

## Research Article

# A Collision Avoidance Method for Intelligent Ship Based on the Improved Bacterial Foraging Optimization Algorithm

Xingzhong Wang , Xinghua Kou, Jinfeng Huang , and Xianchun Tan

*China Ship Development and Design Center, Wuhan, China*

Correspondence should be addressed to Jinfeng Huang; [howardhjf@163.com](mailto:howardhjf@163.com)

Received 7 October 2020; Revised 4 January 2021; Accepted 27 January 2021; Published 10 February 2021

Academic Editor: Keigo Watanabe

Copyright © 2021 Xingzhong Wang 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.

The bacterial foraging optimization algorithm (BFOA) is an intelligent population optimization algorithm widely used in collision avoidance problems; however, the BFOA is inappropriate for the intelligent ship collision avoidance planning with high safety requirements because BFOA converges slowly, optimizes inaccurately, and has low stability. To fix the above shortcomings of BFOA, an autonomous collision avoidance algorithm based on the improved bacterial foraging optimization algorithm (IBFOA) is demonstrated in this paper. An adaptive diminishing fractal dimension chemotactic step length is designed to replace the fixed step length to achieve the adaptive step length adjustment, an optimal swimming search method is proposed to solve the invalid searching and repeated searching problems of the traditional BFOA, and the adaptive migration probability is developed to take the place of the fixed migration probability to prevent elite individuals from being lost in BOFA. The simulation of benchmark tests shows that the IBFOA has a better convergence speed, optimized accuracy, and higher stability; according to a collision avoidance simulation of intelligent ships which applies the IBFOA, it can realize the autonomous collision avoidance of intelligent ships in dynamic obstacles environment is quick and safe. This research can also be used for intelligent collision avoidance of automatic driving ships.

## 1. Introduction

The intelligent navigation is the future development direction of ships and an important part of intelligent ships. It adopts the computer technology, control technology, etc. to analyze and process the sensed and acquired information, design and optimize the ship's route and speed, and navigate autonomously by achieving automatic collision avoidance in open waters, narrow waterways, and complex environmental conditions with the help of shore-based support center [1]. Autonomous collision avoidance planning is an important technology in the field of intelligent navigation [2], the foundation for a ship to perform tasks, and reflects its intelligence [3]. Partial autonomous collision avoidance means that the intelligent ship can effectively and quickly adjust its speed and course and avoid floating obstacles on the water, such as ships and mobile objects, autonomously and safely [4, 5].

At present, experts and scholars principally achieve the collision avoidance planning of intelligent ships with partial planning and intelligent algorithms [6]. An autonomous collision avoidance method for ships, which was put forward by Kuwata and his team and based on speed obstacles in a dynamic environment, has included the Convention on the International Regulations for Preventing Collisions at Sea (COLREGS) and established a collision-avoidance formula [7], but it results in a slow collision avoidance since the collision avoidance formula incorporates dynamic and static obstacles and fails to simplify the rule, which leads to a complex algorithm. Referring to the classic ant colony algorithm, Song has carried out a grid modeling intelligent ship path planning in a complex environment [8]; however, optimization efficiency is low because of the limitation by the classic ant colony algorithm [9]. Zhang et al. came up with a new adaptive intelligent ship avoidance algorithm based on the reinforcement learning of collision avoidance rules. The

algorithm can guide the intelligent ship to avoid obstacles through the local obstacle avoidance module and the adaptive learning module, but the learning process of the algorithm is too complicated and difficult to plan the path in a dynamic and complex environment [10]. Liu [11] suggested a nonlinear decreasing step size, improved the replication and migration, and increased the global optimization ability and convergence speed of the BFOA, but the stability of the algorithm was poor. Scholars later indicated a bacterial foraging algorithm with adaptive fractal dimension step size, which can optimize quickly, but its stability and accuracy are poor [12]. Afterward, scholars optimized the path of the intelligent ship with the improved ant colony algorithm. The parameter is improved by adding the direction angle weight to the selection probability, and then, the algorithm can converge faster and its searchability is enhanced; however, the accuracy and stability of later optimization are still low [13]. Chen [14] improved the artificial potential field method by amending the gravitational and repulsive potential field functions and conducted the path planning of the intelligent ship, but a local minimum point problem still existed. Liu [15] used the fast-travel method to plan the collision avoidance of intelligent ships, but the generated path is too far away from obstacles and too long. The velocity obstacle (VO) approach was first used in the robotic field and now is widely applied in the collision avoidance of intelligent ship [16]. According to the method, a cone-shaped space is generated on the obstacles and an intelligent ship outside the space, which is impossible to crash. Since the method is efficient, a combination of VO algorithm and COLREGS rules is now the direction of many scholars' researches [17–19]. Liu [20] and Ma [21] illustrate two particle swarm bacterial foraging intelligent algorithm in ship collision avoidance, but the motion of obstacles in the references is simple uniform rectilinear motion which has no change in velocity and heading and has the shortcomings of slow convergence speed, low optimization precision, and low stability. Yoshiaki [22] presented a theory to generate conic obstacles in the velocity space of the intelligent ship and measure the risk of collision in different conic zones based on the COLREGS; the results of the collision avoidance experiment in a multiobstacle environment show that the method has a considerable success rate, and the convergence speed can be improved a lot. In view of the existing shortcomings, the author proposed the intelligent ship autonomous collision avoidance algorithm with IBFOA, which improves the chemotaxis step size, swimming, and migration operations of the traditional BFOA. The simulation of the function test shows that the convergence speed, optimization accuracy, and stability of the improved algorithm are improved. The result of collision avoidance simulation of smart ships after IBFOA is used and shows that the improved algorithm can realize the autonomous collision avoidance of smart ships under dynamic obstacles quickly and safely.

In Section 2, the motion model based on the VO algorithm was briefly introduced. In Section 3, the IBFOA is discussed in detail, and the effectiveness of the algorithm was verified by simulation experiments and the results and detail analysis are presented in Section 4. Section 5 gives the conclusion of the article.

## 2. Collision Avoidance Model Based on Speed Obstacles

The automatic identification system (AIS) is developed from the identification friend-or-foe system ships and aircraft, and it cooperates with the global positioning system (GPS) to integrate static data such as the ship's name, call sign, length, width, draft, ship code, communication service identification code, ship type, dangerous goods, and dynamic information such as ship position (latitude and longitude), ship speed, and rate of changing course and course. Broadcasting to shore stations and vessels in adjacent water via very high frequency (VHF) enables local ships to grasp the dynamic and static information of nearby ships in real time and then takes necessary measures to avoid collisions, which is an advanced navigation aid system and has a higher accuracy compared with radar. The intelligent ship considered in this paper obtains the motion information of the obstacle ship through the AIS system, so no observation error exists. As shown in Figure 1, given an intelligent ship with shape A and a moving ship with shape B and their velocity vectors to be  $\mathbf{v}_R$  and  $\mathbf{v}_O$ , respectively. The identification zone of the intelligent ship is defined as C, which is determined by the AIS and the reaction demand for avoidance. The domain of the moving ship is D, which is used to represent the area of an obstacle that needs to be kept away from.  $\mathbf{A} \in \mathbf{C}$  and  $\mathbf{B} \in \mathbf{D}$ . Both C and D are considered as circles because circles are more convenient for designing the obstacle avoidance algorithm, and the moving parameters of the intelligent ship can be calculated faster. As D enters C, the intelligent ship should decide whether and how to avoid the obstacle.

The motion model based on the VO method is shown in Figure 2.  $R$  stands for the intelligent ship; the circle  $O$  stands for the domain around the obstacle. According to Figure 2, the velocity vector of the intelligent ship and the obstacle,  $\mathbf{v}_R$  and  $\mathbf{v}_O$ , could also be represented as  $oX$  and  $(v_o, \beta)$ , respectively.  $\alpha = \sphericalangle(\mathbf{X}, \mathbf{v}_R)$ ,  $\beta = \sphericalangle(\mathbf{X}, \mathbf{v}_O)$ ,  $\theta = \sphericalangle(\mathbf{X}, \mathbf{RO})$ ,  $\psi = \sphericalangle(\mathbf{X}, \Delta\mathbf{v})$ ,  $\gamma = \sphericalangle(\mathbf{RO}, \Delta\mathbf{v})$ ,  $\varphi = \sphericalangle(\Delta\mathbf{v}, \mathbf{v}_R)$ ,  $\eta = \sphericalangle(\mathbf{v}_O, \Delta\mathbf{v})$ ,  $\mu = \sphericalangle(\mathbf{RO}, \mathbf{RT})$ , and  $\lambda = \sphericalangle(\mathbf{v}_O, \mathbf{v}_R)$ , where " $\sphericalangle(\mathbf{a}, \mathbf{b})$ " donates the angle that the vector  $\mathbf{a}$  rotates to the vector  $\mathbf{b}$  through the minor arc side, and it can be either positive or negative depending on whether the rotation is in the anticlockwise direction or clockwise direction.  $\sphericalangle(\mathbf{c}, \mathbf{d})$  donates an invariably positive angle, and it represented the angle between the vector  $\mathbf{c}$  and the vector  $\mathbf{d}$ .  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$ , and  $\mathbf{d}$  are variables used to state the symbols  $\sphericalangle$  and  $\sphericalangle$ , and they are replaced by specific points in the figure to express a certain meaning.  $\mathbf{RT}$  ( $\mathbf{RT}_1$  and  $\mathbf{RT}_2$ ) represents two tangents from  $R$  to the obstacle  $O$ .

Figure 2 shows that the intelligent ship is impossible to collide with the obstacle if  $\psi$ . Based on the velocity vector triangle formed by  $\mathbf{v}_R$ ,  $\mathbf{v}_O$ , and  $\Delta\mathbf{v}$ , we get

$$\begin{cases} v_o \sin(\alpha - \beta) = -\Delta v \sin \varphi, \\ v_R - v_o \cos(\alpha - \beta) = \Delta v \cos \varphi, \\ v_R^2 + v_o^2 - 2v_R v_o \cos(\alpha - \beta) = \Delta v^2, \\ v_R \sin(\beta - \alpha) = \Delta v \sin \eta, \\ v_o - v_R \cos(\beta - \alpha) = \Delta v \cos \eta. \end{cases} \quad (1)$$

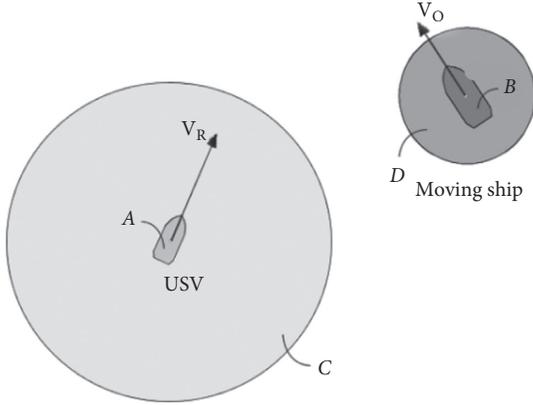


FIGURE 1: The motion model of the intelligent ship and an obstacle boat.

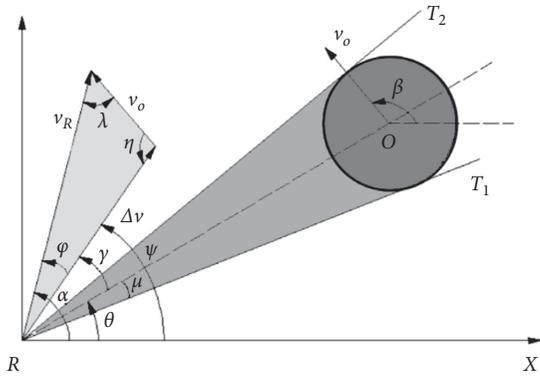


FIGURE 2: The intelligent ship and the velocity obstacle.

From the above equation set, the following equation can be derived:

$$\gamma = \tan^{-1} \frac{v_R \sin(\alpha - \theta) - v_o \sin(\beta - \theta)}{v_R \cos(\alpha - \theta) - v_o \cos(\beta - \theta)}. \quad (2)$$

Taking a derivative of equation (2), the following equation is obtained:

$$\Delta\gamma = \frac{\sin \phi}{\Delta v} \Delta v_R + \frac{v_R \cos \phi}{\Delta v} \Delta\alpha + \frac{-\sin \eta}{\Delta v} \Delta v_o + \frac{v_o \cos \eta}{\Delta v} \Delta\beta. \quad (3)$$

So, if the obstacle velocity  $v_o$ , course  $\beta$ , and their changes  $\Delta v_o$  and  $\Delta\beta$  are acquired from the AIS data,  $\Delta\gamma$  could be altered by adjusting the intelligent ship velocity  $v_R$  and the course  $\alpha$ .

The aim of the algorithm is to adjust  $\gamma$  to satisfy  $|\gamma + \Delta\gamma| \geq \mu$ .

The collision avoidance schemes are described as

$$\begin{cases} \Delta\gamma \geq \mu - \gamma, & \text{if } \gamma > 0, \\ \Delta\gamma \leq -\mu - \gamma, & \text{if } \gamma < 0. \end{cases} \quad (4)$$

### 3. The Autonomous Collision Avoidance Algorithm Established on IBFOA

**3.1. Basic Principles of the Bacterial Foraging Optimization Algorithm (BFOA).** The bacterial foraging optimization algorithm (BFOA) is a new type of swarm intelligence optimization algorithm proposed by K.M. Passino based on *Escherichia coli* (E. coli) foraging behavior in 2002. This algorithm has the advantages of global optimization, easy to jump out of local optimization, parallel search, and strong robustness; in the BFO algorithm, the position of bacteria in the search space corresponds to the solution of the optimization problem, and the fitness value of the optimization function, which is the objective function, represents the optimization degree solution. The BFOA solves the problem by simulating the four behaviors of bacteria: chemotaxis, aggregation, replication, and migration:

- (1) Chemotaxis: the behavior that bacteria tend to the eutrophic area when foraging is called chemotaxis. During the chemotaxis, the bacteria will perform two kinds of movement: veer and swimming. The veer is to move a unit step in a new direction, that is, turn to a new direction; if the fitness value is improved after turning, move a few steps in this direction until the fitness value no longer improves, which is called swimming. The mathematical expression of chemotaxis is shown as follows:

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (5)$$

In the formula,  $\theta^i(j, k, l)$  indicates the position of the bacteria  $i$  after chemotaxis at the  $j$ -th time, replicated at the  $k$ -th time, and dispersed after the  $l$ -th time;  $C(i)$  is the chemotaxis step of bacteria; and  $\Delta(i)$  is a unit vector of bacteria in a random direction in the search space.

- (2) Aggregation: when bacteria are foraging, there are gravitational and repulsive forces between individuals. Gravity causes bacteria to gather more nutrient-rich areas, and repulsion forces each bacterium to have a certain foraging area in the search space. The algorithm expresses bacterial aggregation behavior through the following formula:

$$\begin{aligned} J_{cc}(\theta, P(j, k, l)) &= \sum_{i=1}^S J_{cc}(\theta, \theta^i(j, k, l)) \\ &= \sum_{i=1}^S \left[ -d_a \exp\left(-w_a \sum_{m=1}^p (\theta_m - \theta_i^m)^2\right) \right] \\ &\quad + \sum_{i=1}^S \left[ -h_r \exp\left(-w_r \sum_{m=1}^p (\theta_m - \theta_i^m)^2\right) \right]. \end{aligned} \quad (6)$$

In the formula,  $d_a$  is the depth of gravity,  $w_a$  is the width of gravity,  $h_r$  is the height of repulsion,  $w_r$  is the width of repulsion,  $\theta_i^m$  is the  $m$ -th component of bacteria  $i$ , and  $\theta_m$  is the  $m$ -th component of all other bacteria.

- (3) Replication: bacterial evolution obeys “survival of the fittest,” the weaker foraging ability half of the bacteria (referring to the fitness value, that is, the objective function value is large because the article seeks the minimum value of the objective function) will be eliminated. Another half of the bacteria will replicate:

$$J_{\text{health}}^i = \sum_{i=1}^{N_c+1} J(i, j, k, l). \quad (7)$$

The formula (7) is called the health function (energy function) of bacteria  $i$ . The larger the value, the stronger the foraging ability of bacteria  $i$ . After sorting  $J_{\text{health}}^i$ , the algorithm will discard half of the bacteria with less health and copy the other half with greater health. This behavior improves the foraging ability of bacteria on the premise that the number of bacteria remains unchanged and is more conducive to the algorithm to find the global optimal solution.

- (4) Migration: if the environment of the bacteria gradually changes or mutates, the bacteria would randomly migrate to a new area to cope with this change. The algorithm is given a migration probability  $P_{\text{ed}}$  to simulate this behavior, and any individual bacteria will die and generate new bacteria in a new area when the probability is satisfied. This behavior is conducive to the algorithm to jump out of the local optimal solution.

### 3.2. Improvement of Bacterial Foraging Optimization Algorithm (IBFOA)

**3.2.1. Adaptive Decreasing Fractal Dimension Chemotaxis Step Size.** The chemotaxis step length of the basic BFOA is a constant, so the step length is completely the same in the algorithm iteration process, which affects the convergence speed and optimization accuracy of the algorithm. Two improvement approaches for constant step length are shown in the following equations:

$$C = C_{\min} + (C_{\max} - C_{\min}) \times \cos\left(\frac{\text{gen}}{\text{gen}_m} \pi\right). \quad (8)$$

In the formula,  $C$  is the step length,  $\text{gen}$  is the iterations, and  $\text{gen}_m$  is the maximum iterations:

$$C_d = \cos\left(2\pi\left(\frac{\text{angle}}{360}\right)\right) \times \left(\frac{\text{step}_d}{j+k+l}\right). \quad (9)$$

In the formula,  $C_d$  is the  $d$ -th dimension step length,  $\text{angle}$  is the random number on  $[0, 360]$ ,  $\text{step}_d$  is the original step length of  $d$ -th dimension, and  $j$ ,  $k$ , and  $l$  are current

chemotaxis counts, replication counts, and migration counts, respectively.

The improved chemotaxis step size shown in formula (8) depends on the choice of  $C_{\max}$  and  $C_{\min}$ ; however, a good empirical formula has not been developed for the selection of  $C_{\min}$ , the decreasing change based on cosine is not obvious, and the improvement of the algorithm convergence speed and optimization accuracy is limited. Equation (9) limits the change in step size reduction with  $j+k+l$  and adopts the  $\cos(2\pi(\text{angle}/360))$  control the optimizing forward direction; as a result, a lot of useless calculations are carried out, which makes the algorithm convergence speed decrease. In this paper, the above improvements are integrated according to the idea of adaptive decrement: abandon  $C_{\min}$  and choose  $C_{\max}$  based on experience as a quarter of the  $d$ -th dimension optimization space range; change the decremental change to the multiplication, which makes the step changes faster in the early stage, and approach the optimal value quickly, and more attention would be paid to the small local optimization in the later stage, which can search more accurate; the random step length control of the  $\text{rand}$  function is used to replace the direction control of equation (9), which greatly reduces the computational complexity of the algorithm. The improved self-adaptive decreasing fractal dimension chemotaxis step length formula is shown in the following formula:

$$C_d = \left(\frac{C_{\max}}{j \times k \times l}\right) \times \text{rand}(). \quad (10)$$

In the formula,  $C_{\max} = ((\text{MAX}_d - \text{MIN}_d)/4)$ ,  $\text{MAX}_d$  is the maximum value of the  $d$ -dimensional optimization range,  $\text{MIN}_d$  is the minimum value of the  $d$ -th dimension optimization range,  $\text{rand}$  is a random number between  $[0, 1]$ , and  $j$ ,  $k$ , and  $l$  are current chemotaxis counts, replication counts, and migration counts, respectively.

The improved chemotaxis step length allows each bacterium to advance in different dimensions with different step lengths, the early search efficiency is high, and the later search accuracy is high, which improves the convergence speed and optimization accuracy of the algorithm.

**3.2.2. Optimal Swimming Search Method.** The basic BFOA will judge the fitness value after each step of the swimming. If the new fitness value at the new position is increased, the swimming would continue. Although this kind of movement can theoretically perform the algorithm, it has an inevitable shortcoming: in an iteration, once the bacteria swim to a disadvantaged position, they cannot return to the superior position, nor continue swimming, or repeat swimming at the same position, which makes the algorithm convergence slower and the calculation is wasted.

To deal with the above problem, an optimal swimming search method is demonstrated in this article. During the swimming, the bacteria would detect the fitness value of the next swimming position, which is called the intelligent search; if the fitness value is better than the value of the previous position, the bacteria will move to the new position, or it would stop swimming. This method can prevent

bacteria from moving to a bad position and increase the calculation speed greatly; the improved swimming method is shown in Figure 3.

**3.2.3. Adaptive Migration.** The migration probability of the traditional BFOA is a certain value  $P_{ed}$ , and the bacteria of different health levels are all migrated according to  $P_{ed}$ , which may force the bacteria near the optimal position to migrate. This situation would cause the loss of elite individuals, solution degradation, and the decreasing convergence speed, accuracy, and stability. An adaptive migration probability referring to the idea of roulette is proposed in this article, as shown in the following equation:

$$P_{self}(i) = \frac{J_{health}^{max} - J_{health}^i}{J_{health}^{max} - J_{health}^{min}} \times P_{ed}. \quad (11)$$

Adopting the adaptive migration probability makes the migration probability of bacteria with a higher health degree smaller, and the probability of bacteria migration with a lower health degree higher. According to the roulette theory, the bacteria with the smallest  $J_{health}$  would migrate, which greatly increases the ability for searching the optimal swimming, and the convergence speed, accuracy, and the stability of the algorithm.

**3.3. Developing the Collision Avoidance Strategy.** In the actual process of autonomous collision avoidance of intelligent ships, it is not easy to achieve collision avoidance by simply changing the course or speed, so the speed and course of the smart ship are simultaneously adjusted to avoid obstacles. When avoiding obstacles, the IBFOA is applied to solve the optimal  $dv_R$  and  $d\alpha$  that satisfy formula (5) at any collision avoidance timing, that is, to solve the optimization problem, as follows:

$$\begin{cases} f(x_1, x_2) = \min(m_1 x_1 + m_2 x_2), \\ \frac{\sin \varphi}{\Delta v} x_1 + \frac{v_R \cos \varphi}{\Delta v} x_2 \geq \mu - \gamma, & \text{if } \gamma \geq 0, \\ \frac{\sin \varphi}{\Delta v} x_1 + \frac{v_R \cos \varphi}{\Delta v} x_2 \leq -\mu - \gamma, & \text{if } \gamma < 0. \end{cases} \quad (12)$$

In the formula,  $x_1 = dv_R$  and  $x_2 = d\alpha$  are the unknowns we need to figure out, and other parameters are known quantities.  $f(x_1, x_2)$  is the objective function (fitness function), which is defined as the comprehensive change in the speed and direction of the intelligent ship in each cycle. The smaller the change, the smoother the movement, so under the premise of ensuring the safety of collision avoidance, the lower the fitness value, the better the outcome.  $m_1, m_2$  is the weight of  $x_1, x_2$  in the objective function, and the value of  $m_1, m_2$  reflects the priority of speed adjustment and course adjustment; since the changing course is more effective in avoidance,  $m_2$  is much larger than  $m_1$ .

The IBFOA is used to develop the collision avoidance strategy, and the optimal  $dv_R$  and  $d\alpha$  should follow the

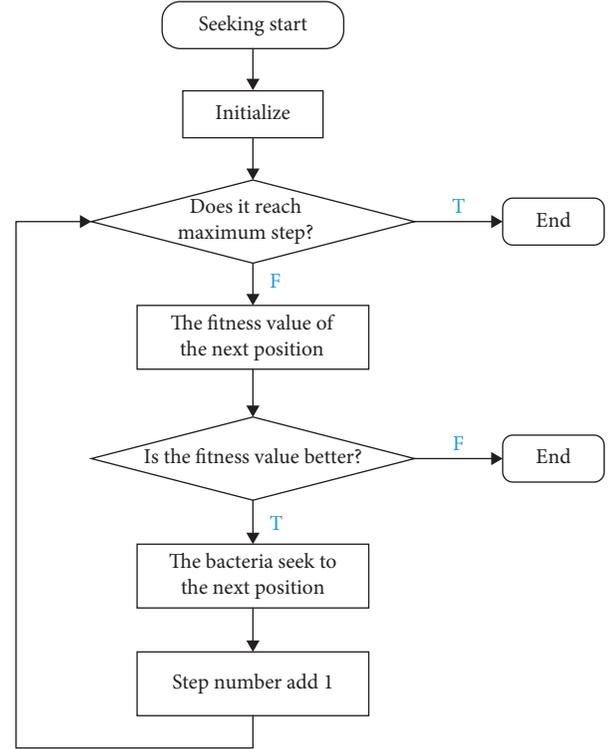


FIGURE 3: Seeking better fitness value ahead of time for moving.

actual acceleration and steering capabilities of the intelligent ship, which may cause the intelligent ship to be unable to successfully avoid obstacles in a collision-avoidance cycle, so the expansion of the arena and ship obstacles is considered to ensure that the smart ship has sufficient time to complete the obstacle avoidance action during the avoidance of collisions.

## 4. Simulation Experiment and Result Analysis

By adopting the MatlabR2016a to conduct the numerical simulation and collision avoidance simulation, an intelligent ship autonomous collision avoidance system is built.

### 4.1. The Verification of Numerical Simulation

**4.1.1. Test Function.** Four standard benchmark functions are selected and shown in Figure 4 and Table 1 to carry out tests for three algorithms: the BFOA mentioned above, the reformed step size in BFOA according to equations (8) and (9) (indicated by BFO in the graph), and the ant colony optimization algorithm with improved parameters in the literature (ant colony optimization algorithm, referred to as ACO algorithm). These four benchmark functions are all multimodal functions and have multiple local minima, so they are difficult to find the global optimal value and credible for evaluating the optimization level of the algorithm. The ACO algorithm is referred for comparison because the ACO algorithm is widely used in the field of collision avoidance and has a high reference.

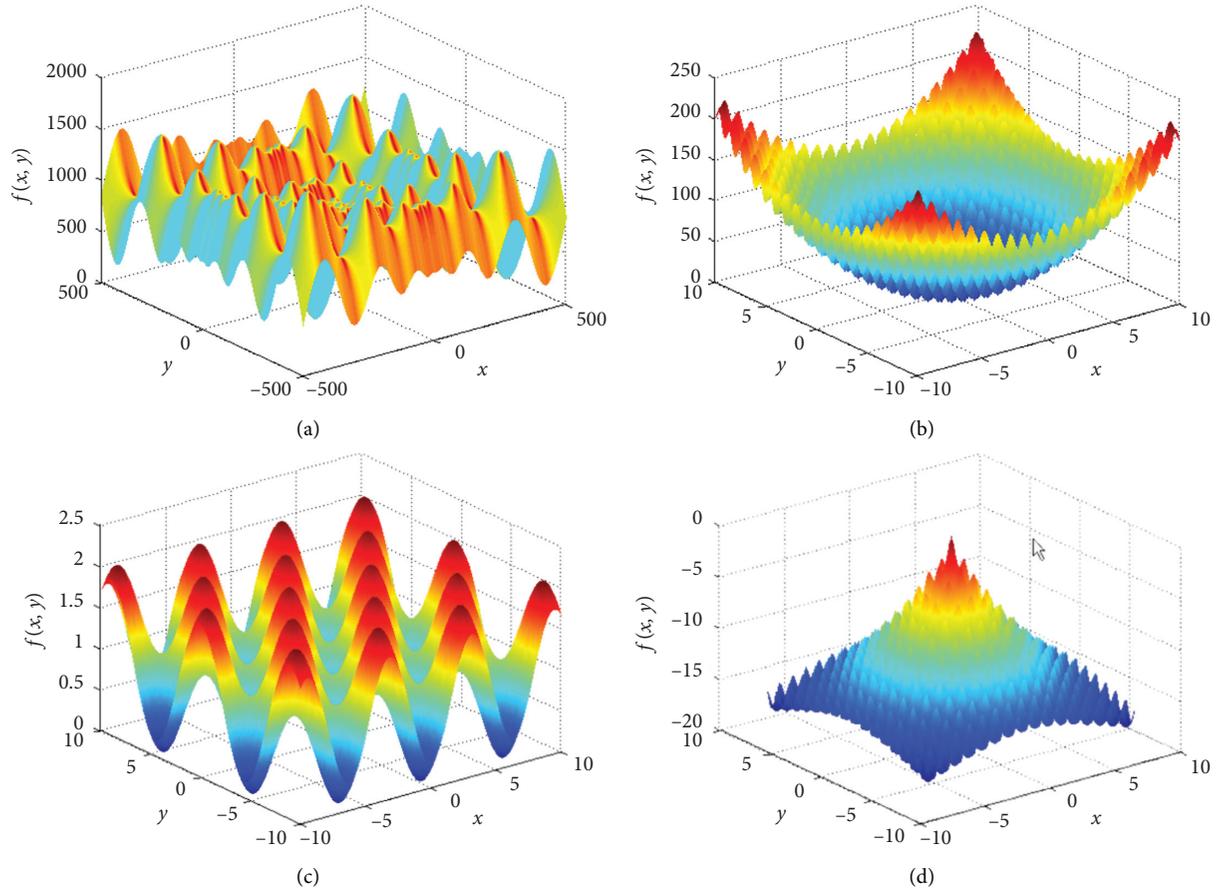


FIGURE 4: 3D graph of test functions: (a) Schwefel; (b) Rastrigin; (c) Griewank; (d) Ackley.

TABLE 1: Standard test functions.

Function name	Function expression	Search range	The optimal position	The optimal solution
Schaffer	$\min f(x_1, x_2) = 0.5 + ((\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5 / (1 + 0.001(x_1^2 + x_2^2)))^2$	$[-10, 10]$	$x_i = 0$	0
Rastrigin	$\min f(x_i) = \sum_{i=1}^n \left[ \frac{x_i^2 - 10}{\cos(2\pi x_i)} + 10 \right]$	$[-5.12, 5.12]$	$x_i = 0$	0
Griewank	$\min f(x_i) = \sum_{i=1}^n (x_i^2 / 4000) - \prod_{i=1}^n \cos(x_i / \sqrt{i}) + 1$ $\min f(x_i) = -c_1 \exp(-0.2 \sqrt{(1/n) \sum_{i=1}^n x_i^2})$	$[-600, 600]$	$x_i = 0$	0
Ackley	$-\exp((1/n) \sum_{i=1}^n \cos(2\pi x_i)) + c_1 + e$	$[-8, 8]$	$x_i = 0$	0

**4.1.2. Parameter Settings.** The parameters of the BFOA are set as the value in Table 2:  $p$  is the search dimension of the algorithm,  $s$  is the size of the bacterial population,  $N_c$  is the counts of bacterial chemotaxis,  $N_s$  is the maximum number of steps in the one-way movement of the bacteria in the chemotaxis,  $N_{re}$  is the counts of bacterial replication,  $N_{ed}$  is the counts of bacterial migration, and  $P_{ed}$  is the basic migration probability of bacteria.

TABLE 2: Algorithm parameters of BFO.

Parameter name	Parameter value	Parameter name	Parameter value
$p$	2	$N_{re}$	4
$s$	20	$N_{ed}$	2
$N_c$	25	$P_{ed}$	0.25
$N_s$	4	—	—

The parameters of the ACO algorithm are shown in Table 3:  $p$  is the algorithm search dimension, Ant is the number of ants, Times is the counts of ants moving, Rou is the pheromone volatilization coefficient, and  $P_0$  is the transition probability.

*4.1.3. Simulation Results.* Figure 5 is a comparison diagram of the optimal fitness values obtained by applying these three algorithms to the four standard test functions. The horizontal axis is the algorithm iteration counts, and the vertical axis is the fitness value.

*4.1.4. Result Analysis.* It can be concluded from Figure 5 and Table 4 that, for the four standard test functions, the IBFOA has better global optimization and optimization accuracy than the ACO algorithm with improved parameter and the BFO algorithm with improved step size, and for the precocious Griewank and Rastrigrin functions and the Schaffer and Ackley functions that are difficult to find the global optimal value, the optimal position obtained by the IBFOA reaches the optimal value (0, 0), and the convergence accuracy and the global optimization of the algorithm are increased remarkably.

Compared with the ACO algorithm with improved parameters, the IBFOA has better global optimization and convergence accuracy; compared with the BFOA with improved step size, the IBFOA has advantages in convergence accuracy, speed, and algorithm stability. The BFOA has a slower convergence speed in the early stage; it may be limited by a local optimum, then gradually approaches the optimal value after a single dispersal, and the stability of the algorithm is insufficient in the later stage. The algorithm's early optimization ability and later optimization accuracy are increased through the improvement of adaptive step size and swimming operation, which makes the algorithm convergence faster and more accurate, and the selection of adaptive migration probability enhances the global optimization ability and algorithm stability. Analyzing the changes in the fitness value in Figure 5, it can be found that the IBFOA has better convergence speed, accuracy, global optimization, and stability.

*4.2. The Analysis of Single Obstacle Avoidance Simulation.* Figure 6 is a simulation of a single obstacle autonomous collision avoidance solved by the improved BFO algorithm, which shows that the intelligent ship completes collision avoidance and reaches the target point. We did not discuss the subtarget points calculated in the global planning process but considered the collision avoidance planning of the intelligent ship against dynamic obstacles when the starting point and target points are known. The uniform velocity obstacle means that the obstacle speed and heading do not change, so the speed and heading obtained by AIS at the time are used as the reference for the collision avoidance judgment and strategy solution in each cycle, and the transient changes in the obstacle are not considered.

TABLE 3: Algorithm parameters of ACO.

Parameter name	$p$	Ant	Times	Rou	$P_0$
Parameter value	2	100	200	0.8	0.2

The data of dynamic performance of the USV used in this paper are measured by the USV platform of Harbin Engineering University in China; specifically, the cruising speed is 20 knots and the minimum avoidance velocity is set as 15 knots. During navigation, the operational sensors for detecting the obstacles on the platform include a radar, a binocular vision, a stereo vision, a monocular vision, infrared cameras, and laser range finders, and the maximum sensor detecting distance is 1000 m. The radar information for boat updates every 5 s, so the period for collision avoidance in this paper is set as 5 s. According to experiments, no matter what the speed is, the maximum steering angle of the boat in one cycle is not less than  $8^\circ$ . To ensure the feasibility of collision avoidance, we set the steering angle of collision avoidance to be no more than  $3^\circ$  under general cases and no more than  $5^\circ$  in emergency cases in this paper, and the coupling relationship between speed and heading during the steering process is ignored for simplicity. The motion capacity is controlled and implemented, dynamic performance for the above is set as the standard value in this study, and parameters set above are within the limit power range of the boat. The algorithm solves fast, so there is enough time to complete the collision avoidance in the cycle. The overtaking situation was selected for the simulation. And the obstacle boat kept a fixed course and speed as COLREGS required. The unit of the distance was meter. The initial position of the USV was [0, 0], and the initial velocity was 38 kN, while the initial velocity of the obstacle boat was 23 kN, and both these two velocities were eastward. The collision range of the intelligent ship is 1000. The initial position of the obstacle boat was [1500, 100]. It can be seen from Figure 6 that the smart ship can successfully complete the autonomous collision avoidance under the condition of the obstacles at a constant speed or a variable speed. The safety expansion circle in Figure 6 no longer shows that the intelligent ship has successfully completed the collision avoidance.

*4.3. The Analysis of Multiobstacle Collision Avoidance Simulation.* The intelligent ship has an initial position of (0, 0) and an eastward velocity. The obstacle boat I, which is enclosed by the yellow circle, is located at (1500, -500), and its velocity points to the north by west. The obstacle boat II, which is enclosed by the red circle, is located at (2000, 500), and its velocity points to the east by south. The obstacle boat III, which is enclosed by the blue circle, is located at (6000, -500), and its velocity points to the west by north. The intelligent ship has done the global path planning in the static obstacle environment, and the next subglobal goal is located at (6500, -800). In the first 3 cycles, the relative position of the boats is shown in Figure 7. Figure 7 shows that the intelligent ship had avoided all three obstacles, so in the case of multiple variable-speed obstacles, using the

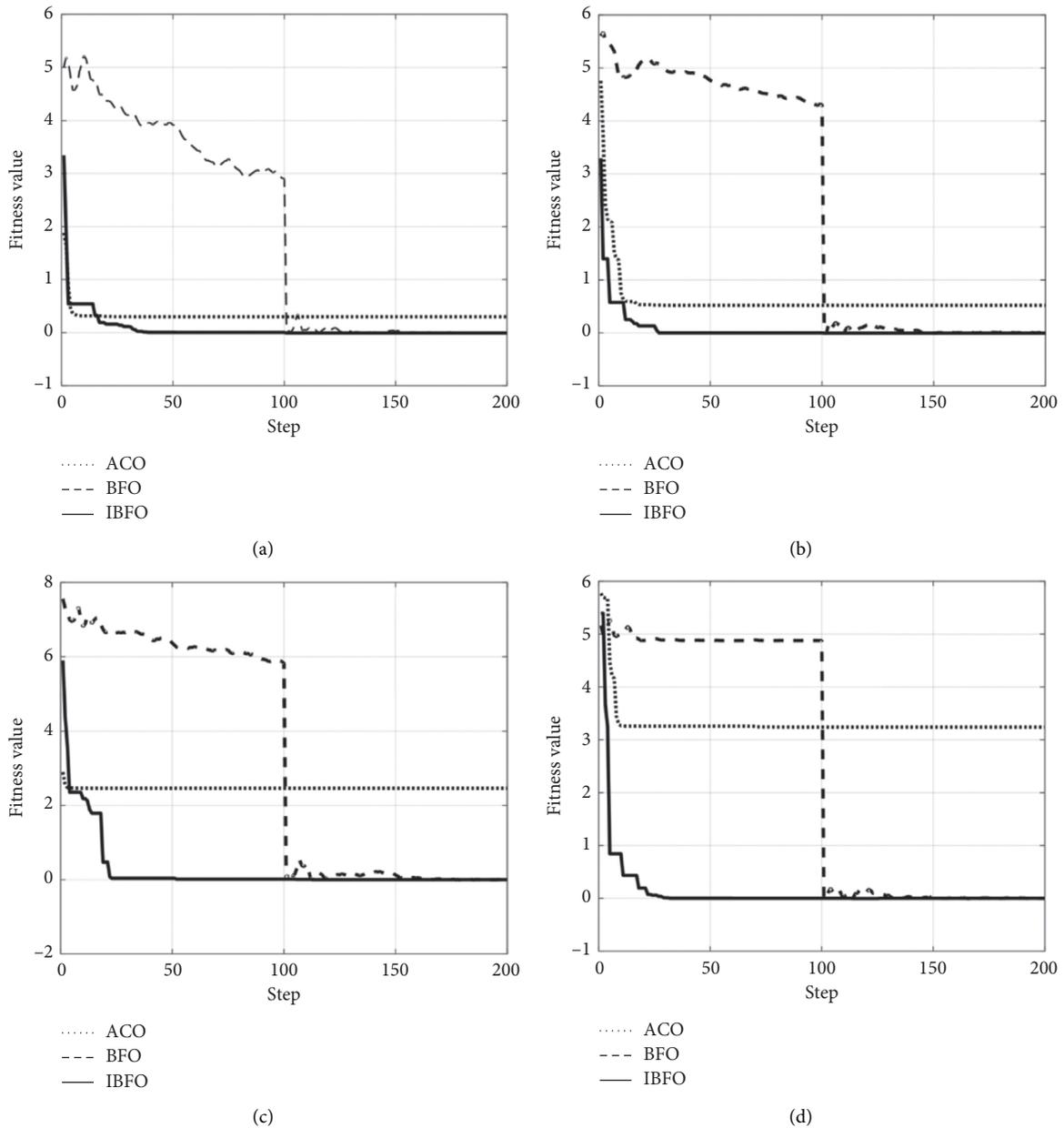


FIGURE 5: Simulation results of the benchmark tests: (a) Schaffer; (b) Rastrigrin; (c) Griewank; (d) Ackley.

TABLE 4: Running results of test functions.

Function name	Algorithm	The optimal position	The optimal value	Convergence
Schaffer	ACO	(-0.0688, 0.0689)	0.3123	10
	BFOA	(-0.0029, -0.0034)	-0.0043	140
	IBFOA	(0, 0)	-2.5238E-05	42
Rastrigrin	ACO	(-0.0953, 0.0954)	0.5039	20
	BFOA	(0.0126, -0.0145)	-0.0058	142
	IBFOA	(0, 0)	-2.5468E-05	28
Griewank	ACO	(-0.1863, 0.1861)	2.4584	8
	BFOA	(0.0007, -0.0350)	0.0046	162
	IBFOA	(0, 0)	-1.1387E-05	52
Ackley	ACO	(-0.2406, 0.2405)	3.2392	13
	BFOA	(0.0042, -0.0035)	0.0035	103
	IBFOA	(0, 0)	-1.1253E-05	36

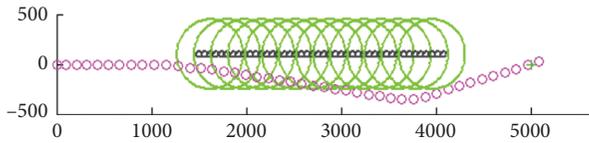


FIGURE 6: Simulation of uniform single speed obstacle collision avoidance.

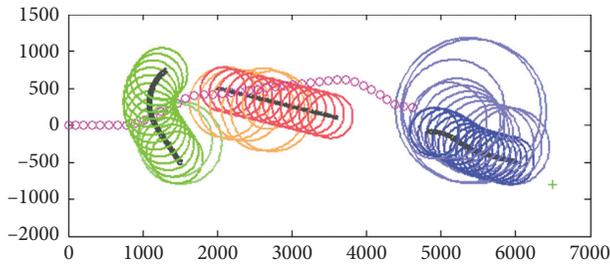


FIGURE 7: Simulation of variable-speed multiple obstacles collision avoidance.

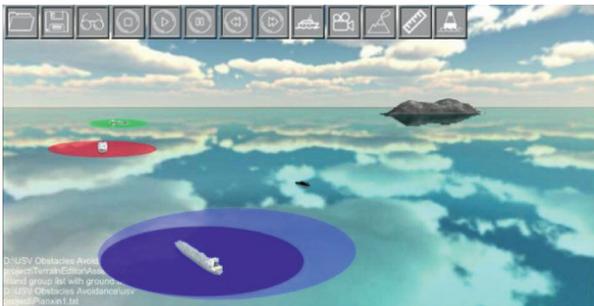


FIGURE 8: The collision avoidance process displayed by Unity3D.

IBFOA to solve the collision avoidance strategy, the intelligent ship can successfully complete the autonomous collision avoidance. The intelligent ship was always in the center of the 3D graph (Figure 8). Figure 8 shows the real-time 3D simulation of collision avoidance. In the figure, the obstacle detected is the real position of the obstacle. The dark-colored area around the obstacle boat indicates its domain, while the light-colored area represents its eccentric expansion. This simulation can provide a clear insight into the avoidance process of USVs in a real-world environment. The finding indicates the accuracy of the proposed method in obstacle avoidance.

## 5. Conclusion

Aiming at the problem of autonomous collision avoidance of intelligent ships, the IBFOA was proposed in this article to use this algorithm to solve the VO model and obtain the autonomous collision avoidance strategy of intelligent ships. Finally, numerical verification and collision avoidance simulation are conducted with MatlabR2016a, and the following are the conclusions: (1) designing an adaptive decreasing fractal dimension chemotaxis step size to realize the adaptive adjustment of the step size and solve the problems of the fixed step size that the algorithm is easy to be limited

by the local optimal solution, the early convergence speed is slow, and the late convergence accuracy is low, which increased the convergence speed and optimization accuracy of the algorithm; (2) the optimal swimming search method is proposed, which solves the defects of invalid swimming and repeated swimming of the basic BFO algorithm and improves the convergence speed of the algorithm; (3) figuring out an adaptive migration probability instead of a fixed migration probability, which solves prevent elite individuals from being lost and improves the accuracy and stability of the algorithm; (4) conducting the numerical simulations of IBFOA and ACO algorithm with improved parameters and the BFO algorithm with improved step size, and the result indicates that the IBFOA has better global optimization capability and accuracy than the ACO algorithm with improved parameters and has better convergence speed, accuracy, and stability than the BFO algorithm with improved step size; (5) applying the IBFOA to the autonomous collision avoidance of intelligent ships, developing the collision avoidance strategy of intelligent ships, and performing collision avoidance simulation; then, the results show that the IBFOA can quickly and safely realize the autonomous collision avoidance of intelligent ships under dynamic obstacles.

In the future, an algorithm of obstacle avoidance, which takes the moving capacity of the USV and the influence of the control system into consideration, will be designed to make it an integrated closed loop in the planning and execution of obstacle avoidance.

## Data Availability

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

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Authors' Contributions

W. X. provided the idea of the algorithms, carried out the simulations, and drafted the manuscript. H. J. supervised the work, arranged the architecture, and contributed to the writing of the manuscript. K. X. supervised the work, and T. X. provided the major direction of the research.

## Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (no. 51809203).

## References

- [1] Y. Zhao, W. Li, and P. Shi, "A real-time collision avoidance learning system for unmanned surface vessels," *Neurocomputing*, vol. 182, pp. 255–266, 2016.
- [2] H. Kim, B. Park, and H. Myung, "Curvature path planning with high-resolution graph for unmanned surface vehicle," in

- Proceedings of the Robot Intelligence Technology and Applications (RiTA)*, pp. 147–154, Gwangju, Korea, 2012.
- [3] V. S. Andrey and W. Chao, “A simple biologically-inspired algorithm for collision free navigation of a unicycle-like robot in dynamic environments with moving obstacles,” *Robotica*, vol. 31, no. 6, pp. 1–11, 2013.
- [4] Y. Yang, S. Wang, Z. Wu, and Y. Wang, “Motion planning for multi-HUG formation in an environment with obstacles,” *Ocean Engineering*, vol. 38, no. 17-18, pp. 2262–2269, 2011.
- [5] L. P. Perera, J. P. Carvalho, and C. Guedes Soares, “Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions,” *Journal of Marine Science and Technology*, vol. 16, no. 1, pp. 84–99, 2011.
- [6] X. G. Peng and D. M. Xu, “Intelligent online path planning for UAVs in adversarial environments,” *International Journal of Advanced Robotic Systems*, vol. 9, no. 3, pp. 1–12, 2012.
- [7] US Department Homeland Security/US Coast Guard, *Navigation Rules*, Paradise Cay Publications, Blue Lake, CA, USA, 2010.
- [8] Y. Kuwata, M. T. Wolf, D. Zarzhitsky et al., “Safe maritime autonomous navigation with COLREGS, using velocity obstacles,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, 2011.
- [9] C. H. Song, “Global path planning method for USV system based on improved ant colony algorithm,” *Applied Mechanics and Materials*, vol. 568, pp. 785–788, 2014.
- [10] R. Zhang, P. Tang, Y. Su et al., “An adaptive obstacle avoidance algorithm for unmanned surface vehicle in complicated marine environments,” *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 4, pp. 385–396, 2014.
- [11] L. Zhen and J. G. Sun, “An improved bacterial foraging optimization algorithm,” *Journal of East China University of Science & Technology*, vol. 12, no. 2, pp. 15–24, 2016.
- [12] Y. Watanabe, A. Calise, and E. Johnson, “Vision-based obstacle avoidance for UAVs,” *AIAA Journal*, vol. 5, pp. 4862–4872, 2007.
- [13] U. Huh and S. R. Chang, “A G2 continuous path-smoothing algorithm using modified quadratic polynomial interpolation,” *International Journal of Advanced Robotic Systems*, vol. 11, no. 1, pp. 1–11, 2014.
- [14] C. Chen, G. E. N. G. Pei-wen, X.-c. Zhang et al., “Path planning for surface unmanned vehicle based on improved artificial potential field method,” *Ship Engineering*, no. 9, pp. 72–75, 2015.
- [15] Y. Liu, R. Song, and R. Bucknall, “A practical path planning and navigation algorithm for an unmanned surface vehicle using the fast marching algorithm,” in *Proceedings of the 2015 MTS/IEEE OCEANS*, pp. 1–7, IEEE, Genova, Italy, October 2015.
- [16] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.
- [17] J. Larson, M. Bruch, R. Halterman et al., *Advances in Autonomous Obstacle Avoidance for Unmanned Surface Vehicles*, Space and Naval Warfare Systems Center, San Diego, CA, USA, 2007.
- [18] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.
- [19] J. van den Berg, M. Lin, and D. Manocha, “Reciprocal velocity obstacles for real-time multi-agent navigation,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1928–1935, IEEE, Pasadena, CA, USA, May 2008.
- [20] H. Liu, S. Liu, L. Zhang, and Z. Gao, “The application research with Particle swarm bacterial foraging intelligent algorithm in ship collision avoidance,” in *Proceedings of the 2014 International Conference on Mechatronics, Control and Electronic Engineering (MCE-14)*, Shenyang, China, August 2014.
- [21] W.-Y. Ma and J.-X. Yang, “Optimization of ship collision avoidance route based on bacterial foraging algorithm,” *Journal of Dalian Maritime University*, vol. 39, no. 2, pp. 21–24, 2013.
- [22] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, 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, 2014.