Journal of
Marine Science and Engineering

MDPI

Article

Local Path Planning with Multiple Constraints for USV Based on Improved Bacterial Foraging Optimization Algorithm

Yang Long [1], Song Liu [1], Da Qiu [1], Changzhen Li [2], Xuan Guo [3], Binghua Shi [4,*] and Mahmoud S. AbouOmar [5]

[1] School of Intelligent Systems Science and Engineering, Hubei Minzu University, Enshi 445000, China
[2] School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China
[3] School of Automation, Wuhan University of Technology, Wuhan 430070, China
[4] School of Information Engineering, Hubei University of Economics, Wuhan 430205, China
[5] Industrial Electronics and Control Engineering Department, Faculty of Electronic Engineering, Menoufia University, Shibin el Kom 32952, Egypt
* Correspondence: shibinghua1988@163.com

Abstract: The quality of unmanned surface vehicle (USV) local path planning directly affects its safety and autonomy performance. The USV local path planning might easily be trapped into local optima. The swarm intelligence optimization algorithm is a novel and effective method to solve the path-planning problem. Aiming to address this problem, a hybrid bacterial foraging optimization algorithm with a simulated annealing mechanism is proposed. The proposed algorithm preserves a three-layer nested structure, and a simulated annealing mechanism is incorporated into the outermost nested dispersal operator. The proposed algorithm can effectively escape the local optima. Convention on the International Regulations for Preventing Collisions at Sea (COLREGs) rules and dynamic obstacles are considered as the constraints for the proposed algorithm to design different obstacle avoidance strategies for USVs. The coastal port is selected as the working environment of the USV in the visual test platform. The experimental results show the USV can successfully avoid the various obstacles in the coastal port, and efficiently plan collision-free paths.

Keywords: unmanned surface vehicle; local path planning; COLREGs; bacterial foraging algorithm; simulated annealing algorithm

check for updates

Citation: Long, Y.; Liu, S.; Qiu, D.; Li, C.; Guo, X.; Shi, B.; AbouOmar, M.S. Local Path Planning with Multiple Constraints for USV Based on Improved Bacterial Foraging Optimization Algorithm. *J. Mar. Sci. Eng.* **2023**, *11*, 489. https://doi.org/10.3390/jmse11030489

Academic Editor: Fausto Pedro García Márquez

Received: 8 February 2023
Revised: 23 February 2023
Accepted: 23 February 2023
Published: 24 February 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Unmanned surface vehicles (USVs) have been successfully used to replace humans in a multitude of potentially dangerous tasks in the military field. With the proliferation of Artificial Intelligence (AI) technology, the value of USVs in the civil field is becoming increasingly evident [1,2]. Potential applications include checking submarine pipelines [3], lake cruising [4], water quality testing [5], being a collaborative platform [6], water depth testing [7], and more.

In the face of unexpected and unpredictable events, USVs must be able to quickly and correctly reach their destination. To achieve this, they must be able to generate both high-precision global paths and real-time local paths that avoid dynamic obstacles. The algorithms for local path planning need to be robust and offer excellent real-time performance. This is an important indicator to measure the quality of USV local path planning. Local path planning is a comprehensive judgment of the unknown or partially unknown dynamic obstacles that USVs obtain from surrounding ship information or sensors based on AIS. Based on this information, the local path can be planned quickly and correctly. The common algorithms for local path planning mainly include the following: the neural network algorithm [8], artificial potential field method [9,10], dynamic window method [11], artificial bee colony algorithm [12], random tree method [13], etc.

Setting constraints on local path-planning methods can help ensure the safe navigation of USVs. Many factors affect the quality of the local path, such as obstacle type, COLREGs rules, working environment, path smoothness, algorithm efficiency, etc. For instance, Ref. [14] proposed an adaptive USV dynamic path-planning algorithm. The safe boundary model, tension boundary condition, and path smoothness are considered in the algorithm. Ref. [15] designed a practical local path-planning method based on the dynamic window approach, taking into account wave and current environmental factors. Ref. [16] presented a solution to motion planning for USVs in a maritime environment, utilizing an A* approach with a circular boundary as a safety distance constraint and accounting for moving obstacles and different ocean current environments. Ref. [17] focused on the security applications within a harbor, devising a simple avoidance algorithm that can be real-time implemented for moving obstacle avoidance.

It is not enough to support the safe navigation of USVs in public waters by considering only the above factors as constraints of local path-planning methods. Once a USV is operating in public waters, it is not only a vehicle for the mission but also a part of many water vehicles. Therefore, the local path-planning method considering traffic rules as constraints is very important. The COLREGs rule is the basic rule of water vehicles, so it must be considered an important constraint for the USV local path-planning algorithm to prevent the USV from posing a safety threat to other ships and to ensure the navigation safety of the USV in public waters. The researchers proposed several works on local path planning considering the COLREGs rule. A COLREGs intelligent collision avoidance algorithm based on deep reinforcement learning is proposed by [18]. The experimental results show that the algorithm can make USVs successfully avoid dynamic obstacles and reach the end point while following the COLREGs rule. Ref. [19] proposed an autonomous navigation algorithm for USVs using fuzzy logic under COLREGs guidelines. An extensive simulation study is used to verify the proposed method. The related research results that only take the rules as constraints include [20–22]. Some research results focus on the efficiency of path-planning algorithms while considering constraint rules. Ref. [23] proposed a guidance strategy that has the functions of global guidance, local COLREGs-compliant anti-collision, and heading control for underactuated USVs. The superiority of the proposed algorithm is shown by comparing it with the comparison algorithms in terms of success rate and voyage time. In summary, the study of local path-planning methods with a fast solving ability and multiple constraints can enable USVs to better complete aquatic tasks, thereby further enhancing its application value.

Intelligent optimization algorithms are an important method for solving the USV local path-planning problem [24–26]. The bacterial foraging optimization (BFO) algorithm, as a typical swarm intelligence optimization algorithm, has the advantages of fine search and strong robustness [27]. In this paper, a simulated annealing-bacterial foraging optimization algorithm (SA-BFO) is proposed, aiming at solving the local path-planning problem of USVs. In the proposed SA-BFO algorithm, the simulated annealing mechanism is introduced in the dispersal operation of the traditional BFO algorithm. The designed operator calculates the fitness value of the dispersal individuals after evolution and accepts the new solution according to the Metropolis criterion so that the proposed algorithm can better escape from local optima and converge to the global optimum. The proposed method considers the COLREGs rules and dynamic obstacles as constraints in designing different collision avoidance strategies for USVs. The coastal port water is selected as the working environment of the USV in the visualization platform with a high degree of simulation. The simulation of the proposed USV local path planning based on the SA-BFO algorithm is realized.

The rest of this paper is organized as follows: Section 2 introduces the structure of the standard BFO and the proposed BFO with the simulated annealing mechanism. The constraints for the USV local path-planning problem according to the encounter situation of USVs are proposed in Section 3. The simulation results based on comparative experiments and the visual platform test experiments are presented and discussed in Section 4. Section 5 gives concluding remarks concerning our works.

2. Improved Bacterial Foraging Optimization Algorithm

BFO mimics the behavior of E. coli foraging in the environment. This physiological property allows the bacteria to obtain food and survive. This algorithm is a bioinspired search algorithm, which can produce the optimal or relative optimal solution through modeling iterative optimization. BFO has good search efficiency because of its three-layer nested structure. Dispersal operators are nested as the outermost layer to perform dispersal operations on bacteria. However, the operator does not evolve iteratively for the current solution. An improved BFO is proposed to improve the optimization efficiency in this paper.

2.1. Bacterial Foraging Optimization Algorithm

BFO was proposed by K. M. Passino in 2002. The algorithm mimics the intelligent behavior of E. coli foraging in the human gut. BFO is one of the important branches of the swarm intelligence optimization algorithm. Because of its outstanding performance, it has been successfully applied in many fields after nearly two decades of development. The three-layer nested structure of BFO realizes the cooperation and competition among individuals, which gives BFO powerful parallel processing ability, efficient search, and good robustness.

The traditional BFO is iteratively optimized by an induction mechanism. The BFO obtains optimal or suboptimal solutions by iterative execution of three main operators: chemotaxis, reproduction, and elimination–dispersal. The basic steps of the algorithm are described as follows:

- **Step 1:** The BFO parameters, maximum number of chemotaxis times N_c, number of reproduction times N_re, number of elimination–dispersal times N_{ed}, population size M and number of swimming times N_s, were initialized.
- **Step 2:** Equation (1) is used to initialize the position of bacteria, and the initial fitness value of bacteria is defined as J, where $Rand$ is a random number uniformly distributed in the interval [0, 1].

$$X = x_{\min} + Rand(x_{\max} - x_{\min}) \tag{1}$$

- **Step 3:** Elimination–dispersal cycle l = 1:N_ed, reproduction cycle k = 1:N_re, and chemotaxis cycle j = 1:N_c.
- **Step 4:** Chemotaxis operation is performed.
- **Step 5:** Reproduction operation is performed. Half of the bacteria with poor fitness value were eliminated, and half of the bacteria with good fitness value cloned themselves.
- **Step 6:** Elimination–dispersal operation is performed. Each bacterium generates a random probability P. This step compares P with a fixed migration probability P_ed. If $P < P_\mathrm{ed}$, the elimination–dispersal operation is performed.
- **Step 7:** The termination conditions are tested. If the conditions are met, the result is output. Otherwise, it returns to step 4.

The chemotaxis operation in Step 4 contains two basic operators, namely, tumble and swim. The position of the bacteria after tumb is defined by Equation (2). The swim refers to continuing the continuous movement in the direction of optimized fitness. The bacterial adaptation value is represented by J_cc, as shown in Equation (3).

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + c(i)\frac{\Delta(i)}{\sqrt{\Delta^\mathrm{T}(i)\Delta(i)}}. \tag{2}$$

$$
\begin{aligned}
J_{\text{cc}}(\theta, P(j,k,l)) &= \sum_{i=1}^{S} J_{\text{cc}}^{i}\big(\theta, \theta^{i}(j,k,l)\big) \\
&= \sum_{i=1}^{S}\left[-d_{\text{attract}} \exp\left(-\omega_{\text{attract}} \sum_{m=1}^{D}\big(\theta_m - \theta_m^{i}\big)^{2}\right)\right] \\
&+ \sum_{i=1}^{S}\left[h_{\text{repellant}} \exp\left(-\omega_{\text{repellant}} \sum_{m=1}^{D}\big(\theta_m - \theta_m^{i}\big)^{2}\right)\right]
\end{aligned}
\tag{3}
$$

where $c(i)$ represents the step length of swimming in the selected direction, and $\Delta(i)$ is a vector in any direction. d_{attract}, ω_{attract}, $h_{\text{repellant}}$, and $\omega_{\text{repellant}}$, respectively, represent the depth of attraction, the width of attraction, the height of repellant, and the width of repellant.

2.2. Simulated Annealing-Bacterial Foraging Optimization Algorithm

The simulated annealing (SA) algorithm performs a random search in the solution space using the Metropolis criterion with probabilistic jump property. SA starts at a high initial temperature and decreases at a constant rate. Every time the temperature drops, the current solution is updated, and the global optimal solution is finally obtained. The Metropolis criterion means that a new state with a large energy difference from the current state can be accepted at high temperatures, while a new state with a small energy difference from the current state can be accepted at low temperatures. Moreover, when the temperature approaches zero, the new states with higher energy than the current state will no longer be accepted. The SA-BFO proposed in this paper retains the three-layer nested structure of the traditional BFO and introduces the simulated annealing mechanism into the elimination–dispersal operation. The improved elimination–dispersal operator realizes the iterative evolution of the current solution. The pseudocode of SA-BFO is shown in Algorithm 1. SA-BFO accepts both the optimization solution and the deterioration solution with a certain probability so that the algorithm can better escape from the local extremum and converge to the global optimum.

Algorithm 1 SA-BFO

```
Input:
  The set of population size, $M$;
  The set of chemotaxis times, $N_c$;
  The set of replication times, $N_{re}$;
  The set of elimination–dispersal times, $N_{ed}$;
  The set of the initial temperature, $T$;
  The set of the temperature iteration number, $L$.
Output:
  Path planning length and trajectory.
while $e < N_{ed}$ do // $e$ denotes the current times of elimination–dispersal
    while $r < N_{re}$ do // $r$ denotes the current times of replication
        while $c < N_c$ do // $c$ denotes the current times of chemotaxis
            Execute chemotaxis operator;
        end while
        Execute replication operator;
    end while
    Select the first $\beta \times M$ bacteria in the order of fitness after replication;
    while $l < L$ do // $l$ denotes the current value
        Execute translocation operator;
        if $K_{global} - K_{new} > 0$ then
            Accept the new solution $S$ with criterion Metropolis
        end if
        $T \leftarrow (a \times T)$ //$a \in (0,1)$
    end while
end while
return Path planning length and trajectory
```

3. Path-Planning Model and Local Path Planning

The USV makes rapid actions in response to surface emergencies to ensure that it can safely avoid dynamic obstacles. Moreover, its local path planning also needs to consider the constraints to ensure the safety of USV navigation in public waters.

3.1. Path-Planning Model

The path-planning model is the basis of USVs to plan the global path and local path. The grid method was used as the path-planning model of USVs in this paper. The two-dimensional working environment of USVs can be established efficiently and intuitively by the grid method. The shape, size, and position of obstacles can be set arbitrarily. The area of the USV is expanded in the grid, as shown in Figure 1.

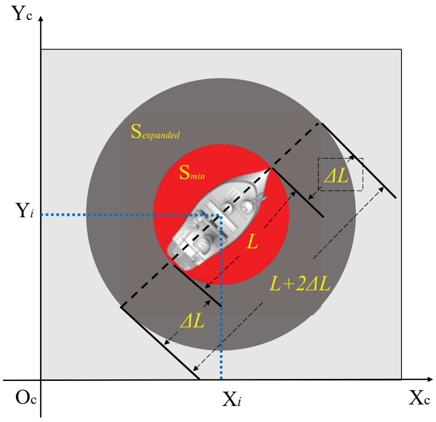


Figure 1. Diagram of USV expansion area.

In Figure 1, L is the length of the USV, $S_{\min}$ is the minimum area occupied by the USV, and S_{expanded} is the area occupied by the USV after expansion treatment. Its mathematical relation is given in Equations (4) and (5).

$$S_{\min} = \pi \times \frac{L^2}{4} \tag{4}$$

$$S_{\text{expanded}} = \pi \times \frac{(L+2\Delta L)^2}{4} \tag{5}$$

The movement rule of the USV in the grid map is shown in Figure 2. In this figure, the gray grids represent feasible grids, the black grids represent obstacles, and the blue grids represent the optimal path chosen by the USV. Figure 2a describes the movable direction of the USV at its position. Figure 2b shows the USV's optimal selection of feasible grids. The center points of the optimal grids are connected, representing the motion trajectory for the USV, as shown in Figure 2c.

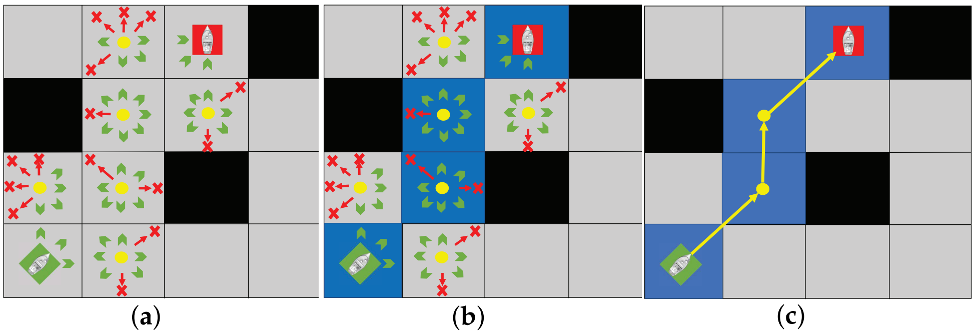
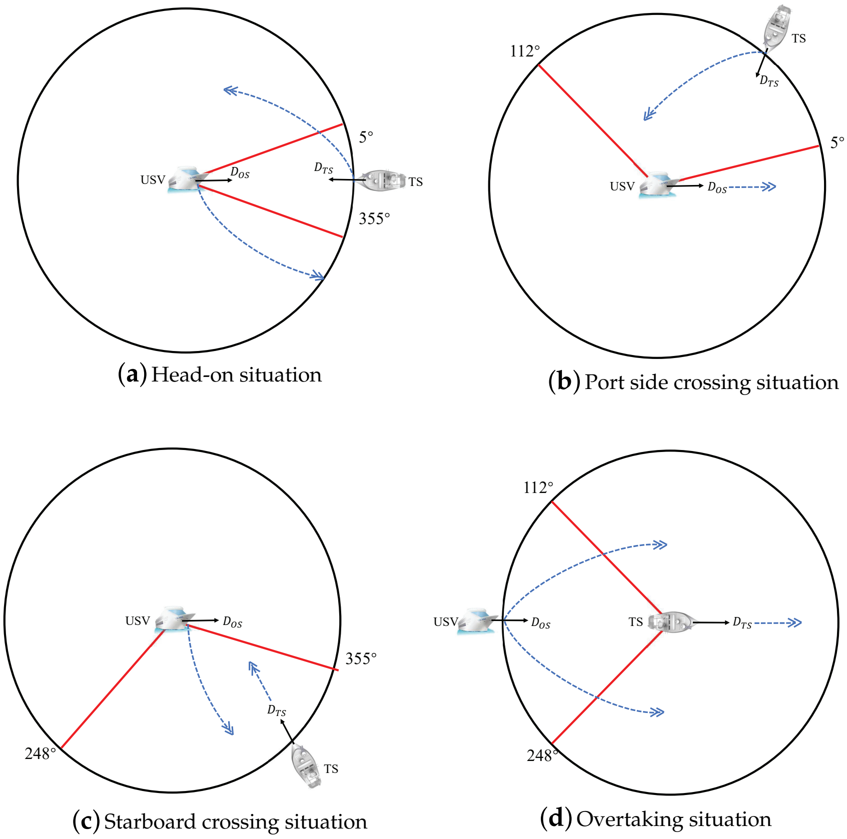


Figure 2. Diagram of USV movement rules and optimal path. (**a**) The USV movement rules in the feasible grids. (**b**) The optimal selection of feasible grids. (**c**) The optimal path.

3.2. The Constraint Conditions

3.2.1. COLREGs

COLREGs is the most important constraint for USV navigation in public waters. COLREGs defines three encounter situations, which are cross, head-on, and overtaking. The diagram of the encounter situation is shown in Figure 3.



(a) Head-on situation

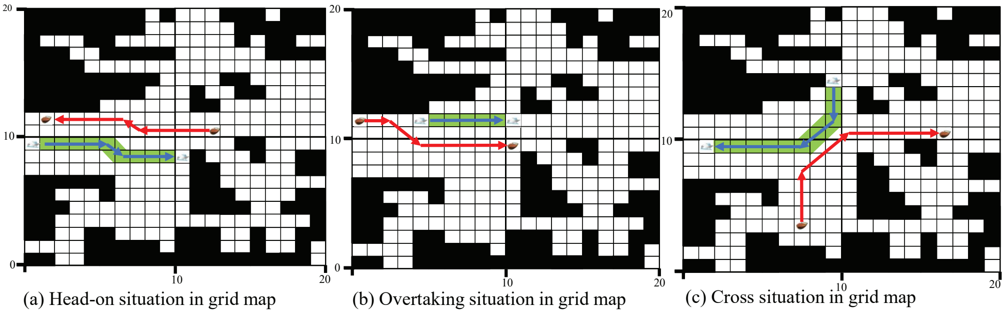
(b) Port side crossing situation

(c) Starboard crossing situation

(d) Overtaking situation

Figure 3. Diagram of encounter situation.

In Figure 3, TS represents other ships, D_{TS} represents the initial direction of other ships, and D_{OS} represents the initial direction of the USV. The encounter situation determined as the relative position between other ships and the USV was $-5°$ to $+5°$. The port side crossing situation defined as the relative position between other ships and the USV's port side was 5° to 112°. The starboard crossing situation was defined as the relative position between other ships and the USV's starboard and was at 5° to 112°. The relative position of other ships behind the USV or the USV behind other ships was 112° to 248°, and there is a certain speed difference between the two ships, which is called overtaking situation. The action of a USV following COLREGs to avoid other ships is shown in Figure 4.



(a) Head-on situation in grid map

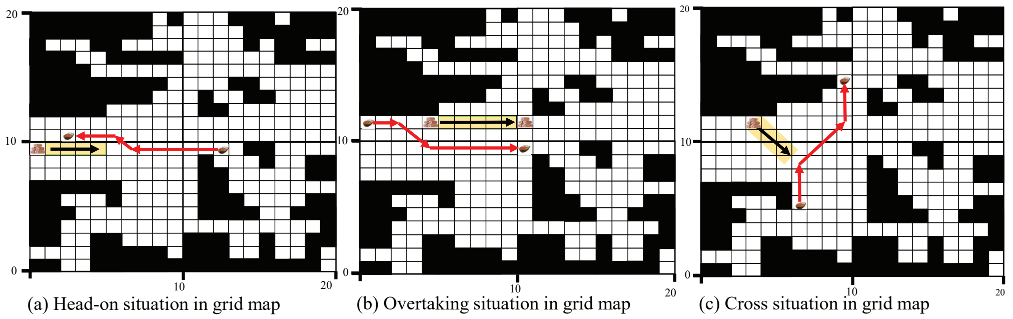
(b) Overtaking situation in grid map

(c) Cross situation in grid map

Figure 4. Diagram of USV encounters dynamic obstacles with COLREGs in grid map.

3.2.2. Dynamic Obstacle Division

Dynamic obstacles encountered by USVs can be divided into maneuvering type and non-motorized type according to their driving forms. When the USV encountered motorized obstacles, its local path planning followed the COLREGs. The floating wood, the ice, the rafts, etc., are dynamic obstacles that move and disorganize. Due to the characteristics of strong maneuvering, the USV actively avoids non-motorized obstacles. The action of a USV avoiding non-motorized obstacles in the grid map is shown in Figure 5.



(a) Head-on situation in grid map (b) Overtaking situation in grid map (c) Cross situation in grid map

Figure 5. Diagram of USV encounter non-motorized obstacles in grid map.

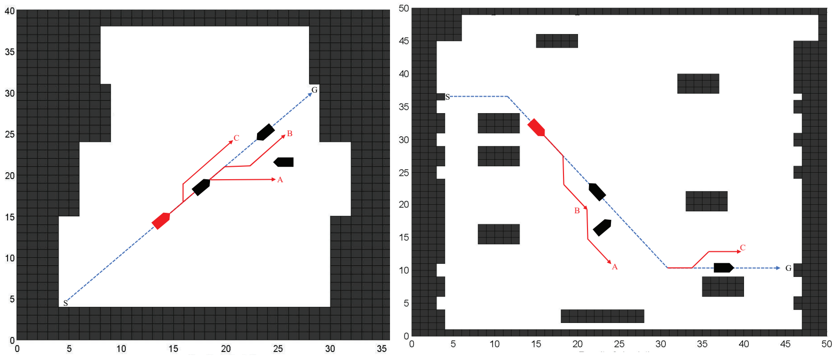
4. Simulation Results

The simulation experiment is divided into two parts: the numerical analysis and the visual platform test. The effectiveness and feasibility of the USV local path-planning method are verified by comparative tests and visual platform tests.

4.1. Contrast Experiment

To verify the advantage of SA-BFO in solving local path problems, comparative experiments of USV local path planning were conducted in various environments. The two sizes of grid maps (Q = 40 × 40 and Q = 50 × 50) were selected as the USV working environment in the experiment. SA-BFO, BFO, GA, and ACO algorithms are used to solve the USV local path. For the two scale maps, the local path planning in different situations is simulated, respectively.

The experiments assumed that the USV would encounter mobile dynamic obstacles, as shown in Figure 6. In this figure, the red ship represents the USV, the black ship represents the mobile dynamic obstacle, the blue dotted line represents the USV global path, and the solid red line is the USV local path. In the process of moving, the USV and the other ships have three encounter situations, namely, crossing situation, head-on situation, and overtaking situation.



(a) Q = 40 × 40 grid size **(b)** Q = 50 × 50 grid size

Figure 6. The experimental simulation environment and the local path-planning diagram.

Tables 1 and 2 show the comparative experimental results in the two environments. ω_t is the number of times the shortest path is searched, $\widetilde{P}$ is the average length of the local path obtained from 20 experiments, and $\widetilde{D}$ is the average iteration number of 20 experiments. According to the data in the above table, SA-BFO has fewer iterations and more times to search the optimal path when solving the USV local path planning compared to the other algorithms in all encounter situations.

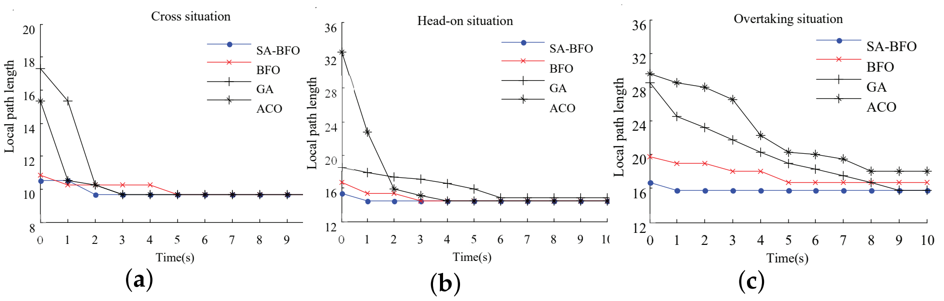
Table 1. Statistical table of experimental results in Q = 40 × 40.

Algorithm	Crossing Situation			Head-on Situation			Overtaking Situation		
	ω_t	$\widetilde{P}$	$\widetilde{D}$	ω_t	$\widetilde{P}$	$\widetilde{D}$	ω_t	$\widetilde{P}$	$\widetilde{D}$
SA-BFO	10	8.3	2.3	19	10.2	2	9	13.6	4
BFO	6	8.6	5.7	19	10.2	5.6	4	14.3	4
GA	8	8.5	61	19	10.2	97.4	7	13.9	81.6
ACO	4	8.7	6.7	18	10.3	3.9	4	14.6	6

Table 2. Statistical table of experimental results in Q = 50 × 50.

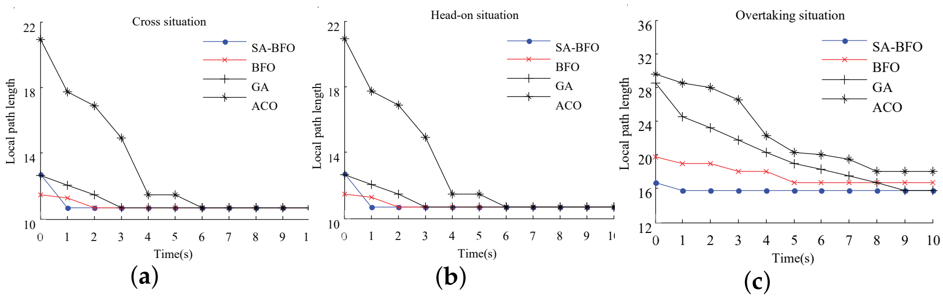
Algorithm	Crossing Situation			Head-on Situation			Overtaking Situation		
	ω_t	$\widetilde{P}$	$\widetilde{D}$	ω_t	$\widetilde{P}$	$\widetilde{D}$	ω_t	$\widetilde{P}$	$\widetilde{D}$
SA-BFO	18	10.8	2.4	12	14.9	2.3	7	17.1	1.8
BFO	16	10.9	2.8	5	18.6	3	1	22.2	4
GA	12	11.1	46.8	6	15.6	47	1	19.8	45
ACO	10	13.4	7.5	5	18.9	6.2	1	27.3	8.2

To further illustrate the better efficiency of the SA-BFO algorithm, this section provides a statistical analysis of the convergence of each algorithm at the same time under three encounter situations in two environments. The convergence curves of the algorithms in different scale environments are shown in Figures 7 and 8.



(**a**) (**b**) (**c**)

Figure 7. The convergence curves of the algorithms in environment 1. (**a**) Local path convergence curve of cross situation. (**b**) Local path convergence curve of head-on situation. (**c**) Local path convergence curve of overtaking situation.



(**a**) (**b**) (**c**)

Figure 8. The convergence curves of the algorithms in environment 2. (**a**) Local path convergence curve of cross situation. (**b**) Local path convergence curve of head-on situation. (**c**) Local path convergence curve of overtaking situation.

4.2. Visual Platform Test Experiment

The path-planning algorithm must be tested on a real ship or simulation platform before being applied in practice. There are some problems in real ship testing, such as long experimental period, high cost, and high experimental risk. Therefore, this paper verifies the proposed path-planning algorithm through a visual test platform. The visual test platform used in this paper is developed based on C++ and Qt graphical interfaces, which can load charts and satellite maps online. The simulation test platform can set the USV's speed and course, the static obstacle position, the dynamic obstacle course and speed, the view mode, and other information.

A coastal port is selected as the working environment of USVs in the visual test platform. The location information of the environment is shown in Table 3. The satellite map and gray map of the environment are shown in Figure 9. The obstacles in environmental information are rasterized. The orange grids are the static obstacles in global path planning. The black grids are the obstacles detected by the moving USV in its perceptual range. The diagram of the environment rasterization is shown in Figure 10.

Table 3. Geographic information of test environment.

Geographic Coordinate System **The Space Resolution**		**GCS_WGS_1984** **2.078396 m per Pixel**
Latitude and longitude of environment map (unit: °)	Top left:	109.898912, 21.405020
	Bottom left:	109.898912, 21.399936
	Top right:	109.908224, 21.405030
	Bottom right:	109.908224, 21.400006

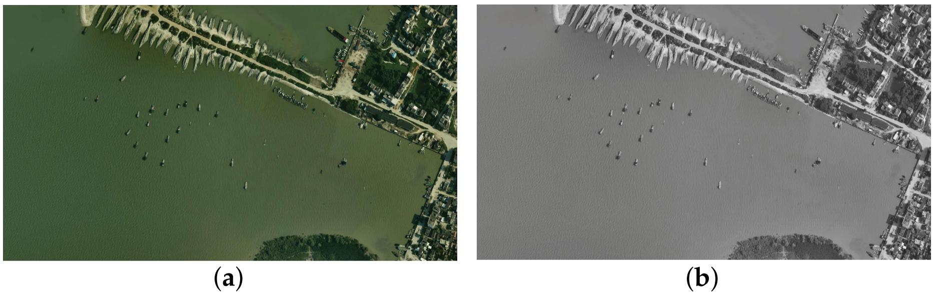


Figure 9. Experimental environment in visual test platform. (**a**) Satellite map of platform experiment environment (source: Bigmap®). (**b**) Grey maps of platform experiment environment.

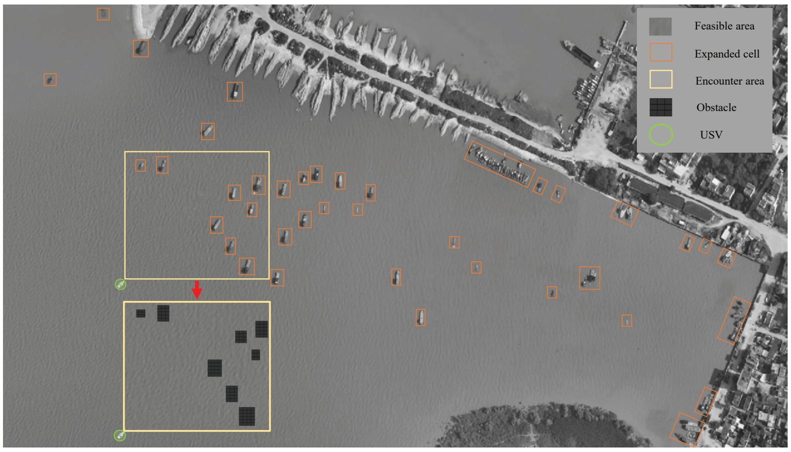
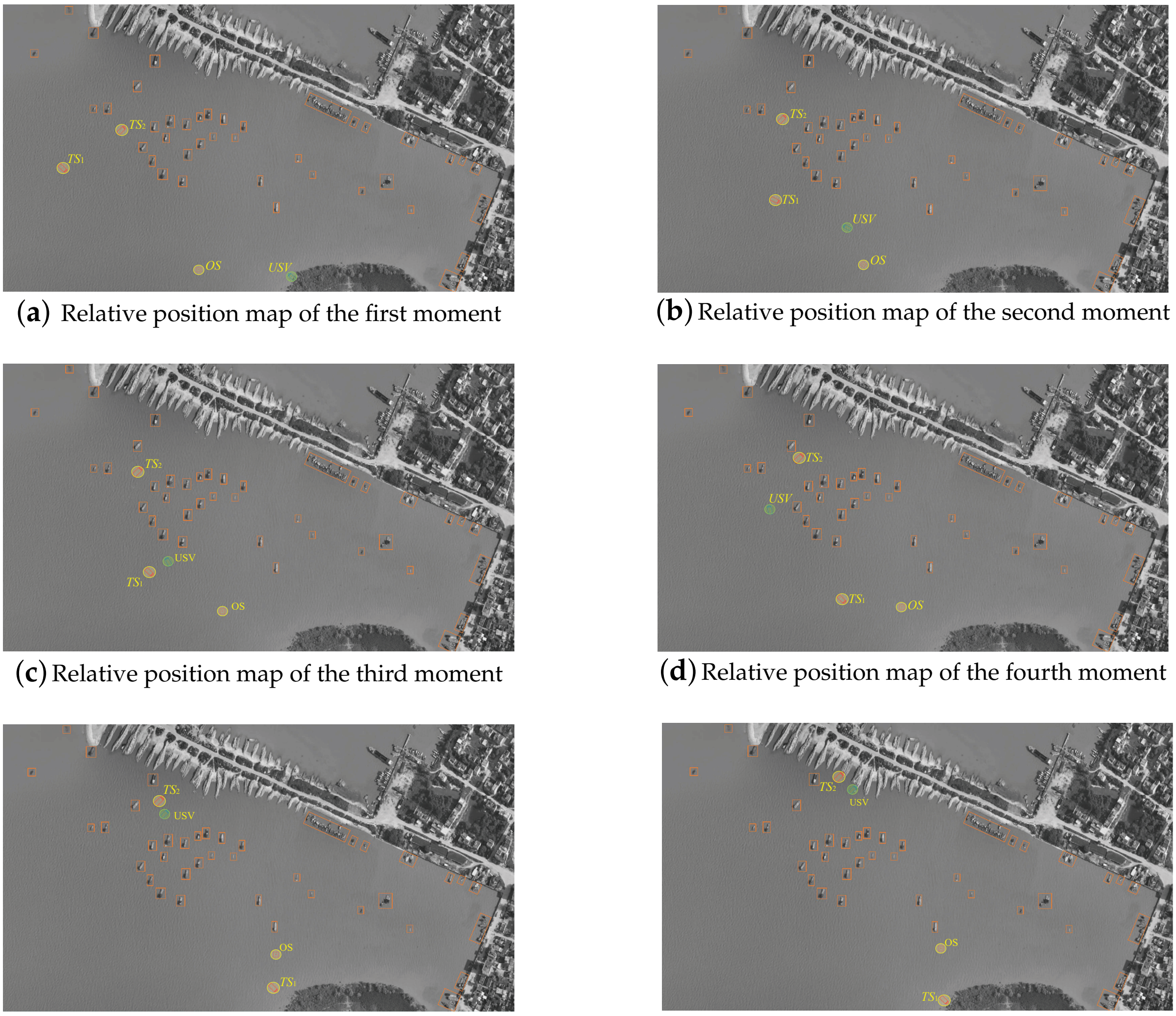


Figure 10. Rasterization of the environment map in the visual test platform.

The experiment simulates the USV's encounter with many ships and non-motorized dynamic obstacles. The initial information about various obstacles and the USV is shown in Table 4. The visual test platform shows the whole process of the USV driving from the starting point to the destination. In the course of the experiment, the USV will cross the uncontrollable raft (OS) first. After the USV completes the cross encounter, it forms a head-on situation with the motorized ship (TS_1). Finally, the USV overtook another motorized vessel (TS_2) at a lower speed until it reached its destination. The simulation experiment records the specific position of the USV at six moments from the start point to the destination. It reflects that the USV local path-planning method can make it safely avoid dynamic obstacles, as shown in Figure 11.

Table 4. The initial information of various dynamic obstacles and USV in the test environment.

	Longitude and Latitude of the Starting Point	Longitude and Latitude of the Destination	Course	Speed
USV	109.90415955, 21.40023232	109.90238206, 21.40387451	315°	20 knots
OS	109.90269470, 21.40039444	109.90326691, 21.40114021	180°	3 knots
TS_1	109.90003967, 21.40194321	109.90406879, 21.40019694	135°	12 knots
TS_2	109.90116882, 21.40297318	109.90213699, 21.40409596	45°	4 knots



(**a**) Relative position map of the first moment

(**b**) Relative position map of the second moment

(**c**) Relative position map of the third moment

(**d**) Relative position map of the fourth moment

(**e**) Relative position map of the fifth moment

(**f**) Relative position map of the sixth moment

Figure 11. Simulation experiment of USV in visual test platform.

4.3. Experimental Analysis

A local path-planning method based on SA-BFO is designed with dynamic obstacle types and COLREGs rules as constraints in this paper. The comparison experiment selects two environments of scale Q = 40 × 40 and Q = 50 × 50 for local path planning to verify the efficiency of the proposed algorithm. In the Q = 40 × 40 scale working environment, the number of iterations taken by the SA-BFO to find the optimal path in the cross situation is 3.7% of that taken by the GA, 34% of that taken by the ACO, and 40% of that taken by the BFO. The number of iterations taken by the SA-BFO in the head-on situation is 2% of that taken by the GA, 51.3% of that taken by the ACO, and 35.7% of that taken by the BFO. The number of iterations taken by the SA-BFO in the overtaking situation is 4.9% of that taken by the GA, and 66.7% of that taken by the ACO. In the Q = 50 × 50 scale working environment, the number of iterations taken by the SA-BFO in the cross situation is 5.1% of that taken by the GA, 32% of that taken by the ACO, and 85.7% of that taken by the BFO. The number of iterations taken by the SA-BFO in the head-on situation is 4.8% of that taken by the GA, 37.1% of that taken by the ACO, and 76.7% of that taken by the BFO. The number of iterations taken by the SA-BFO in the overtaking situation is 4% of that taken by the GA, 22% of that taken by the ACO, and 45% of that taken by the BFO. The above data indicate that the SA-BFO algorithm has obvious advantages over the comparison algorithms in solving accuracy in the two environments. Moreover, the comparative experiments show the convergence of each algorithm in different situations within 10 s. The simulation results illustrate that SA-BFO can quickly converge to the global optimum or relative optimum.

In the experiment of the visual test platform, the USV safely avoided all kinds of obstacles from the starting point to the endpoint. It can be seen from the simulation diagrams at different times of the experiment that the USV follows the constraints described above, safely avoids dynamic obstacles, and plans a reasonable local path.

5. Conclusions

Aiming at the local path-planning problem of USVs, a solution method based on the SA-BFO is proposed. The SA-BFO algorithm introduces the simulated annealing mechanism into the elimination–dispersal operation. The improved elimination–dispersal operator realizes the iterative evolution of the current solution. SA-BFO accepts both the optimization solution and the deterioration solution with a certain probability so that the algorithm can better escape from the local extremum and converge to the global optimum. The simulation results show that the proposed method can make the USV make rapid and reasonable obstacle avoidance actions when facing various obstacles. Furthermore, the USV has simulated many encounter situations with various obstacles in public waters and achieved a good obstacle avoidance effect in the visual test platform.

The research results of the USV path planning in this paper are all validated using the numerical simulation analysis and the computer simulation experiment stage. The proposed method and the experimental results can provide reference and technical support for intelligent ships. In the future, the research results need to be verified by real ships. Moreover, in future work, the USV energy consumption, channel constraints, multitask planning, and other issues will be considered.

Author Contributions: Y.L.: Conceptualization, data curation, formal analysis, investigation, methodology, funding acquisition, validation, visualization, writing—original draft, and writing—review and editing. S.L.: Conceptualization, resources, software, and writing—review and editing. D.Q.: Conceptualization, resources, software, and writing—review and editing. C.L.: Measurement, investigation, methodology, and software. X.G.: Investigation, methodology, and software. B.S.: Conceptualization, formal analysis, funding acquisition, methodology, project administration, resources, software, supervision, and writing—review and editing. M.S.A.: Conceptualization, measurement, methodology, and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the National Natural Science Foundation of China (52201363) and the Hubei Provincial Education Department Scientific Research Program Project (NO. Q20222202).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated and analyzed during this study are available from the corresponding author upon request.

Acknowledgments: We thank the Hubei Minzu University and Hubei University of Economics for their support.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of this study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Li, C.; Zhang, D. A Global Dynamic Path Planning Algorithm Based on Optimized A* Algorithm and Improved Dynamic Window Method. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 7515–7519.
2. Zhou, C.; Gu, S.; Wen, Y.; Du, Z.; Xiao, C.; Huang, L.; Zhu, M. The review unmanned surface vehicle path planning: Based on multi-modality constraint. *Ocean Eng.* **2020**, *200*, 107043. [CrossRef]
3. Rumson, A.G. The application of fully unmanned robotic systems for inspection of subsea pipelines. *Ocean Eng.* **2021**, *235*, 109214. [CrossRef]
4. Long, Y.; Su, Y.; Zhang, H.; Li, M. Application of improved genetic algorithm to unmanned surface vehicle path planning. In Proceedings of the 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), Enshi, China, 25–27 May 2018; pp. 209–212.
5. Chen, W.; Hao, X.; Yan, K.; Lu, J.; Liu, J.; He, C.; Zhou, F.; Xu, X. The Mobile Water Quality Monitoring System Based on Low-Power Wide Area Network and Unmanned Surface Vehicle. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 1609612. [CrossRef]
6. Zhang, H.; He, Y.; Li, D.; Gu, F.; Li, Q.; Zhang, M.; Di, C.; Chu, L.; Chen, B.; Hu, Y. Marine UAV–USV marsupial platform: System and recovery technic verification. *Appl. Sci.* **2020**, *10*, 1583. [CrossRef]
7. Kum, B.C.; Shin, D.H.; Jang, S.; Lee, S.Y.; Lee, J.H.; Moh, T.; Lim, D.G.; Do, J.D.; Cho, J.H. Application of Unmanned Surface Vehicles in Coastal Environments: Bathymetric Survey using a Multibeam Echosounder. *J. Coast. Res.* **2020**, *95*, 1152–1156. [CrossRef]
8. Liu, X.h.; Zhang, D.; Zhang, J.; Zhang, T.; Zhu, H. A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm. *Clust. Comput.* **2021**, *24*, 1901–1915. [CrossRef]
9. Azmi, M.Z.; Ito, T. Artificial potential field with discrete map transformation for feasible indoor path planning. *Appl. Sci.* **2020**, *10*, 8987. [CrossRef]
10. Sang, H.; You, Y.; Sun, X.; Zhou, Y.; Liu, F. The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations. *Ocean Eng.* **2021**, *223*, 108709. [CrossRef]
11. Liu, L.; Yao, J.; He, D.; Chen, J.; Huang, J.; Xu, H.; Wang, B.; Guo, J. Global dynamic path planning fusion algorithm combining jump-A* algorithm and dynamic window approach. *IEEE Access* **2021**, *9*, 19632–19638. [CrossRef]
12. Contreras-Cruz, M.A.; Ayala-Ramirez, V.; Hernandez-Belmonte, U.H. Mobile robot path planning using artificial bee colony and evolutionary programming. *Appl. Soft Comput.* **2015**, *30*, 319–328. [CrossRef]
13. Wen, N.; Zhang, R.; Wu, J.; Liu, G. Online planning for relative optimal and safe paths for USVs using a dual sampling domain reduction-based RRT* method. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 2665–2687. [CrossRef]
14. Chen, R.; Yang, C.; Han, S.; Zheng, J. Dynamic path planning of USV with towed safety boundary in complex ocean environment. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 71–76.
15. Wang, Z.; Liang, Y.; Gong, C.; Zhou, Y.; Zeng, C.; Zhu, S. Improved Dynamic Window Approach for Unmanned Surface Vehicles' Local Path Planning Considering the Impact of Environmental Factors. *Sensors* **2022**, *22*, 5181. [CrossRef] [PubMed]
16. 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]
17. Casalino, G.; Turetta, A.; Simetti, E. A three-layered architecture for real time path planning and obstacle avoidance for surveillance USVs operating in harbour fields. In Proceedings of the Oceans 2009-Europe, Bremen, Germany, 11–14 May 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1–8.
18. Xu, X.; Lu, Y.; Liu, X.; Zhang, W. Intelligent collision avoidance algorithms for USVs via deep reinforcement learning under COLREGs. *Ocean Eng.* **2020**, *217*, 107704. [CrossRef]

19. Lee, S.M.; Kwon, K.Y.; Joongseon, J. A fuzzy logic for autonomous navigation of marine vehicles satisfying COLREG guidelines. *Int. J. Control. Autom. Syst.* **2004**, *2*, 171–181.
20. Naeem, W.; Irwin, G.W.; Yang, A. COLREGs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics* **2012**, *22*, 669–678. [CrossRef]
21. Wang, H.; Guo, F.; Yao, H.; He, S.; Xu, X. Collision avoidance planning method of USV based on improved ant colony optimization algorithm. *IEEE Access* **2019**, *7*, 52964–52975. [CrossRef]
22. Zhuang, J.y.; Su, Y.m.; Liao, Y.l.; Sun, H.b. Motion planning of USV based on Marine rules. *Procedia Eng.* **2011**, *15*, 269–276. [CrossRef]
23. Han, S.; Wang, L.; Wang, Y. A COLREGs-compliant guidance strategy for an underactuated unmanned surface vehicle combining potential field with grid map. *Ocean Eng.* **2022**, *255*, 111355. [CrossRef]
24. Miao, C.; Chen, G.; Yan, C.; Wu, Y. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. *Comput. Ind. Eng.* **2021**, *156*, 107230. [CrossRef]
25. Yun, S.; Won, M. Genetic Algorithm Based 3D Environment Local Path Planning for Autonomous Driving of Unmanned Vehicles in Rough Terrain. *J. Korea Inst. Mil. Sci. Technol.* **2017**, *20*, 803–812.
26. Wang, Z.; Li, G.; Ren, J. Dynamic path planning for unmanned surface vehicle in complex offshore areas based on hybrid algorithm. *Comput. Commun.* **2021**, *166*, 49–56. [CrossRef]
27. Long, Y.; Su, Y.; Shi, B.; Zuo, Z.; Li, J. A multi-subpopulation bacterial foraging optimisation algorithm with deletion and immigration strategies for unmanned surface vehicle path planning. *Intell. Serv. Robot.* **2021**, *14*, 303–312. [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.