head-on	Give-way	Give-way
$247.5 \leq \theta_r < 354$	$67.5 \leq \Delta C < 174$	Left-Crossing	Stand-on	Give-way
$6 < \theta_r \leq 112.5$	$186 < \Delta C \leq 292.5$	Right-Crossing	Give-way	Stand-on
$112.5 < \theta_r < 247.5$	$\Delta C < 67.5 \cup \Delta C > 292.5$	Overtaking	Stand-on	Give-way

3. Improved Adaptive Differential Evolution Algorithm Model

This section introduces an adaptive DE algorithm model that incorporates the analytic hierarchy process. Initially, an elite archive strategy [39] is integrated into the mutation operation of the DE, and the control factors F and CR are adaptively improved [40], enhancing the population diversity and avoiding local optima. Simultaneously, a penalty mechanism is introduced to address new vectors that do not meet the boundary constraints, ensuring their effectiveness. Furthermore, the CRI, COLREGs, and the optimal collision avoidance distance are incorporated into the assessment of the fitness function, constituting an evaluation set with safety, economy, the COLREGs, and optimal collision avoidance distance as the evaluating factors, where the safety is determined using the CRI, and optimized for the traditional CRI model. Finally, the weights of these evaluation factors are determined using the AHP, further enhancing the rationality and accuracy of the fitness function.

3.1. Algorithm Improvements

3.1.1. Elite Archive Strategy

In this study, the selected mutation strategy is *DE/rand/2*, which performs mutation operations using two different differential vectors. Compared to other mutation strategies, this enables a broader exploration of the search space, making the algorithm more likely to escape local optima. Additionally, this strategy offers strong flexibility and adaptability, and it is relatively simple to implement. However, the strategy lacks the tendency of searching for better solutions during the mutation process. Therefore, to overcome this drawback, an elite archive strategy is added to *DE/rand/2*, offering both the precision of local search and the breadth of global search.

First, classify the M individuals in the current population based on their fitness. The OR individuals with better fitness are placed in set P to form the elite population, while the $(M - OR)$ individuals with poorer fitness are placed in set Q to form the non-elite population. P and Q should satisfy $P \cup Q = M$ and $P \cap Q = \emptyset$. The improvement to *DE/rand/2* is as follows:

$$V_i(G) = X_{r1}^P(G) + F \times (X_{r2}^P(G) - X_{r3}^Q(G)) + F \times (X_{r4}^P(G) - X_{r5}^Q(G)) \tag{13}$$

In this context, $X_{r1}^P(G)$, $X_{r2}^P(G)$, and $X_{r4}^P(G)$ are individuals randomly selected from the elite population P, while $X_{r3}^Q(G)$ and $X_{r5}^Q(G)$ are randomly selected from the non-elite population Q. The principle of the strategy is simple: P is used to guide the mutant individuals towards better solutions in the population, improving the convergence speed, while Q is employed to adjust the diversity of the population. During the population iteration process, if the fitness value of $U_i(G)$ is greater than that of $X_i(G)$, P and Q remain unchanged. If the fitness value of $U_i(G)$ is less than that of $X_i(G)$, P and Q are dynamically updated. The updates include the following two situations:

(1) If the individual corresponding to $U_i(G)$ is in the elite population P, then $U_i(G)$ directly replaces the corresponding individual in P.

(2) If the individual corresponding to $U_i(G)$ is not in P, and the fitness of $U_i(G)$ is better than the worst individual in P, then $U_i(G)$ replaces the worst individual in P. The worst individual from P is added to Q, and the corresponding individual of $U_i(G)$ in Q is removed.

3.1.2. Adaptive Factors

The scaling factor F can be viewed as a factor that regulates the degree of individual perturbation, influencing the step length of an individual's movement in space. When F is large, the perturbation of individuals is greater, which increases the population diversity and makes the algorithm more likely to escape local optima, thereby enhancing the global search capabilities. However, this comes with a slower convergence. Conversely, when F is small, the perturbation of individuals is reduced, leading to shorter movement steps in space. This enhances the local search capabilities and speeds up convergence. However, it also increases the likelihood of becoming trapped in local optima. Therefore, this paper proposes an adaptive F strategy that enhances the global search capabilities while increasing the convergence speed, with improvements as follows:

$$F_G = \begin{cases} F_1, F_0 \times \sum\limits_{i=1}^{M} f(x_i^G) / \sum\limits_{i=1}^{M} f(x_i^0) \leq F_1 \\ F_0 \times \frac{\sum\limits_{i=1}^{M} f(x_i^G)}{\sum\limits_{i=1}^{M} f(x_i^0)}, F_0 \times \sum\limits_{i=1}^{M} f(x_i^G) / \sum\limits_{i=1}^{M} f(x_i^0) > F_1 \end{cases} \tag{14}$$

In this context, F_0 and F_1 represent the scaling factors during the first and second iterations, respectively, while F_G is the scaling factor during the G-th iteration. $f(x_i^G)$ and $f(x_i^0)$ denote the fitness values of the individual during the G-th iteration and the first iteration, respectively.

The crossover factor CR determines the likelihood that the vectors of the initial individual will be replaced by those of the mutant individual during the crossover operation. When CR is high, the information from the mutant individual is better transferred to the initial individual. Conversely, when CR is low, less information is transferred from the mutant to the initial individual, but this increases the independence among individuals. Therefore, an adaptive CR mechanism is proposed to balance these two effects, with improvements as follows:

$$CR_n = \begin{cases} CR_1, f(x_n^G) > f(x_{avg}^G) \\ CR_0 \times \frac{(CR_1 - CR_0)(f(x_{avg}^G) - f(x_n^G))}{f(x_{avg}^G) - f(x_{\min}^G)}, f(x_n^G) \leq f(x_{avg}^G) \end{cases} \tag{15}$$

$f(x_n^G)$ and $f(x_{avg}^G)$ represent the fitness value of the n-th individual and the average fitness value of all individuals, respectively, while $f(x_{min}^G)$ is the minimum fitness value.

To prevent newly generated individual vectors from violating the boundary constraints, a penalty mechanism is introduced, causing the individual vectors to change according to the following formula:

$$U_{i,j}(G) = \min(\max(U_{i,j}(G), U_{i,j}^{\min}), U_{i,j}^{\max}) \tag{16}$$

Here, j represents the dimension of the current vector, and $U_{i,j}^{\min}$ and $U_{i,j}^{\max}$ denote the lower and upper bounds of the j-th dimensional vector, respectively. The individual vectors are eventually made to satisfy the boundary constraints through this penalty mechanism.

3.2. Collision Risk Index

The CRI [34] is typically calculated through a quantitative analysis of the relative positions, velocities, and headings of two or more ships, along with other environmental factors such as weather and visibility. This index is used to assess the probability of collisions between ships within a specified time and spatial range. When the value of the CRI is high, ships must take emergency avoidance actions to prevent collisions with other ships. This study selects $DCPA$, $TCPA$ (the time to closest point of approach), the distance between two ships (D), the relative bearing (A), and the speed ratio (K) as factors to construct the collision risk model. Establishment of the collision risk index factor set is $U = \{DCPA, TCPA, D, A, K\}$

Define the membership functions for each factor:

(1) Membership function for $DCPA$

Establish a coordinate system with the own ship (OS) as the origin, where the positive x-axis points east and the positive y-axis points north. The coordinates of the own ship are (x_O, y_O), with a speed of v_O and a heading of φ_O. The target ship (TS)'s coordinates are

(x_T, y_T), with a speed of v_T and a heading of φ_T. The true bearings from the OS to the TS and from the TS to the OS are a_{OT} and a_{TO}, respectively. The relative speed is v_R.

Previously, when selecting the membership function for $DCPA$, only the influence of the minimum safe encounter distance r_1 and the safe passing distance r_2 were considered, without taking into account whether the ship domain of the OS and the TS are intruded upon. Figure 3 illustrates several scenarios where the ship domains are intruded upon. Therefore, an improvement is made to the membership function of $DCPA$ to address this issue.

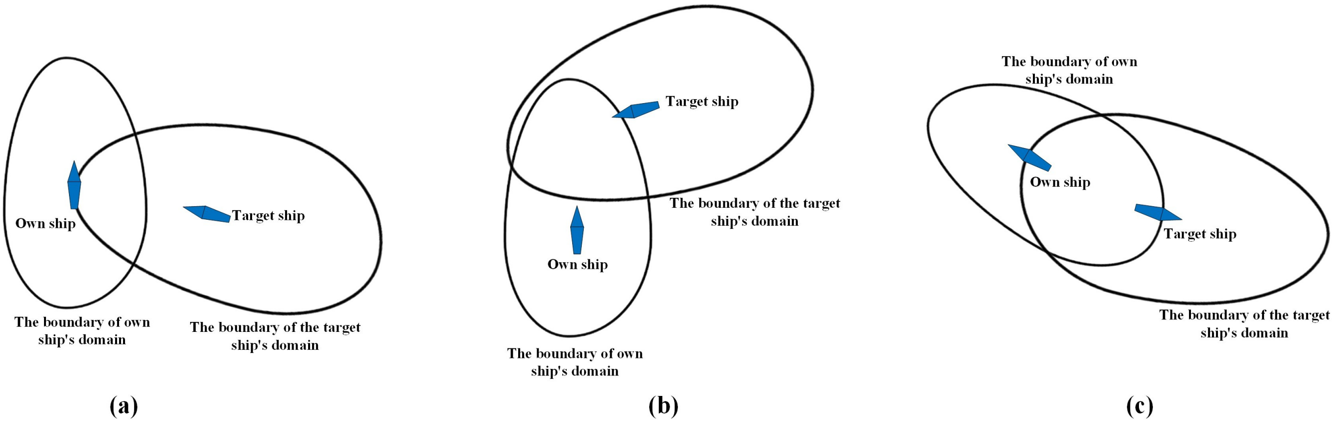
Own ship
Target ship
The boundary of own ship's domain
The boundary of the target ship's domain
(a)
Target ship
The boundary of the target ship's domain
Own ship
The boundary of own ship's domain
(b)
The boundary of own ship's domain
Own ship
Target ship
The boundary of the target ship's domain
(c)

Figure 3. Various situations of OS and the TS domains being intruded upon: (**a**) the TS does not intrude into the OS's domain, but the OS intrudes into the TS's domain; (**b**) the OS does not intrude into the TS's domain, but the TS intrudes into the OS's domain; (**c**) both ships intrude into each other's domains.

Establish a coordinate system with the TS as the origin, with the direction of the bow serving as the positive y-axis and the direction perpendicular to the bow to the right as the positive x-axis. Perform a coordinate transformation for the position of the OS:

$$x_{O1} = D\sin\beta_0, y_{O1} = D\cos\beta_0, \beta_0 = a_{OT} - \varphi_T + \gamma_1, \tag{17}$$

$$\gamma_1 = \begin{cases} 360, a_{OT} - \varphi_T \leq 0 \\ 0, a_{OT} - \varphi_T > 0 \end{cases} \tag{18}$$

Based on the transformed coordinates (x_{O1}, y_{O1}), derive the equation for the relative motion line of the OS with respect to the TS:

$$y = \cot(\varphi_R - \varphi_T)x + (y_{O1} - x_{O1}\cot(\varphi_R - \varphi_T)) \tag{19}$$

$$\varphi_R = \begin{cases} \arctan\frac{v_{OTx}}{v_{OTy}} + \theta, & otherwise \\ 90, & v_{OTx} \geq 0, v_{OTy} = 0 \\ 270, & v_{OTx} < 0, v_{OTy} = 0 \end{cases} \tag{20}$$

$$\theta = \begin{cases} 0 & v_{OTx} \geq 0, v_{OTy} > 0 \\ 180 & v_{OTx} \geq 0, v_{OTy} < 0 \quad or \quad v_{OTx} < 0, v_{OTy} < 0 \\ 360 & v_{OTx} < 0, v_{OTy} > 0 \end{cases} \tag{21}$$

Due to the change in the coordinate system, the equation for the relative motion line also needs to be transformed:

$$x = \cot(\varphi_R - \varphi_T)y + (y_{O1} - x_{O1}\cot(\varphi_R - \varphi_T)) \tag{22}$$

As illustrated in Figure 4, when the OS intrudes into the ship domain of the TS, the relative motion line of the OS with respect to the TS will intersect with the boundary of the

TS's ship domain. Therefore, it is possible to determine whether the OS has intruded into the TS's domain by calculating if an intersection exists.

Similarly, by analyzing whether there are intersections between the relative motion line of the TS with respect to the OS and the boundary of the OS's domain, it can be determined whether the TS has intruded into the domain of the OS.

The improved membership function is

$$k_{DCPA} = \begin{cases} 1, \quad DCPA < r_1 || p_1 > 0 || p_2 > 0 \\ \frac{1}{2} - \frac{1}{2}\sin\left(\frac{180^\circ}{r_2 - r_1}\left(DCPA - \frac{r_2 + r_1}{2}\right)\right), r_1 < DCPA < r_2 \\ 0, \quad DCPA \geq r_2 \end{cases} \tag{23}$$

Here, p_1 is the number of intersections between the relative motion line of the OS with respect to the TS and the boundary of the TS's domain, and p_2 is the number of intersections between the relative motion line of the TS with respect to the OS and the boundary of the OS's domain.

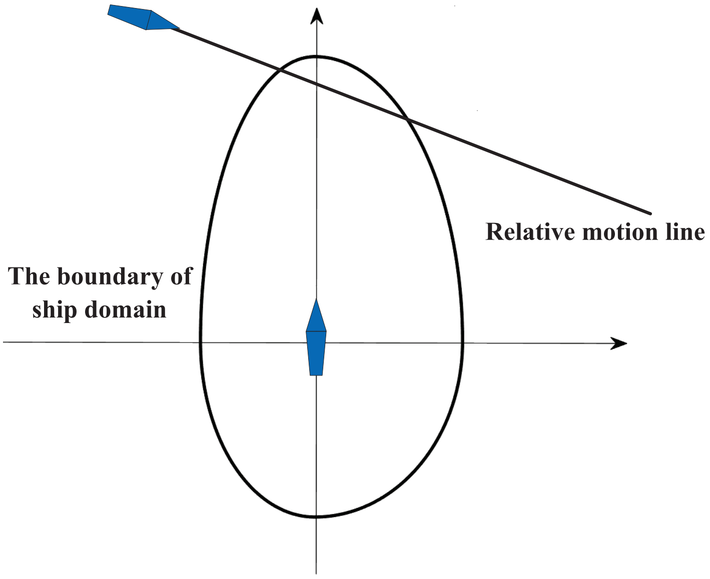

Figure 4. The relative motion lines intersect at the boundary of the ship domain.

(2) Membership function for $TCPA$

$$k_{TCPA} = \begin{cases} 1, \quad TCPA \leq T_1 \\ \left(\frac{T_2 - TCPA}{T_2 - T_1}\right)^2, \quad T_1 < TCPA \leq T_2 \\ 0, \quad TCPA > T_2, DCPA > d_4 \end{cases} \tag{24}$$

$$T_1 = \begin{cases} \frac{\sqrt{d_3^2 - DCPA^2}}{v_R}, \quad DCPA \leq d_3 \\ \frac{DCPA - d_3}{v_R}, \quad DCPA > d_3 \end{cases} \tag{25}$$

$$T_2 = \frac{\sqrt{d_4^2 - DCPA^2}}{v_R} \tag{26}$$

Here, d_3 represents the latest distance at which the give-way ship must take evasive action, and d_4 is the distance within which a ship can take measures to avoid a collision.

(3) Membership function of the distance between two ships (D)

$$k_D = \begin{cases} 1, \quad 0 \le D \le d_3 \\ \left(\frac{d_4 - D}{d_4 - d_3}\right)^2, \quad d_3 < D \le d_4 \\ 0, \quad D > d_4 \end{cases} \tag{27}$$

(4) Membership function of relative bearing (A)

$$k_A = \frac{1}{2}\left[\cos(a_{TO} - 19^\circ) + \sqrt{\frac{440}{289} + \cos^2(a_{TO} - 19^\circ)}\right] - \frac{5}{17} \tag{28}$$

(5) Membership function of speed ratio (K)

$$k_K = \frac{1}{1 + \left(\frac{K_0}{K}\right)^2} \tag{29}$$

In the above formula, K_0 is the threshold of K, taking $K_0 = 1$.

Establish a set of weights W based on the importance of each factor in the calculation of the CRI.

$$W = \{W_{DCPA}, W_{TCPA}, W_D, W_A, W_K\} \tag{30}$$

$$CRI = W_{DCPA}k_{DCPA} + W_{TCPA}k_{TCPA} + W_D k_D + W_A k_A + W_K k_K \tag{31}$$

3.3. Fitness Function

The CRI, COLREGs, and optimal avoidance distance are incorporated into the assessment of the fitness function, creating an evaluation set F defined by the factors of safety, economy, COLREGs, and optimal avoidance distance. Here, the safety of the path points is determined using the CRI, while the economy of the path points is determined using the voyage distance and the degree of turning. The COLREGs are used to assess encounter situations and determine whether the ship needs to take evasive action.

The objective function with respect to CRI is defined as follows:

$$f_1 = CRI \tag{32}$$

During path planning, the total voyage is an economic evaluation index. Assuming that the coordinates of three adjacent points in the path are (x_{i-1}, y_{i-1}), (x_i, y_i), and (x_{i+1}, y_{i+1}), respectively, where (x_i, y_i) is the current point. The total number of path points is m, and the destination point is (x_m, y_m). When assessing the total voyage, it is evident that (x_{i+1}, y_{i+1}) is unknown. Therefore, the total voyage is evaluated by examining the relationship between the current point, the previous path point, and the destination point. The objective function with respect to the total voyage is defined as follows:

$$\begin{aligned} f_2 = & \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{\sqrt{(x_m - x_{i-1})^2 + (y_m - y_{i-1})^2}} \\ & + \frac{\sqrt{(x_m - x_i)^2 + (y_m - y_i)^2}}{\sqrt{(x_m - x_{i-1})^2 + (y_m - y_{i-1})^2}} \end{aligned} \tag{33}$$

The objective function based on the degree of turning is

$$f_3 = \arccos\left(\frac{(x_i - x_{i-1}, y_i - y_{i-1}) \cdot (x_m - x_i, y_m - y_i)^T}{||(x_i - x_{i-1}, y_i - y_{i-1}) \cdot (x_m - x_i, y_m - y_i)||}\right) \tag{34}$$

When the OS is a stand-on ship, it does not need to take evasive actions. However, when the OS is a give-way ship, it should take appropriate evasive measures to avoid collision risks. Based on the previously described encounter situations and responsibility allocations, the objective function constructed according to the COLREGs is as follows:

$$f_4 = \begin{cases} 1 & 000^\circ \leq \theta_r \leq 112.5^\circ \; or \; 354^\circ \leq \theta_r \leq 006^\circ \\ 0 & otherwise \end{cases} \tag{35}$$

The optimal avoidance distance is crucial for determining when a ship should take evasive actions. Therefore, this paper employs a fuzzy evaluation method to calculate the optimal avoidance distance, making the decision-making process more accurate and efficient.

Establish a set of factors γ and a set of corresponding evaluation results φ, where $\gamma = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6, \gamma_7\}$, and $\varphi = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5\}$. γ represents a group of judgment factors, i.e., the set of factors influencing the optimal avoidance distance. Specifically, $\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6,$ and γ_7 represent the relative motion velocity judgment vector, bearing judgment vector, speed ratio judgment vector, ship density judgment vector, navigation water condition judgment vector, meteorological condition judgment vector, and ship size judgment vector, respectively. φ corresponds to the values of the optimal avoidance distance, where $\varphi_1 = 2$, $\varphi_2 = 3$, $\varphi_3 = 4$, $\varphi_4 = 5$, $\varphi_5 = 6$.

Construction and determination of the judgment matrix for the optimal avoidance distance is as follows:

$$E = \begin{pmatrix} \gamma_{11}, \gamma_{12}, \gamma_{13}, \gamma_{14} \,,\gamma_{15} \\ \gamma_{21}, \gamma_{22}, \gamma_{23}, \gamma_{24}, \gamma_{25} \\ \gamma_{31}, \gamma_{32}, \gamma_{33}, \gamma_{34}, \gamma_{35} \\ \gamma_{41}, \gamma_{42}, \gamma_{43}, \gamma_{44}, \gamma_{45} \\ \gamma_{51}, \gamma_{52}, \gamma_{53}, \gamma_{54}, \gamma_{55} \\ \gamma_{61}, \gamma_{62}, \gamma_{63}, \gamma_{64}, \gamma_{65} \\ \gamma_{71}, \gamma_{72}, \gamma_{73}, \gamma_{74}, \gamma_{75} \end{pmatrix} \tag{36}$$

In previous research and surveys [41], the authors of the paper carried out further research and provided solutions. Therefore, the weight values of each influencing factor on the optimal avoidance distance are known under different conditions.

Determine the evaluation vector γ_1:

The greater the relative motion speed, the longer the time a ship takes to avoid a collision by decelerating or turning. Therefore, evasive actions should be taken as early as possible, and the avoidance distance should be correspondingly increased.

$$\begin{array}{l} if \;\; 0 \text{ kn} \leq V_R < 5 \text{ kn}, \quad \gamma_1 = (0.8, 0.2, 0, 0, 0) \\ if \;\; 5 \text{ kn} \leq V_R < 15 \text{ kn}, \quad \gamma_1 = (0.5, 0.3, 0.2, 0, 0) \\ if \;\; 15 \text{ kn} \leq V_R < 25 \text{ kn}, \quad \gamma_1 = (0.2, 0.6, 0.2, 0, 0) \\ if \;\; 25 \text{ kn} \leq V_R < 35 \text{ kn}, \quad \gamma_1 = (0, 0.2, 0.6, 0.2, 0) \\ if \;\; 35 \text{ kn} \leq V_R < 45 \text{ kn}, \quad \gamma_1 = (0, 0, 0.2, 0.3, 0.5) \\ if \;\; 45 \text{ kn} \leq V_R, \quad \gamma_1 = (0, 0, 0, 0.2, 0.8) \end{array} \tag{37}$$

Determine the evaluation vector γ_2:

$$\begin{array}{l} if \;\; -10^\circ \leq Q < 10^\circ, \quad \gamma_2 = (0, 0, 0.1, 0.3, 0.6) \\ if \;\; 10^\circ \leq Q < 60^\circ, \quad \gamma_2 = (0.1, 0.2, 0.3, 0.3, 0.1) \\ if \;\; 60^\circ \leq Q < 112^\circ, \quad \gamma_2 = (0.1, 0.2, 0.4, 0.2, 0.1) \\ if \;\; 112^\circ \leq Q < 248^\circ, \quad \gamma_2 = (0.5, 0.4, 0.1, 0, 0) \\ if \;\; 248^\circ \leq Q < 270^\circ, \quad \gamma_2 = (0.1, 0.4, 0.4, 0.1, 0) \\ if \;\; 270^\circ \leq Q < 350^\circ, \quad \gamma_2 = (0, 0.1, 0.2, 0.3, 0.4) \end{array} \tag{38}$$

Determine the evaluation vector γ_3:

$$\begin{array}{l} if \;\; K < 0.8, \quad \gamma_3 = (0, 0, 0.1, 0.3, 0.6) \\ if \;\; 0.8 \leq K < 1.2, \quad \gamma_3 = (0, 0.2, 0.6, 0.2, 0) \\ if \;\; 1.2 \leq K, \quad \gamma_3 = (0.6, 0.2, 0.2, 0, 0) \end{array} \tag{39}$$

Determine the evaluation vector γ_4:

When there is less than one ship within two nautical miles,

$$\gamma_4 = (0, 0.1, 0.2, 0.3, 0.4) \tag{40}$$

When more than one ship is within one nautical mile,

$$\gamma_4 = (0.4, 0.3, 0.2, 0.1, 0) \tag{41}$$

Otherwise,

$$\gamma_4 = (0.1, 0.2, 0.4, 0.2, 0.1) \tag{42}$$

Determine the evaluation vector γ_5:

$$\begin{array}{l} in \;\; open \;\; waters, \quad \gamma_5 = (0, 0.1, 0.2, 0.3, 0.4) \\ in \;\; general \;\; waters, \quad \gamma_5 = (0, 0.3, 0.4, 0.3, 0) \\ in \;\; narrow \;\; waters, \quad \gamma_5 = (0.4, 0.3, 0.2, 0.1, 0) \end{array} \tag{43}$$

Determine the evaluation vector γ_6:

As meteorological conditions worsen, visibility at sea decreases, making ships more susceptible to external influences and impairing their ability to evaluate unknown risks, so the distance for collision avoidance should be increased.

When weather conditions are good,

$$\gamma_6 = (0.4, 0.3, 0.2, 0.1, 0) \tag{44}$$

When weather conditions are normal,

$$\gamma_6 = (0.1, 0.2, 0.4, 0.2, 0.1) \tag{45}$$

When weather conditions are poor,

$$\gamma_6 = (0, 0.1, 0.2, 0.3, 0.4) \tag{46}$$

Determine the evaluation vector γ_7:

The larger a ship's size, the poorer its maneuverability, and thus a greater avoidance distance should be allowed.

$$\begin{array}{l} if \;\; L \leq 50 \text{ m}, \quad \gamma_7 = (0.4, 0.3, 0.2, 0.1, 0) \\ if \;\; 50 \text{ m} \leq L < 100 \text{ m}, \quad \gamma_7 = (0.1, 0.3, 0.3, 0.2, 0.1) \\ if \;\; 100 \text{ m} \leq L < 250 \text{ m}, \quad \gamma_7 = (0.1, 0.2, 0.3, 0.3, 0.1) \\ if \;\; 250 \text{ m} \leq L, \quad \gamma_7 = (0, 0.1, 0.2, 0.3, 0.4) \end{array} \tag{47}$$

When calculating the optimal avoidance distance, it is necessary to select appropriate evaluation vectors according to the actual conditions of each factor and to construct a judgment matrix E. Based on expert evaluations, the weights of each evaluation factor are obtained as $X = (0.4, 0.2, 0.15, 0.1, 0.05, 0.05, 0.05)$. The evaluation result H is calculated as $H = X \bullet E$, and the optimal avoidance distance φ_0 is determined according to the principle of maximum membership.

Construct an objective function based on the optimal avoidance distance:

$$f_5 = \begin{cases} 1 & D < \varphi_0 \\ 0 & D \geq \varphi_0 \end{cases} \tag{48}$$

$$Fitness = W_1 f_1 + W_2 f_2 + W_3 f_3 + W_4 f_4 + W_5 f_5 \tag{49}$$

where W_1, W_2, W_3, W_4, and W_5 are the weights for safety, total voyage, degree of turning, COLREGs, and optimal avoidance distance, respectively.

3.4. Determining Weight Values Through the Analytic Hierarchy Process

The analytic hierarchy process (AHP) is a decision-support tool for calculating weights in multi-objective complex issues. Due to its systematic and flexible nature, the AHP has been extensively utilized across various sectors, including risk assessment, resource allocation, and project evaluation.

To enhance the rationality of the fitness function value in evaluating the merits of individual path points, this paper employs the AHP to calculate the weights. Expert scoring, constructing a judgment matrix, calculating eigenvectors and weight values, and consistency checks are used to ultimately determine the weight of each factor.

3.4.1. Establishment of the Judgment Matrix

(1) Expert scoring

After scoring by multiple experts, an initial judgment matrix was established as depicted in Table 2.

Table 2. Initial judgment matrix.

	CRI	**Voyage**	**Degree of Turning**	**COLREGs**	**Avoidance Distance**
CRI	1	3	2	4	1
Voyage	1/3	1	1/2	2	1/3
Degree of turning	1/2	2	1	3	1/2
COLREGs	1/4	1/2	1/3	1	1/4
Avoidance distance	1	3	2	4	1
CRI	1	6	3	3	2
Voyage	1/6	1	1/2	1/2	1/3
Degree of turning	1/3	2	1	1	2/3
COLREGs	1/3	2	1	1	2/3
Avoidance distance	1/2	3	3/2	3/2	1
CRI	1	4/3	2	3	5
Voyage	3/4	1	4/3	2	4
Degree of turning	1/2	3/4	1	4/3	3
COLREGs	1/3	1/2	3/4	1	2
Avoidance distance	1/5	1/4	1/3	1/2	1

(2) Combining judgment matrices

The geometric mean was calculated from the scores of multiple experts, and the consolidated judgment matrix is shown in Table 3.

Table 3. The combined judgment matrix.

	CRI	Voyage	Degree of Turning	COLREGs	Avoidance Distance
CRI	1	2.8851	2.289	3.301	2.154
Voyage	0.3466	1	0.693	1.260	0.763
Degree of turning	0.4368	1.4422	1	1.587	1
COLREGs	0.3029	0.7937	0.6300	1	0.693
Avoidance distance	0.4642	1.3104	1	1.4422	1

3.4.2. Calculation of Weight Values

(1) Normalize the judgment matrix.
(2) Sum the processed matrix row-wise.
(3) Normalize again to obtain the eigenvector and the weight values of each factor. The results are presented in Table 4.

Table 4. AHP hierarchical analysis results.

	Eigenvector	Weight Value	Maximum Eigenvalue	Value of CI
CRI	1.956	39.129%	5.004	0.001
Voyage	0.677	13.533%		
Degree of turning	0.906	18.130%		
COLREGs	0.578	11.556%		
Avoidance distance	0.883	17.652%		

As indicated by Table 4, a 5-order judgment matrix was constructed for research on CRI, voyage, degree of turning, COLREGs, and avoidance distance (calculated using the sum-product method). The analysis yielded an eigenvector of (1.956, 0.677, 0.906, 0.587, 0.883), with corresponding weight values of 0.39129, 0.13533, 0.18130, 0.11556, and 0.17652, respectively. Additionally, combining the eigenvector allowed the calculation of the $maximum\ eigenvalue$ (5.004). Subsequently, this value was used to obtain the consistency index (CI) value 0.001, where $CI = (maximum\ eigenvalue - n)/(n-1)$, which was used for the consistency check described below.

3.4.3. Consistency Check

The random consistency index (RI) value corresponding to the 5-order judgment matrix was 1.120. The consistency index (CR) calculated ($CR = CI/RI$) was 0.001 < 0.1. This demonstrates that the judgment matrix used in this study passed the consistency check, ensuring that the calculated weights are reliable. The results of the consistency check are shown in Table 5.

Table 5. Summary of results of consistency check.

Maximum Eigenvalue	CI	RI	CR	Consistency Check
5.004	0.001	1.120	0.001	pass

As illustrated in Figure 5, the weight values for the factors constituting the fitness function are as follows: $W_1 = 0.3913, W_2 = 0.1353, W_3 = 0.1813, W_4 = 0.1156, W_5 = 0.1765$.

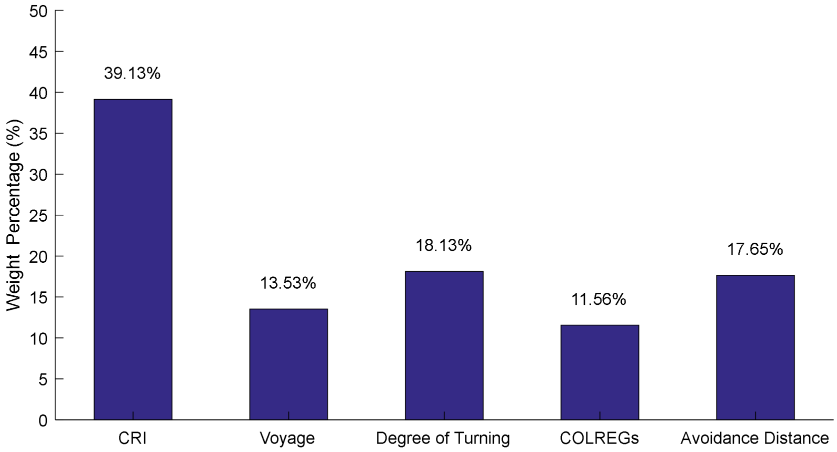

Figure 5. Weights of evaluation factors.

The process of the improved differential evolution algorithm in this paper is illustrated in Figure 6, and the pseudocode is provided in Algorithm 1.

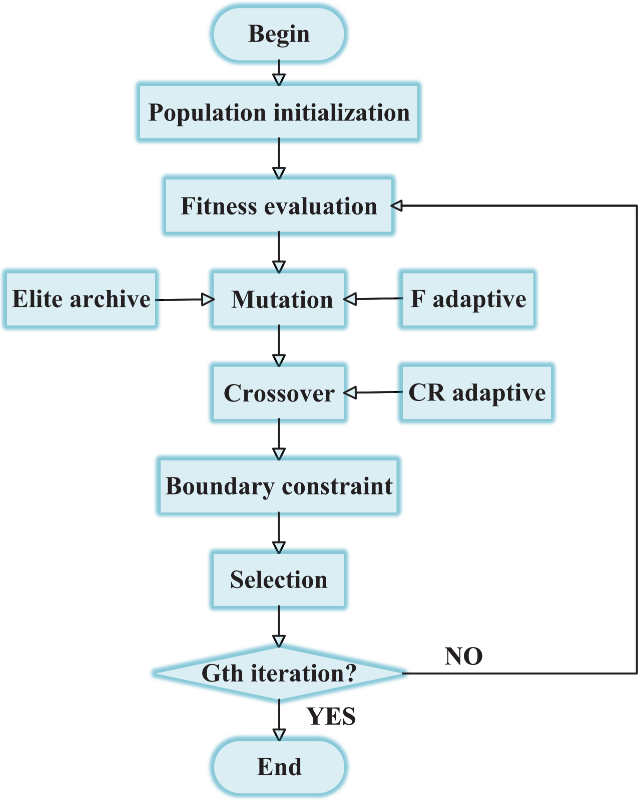

Figure 6. Algorithmic process.

Algorithm 1: Differential Evolution Algorithm

```
Input: population size $M$, crossover rate $CR$, scale factor $F$, maximum generation $K$
Output: optimal solution
for $i \leftarrow 1$ to $M$ do
    initialize population $X_i$
end
for $g \leftarrow 1$ to $K$ do
    for $i \leftarrow 1$ to $M$ do
        evaluate the fitness value $X_i$
    end
    for $i \leftarrow 1$ to $M$ do
        mutation operation;
        select 3 distinct individuals $a, b, c$ from elite population;
        select 2 distinct individuals $d, e$ from Non-elite population;
        $F_G = \begin{cases} F_1, F_0 \times \sum_{i=1}^{M} f(x_i^G) / \sum_{i=1}^{M} f(x_i^0) \leq F_1 \\ F_0 \times \frac{\sum_{i=1}^{M} f(x_i^G)}{\sum_{i=1}^{M} f(x_i^0)}, F_0 \times \sum_{i=1}^{M} f(x_i^G) / \sum_{i=1}^{M} f(x_i^0) > F_1 \end{cases}$
        generate trial vector $V_i = a + F(b - d) + F(c - e)$
        crossover operation;
        $CR_n = \begin{cases} CR_1, f(x_n^G) > f(x_{avg}^G) \\ CR_0 \times \frac{(CR_1 - CR_0)(f(x_{avg}^G) - f(x_n^G))}{f(x_{avg}^G) - f(x_{\min}^G)}, f(x_n^G) \leq f(x_{avg}^G) \end{cases}$
        $U_{i,j}(G) = \begin{cases} V_{i,j}(G), rand[0,1) < CRorj = jrand \\ X_{i,j}(G), otherwise \end{cases}$
        penalty mechanism;
        $U_{i,j}(G) = \min(\max(U_{i,j}(G), U_{i,j}^{\min}), U_{i,j}^{\max})$
        apply crossover operation to create offspring $U_i$;
        evaluate offspring $U_i$;
        if $fitness(U_i) < fitness(X_i)$ then
            replace $X_i$ with $U_i$;
        end
    end
end
```

4. Simulation Experiments

The simulation experiments were carried out in Matlab 2023b using standard hardware, specifically a computer with an Intel Core i5 CPU and 16 GB of RAM. The experimental environment consisted of open water with good visibility, where the effects of wind, waves, and currents were neglected. In the absence of external factors such as water currents, the difference between the ship's course and heading is considered negligible. Therefore, the course used in this model was effectively equivalent to the heading. To more accurately assess the effectiveness of the algorithms, the experiments simulated scenarios of two-ship encounters and four-ship encounters, incorporating static obstacles in the two-ship encounter scenario to evaluate the algorithm's obstacle avoidance capabilities. Considering the limitations of the TS in simulation experiments and to better assess the real-time avoidance capability of the OS, the TSs were set as stand-on ships while the OS acted as the give-way ship. The parameters and initial states of the experimental subjects are presented in Tables 6–8. The various parameters used during the experiment were uniformly set as follows: number of iterations of the algorithm K = 100, population size M = 50, and individual dimension n = 10.

Table 6. Ship parameters.

Parameter	OS	TS
Length Overall/m	41.38	153.80
Beam/m	7.20	23.20
Draft/m	3.2	8.200
Displacement/t	615.01	12,000.0
Water density/m^3	1.025	1.025

Table 7. Initial state of experimental subjects in the two-ship encounter.

Ship	Initial Heading/°	Initial Speed/kn	Distance from OS/n Mile
OS	135°	8	0
TS	0°	8	5.10
Obstacle	none	none	2.95

Table 8. Initial state of experimental subjects in the four-ship encounter.

Ship	Initial Heading/°	Initial Speed/kn	Distance from OS/n Mile
OS	225°	8	0
TS1	0°	8	6.10
TS2	85°	8	6.30
TS3	50°	8	7.00

Article 8 of the COLREGs explicitly states that if sufficient water is available, the most effective way to avoid a close-quarters situation might be to change course, as long as the change is performed in a timely and significant manner and does not lead to another close-quarters situation. Therefore, for the purposes of this study, when considering encounter scenarios and avoidance strategies, and in accordance with the rules of collision avoidance and the usual practices followed by crews when navigating at sea, this paper mainly adopted a change of course for avoidance, without slowing down or stopping.

Given the relatively few applications of the DE algorithm in USV path planning, this paper compared the improved differential evolution algorithm (AHP-ADE) with the improved particle swarm algorithm (I-PSO) [42], which includes adaptive improvements to the acceleration coefficients and introduces a shared learning factor to correct the speed update, which improves the global search ability and group collaboration ability of the PSO. The effectiveness of each algorithm was verified using factors such as safety, economy, and algorithm operation efficiency as evaluation indexes.

4.1. Simulation of Two-Ship Encounter

In ship collision scenarios, crossing situations constitute a significant proportion. Therefore, in the two-ship encounter scenario, the OS's heading was set to 135°, and the TS's heading was set to 000°, forming a crossing encounter situation, with an added static obstacle in the path. The results of the path planning are depicted in Figures 7 and 8, where Figure 7 displays the complete planned path and Figure 8 shows the ship's progress at specific time intervals. Figure 9 reflects the real-time distance between the OS and the TS and the static obstacles, providing data support for evaluating the safety of the path. In the simulation result diagrams, the blue ship represents the OS, the black ship represents the TS, the black hexagon is the static obstacle, the red path indicates the path planned by the AHP-ADE, the blue path is that planned by the I-PSO, the black line shows the TS's trajectory, and the black dashed line indicates the original path of the OS. Table 9 presents the simulation results of each algorithm. It includes the closest distances between the OS and the TS, and between the OS and the obstacle. These distances reflect the safety of the path. The total deviation distance between the planned and original paths, along with the maximum deviation distance, reflects the economic efficiency of the path. Additionally, the run-time reflects the operational efficiency of the algorithms.

The findings from the experiments indicate that the I-PSO performed poorly in terms of path safety. Although the closest distance to obstacles was comparable to that of the AHP-ADE, the closest distances to the TS, as shown in Figure 8 and Table 9, indicates that the path planned by the I-PSO involved periodic close encounters with the TS during navigation, increasing the risk of collision and thus lowering the safety. In contrast, the AHP-ADE ensured that a large safe distance was always maintained between the OS and the obstacle and the TS. The closest distance between the OS and the TS was approximately 1.98 nautical miles, remaining within a safe range and effectively ensuring navigational safety.

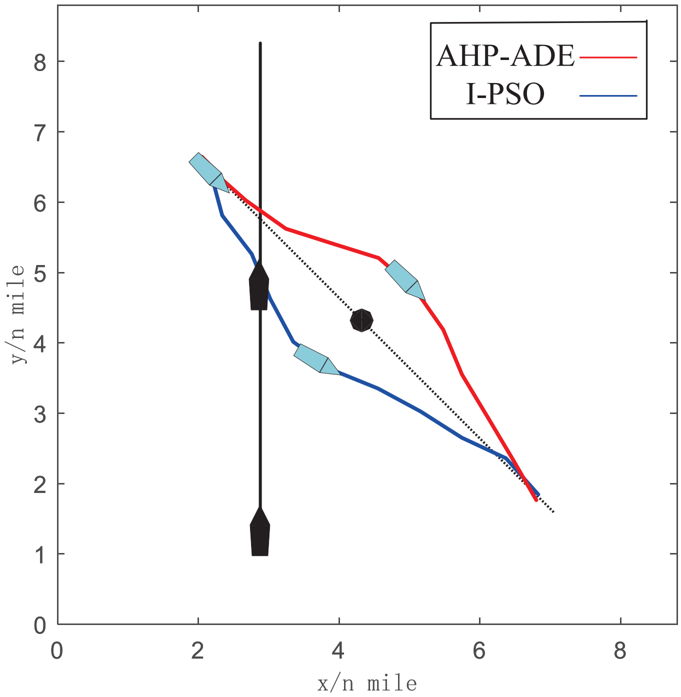

Figure 7. Simulation results of the two-ship encounter.

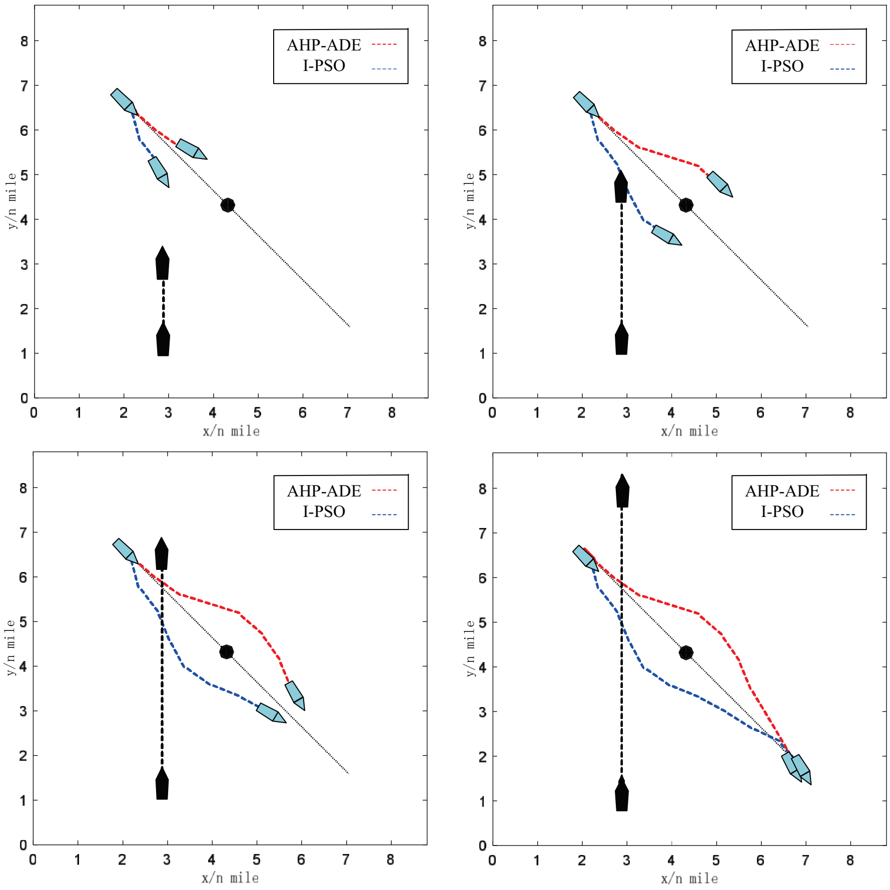

Figure 8. The state of the two-ship encounter ship at specific time intervals.

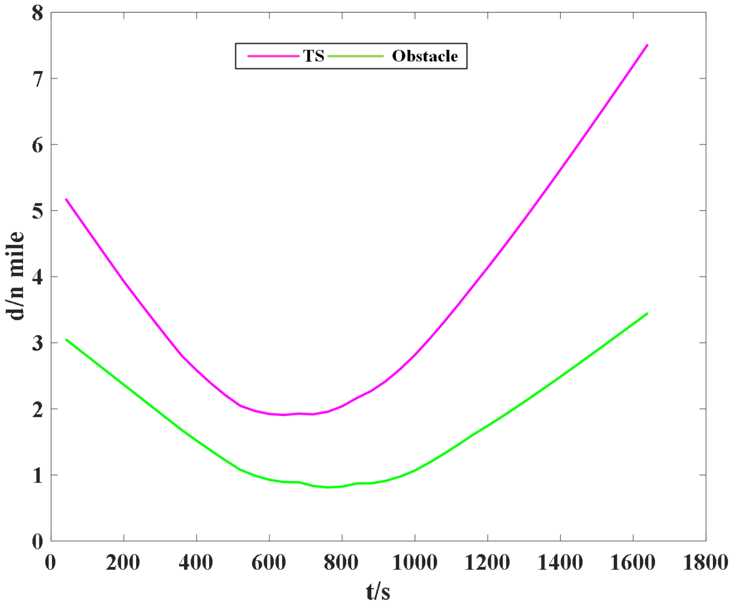

Figure 9. Real-time distance of the OS from the TS and static obstacles.

The initial position of the OS was 2.95 nautical miles from the obstacle, within a safe distance; thus, the ship should have continued along its original path, without needing to take evasive actions. However, the I-PSO algorithm caused the ship to deviate from its intended course right at the start of navigation. This premature maneuvering resulted in a significant deviation distance, thereby reducing the path's economic efficiency. In contrast, the AHP-ADE allowed the ship to continue along its original path during the initial phase, and it started taking evasive actions when approximately 2.1 nautical miles away from the obstacle. It effectively avoided collisions with the obstacle by adjusting the heading, and gradually returned to the original path after safely passing. The data in Table 9 show that the total deviation distance caused by the I-PSO was larger, indicating a lower economic efficiency, while the AHP-ADE had a relatively smaller total deviation distance, thus performing slightly better in terms of economy. In addition, the running time of the AHP-ADE proposed in this paper was shorter than that of the I-PSO, and the running efficiency was higher.

Table 9. Results of the operation of the algorithm in the two-ship encounter.

Algorithms	AHP-ADE	I-PSO
Min Dis to TS/n mile	1.978486	0.588198
Min Dis to obstacle/n mile	0.811146	0.810035
Sum deviation Dis/n mile	19.67959	21.50712
Max deviation Dis/n mile	0.854862	0.893951
Runtime/s	8.9933	9.4854

4.2. Simulation of Four-Ship Encounter

Similarly, in the four-ship encounter scenario, the OS's heading was set at 225°, while the TSs' headings were set at 000°, 085°, and 050°, respectively. The results of the path planning are depicted in Figures 10–12, displaying the real-time distances between the OS and each TS. The results of the simulation experiments are presented in Table 10.

Table 10. Results of the operation of the algorithm in the four-ship encounter.

Algorithms	AHP-ADE	I-PSO
Min Dis to TS1/n mile	2.308743	1.193513
Min Dis to TS2/n mile	0.459320	0.650301
Min Dis to TS3/n mile	1.081382	0.252936
Sum deviation Dis/n mile	16.14991	15.79813
Max deviation Dis/n mile	0.823841	0.794038
Runtime/s	14.7038	14.48

The findings from the experiments indicate that the AHP-ADE and the I-PSO performed comparably in terms of economic efficiency, and the total deviation distances of 16.15 nautical miles and 15.8 nautical miles guaranteed that both paths had good economy.

However, the I-PSO algorithm performed poorly in terms of path safety. According to the data in Figure 11 and Table 10, this algorithm caused the OS to pass ahead of the TS2 during their encounter, violating the COLREGs. Additionally, during the encounter with the TS3, the OS failed to maintain a safe distance, with the closest proximity being only 0.25 nautical miles, thereby increasing the risk of collision. In contrast, the AHP-ADE, during the encounter with TS2, adjusted the course, allowing the OS to pass from the stern, adhering to the COLREGs. Throughout the navigation, the algorithm also maintained greater safety distances with all TSs, effectively ensuring navigational safety.

In summary, the AHP-ADE and the I-PSO models were simulated and evaluated in scenarios of two-ship and four-ship encounters. The effectiveness of the algorithms was verified by comprehensively evaluating the total deviation distance, the maximum deviation distance, the nearest distances to the TS and the obstacles, and the algorithm's operation time. The findings from the experiments indicate that the AHP-ADE model showed better performance in the scenarios of two-ship and four-ship encounters, and at the same time conformed to the principle of early, largely, widely and clearly, thus effectively verifying the applicability and superiority of the AHP-ADE algorithm.

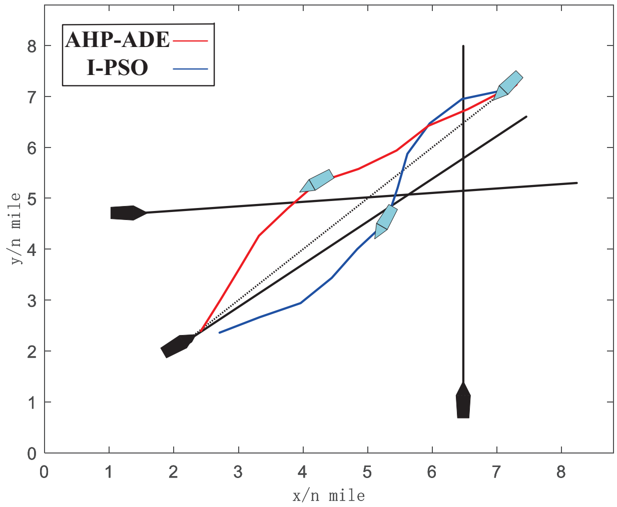

Figure 10. Simulation results of the four-ship encounter.

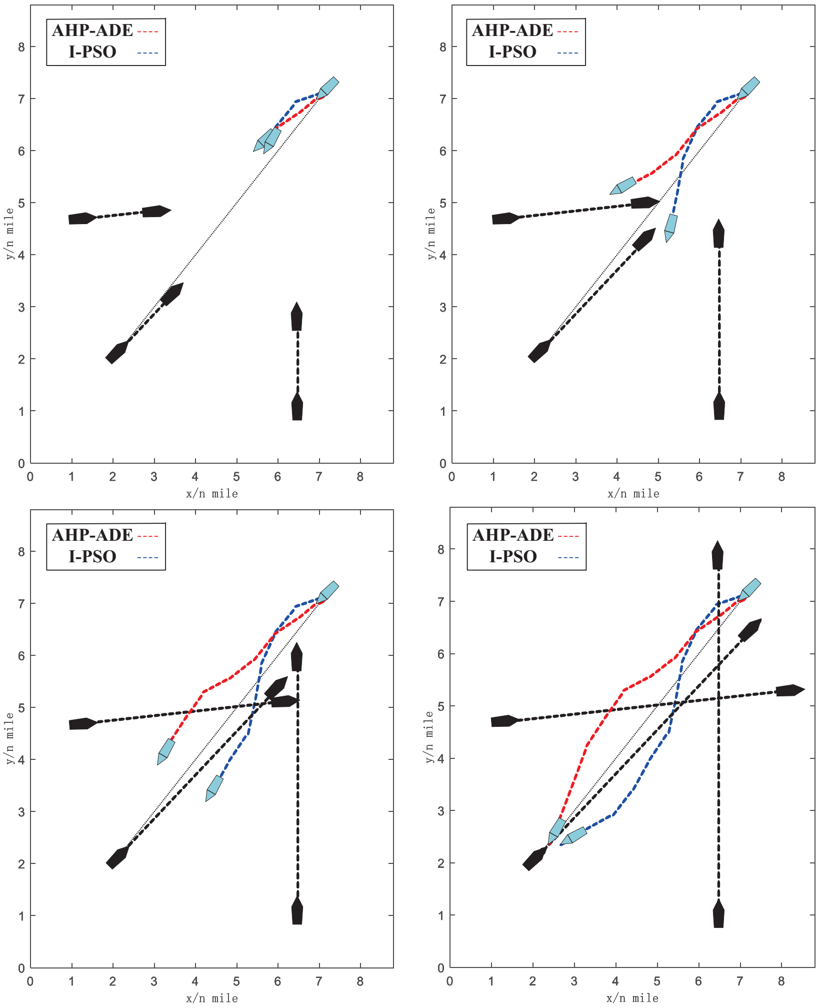

Figure 11. The state of a four-ship encounter ship at specific time intervals.

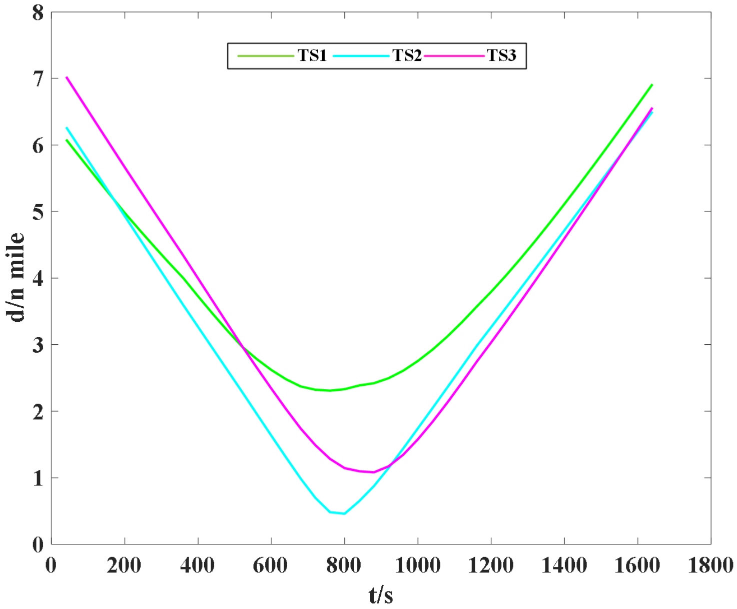

Figure 12. Real-time distance of the OS from the TSs.

5. Conclusions

In this paper, an adaptive differential evolutionary algorithm model combined with hierarchical analysis (AHP-ADE) was proposed for the path planning and collision avoidance problem of USVs in open water. This model combines the breadth of global search and the accuracy of local search by introducing an elite archiving strategy into the crossover operation. In order to enhance the efficiency of the algorithm, the control factors F and CR were adaptively improved. This adjustment increases the global search capabilities in the early stage to prevent premature convergence and enhances the local search capabilities in the later stage to improve the search accuracy. Additionally, this paper improved the traditional collision risk model by incorporating a limiting factor in the membership function by selecting the DCPA, aligning the CRI calculations more closely with maritime practices. The CRI, COLREGs, and optimal collision avoidance distance were incorporated into the evaluation of the fitness function. And the weights of each evaluation factor were obtained through hierarchical analysis, so as to more accurately assess the quality of the individual path points.

Simulation experiments in multi-ship encounter scenarios compared the AHP-ADE and I-PSO models. The total deviation distance and maximum deviation distance were used to evaluate the economy of the path, the closest distance to the target ship and obstacles was used to evaluate the safety of the path, and the running time was used to evaluate the operational efficiency of the algorithm. The simulation results showed that considering safety, economy, and operation efficiency, the AHP-ADE showed better performance in multi-ship encounter situations, which fully verified the effectiveness of the algorithm in this paper. The algorithm model in this study ignored the effects of factors such as wind, waves, and current. Therefore, the course used in this model was effectively equivalent to the heading. In the absence of external factors like water currents, the difference between a ship's heading and course is considered negligible. This simplification enhanced the computational efficiency and provided a feasible solution for the simulation experiments.

Although this study achieved certain results, there were still some limitations, as the ship's navigation in high-density waters, the ship's maneuverability, and the interference of external factors like wind, waves, and currents were not fully considered, which provide directions for further research on this algorithm.

Author Contributions: Conceptualization, Z.X. and B.H.; methodology, Z.X. and Z.L.; software, J.N. and B.L.; validation, J.N., B.L. and Z.L.; formal analysis, B.H.; investigation, Z.X.; resources, J.N. and B.L.; data curation, Z.X. and B.H.; writing—original draft preparation, Z.X. and B.H.; writing—review and editing, J.N., B.L. and Z.L.; visualization, B.L.; supervision, J.N., B.L. and Z.L.; project administration, J.N.; funding acquisition, J.N. and B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 51939001, 62371085, the Fundamental Research Funds for the Central Universities, grant number 3132024137, and the 2023 DMU navigation college first-class interdisciplinary research project, grant number 2023JXA09.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Campbell, S.; Naeem, W.; Irwin, G.W. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annu. Rev. Control.* **2012**, *36*, 267–283. [CrossRef]
2. Caccia, M.; Bibuli, M.; Bono, R.; Bruzzone, G.; Bruzzone, G.; Spirandelli, E. Unmanned surface vehicle for coastal and protected waters applications: The Charlie project. *Mar. Technol. Soc. J.* **2007**, *41*, 62–71. [CrossRef]

3. Bălănescu, M.; Suciu, G.; Bădicu, A.; Birdici, A.; Pasat, A.; Poenaru, C.; Zătreanu, I. Study on unmanned surface vehicles used for environmental monitoring in fragile ecosystems. In Proceedings of the 2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME), Pitesti, Romania, 21–24 October 2020; pp. 94–97.
4. Cao, H.; Guo, Z.; Wang, S.; Cheng, H.; Zhan, C. Intelligent wide-area water quality monitoring and analysis system exploiting unmanned surface vehicles and ensemble learning. *Water* **2020**, *12*, 681. [CrossRef]
5. Chang, Y.; Cui, Y.; Ren, J. Research on the Simulation Method of the Drift Trajectory of Persons Overboard Under the Sea Search Mode of Unmanned Surface Vehicle (USV). In *Advanced Intelligent Technologies for Industry*; Springer: Singapore, 2022; pp. 321–329.
6. Murphy, R.R.; Steimle, E.; Griffin, C.; Cullins, C.; Hall, M.; Pratt, K. Cooperative use of unmanned sea surface and micro aerial vehicles at Hurricane Wilma. *J. Field Robot.* **2008**, *25*, 164–180. [CrossRef]
7. Goulon, C.; Le Meaux, O.; Vincent-Falquet, R.; Guillard, J. Hydroacoustic Autonomous boat for Remote fish detection in LakE (HARLE), an unmanned autonomous surface vehicle to monitor fish populations in lakes. *Limnol. Oceanogr. Methods* **2021**, *19*, 280–292. [CrossRef]
8. Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D.; Khan, A. A constrained A* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents. *Ocean. Eng.* **2018**, *169*, 187–201. [CrossRef]
9. Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D.; Khan, A. Feasibility study of a constrained Dijkstra approach for optimal path planning of an unmanned surface vehicle in a dynamic maritime environment. In Proceedings of the 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Torres Vedras, Portugal, 25–27 April 2018; pp. 117–122.
10. Lazarowska, A. Ship's trajectory planning for collision avoidance at sea based on ant colony optimisation. *J. Navig.* **2015**, *68*, 291–307. [CrossRef]
11. Ntakolia, C.; Kladis, G.P.; Lyridis, D.V. A fuzzy logic approach of pareto optimality for multi-objective path planning in case of unmanned surface vehicle. *J. Intell. Robot. Syst.* **2023**, *109*, 21. [CrossRef]
12. MahmoudZadeh, S.; Abbasi, A.; Yazdani, A.; Wang, H.; Liu, Y. Uninterrupted path planning system for Multi-USV sampling mission in a cluttered ocean environment. *Ocean Eng.* **2022**, *254*, 111328. [CrossRef]
13. Praczyk, T.; Szymak, P. Using genetic algorithms to fix a route for an Unmanned Surface Vehicle. In Proceedings of the 2012 17th International Conference on Methods & Models in Automation & Robotics (MMAR), Miedzyzdroje, Poland, 27–30 August 2012; pp. 487–492.
14. Ning, J.; Chen, H.; Li, T.; Li, W.; Li, C. COLREGs-compliant unmanned surface vehicles collision avoidance based on multi-objective genetic algorithm. *IEEE Access* **2020**, *8*, 190367–190377. [CrossRef]
15. Chen, X.; Liu, Y.; Hong, X.; Wei, X.; Huang, Y. Unmanned ship path planning based on RRT. In Proceedings of the Intelligent Computing Theories and Application: 14th International Conference, Part I, Wuhan, China, 15–18 August 2018; pp. 102–110.
16. Luan, T.; Tan, Z.; You, B.; Sun, M.; Yao, H. Path planning of unmanned surface vehicle based on artificial potential field approach considering virtual target points. *Trans. Inst. Meas. Control.* **2024**, *46*, 1190–1202. [CrossRef]
17. Lin, X.; Fu, Y. Research of USV obstacle avoidance strategy based on dynamic window. In Proceedings of the 2017 IEEE International Conference on Mechatronics and Automation (ICMA), Kagawa, Japan, 6–9 August 2017; pp. 1410–1415.
18. Chen, C.; Ma, F.; Liu, J.L.; Yan, X.P.; Chen, X.Q. A novel path planning approach for unmanned ships based on deep reinforcement learning. In *Data Science and Knowledge Engineering for Sensing Decision Support*; World Scientific: Singapore, 2018; pp. 626–633.
19. Guo, S.; Zhang, X.; Zheng, Y.; Du, Y. An autonomous path planning model for unmanned ships based on deep reinforcement learning. *Sensors* **2020**, *20*, 426. [CrossRef] [PubMed]
20. Xing, B.; Wang, X.; Yang, L.; Liu, Z.; Wu, Q. An algorithm of complete coverage path planning for unmanned surface vehicle based on reinforcement learning. *J. Mar. Sci. Eng.* **2023**, *11*, 645. [CrossRef]
21. Zhang, H.; Tao, Y.; Zhu, W. Global path planning of unmanned surface vehicle based on improved A-Star algorithm. *Sensors* **2023**, *23*, 6647. [CrossRef]
22. Mao, S.; Yang, P.; Gao, D.; Bao, C.; Wang, Z. A motion planning method for unmanned surface vehicle based on improved rrt algorithm. *J. Mar. Sci. Eng.* **2023**, *11*, 687. [CrossRef]
23. Wang, J.; Wang, R.; Lu, D.; Zhou, H.; Tao, T. USV dynamic accurate obstacle avoidance based on improved velocity obstacle method. *Electronics* **2022**, *11*, 2720. [CrossRef]
24. Guo, H.; Mao, Z.; Ding, W.; Liu, P. Optimal search path planning for unmanned surface vehicle based on an improved genetic algorithm. *Comput. Electr. Eng.* **2019**, *79*, 106467. [CrossRef]
25. Xu, D.; Yang, J.; Zhou, X.; Xu, H. Hybrid path planning method for USV using bidirectional A* and improved DWA considering the manoeuvrability and COLREGs. *Ocean Eng.* **2024**, *298*, 117210. [CrossRef]
26. Yang, C.; Pan, J.; Wei, K.; Lu, M.; Jia, S. A Novel Unmanned Surface Vehicle Path-Planning Algorithm Based on A* and Artificial Potential Field in Ocean Currents. *J. Mar. Sci. Eng.* **2024**, *12*, 285. [CrossRef]
27. Li, L.; Wu, D.; Huang, Y.; Yuan, Z.M. A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. *Appl. Ocean Res.* **2021**, *113*, 102759. [CrossRef]
28. HuChunAn.; WenHao. A differential evolution SAF-DE algorithm which jumps out of local optimal. In Proceedings of the 2020 16th International Conference on Computational Intelligence and Security (CIS), Guangxi, China, 27 October–30 November 2020; pp. 333–337.

29. Tien, C.H.; Hsu, C.Y.; Chen, M.H.; Chang, P.C. Differential evolutionary algorithms with novel mutation operator for solving the permutation flowshop scheduling problem. In Proceedings of the 2015 International Conference on Control, Automation and Robotics, Singapore, 20–22 May 2015; pp. 191–194.
30. Jitkongchuen, D. A hybrid differential evolution with grey wolf optimizer for continuous global optimization. In Proceedings of the 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE), Chiang Mai, Thailand, 29–30 October 2015; pp. 51–54.
31. Seo, C.; Noh, Y.; Abebe, M.; Kang, Y.J.; Park, S.; Kwon, C. Ship collision avoidance route planning using CRI-based A* algorithm. *Int. J. Nav. Archit. Ocean Eng.* **2023**, *15*, 100551. [CrossRef]
32. Hu, Y.; Zhang, A.; Tian, W.; Zhang, J.; Hou, Z. Multi-ship collision avoidance decision-making based on collision risk index. *J. Mar. Sci. Eng.* **2020**, *8*, 640. [CrossRef]
33. Wang, N. An intelligent spatial collision risk based on the quaternion ship domain. *J. Navig.* **2010**, *63*, 733–749. [CrossRef]
34. Tsou, M.C. Multi-target collision avoidance route planning under an ECDIS framework. *Ocean Eng.* **2016**, *121*, 268–278. [CrossRef]
35. Kang, Y.T.; Chen, W.J.; Zhu, D.Q.; Wang, J.H.; Xie, Q.M. Collision avoidance path planning for ships by particle swarm optimization. *J. Mar. Sci. Technol.* **2018**, *26*, 3.
36. Storn, R.; Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
37. Fujii, Y.; Tanaka, K. Traffic capacity. *J. Navig.* **1971**, *24*, 543–552. [CrossRef]
38. Goodwin, E. A statistical study of ship domains. *J. Navig.* **1973**, *26*, 130. [CrossRef]
39. Wang, S.; Li, Y.; Yang, H.; Liu, H. Self-adaptive differential evolution algorithm with improved mutation strategy. *Soft Comput.* **2018**, *22*, 3433–3447. [CrossRef]
40. Zhang, B.; Sun, X.; Liu, S.; Deng, X. Adaptive differential evolution-based distributed model predictive control for multi-UAV formation flight. *Int. J. Aeronaut. Space Sci.* **2020**, *21*, 538–548. [CrossRef]
41. Ke, Q.; Renxiang, B.; Yong, L.; Kaige, D. Ship collision avoidance timing based on subjective collision risk. In Proceedings of the 2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 3–5 September 2018; pp. 170–173.
42. Guo, X.; Ji, M.; Zhao, Z.; Wen, D.; Zhang, W. Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm. *Ocean Eng.* **2020**, *216*, 107693. [CrossRef]

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