The Application Research with Particle Swarm Bacterial Foraging Intelligent Algorithm in Ship Collision Avoidance

Liu Hongdan
College of Automation
Harbin Engineering University
Harbin，China

Liu Sheng
College of Automation
Harbin Engineering University
Harbin，China

Zhang Lanyong
College of Automation
Harbin Engineering University
Harbin，China

Gao Zhengguo
College of Automation
Harbin Engineering University
Harbin，China

Abstract: Bacterial foraging algorithm has defects of low convergence rate and high computation complexity, so it will increase the optimization time when used to produce ship collision avoidance path. In order to improve the optimization performance, this paper proposes that intelligent bacterial foraging algorithm of particle swarm can be used for ship collision avoidance, for optimization of collision avoidance path, and for shortening of the generation time of collision avoidance strategies. Besides, it also verifies effectiveness and superiority of this algorithm when it is used to generate ship collision avoidance path under three ship approach situations including head-on-head, crossover, and overtaking through the ship collision avoidance simulation interface designed by VS2010 development environment. This has provided timely and effective information for collision avoidance operation of seaman.

Key word: ship collision avoidance；collision risk；route optimization ； hybrid optimization algorithm; dynamic simulation

Ⅰ. INTRODUCTION

Intellectualization of ship collision avoidance is one of the advanced and hot research issues in the academic circle of international maritime. Fig. 1 shows the implementation process of single ship collision avoidance system; involving reception and integration of collision avoidance information, target tracking, collision risk index analysis, judgment for approach situations, generation of optimized path, and dynamic display of collision avoidance path. It is a multi-objective and nonlinear planning problem, so it covers research and application of fuzzy logic, neural network (fuzzy neural network), expert system, genetic algorithm, danger model immune algorithm, modified PSOIM, and artificial intelligence technology like subject and case-based reasoning technology in ship collision avoidance field [1-4].

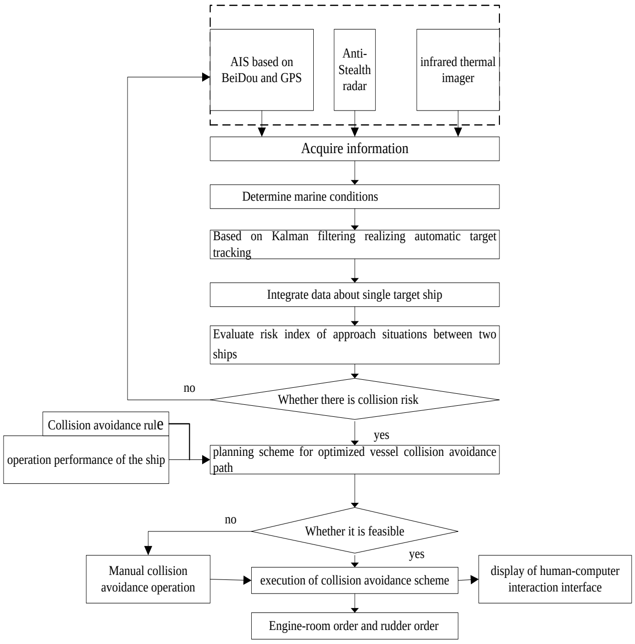

AIS based on BeiDou and GPS
Anti-Stealth radar
infrared thermal imager
Acquire information
Determine marine conditions
Based on Kalman filtering realizing automatic target tracking
Integrate data about single target ship
Evaluate risk index of approach situations between two ships
no
Whether there is collision risk
yes
Collision avoidance rule
operation performance of the ship
planning scheme for optimized vessel collision avoidance path
no
Whether it is feasible
yes
Manual collision avoidance operation
execution of collision avoidance scheme
display of human-computer interaction interface
Engine-room order and rudder order

Figure1. Implementation Process of Single Ship Collision Avoidance System

In which bacterial foraging algorithm (BFO) proposed by K. M. Passino in 2002 is a meta-heuristic search algorithm that simulates the foraging behavior of escherichia coli. Compared with other intelligent algorithms, this algorithm possesses strong global searching ability, and it can efficiently solve many optimization problems in the real world. In order to overcome defects of this algorithm like slow convergence rate and being caught in precocity and mistakes, it is improved on the basis of particle swarm optimization. In this way, the convergence rate in the bacterial foraging process can be increased [5] and the search time will be shortened without affecting the search quality. Therefore, this paper proposes that the bacterial foraging algorithm of particle swarm optimization can be applied to generation of intelligent ship collision avoidance path. Moreover, effectiveness and superiority of this algorithm in optimization of ship collision avoidance path are verified.

Ⅱ. COLLISION RISK INDEX

Collision risk index (CRI) is measurement for the possibility of collision among ships, and its value is 0～1 under general situations. CRI=0 means that it is safe between two ships and there is no collision possibility; CRI=1 shows that if no measure is taken, the two ships will inevitably collide, and when the collision risk index exceeds a certain threshold value, some measures must be taken. Parameters related to ship collision risk index include distance of closest point of approach (DCPA) between two ships, time of closest point of approach (TCPA) between two ships, distance between two ships (D), direction of the objective ship(B), and ship velocity ratio (K). In order to consider the above factors, this paper determines CRI among single ships on the basis of CRI of time (U_{tT}) and CRI of space (U_{dT}).

$$\text{CRI} = U_{tT} \oplus U_{dT} \tag{1}$$

Where $\oplus$ is composition operator, and the operation is defined as:

$$\begin{aligned} &U_{dT} = 0\text{......CRI=0;} \\ &U_{dT} \neq 0, U_{tT} = 0\text{...........CRI=0;} \\ &U_{dT} \neq 0, U_{tT} \neq 0\text{....CRI=max(}U_{dT}, U_{tT}\text{).} \end{aligned} \tag{2}$$

Therefore, only when CRI of time (U_{tT}) and CRI of space (U_{dT}) are not zero, does ship collision risk exist. According to the judgment level of risk index, this paper chooses to call collision avoidance path algorithm to prevent the collision when CRI=0.7(relatively high).

Ⅲ. ESTABLISHMENT FOR OBJECTIVE FUNCTION OF SHIP COLLISION AVOIDANCE

Suppose that our ship is the give-way ship, and safe collision avoidance should be guaranteed at first and then “economical” collision avoidance must be considered in the process of planning collision avoidance path. Therefore, the collision avoidance path optimization algorithm is required to complete collision avoidance in the shortest path on the premise of ensuring safety; the objective function is as follows [5-6].

$$J = \min(d_s + d_r) \tag{3}$$

Where, d_s is the sailing distance of our ship after collision avoidance; d_r is the sailing distance of our ship at the resumption stage. Suppose that velocity of our ship is V, steering angle of avoidance is C_s, steering angle at the resumption stage is C_r, and sailing time after avoidance is t_a. Then d_s and d_r in the above objective function meet the following expression:

$$d_s = t_a * V \tag{4}$$

$$d_r = t_a * V * \sin C_s / |\sin C_r| \tag{5}$$

In the process of path planning, constraint condition should be provided for variables in Formula (4) and (5), as shown in Formula (6):

$$\begin{aligned} &30^\circ \leq C_s \leq 90^\circ \\ &-60^\circ \leq C_r \leq -30^\circ \\ &t_{CPA1} \leq t_a \leq 40\min \\ &d_{CPA1} \geq d_G, d_{CPA2} \geq d_G \end{aligned} \tag{6}$$

Where d_{CPA1} and d_{CPA2} are the new DCPA after collision avoidance and course restoration, and the value of d_G is determined by the zero boundary of space risk index. The sailing time after collision avoidance is t_a, and it should be equal to the new DCPA t_{CPA1} after collision avoidance behavior is adopted at least. The calculation process for t_a is determined according to the actual simulation, as shown in the following:

(1) If $d_{CPA1} < d_G$, then $t_a = 40\min$; thus the objective function will be too large and the optimization process is directly eliminated;

(2) Under the condition of $d_{CPA1} \geq d_G$, if $t_{CPA1} > 0$, then $t_{CPA1} = t_a$ is the starting point; besides, if $d_{CPA2} \geq d_G$ is not satisfied, t_a will increase by a certain step value;

(3) Under the condition of $d_{CPA1} \geq d_G$, if $t_{CPA1} < 0$, then the initial value $t_a = 0.05*60$;besides, if $d_{CPA2} \geq d_G$ is not satisfied, t_a will increase by a certain step value (search step value adopted by simulation of this paper is equal to 0.02*60).

Ⅳ. SHIP COLLISION AVOIDANCE PATH STRATEGY BASED ON BACTERIAL FORAGING ALGORITHM OF PARTICLE SWARM

The implementation process for bacterial foraging algorithm of particle swarm optimization is presented in Fig.2 as in [7]. Parameter determination in the algorithm implementation process includes as in [8] search space dimensionality of bacterial foraging algorithm $P = 2$; bacterial quantity used for search $S = 26$; and number of convergence $N_c = 50$; the maximum step of movement is required to be 3-8, which can guarantee a good search effect as in [9], in this paper, ; and number of replication $N_{re} = 4$; number of migration $N_{ed} = 2$; value range of migration possibility should be 0.05-0.3, and it is $P_{ed} = 0.25$ in this paper; In particle swarm optimization, C_1, C_2, R_1 and R_2 are random parameters.

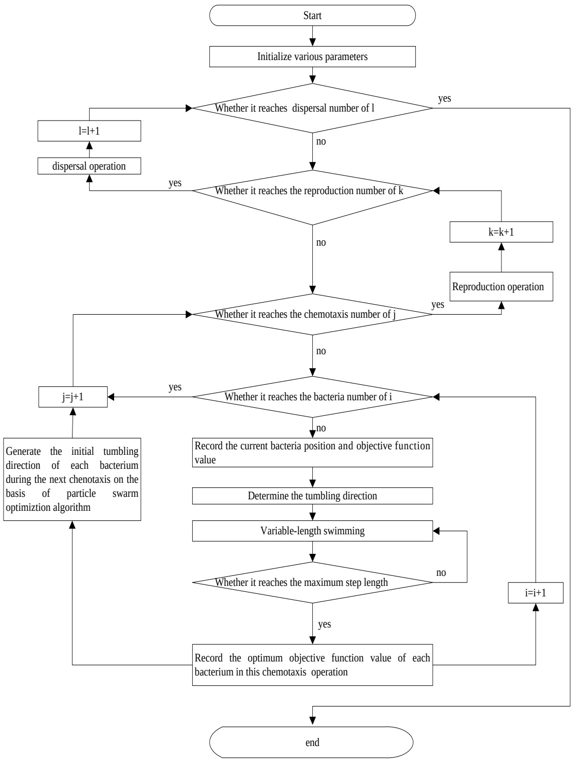

Start
Initialize various parameters
Whether it reaches dispersal number of l
yes
no
l=l+1
dispersal operation
yes
Whether it reaches the reproduction number of k
no
k=k+1
Reproduction operation
Whether it reaches the chemotaxis number of j
yes
no
yes
j=j+1
Whether it reaches the bacteria number of i
no
Generate the initial tumbling direction of each bacterium during the next chenotaxis on the basis of particle swarm optimiztion algorithm
Record the current bacteria position and objective function value
Determine the tumbling direction
Variable-length swimming
Whether it reaches the maximum step length
no
i=i+1
yes
Record the optimum objective function value of each bacterium in this chemotaxis operation
end

Figure2. Bacterial Foraging Algorithm Process of Particle Swarm Optimization

By combining with Fig. 2, the steps of implementing ship collision avoidance path optimization are as follows:

(1) Initialize parameters, arrays, and positions, and set the initial step C=0.05 .

(2) Start the circulation process

```
For( int $l = 0$; $l < N_{re}$; $l++$ )
    For ( int $k = 0$; $k < N_{re}$; $k++$ )
        For ( int $j = 0$; $j < N_c$; $j++$ )
            For( int $i = 0$; $i < S$; $i++$ )
```

//Initialize the random position of bacteria in 2D space;

$P_x = 60 + 30*(2.0*((double)rand())/RAND\_MAX - 1$

$P_y = -45 + 15*(2.0*((double)rand())/RAND\_MAX - 1$

//Call livefun function to calculate the initial value of objective function;

$J[l][k][j][i] = \text{livefun}$ (Calculate the objective function for the shortest ship collision avoidance path, and save the current value of fitness $Jlast = J$)

// Turn it over in 2D space to generate a random direction

$$Delta = 2*((double)rand())/\mathrm{RAND\_MAX} - 1.0$$

$$\mathrm{cov} += \mathrm{pow}(Delta, 2.0)$$

//Move toward a new direction within the maximum step range to produce a new 2D position (steering angle of collision avoidance and resumption) and objective function

$$P = P + C * Delta / sqrt(\mathrm{cov})$$

$$J[l][k][j][i] = \mathrm{livefun} \text{ (objective function)}$$

//Seek the globally optimal solution P_{gbest} and single optimal solution P_{lbest} as well as the corresponding particle position during this convergence circulation; calculate the current optimal fitness value $Jbest = J$

// Update the particle state according to particle swarm optimization and select the new direction and position of the next convergence

$$V = w*V + C_1*R_1*(P_{lbest} - P) + C_2*R_2*(P_{gbest} - P)$$

$$P = P + V$$

// Rank the objective function values, and eliminate some bacteria

$$\mathrm{Quicksort}(Jhealth, S)$$

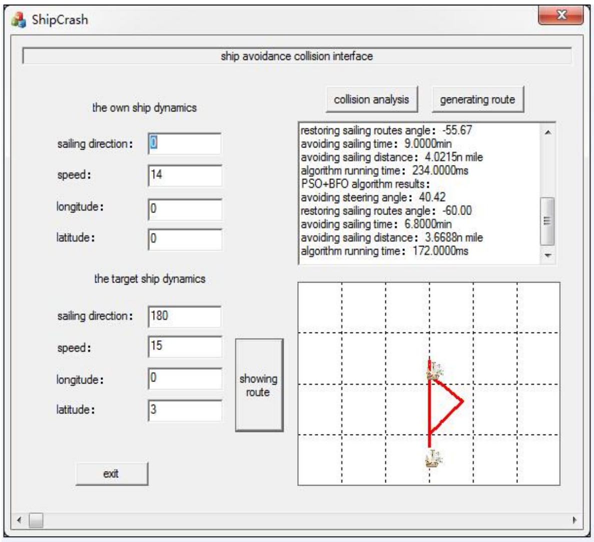
// Execute dispersal operation according to migration possibility, gain new random positions in the 2D space, and execute the next circulation.

V SIMULATION VERIFICATION AND RESULT ANALYSIS

Dynamic ship collision avoidance simulation is realized via intelligent bacterial foraging algorithm of particle swarm under the interface of VC++9.0 environment based on VS2010 as in [10]. Due to the restriction of display interface, the sailing time is calculated when CRI is greater than 0.4.

(1) Under head-on-head approach of single ship, initial motion parameters of our ship are: $V = 14$, $B = 0^{\circ}$, and the coordinate is (0,0); initial motion parameters of the target ship are: $V = 15$, $B = 180^{\circ}$, and the coordinate is (0,3). According to collision risk index analysis, CRI=1, DCPA=0, and TCPA=-0.10. There is collision risk at the preliminary stage, and the collision avoidance process is implemented directly, as shown in Fig. 3.

(2) Under crossover approach of single ship, initial motion parameters of our ship are: $V = 14$, $B = 0^{\circ}$, and the coordinate is (0,0); initial motion parameters of the target ship are: $V = 15$, $B = 300^{\circ}$, and the coordinate is (4,0). According to collision risk index analysis, under the initial state of the system, CRI=0.45, DCPA=1.79, and TCPA=-0.25. After sailing straightly for 1.8 minutes, the collision risk threshold value CRI=0.7 is reached, so the collision avoidance process is implemented, as shown in Fig. 4.



the own ship dynamics

sailing direction: 0

speed: 14

longitude: 0

latitude: 0

the target ship dynamics

sailing direction: 180

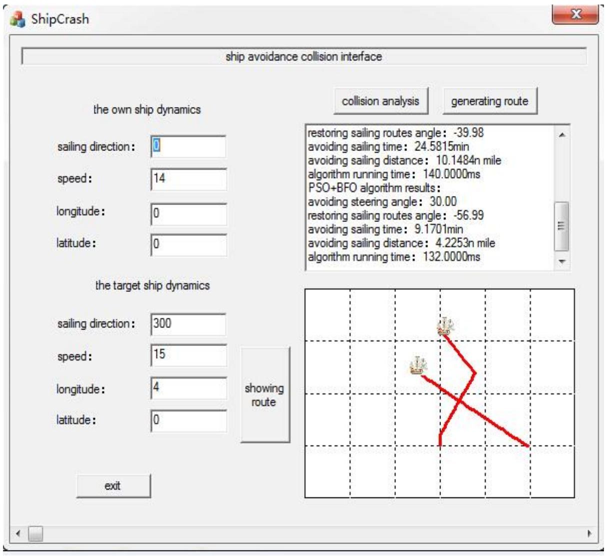
speed: 15

longitude: 0

latitude: 3

```
restoring sailing routes angle: -55.67
avoiding sailing time: 9.0000min
avoiding sailing distance: 4.0215n mile
algorithm running time: 234.0000ms
PSO+BFO algorithm results:
avoiding steering angle: 40.42
restoring sailing routes angle: -60.00
avoiding sailing time: 6.8000min
avoiding sailing distance: 3.6688n mile
algorithm running time: 172.0000ms
```

Figure3. Interface for Path Optimization Under Head-on-head Approach



the own ship dynamics

sailing direction: 0

speed: 14

longitude: 0

latitude: 0

the target ship dynamics

sailing direction: 300

speed: 15

longitude: 4

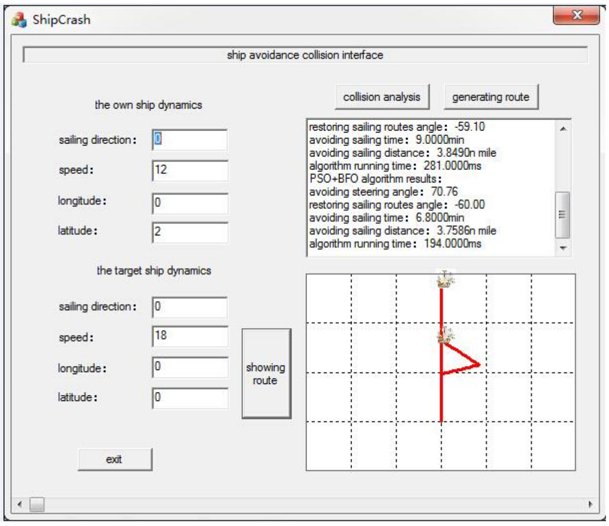
latitude: 0

```
restoring sailing routes angle: -39.98
avoiding sailing time: 24.5815min
avoiding sailing distance: 10.1484n mile
algorithm running time: 140.0000ms
PSO+BFO algorithm results:
avoiding steering angle: 30.00
restoring sailing routes angle: -56.99
avoiding sailing time: 9.1701min
avoiding sailing distance: 4.2253n mile
algorithm running time: 132.0000ms
```

Figure4. Interface for Path Optimization under Crossover Approach

(3) Under overtaking approach of single ship, initial

motion parameters of our ship are: $V = 12$, $B = 0^{\circ}$, and the coordinate is (0,2); initial motion parameters of the target ship are: $V = 18$, $B = 0^{\circ}$, and the coordinate is(0,0). According to collision risk index analysis, there is collision risk at the preliminary stage, and the collision avoidance process is implemented directly, as shown in Fig. 5.



the own ship dynamics

sailing direction: 0

speed: 12

longitude: 0

latitude: 2

the target ship dynamics

sailing direction: 0

speed: 18

longitude: 0

latitude: 0

```
restoring sailing routes angle: -59.10
avoiding sailing time: 9.0000min
avoiding sailing distance: 3.8490n mile
algorithm running time: 281.0000ms
PSO+BFO algorithm results:
avoiding steering angle: 70.76
restoring sailing routes angle: -60.00
avoiding sailing time: 6.8000min
avoiding sailing distance: 3.7586n mile
algorithm running time: 194.0000ms
```

Figure5. Interface for Path Optimization under Overtaking Approach

In the above collision avoidance interfaces, by clicking the path generation buttons, the comparison between bacterial foraging algorithm and intelligent bacterial foraging algorithm of particle swarm in optimized collision avoidance parameters will be presented, as shown in table 1: according to table 1, on the one hand, intelligent bacterial foraging algorithm of particle swarm can reduce the ship avoidance angle, reasonably select resumption angle of the ship, decrease the sailing time of avoidance, and shorten the sailing distance, so efficiency of intelligent ship avoidance is enhanced. On the other hand, particle swarm optimization can increase the convergence rate and reproduction quality of bacteria, so time taken by the algorithm is obviously decreased. Besides, more collision avoidance time is provided for seaman and the possibility of improper operation is reduced.

Ⅵ. CONCLUSION

Intelligent bacterial foraging algorithm of particle swarm has advantages of strong global searching ability and high local convergence rate. Therefore, this algorithm is applied to optimization of ship collision avoidance strategies in this paper. Besides, the dynamic collision avoidance processes based on this algorithm under three approach situations are demonstrated via VS2010 simulation interface. By displaying the comparison parameters of collision avoidance, feasibility and superiority of applying this algorithm to intelligent ship collision avoidance are verified directly and visually. This has laid a foundation for implementation of multi-ship collision avoidance process in the next step.

TABLE I. Optimized Collision Avoidance Parameters

	avoiding steering angle (°)	resumption sailing routes angle (°)	Avoiding sailing time (min)	Avoiding sailing distance (n mile)	algorithm running time (ms)
BFO head-on-head	49.08	-55.67	9	4.0215	234
PSO+BFO head-on -head	40.42	-60	6.8	3.6688	172
BFO crossing	39.62	-39.98	24.5815	10.1484	140
PSO+BFO crossing	30	-56.99	9.1701	4.2253	132
BFO over-taking	77.62	-59.10	9	3.8490	281
PSO+BFO over-taking	70.62	-60	6.8	3.7586	194

REFERENCES

[1] SHI Pingan and LI Yongji, “Artificial Intelligence Application in the field of warships collision avoidance,”Guang Zhou maritime college journals, vol. 2, 2007, pp.1-3, doi:10.1127/science.1066467.

[2] CHEN Xide and LIU Yuan, “Study on Ship Collision Avoidance Trajectory Optimization in Inland Waterways with Genetic Algorithm,” Journal of Navigation of China, vol. 2, 2006, pp.38-46,doi:675.96.

[3] BAI Yiming, HAN Xingjie and MENG Xianrao, “Optimization of strategy for Ship Collision Avoidance with Danger Model Immune Algorithm,” Journal of Navigation of China, vol. 35(2), 2012, pp.29-32.

[4] TIAN Yubo and PAN Pengpeng, “Application of Improved Immune Particle Swarm Algorithm in Ship Collision Avoidance,” Journal of Navigation of China, vol. 34(1), 2011, pp.48-51.

[5] M.D.Nguyen,V.T.Nguyen and H.Tamaru, “Automatic Collision Avoiding Support System for Ships in Congested Waters and at Open Sea,”. International conference on Control, Automation and Information Sciences, 2012, pp.96-101, doi: 10.1109/ICCAIS.2012.6466638.

[6] MA Wenyao and YANG Jiaxuan, “Optimization of Ship Collision Avoidance Route based on Bacterial Foraging Algotithm,”Journal of Da lian Maritime University, vol. 39(2), 2013, pp. 21-24.

[7] ArijitBiswas, SambartaDasgupta, Swagatam Das and Ajith Abraham, “Synergy of PSO and Bacterial Foraging Optimization A Comparative Study on Numerical Benchmarks,” Innovations in Hybrid Intelligent Systems, ASC 44, 2007, pp. 255-263, doi:10.1109/WICT.2012.6409165.

[8] Yosra JARRAYA, Souhir BOUAZIZ, Adel M. ALIMI, and Ajith Abraham, “A Hybrid Computational Chemotaxis in Bacterial Foraging Optimization Algorithm for Global Numerical Optimization,” 2013 IEEE International Conference on Cybernetics,pp. 213-218,doi: 10.1109/CYBConf.2013.6617428.

[9] K. M. Passino, “Biomimicry of bacterial foraging for distributed optimization and control,” IEEE Control Systems Magazine, vol. 6(2), 2002, pp. 52–67, doi: 10.1109/MCS.2002.1004010.

[10] Xue Yanzhuo, “Automatic Simulation of Ship Navigation in Confined Waterways, D” The University of Strathclyde in ship building and ocean engineering. 2010.