

# Distributed Multi-Objective Algorithm for Preventing Multi-Ship Collisions at Sea

Jinxin Li, Hongbo Wang, Zhiying Guan and Chong Pan

*(State Key Laboratory of Integrated Optoelectronics, College of Electronic Science and  
Engineering, Jilin University, Changchun, China)*

(E-mail: [wang\\_hongbo@jlu.edu.cn](mailto:wang_hongbo@jlu.edu.cn))

Avoidance of collisions at sea is crucial to navigational safety. In this paper, we use a distributed algorithm to communicate the entire collision avoidance trajectory information for each ship. In each communication, we suggest a new improvement function considering safety and efficiency to identify the avoidance ship in each cycle. Considering the nonlinear collision avoidance trajectory of ships, a new method for calculating the degree of danger using a velocity obstacle algorithm is proposed. Therefore, in each communication, each ship considers the avoidance behaviours of other ships in planning its avoidance trajectory. Additionally, we combine bi-criterion evolution (BCE) and the ant lion optimiser to plan the entire collision avoidance path. Three scenarios are designed to demonstrate the performance of this method. The results show that the proposed method can find a suitable collision-free solution for all ships.

## KEY WORDS

1. Bi-Criterion Evolution.
2. Distributed Local Search Algorithm.
3. Velocity Obstacle Algorithms.
4. Multiple Ships Collision Avoidance.

Submitted: 28 May 2019. Accepted: 30 October 2019. First published online: 3 March 2020.

1. INTRODUCTION. Although navigation technology has considerably improved, incidents of collisions between ships still occur. Finding a collision-free path is the main task in collision avoidance problems. Marine accident surveys have shown that more than 80% of collision accidents are caused by the negligence and irresponsibility of navigators (Zhang et al., 2013; Sormunen et al., 2016). Therefore, intelligent collision avoidance technology has become one of the key technologies for achieving ship automation in recent years.

Although considerable investments in intelligent collision avoidance have been made, there are still many shortcomings in collision avoidance decisions involving multiple ships. Most current collision avoidance studies have assumed that one ship is implementing collision avoidance strategies while other target ships maintain their headings. To learn the intentions of other ships, communication between ships becomes important. Kim et al. applied distributed algorithms in collision avoidance research and proposed some improvements over traditional methods. For example, in the distributed local search algorithm

(DLSA) (Kim et al., 2014) ships exchange intentions with each other. The distributed tabu search algorithm (Kim et al., 2015) introduced a tabu search technique to avoid local optima. The distributed stochastic search algorithm (Kim et al., 2017) changes ship intentions in a random manner to reduce the number of communications between ships. In this paper, a distributed algorithm principle is applied, and information for the entire route is used in communication between ships. We suggest a new improvement function that considers the safety and economy of the entire trajectory as a standard for avoidance in scenarios with multiple ships. This approach can not only reduce the collision avoidance operations of ships but also reduce the number of communications between ships in the whole collision avoidance planning.

Communication between ships can effectively simplify the complexity of multi-ship encounters, but detection of danger of collision between ships remains difficult. Szlapczynski and Szlapczynska (2016) proposed using the ship domain for risk calculations considering the shortcomings of closest point of approach (CPA) calculations. The ‘collision diameter’ used by Altan (2019) considers the probability of maritime accident. Kozynchenko and Kozynchenko (2018) applied fuzzy theory and neural networks to predict collision risks. These methods may struggle to make accurate judgements if the trajectory of a ship is nonlinear, however, so it is necessary to adopt a more effective method to assess risk. Huang et al. (2018) used the velocity obstacle algorithm to monitor the risk of ship collisions. Shi et al. (2018) applied the velocity obstacle algorithm to solve a multi-agent collision avoidance problem. The above studies found that the velocity obstacle algorithm can assess the collision risk of a ship and rapidly find a collision-free solution in multiple-obstacle scenes. Therefore, in this paper, combined with the distributed algorithm principle, the velocity obstacle algorithm is used to solve the collision risk assessment problem in multi-ship scenarios when the trajectory intention of the target ship is known.

There are currently many methods to plan a collision avoidance path. For example, Lisowski (2018) applied differential games for collision avoidance planning. Additionally, evolutionary computation methods have been used to find the optimal path (Szlapczynski and Szlapczynska, 2012), and genetic, artificial potential fields and particle swarm optimisation algorithms have been applied to find the optimal path in automatic collision avoidance (Kang et al., 2018; Ni et al., 2018; Lyu and Yin, 2019). In this paper, considering the need to plan entire collision avoidance paths at a given time and meet the relevant safety and economic objectives, a multi-objective algorithm is used to select the collision avoidance path. In the case of multiple ships, however, the Pareto front may be discontinuous. Therefore, it is difficult to find all Pareto-optimal solutions using the algorithm based on a non-Pareto criterion, and an algorithm based on the Pareto criterion will have very slow convergence. The bi-criterion evolution (BCE) algorithm can effectively solve these problems (Li et al., 2016), because the entire trajectory can be represented by four variables, namely: the time and range of action at a given avoidance point and the time and range of action at a given resumption point. These four variables are taken as the parameters of the algorithm optimisation to find the desired avoidance path.

In summary, in this paper, we use a distributed algorithm to communicate the intentions of avoidance among ships. Under the condition of knowing the trajectory of the target ship, the velocity obstacle algorithm is used to assess the corresponding danger. Additionally, based on the velocity obstacle method, we propose a new method to calculate the degree of risk, which is regarded as a goal of path optimisation. Because the essence of the velocity

obstacle method is related to the ship domain, and the shape of the ship domain is irregular, we apply ship domains related to the ship itself for all ships here instead of expanding the ship domain of the target ship. BCE is used to select the trajectory of the ship, and a population updating method based on the ant lion algorithm is introduced to replace and improve upon the crossover operator in BCE. We suggest a new improvement function considering safety and efficiency to discriminate the action ship in each communication. Therefore, the method adopted in this paper reduces the number of communications in the process of distributed algorithm, and the velocity obstacle method in the field of aggregate ships can better judge the danger between ships. In addition, a new dangerous calculation method is proposed to make it more suitable for a multi-ship collision avoidance environment. By using the multi-objective algorithm instead of deterministic action time as the distributed algorithm, the collision avoidance trajectory has greater autonomy.

The rest of the paper is organised as follows. Section 2 mainly explains the principles of the velocity obstacle algorithm and the distributed algorithm and the calculation of the parameters required for collision avoidance. The principle of the BCE algorithm embedded in the ant lion algorithm and the specific application of the method in ship collision avoidance are explained in Section 3. In Section 4, the simulation examples based on this method are given for multi-ship encounters. Finally, a summary and conclusions are given in Section 5.

## 2. DESCRIPTION OF THE COLLISION AVOIDANCE METHOD.

2.1. *Danger model based on velocity obstacle algorithm.* The velocity obstacle (VO) algorithm has been applied in many fields, such as vehicles, robots, and unmanned aerial vehicles, to avoid collisions (Mercado et al., 2015; Wang, 2015; Lee et al., 2017). Because the VO algorithm itself provides not only conflict detection results but also an optional speed set, this method has gradually been applied in the ship field in recent years (Huang et al., 2018; Song et al., 2018). In the velocity space, the collision risk can be converted into visible velocity contours. Figure 1 shows the principle of the VO algorithm. The speed of the ‘own ship’ (OS) is  $\vec{v}_o$ , its position is  $P_o(t)$ , and the speed of the target ship (TS) is  $\vec{v}_t$ , and its position is  $P_t(t)$ . Therefore, at time  $t$ , the collision between OS and TS is defined as follows:

$$P_o(t) \in P_t(t) \oplus ConfP(O, R) \tag{1}$$

where  $ConfP$  is an abbreviation of ‘conflict position’, which depends on the shape and size of the ship. The operation  $\oplus$  is the Minkowski addition. Therefore, for a course change of TS in the collision process, OS is informed, and the VO set can be expressed as follows:

$$VO = \bigcup_{t_f}^{\infty} \left( \frac{P_j(t_f) - P_j(t_0)}{t_f - t_0} \oplus \frac{ConfP(O, R)}{t_f - t_0} \right) \tag{2}$$

where  $t_f$  is the time when two ships collide and  $t_0$  is the initial time. As shown in Figure 1, a VO is a cone in the velocity space. To determine whether there is a risk of collision between two ships, it is necessary only to determine whether the velocity vector  $\vec{v}_o$  of OS is in the above VO set.

In this paper, the VO algorithm performs well because the communication between ships can determine the behaviour of each ship. According to Szlapczynski and

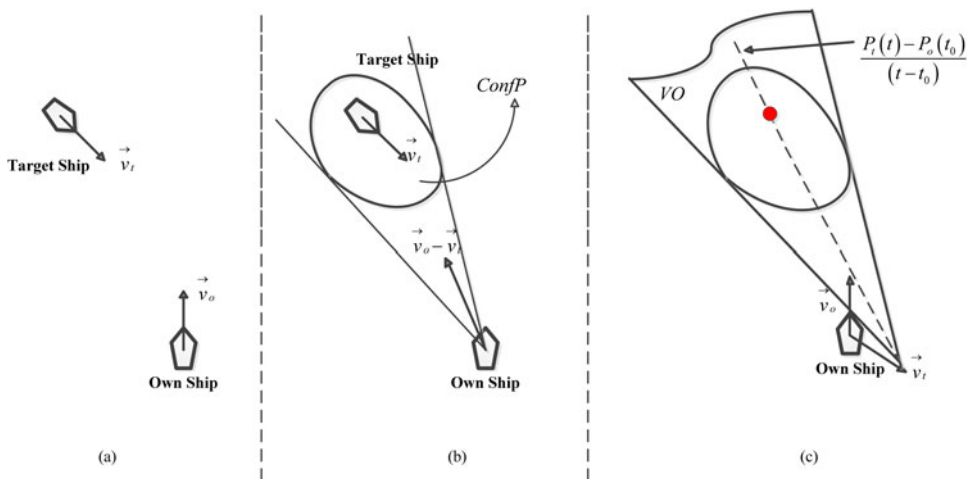


Figure 1. Graphical interpretation of the VO set: (a) scenario in geographical space; (b) VO set composed of the relative velocities; the ship domain is treated as *ConfP*; (c) transformation of the VO set through  $\vec{v}_t$ .

Szlapczynska (2016), there are certain defects in CPA discrimination. Thus, *ConfP* is represented as the ship domain here, and the quaternion ship domain is adopted (Wang, 2013), as shown in Figure 1. Because the ship domain is irregular, it is unreasonable to consider only a certain proportion of the TS to be enlarged. Therefore, the ship domain of each ship should be retained in the form of a dual ship domain. In this way, there are two collision risk values for the two ships. If one value is beyond the collision limit, it is considered that a collision will occur. The specific formula for the ship domain is as follows (Wang, 2013):

$$\begin{cases} R_{\text{fore}} = \left( 1 + 1.34\sqrt{A_D^2 + (D_T/2)^2} \right) L \\ R_{\text{aft}} = \left( 1 + 0.67\sqrt{A_D^2 + (D_T/2)^2} \right) L \\ R_{\text{starb}} = (0.2 + D_T/L) L \\ R_{\text{port}} = (0.2 + 0.75D_T/L) L \end{cases} \quad (3)$$

where  $R_{\text{fore}}$ ,  $R_{\text{aft}}$ ,  $R_{\text{starb}}$  and  $R_{\text{port}}$  represent the four radii of the ship domain,  $A_D$  and  $D_T$  are the advance and the tactical diameter, respectively.  $A_D$  and  $D_T$  are related to the speed of the ship, so the radius is related to the length and speed of the ship.

In the case of multi-ship encounters, it is difficult to calculate the danger between ships via CPA because the TS may have a nonlinear trajectory. The VO algorithm converts the trajectory of the TS in geographic space into velocity space and can clearly identify the VO set in which the velocity vector of OS is located. Therefore, the risk between ships is quantitatively calculated based on the velocity space of the VO algorithm. As shown in Figure 2, the degree of risk between ships can be seen as the degree to which the velocity is in the VO set. Because the centreline is the motion trajectory of the TS in geographic space mapped to velocity space, the closer to the centreline, the greater the corresponding risk, and the closer to the boundary, the lower the risk. The formula for calculating the degree

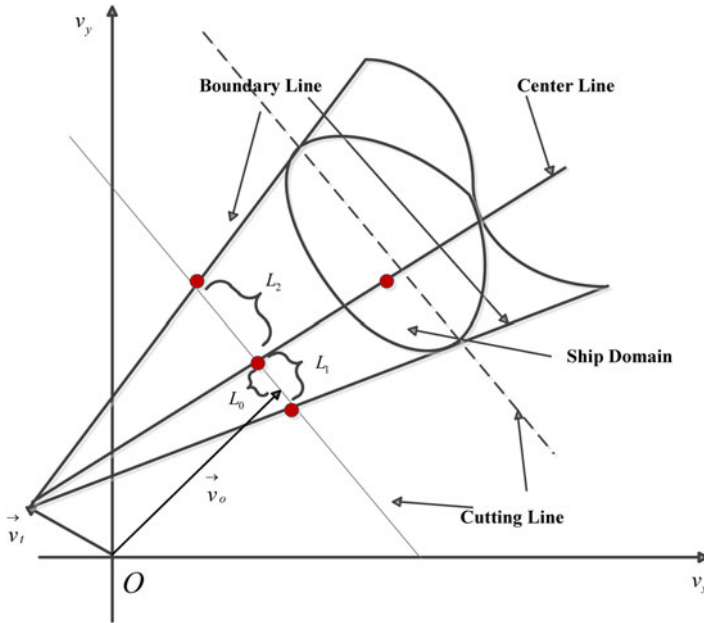


Figure 2. Principle of calculating risk based on the VO algorithm.

of risk is as follows:

$$\text{Risk} = \begin{cases} \frac{2}{1 + e^{(\log 3 \cdot (L_0/L_1))}}, & x_0 \cdot v_y - y_0 \cdot v_x > 0 \\ \frac{2}{1 + e^{(\log 3 \cdot (L_0/L_2))}}, & x_0 \cdot v_y - y_0 \cdot v_x \leq 0 \end{cases} \quad (4)$$

where  $L_0$ ,  $L_1$ , and  $L_2$  are the distances from the centre intersection to the velocity point and the intersections to the two sides in the velocity space, respectively, as shown in Figure 2;  $(x_0, y_0)$  are the coordinates of the ship, and  $(v_x, v_y)$  are the components of the ship velocity.  $x_0 \cdot v_y - y_0 \cdot v_x$  is mainly used to determine whether the speed of the ship is above or below the centreline. The value of the risk calculated by Equation (4) is 1 at the centreline and 0.5 at the boundary lines. The farther from the VO set the risk value is, the closer the value is to 0; therefore, the range of Risk is  $[0, 1]$ .

The method of calculating the feasible risk in this study is in contrast to the degree of domain violation (DDV) method proposed by Szlapczynski and Szlapczynska (2016). In the scenario considered, the distance at the closest point of approach (DCPA) is 1 n mile between all ships, as shown in Figure 3. A comparison between the two methods noted above is presented in Table 1. The DDV used here is a calculation involving the entire space, that is, the range of DDV is  $[0.5, 1]$  within the ship domain, and the range of DDV is  $[0, 0.5)$  outside the ship domain. In Table 1, in the case where the DCPA is 1 n mile, both the DDV and Risk methods can calculate the degree of danger. The results obtained with the two methods are similar, and it is possible to distinguish whether a ship has entered the domain of the TS. At the same time, the method used in this paper can perform a risk calculation for nonlinear navigation of the TS. Therefore, it is feasible to use the risk calculation method in this paper for collision avoidance optimisation. In addition,

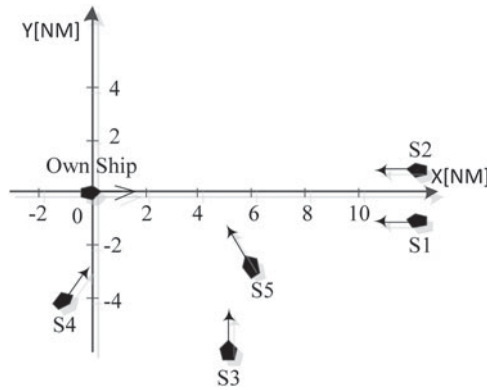


Figure 3. Illustration of all scenarios in Table 1.

Table 1. Comparison of DDV and Risk – results.

Number	Location [n mile]	Velocity [kn]	Course [°]	DCPA [n mile]	DDV [/]	Risk [/]	VO set
S1	[12, -1]	12	270	1	0.6048	0.5939	IN
S2	[12, 1]	12	270	1	0.4706	0.4822	OUT
S3	[5, -6]	12	0	1	0.8295	0.8032	IN
S4	[-1, -4]	15	53	1	0.7315	0.7136	IN
S5	[6, -2.89]	12	330	1	0.7825	0.7715	IN

the velocity obstacle algorithm applied in this paper considers the VO set constructed for collisions within 40 min; that is, the collision danger within the range of 10 n mile will be detected when the ship speed is 15 kn.

2.2. *Distributed algorithm and related parameter calculations.* In this paper, to reduce the number of collision avoidance actions for a ship, each communication involves the exchange of all collision avoidance trajectory information for the ship, so the principle of DLSA is adopted here. Only one avoidance is considered, so the entire avoidance track can be represented by four variables, as shown in Figure 4. This study defines two points: the avoidance point and restoration point (Tsou and Hsueh, 2010; Wei et al., 2015; Li et al., 2019). The action information for these points is optimised, as shown below.

1. The time of arrival  $T_o$  for the collision avoidance points indicates the time at which the reference ship is detected by the TS to avoid collision action (unit: min).
2. The avoiding range  $C_o$  of the relative original route at the avoidance point indicates that the ship is in a position to avoid the point requiring action (unit: degree).
3. The time  $T_a$  between a turn for collision avoidance and a turn for navigational restoration is the duration of the collision avoidance navigation (unit: min).
4. The return range  $C_a$  of the relative avoidance route at the restoration point should be at least larger than  $C_o$ ; otherwise, it cannot go back to the original route (unit: degree).

By considering the speed of the ship, the length of the avoidance route and the restriction on ship movement, the scope of the four parameters ( $T_o, C_o, T_a, C_a$ ) is limited to  $[0, 90]$ . The above four variables are transmitted through the communication between ships, and each ship has a specific and known collision avoidance strategy; thus, the behaviour of

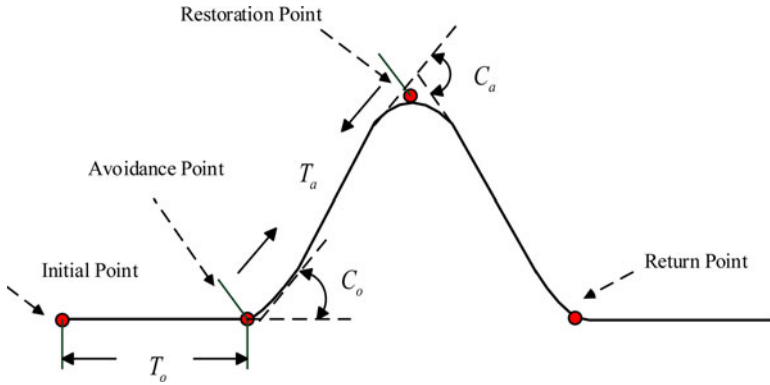


Figure 4. Schematic diagram of the obstacle avoidance trajectory.

other ships can be considered when planning the route. Because it takes time to exchange information and plan the trajectory of each ship and the ships communicate with each other to determine the collision avoidance strategy, we set 3 min as the time step (Kim et al., 2017). Therefore, when a ship identifies a potential risk, the collision avoidance strategy is updated every 3 min. This communication cycle continues until each ship determines its own avoidance collision trajectory.

After learning the intention of each ship and planning the corresponding avoidance behaviour, a parameter must be established to determine which ship avoidance behaviour is best given the current state of the domain. Thus, an improvement function is needed to compare the avoidance trajectory of each ship. An improvement function is proposed here considering the safety of the avoidance trajectory and the smoothness of the entire trajectory, as shown in Equation (5).

$$ImpRS = \left( \sum_{i=1}^n (Risk_O^i - Risk_A^i) + Time \right) + \frac{(\cos(C_o) + \cos(C_b))}{2} \tag{5}$$

where  $Risk_O^i$  and  $Risk_A^i$  are the degrees of danger of the original route and the avoidance route for the  $i$ th ship, respectively, and  $n$  is the number of TSs. The value of Risk is the maximum value of the collision risk at three points in the trajectory: the avoidance point, the restoration point and the return point. Because a ship close to colliding should have a higher probability of being selected than other ships, time information is introduced via the variable Time. The closer a ship is to experiencing a collision, the larger the Time value, which increases the probability that the ship will be selected, as shown in Equation (6).

$$Time = \exp(-\alpha(t_f - Kt)) \tag{6}$$

where  $t_f$  is the collision time initially calculated for a ship, which generally takes the minimum value of the collision time calculated for other ships;  $t$  is the time step;  $K$  is the  $K$ th cycle; and  $\alpha$  is a constant, so that the value of the function reflects the requirements of the scene. We consider the speed of the ship and the point of collision, and  $\alpha$  is set to 0.3 (trial-and-error method). In this approach, the ships must exchange their  $ImpRS$  values after planning their collision avoidance routes, and these values are compared to determine

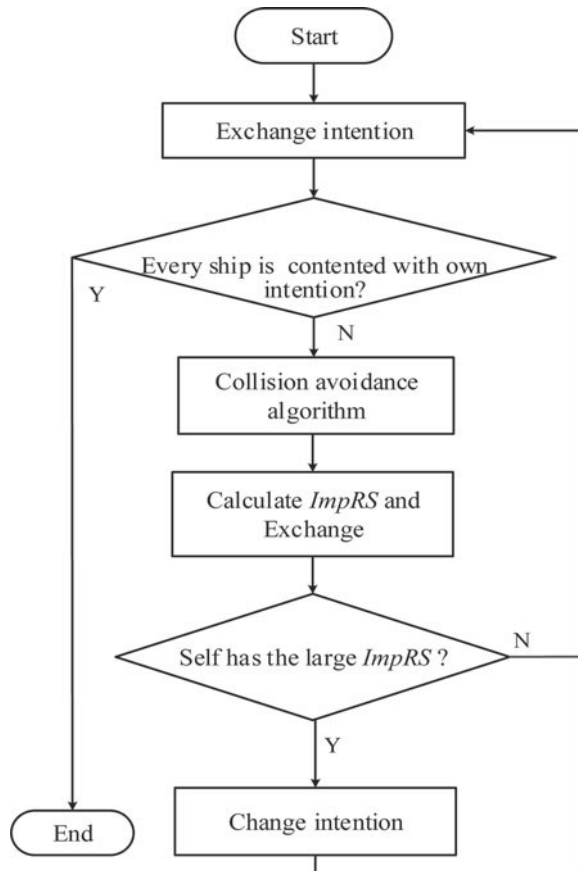


Figure 5. Procedure for ship avoidance selection.

which ship is the avoidance action ship in a given cycle. After the action ship is determined, the other ships will maintain their original routes and enter the next cycle of communication until there is no danger among ships, as shown in Figure 5.

### 3. THE DECISION-MAKING PROCESS OF SHIP COLLISION AVOIDANCE.

3.1. *BCE algorithm.* In ship collision avoidance path planning, we should consider not only safety among ships but also whether the planned ship collision avoidance trajectory is reasonable, that is, whether the length and smoothness of the path meet the physiological requirements of maritime personnel. Therefore, the planning of a ship collision avoidance trajectory should be regarded as a multi-objective problem so that the optimal solution set can provide reasonable choices for maritime personnel. In this paper, because a complex situation involving multi-ship encounters may result in a non-continuous Pareto-optimal front, we adopt Li's BCE framework for the Pareto criterion (PC) and non-Pareto criterion (NPC), which encompasses the advantages of the PC and NPC (Li et al., 2016). BCE is mainly divided into two parts: NPC evolution and PC evolution, and this approach can work with any non-Pareto evolutionary multi-objective



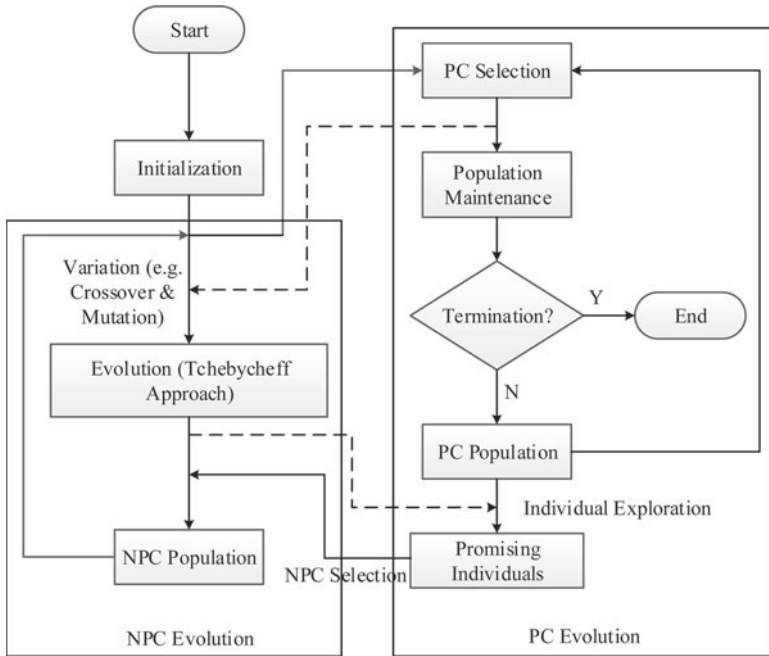


Figure 6. Procedure for BCE.

optimisation (EMO) algorithm. In this paper, the MOEA/D (multi-objective evolutionary algorithm based on decomposition) algorithm is taken as the evolution part of the NPC, and the PC evolution part mainly uses the PC to select the relevant solution set. This method can overcome the poor convergence of PC evolution and the selection pressure of NPC evolution (Li et al., 2016). The PC evolution process mainly finds the Pareto-optimal solution set and maintains the original population size. The NPC evolution process is consistent with that of the NPC algorithm framework shown in Figure 6.

The key to the complementary advantages of NPC and PC in BCE lies in the selection of promising individuals based on the PC. Promising individuals are mainly non-dominated solutions of unselected or unexplored regions in the population of the NPC. The selection criterion for promising individuals is as follows: there are no NPC individuals or only one NPC individual in the niche range of non-dominated individuals in the PC population. Obviously, such a constraint avoids exploration of undeveloped and low-density regions in the NPC evolution.

The NPC evolution is based on the Tchebycheff approach of the MOEA/D algorithm. In this paper, considering that the evolution of the NPC is closely related to its neighbours and that the BCE algorithm communicates non-dominant individuals of PC evolution with NPC population, the ant lion algorithm is used here to update the population and improve individual crossover in NPC evolution. The ant lion algorithm is a novel nature-inspired algorithm that was proposed by Mirjalili based on the process of an ant lion preying on ants in nature (Mirjalili, 2015). In the population updating scheme of the ant lion algorithm, the elite ant lion and a random ant lion are taken as the centre ants, and the ants around them conduct a random walk to generate individuals. In the multi-objective algorithm, the elite

ant lion is selected as the non-dominated solution, so the population updating process is directional. Therefore, according to the characteristics of population updating with the ant lion algorithm, the population updating strategy in BCE is improved to produce a better solution than those of traditional methods.

In this paper, because the NPC part of the BCE uses the MOEA/D algorithm, a random ant in the updated population is randomly selected from the neighbours. Because part of the initial population of PC evolution comes from the population generated by NPC evolution, the non-dominated solutions obtained from PC evolution are used in NPC evolution to determine whether there are non-dominated solutions among the neighbours of the currently updated individuals. If a solution exists, it may be randomly selected as an elite ant lion. If no solution exists, the elite ant lion randomly chooses among its neighbours. The new individual is as follows:

$$\text{Ant}_j^t = \frac{R_A^h + R_E^h}{2} \quad (7)$$

where  $\text{Ant}_j^t$  is the  $j$ th ant obtained from the  $h$ th iteration,  $R_A^h$  is the solution obtained by randomly walking around a random ant lion, and  $R_E^h$  is the solution obtained by randomly walking around the elite ant lion. The updating process in the ant lion algorithm is used to replace the cross-operations and improve the solution. Additionally, the mutation operation adopted is polynomial mutation (Wang et al., 2018). In this manner, the above four variables representing the entire collision avoidance trajectory are taken as the targets of algorithm optimisation, and the desired solution of the collision avoidance trajectory is obtained based on an objective function that safety and costs.

**3.2. The objective function.** In the process of planning the avoidance trajectory, a ship should consider the safety among ships and the economics of the collision avoidance trajectory. Therefore, two objective functions for security and economy are adopted to reflect the quality of the solution obtained by the BCE algorithm in this paper. The VO algorithm is combined with the ship domain to construct a method of assessing the corresponding risk, and this method is applied to calculate the risk among ships with the following safety objective function:

$$F_{\text{safe}} = \text{Risk}_M \quad (8)$$

Therefore, the range of the safety objective function  $F_{\text{safe}}$  is  $[0, 1]$ , where  $\text{Risk}_M$  is the maximum value of the collision risk at the avoidance point, restoration point and return point.

For the economic objective function, the general considerations are the time of the voyage, the length of the path and the fuel consumption of the voyage. Because collision avoidance planning occurs over a very short period compared with the time it takes to complete an entire route, it is not a good choice to consider the fuel consumption problem in collision avoidance. Therefore, treating the sailing time and sailing distance of a ship as linear variables, the path length of ship collision avoidance is taken as the economic objective function:

$$F_{\text{economy}} = \sum_k \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \quad (9)$$

where  $F_{\text{economy}}$  represents the distance between the initial point and the return point, as shown in Figure 4,  $(x, y)$  are the coordinates of the ship, and  $k$  is the index of the position

of a ship at each time. The navigation simulation is carried out through the aforementioned ship motion mathematical model. In this paper, we choose the solution according to the range of the safety objective function, that is, a solution is randomly selected in the solution set with the value of  $F_{\text{safe}}$  in the range of [0.2, 0.5] (Li et al., 2019).

3.3. *Hypothesis for collision avoidance.* In a multi-ship collision avoidance plan, if it is determined that the ship will collide with another ship, the ships communicate with each other to clarify their respective sailing intentions. Here, the provisions of the International Regulations for Preventing Collisions at Sea 1972 (COLREGS) are used to judge the direction of ship actions (Rule 13, 14, 15). The TS with the greatest risk is evaluated based on the degree of danger, and the ship is steered based on the encountered situation involving the TS and another ship. If it is determined that the OS is the stand-on ship, the OS's steering is judged according to the encounter situation between the second dangerous ship and the OS, and so on, until the OS is found to give way. Alternatively, if all the TSs are judged and the OS is still the stand-on ship, and then the OS will not carry out the avoidance plan. Considering that the speed of the ship during the large angle steering ( $60^\circ$ – $80^\circ$ ) will decrease, the return range  $C_a$  is limited to below 60 degrees.

According to the sailing intention of each ship, the multi-objective algorithm is used to plan each ship's avoidance route, and the improvement degree of this route relative to the original course is calculated. Through the communication between ships regarding their respective route improvements, the ship with the greatest improvement sails along the avoidance route. Other ships maintain their original routes, enter the next cycle, and again perform collision avoidance planning. This process continues until there is no risk of collision between ships. Taking into account the communication events and the run time of the algorithm, the time step is set to 3 min. After reaching the 3-min threshold, the ship will sail along the route planned in the cycle and enter the next time step. Since the ship domain itself takes into account the length of the ship, the ship is considered as a point, and the location of the point is the centre of the ship. The simulation program is MATLAB 2016b.

#### 4. CASE STUDY.

4.1. *Four-ship encounter – situation 1 (mainly for crossing and head-on encounters).* In this scenario, the encounters between ships are dominated by crossing encounters and head-on encounters, as shown in Figure 7(a). The respective parameter information for the ships is shown in Table 2. In judging whether OS is the stand-on ship or the give-way ship, according to COLREGS, as long as a TS requires OS to give way, then the state of OS is the give-way ship. Therefore, when assessing the state of  $S_2$ , although the risk associated with  $S_1$  is the greatest, according to COLREGS,  $S_2$  is the stand-on ship, and the second-most-dangerous ship for  $S_2$  is  $S_4$ . There is a crossing encounter between the two ships. Notably,  $S_4$  is on the starboard side of  $S_2$ , and  $S_2$  is the give-way ship. After the action status of each ship is defined, we use the BCE algorithm to plan the avoidance trajectory of each ship and calculate the route improvement degrees to identify the avoidance ship in the current cycle. The planned trajectory is shown in Figure 7(b), where the circle around the ship is the scope of the ship domain.

The distance between the ships during collision avoidance is shown in Figure 8. The numbers on the lines in the figure represent the closest distance between ships in the collision avoidance trajectories (calculate the closest distance based on the centre point of the ship). The closest distance between the ships is greater than the radius of the ship domain

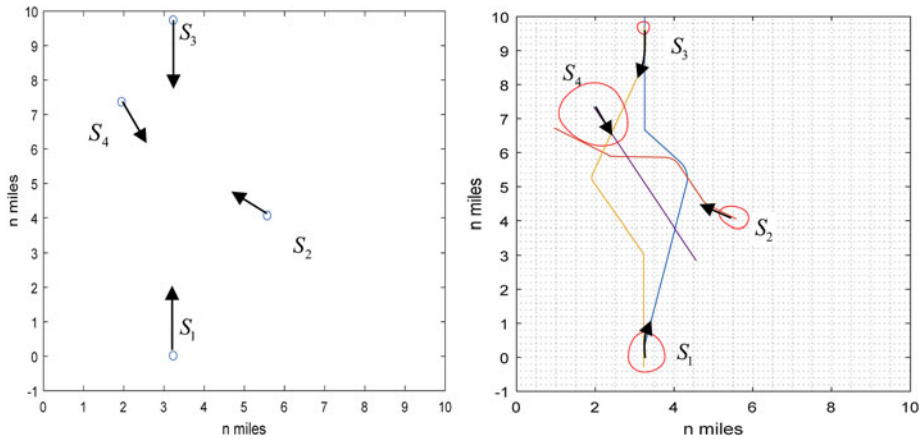


Figure 7. Multi-ship encounter situation (scenario 1) (left) and ship avoidance trajectories (right).

Table 2. Initial multi-ship encounter situation (crossing and head-on encounters).

Ship	Position (n mile)	Length (m)	Speed (kn)	Course (°)	Risk to others	Ship domain (m) ( $R_{fore}, R_{aft}, R_{starb}, R_{port}$ )
$S_1$	(3.24, 0)	250	15	0	(0.99, 1, 1)	(1359, 804, 974, 743)
$S_2$	(5.58, 4.05)	250	7.5	300	(0.99, 0.14, 0.84)	(886, 568, 620, 477)
$S_3$	(3.24, 9.72)	60	15	180	(1, 0.52, 0.87)	(422, 251, 323, 245)
$S_4$	(1.96, 7.35)	350	7.5	150	(0.99, 0.58, 0.36)	(2244, 1297, 1739, 1322)

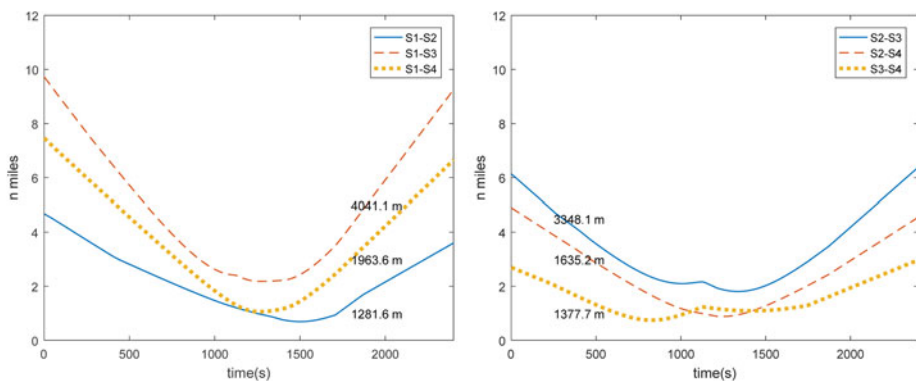


Figure 8. Relative distance between ships.

on the side where the ships pass, which ensures that safety is maintained between the ships. In Figure 9, the ship navigation at the critical time node is illustrated. The presence of the ship domain around each ship rather than the expansion of the ship domain not only ensures safety but also avoids the unnecessary oversteering of the ship. Among them,  $S_1$ ,  $S_2$  and  $S_3$  all turn to the starboard side, mainly according to COLREGS. Table 3 shows the trajectory improvement degree of each ship in each cycle. The order in which the ship collision

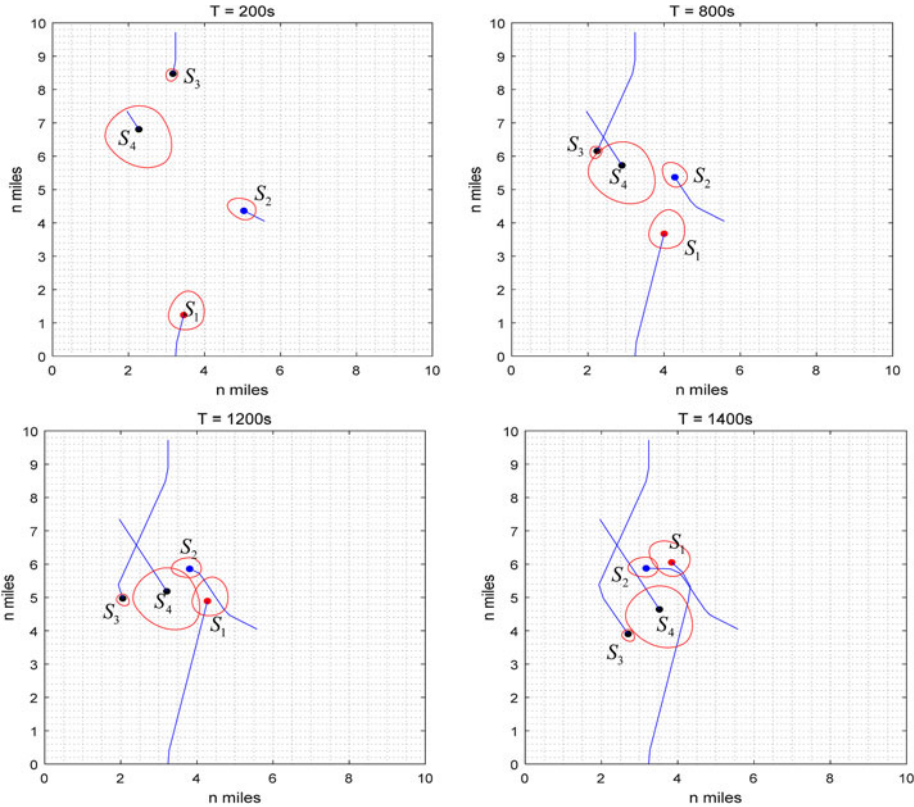


Figure 9. Trajectory of each ship at given times.

Table 3. The value of *ImpRS* in each cycle.

	$S_1$	$S_2$	$S_3$	$S_4$	$T_o$	$C_o$	$T_a$	$C_a$
First cycle	1.98	1.21	1.58	0.91	0.6	13.4	19.9	57.8
Second cycle	0	0.4794	0.6079	0.5899	4	21.5	14.3	53.2
Third cycle	0	0.9746	0	0.1939	6.3	30.4	11.2	60.0
Fourth cycle	0	0	0	0	—	—	—	—

avoidance trajectory is selected can be observed. The ship with a planned avoidance trajectory does not participate in avoidance planning in the next cycle, and the improvement degree is zero. The action ship in the first cycle is  $S_1$ , and the action ship in the second cycle is  $S_3$ , followed by  $S_2$  in the third cycle. The collision avoidance parameters of these ships are shown in Table 3. Due to the avoidance behaviour of the first three ships,  $S_4$  is not in danger at this time, so it maintains its original course. In this scenario, the avoidance of each ship will not generate new collisions, and the planning of the following ships will consider the avoidance of the previous ships so that all ships can sail safely.

4.2. *Four-ship encounter – situation 2 (mainly for crossing and overtaking encounters).* In this scenario, an overtaking encounter situation is added, as shown in Figure 10(a). In Table 4, there is a danger of collision between  $S_4$  and  $S_1$ , and the two

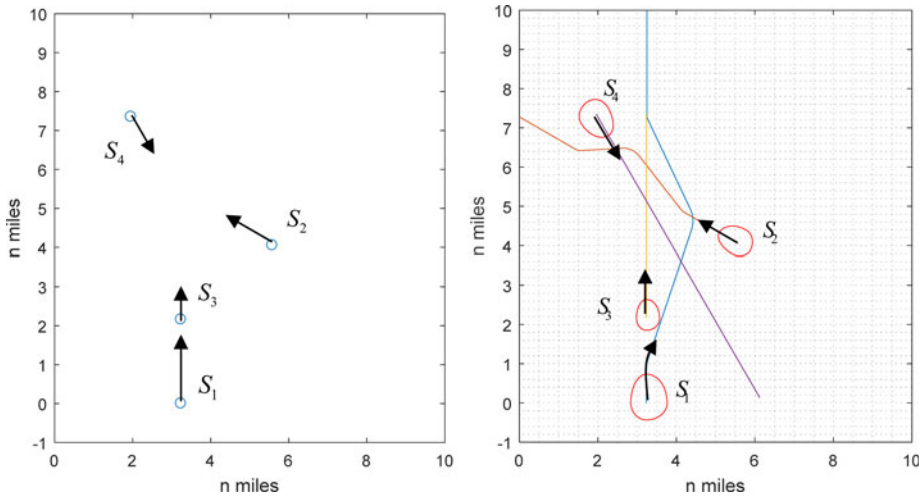


Figure 10. Multi-ship encounter situation: scenario 2 (left) and ship avoidance trajectories (right).

Table 4. Initial multi-ship encounter situation (crossing and overtaking encounters).

Ship	Position	Length (m)	Speed (kn)	Course (°)	Risk to others	Ship domain (m)
	(n mile)					( $R_{fore}, R_{aft}, R_{starb}, R_{port}$ )
S <sub>1</sub>	(3.24, 0)	250	15	0	(0.11, 1, 0.43)	(1359, 804, 974, 743)
S <sub>2</sub>	(5.58, 4.05)	250	7.5	300	(0.29, 0.10, 0.53)	(886, 568, 620, 477)
S <sub>3</sub>	(3.24, 2.16)	250	7.5	0	(1, 0.03, 0.42)	(886, 568, 620, 477)
S <sub>4</sub>	(1.96, 7.35)	250	12	150	(0.58, 0.49, 0.41)	(1169, 710, 833, 637)

ships experience a crossing encounter. Additionally, since the length of the ships is identical, the difference in ship speed leads to different sizes of the ship domain. The parameters of each ship are shown in Table 4. The trajectory of each ship after planning can be seen in Figure 10(b). S<sub>1</sub> and S<sub>2</sub> conducted avoidance operations, and S<sub>3</sub> and S<sub>4</sub> maintained their original courses. The ship avoidance and steering manoeuvres were in accordance with COLREGS.

The distance between the ships in each voyage is shown in Figure 11, and the numbers on the lines indicate the closest distance between the ships (calculate the closest distance based on the centre point of the ship). The closest distance between the ships is greater than the radius of the ship domain on the side where the ship passes. For S<sub>3</sub> and S<sub>4</sub>, in Figure 10(b), both ships are sailing based on their original headings. In addition, in Figure 12, at 900 s, S<sub>3</sub> and S<sub>4</sub> pass on the starboard side. The minimum distance between the two ships is 854.6 m, which is greater than the starboard radius of the two ships. Thus, it is safe for the ships to pass at that distance. Figure 12 shows the navigation status of the ships at each point in time. In Table 5, we can see the improvement degree of each ship in each time cycle and the avoidance action for that cycle. The avoidance parameter values of each ship are shown in Table 5.

4.3. *Multi-ship and static obstacle encounters.* Static obstacles are introduced in this scene. Considering the convenience in the algorithm, obstacles are assigned circular shapes, as shown in Figure 13(a). And the parameters of each ship are shown in Table 6. Here, the

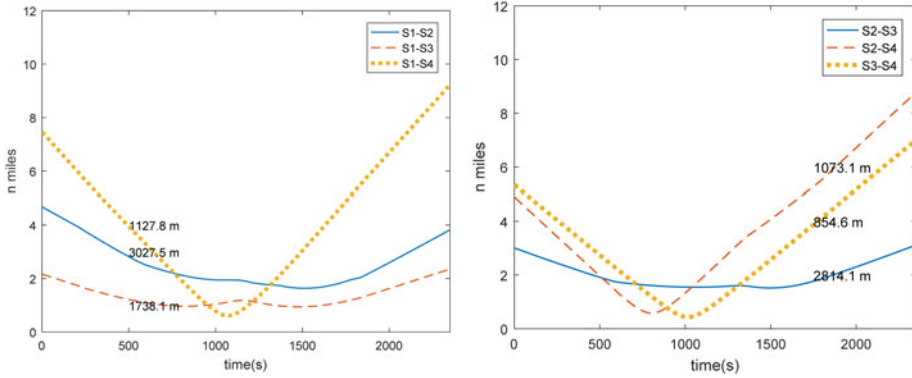


Figure 11. Relative distance between ships.

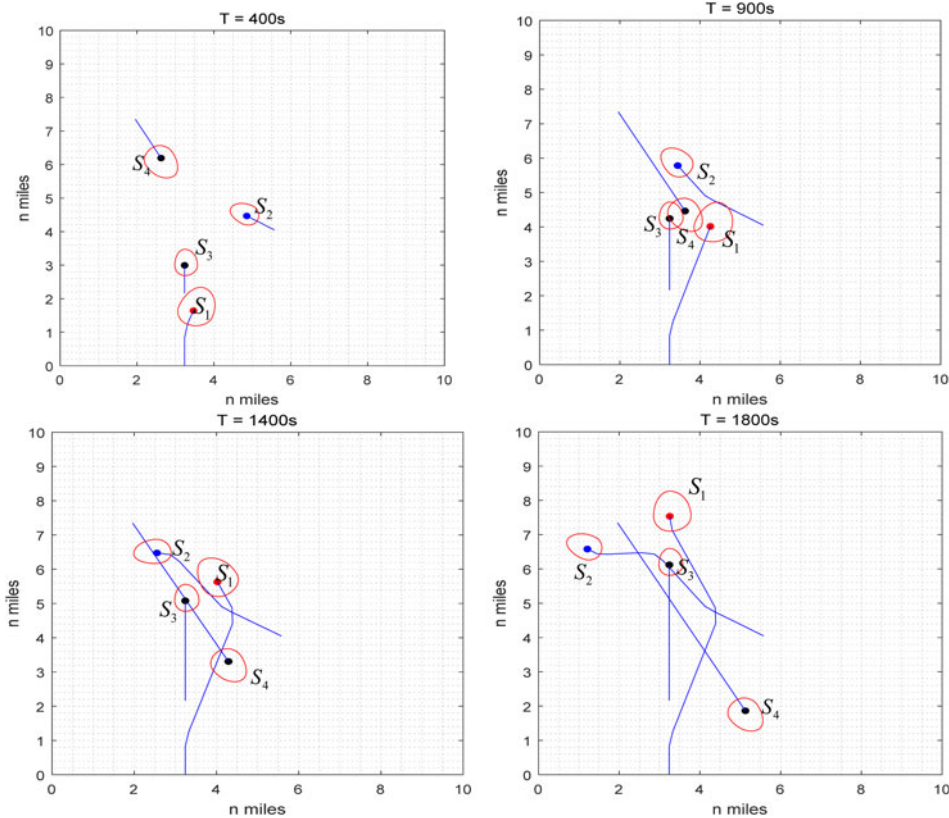


Figure 12. Trajectory of each ship at given times.

radius of the static obstacle is set to 1180 m, which can be regarded as a stationary ship or buoy not considered in the navigation plan. The avoidance trajectory of each ship is shown in Figure 13(b). In considering the ship avoidance direction for static obstacles, only the direction of the minimum steering amplitude is considered here to determine which

Table 5. The value of *ImpRS* in each cycle.

	$S_1$	$S_2$	$S_3$	$S_4$	$T_o$	$C_o$	$T_a$	$C_a$
First cycle	0.903	0.385	0	0.8537	3.3	19.3	14.6	45.5
Second cycle	0	0.2985	0	0	9.6	22.2	11.2	57.2
Third cycle	0	0	0	0	—	—	—	—

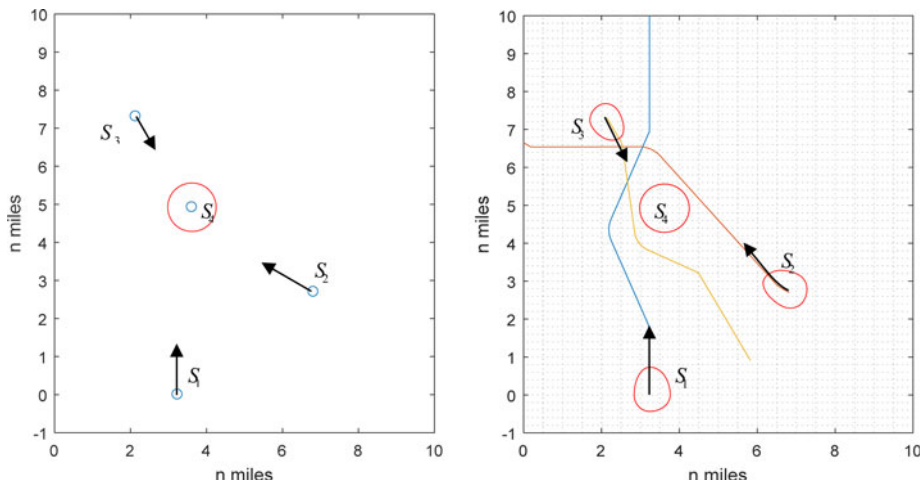


Figure 13. Multi-ship and static obstacle encounter situation (scenario 3) (left) and ship avoidance trajectories (right).

Table 6. Initial multi-ship and static obstacle encounter situation.

Ship	Position (n mile)	Length (m)	Speed (kn)	Course (°)	Risk to others	Ship domain (m) ( $R_{fore}, R_{aft}, R_{starb}, R_{port}$ )
$S_1$	(3.24, 0)	250	15	0	(0.39, 0.29, 0.68)	(1359, 804, 974, 743)
$S_2$	(6.82, 2.69)	250	15	300	(0.59, 0.83, 0.73)	(1359, 804, 974, 743)
$S_3$	(2.14, 7.31)	250	12	150	(0.45, 0.85, 0.92)	(1169, 710, 833, 637)
$S_4$	(3.62, 4.92)	—	0	—	(0, 0, 0)	(1180, 1180, 1180, 1180)

direction to turn the ship, port or starboard. In avoidance steering tasks, the requirements of COLREGS are still considered.

Figure 14 shows the distance between the ships during their voyages (calculate the closest distance based on the centre point of the ship). As shown, the distance between each ship and the static obstacle is greater than each ship domain radius. In Figure 15, the navigation status of the ship is used in each time period. At 1,000 s,  $S_1$  and  $S_3$  pass on their respective port sides. In Figure 14, the closest distance between the two ships is 1144.8 m, which is greater than the port radius of their respective ship domains, so they are safe from colliding. At 1,400 s,  $S_1$  and  $S_2$  pass on their respective port sides. In Figure 14, the closest distance between the two ships is 1530.2 m, which is greater than the radius on the port side of each ship domain. Then, the ships complete their avoidance operations and return to their original routes to continue sailing. Table 7 shows the improved values for each ship



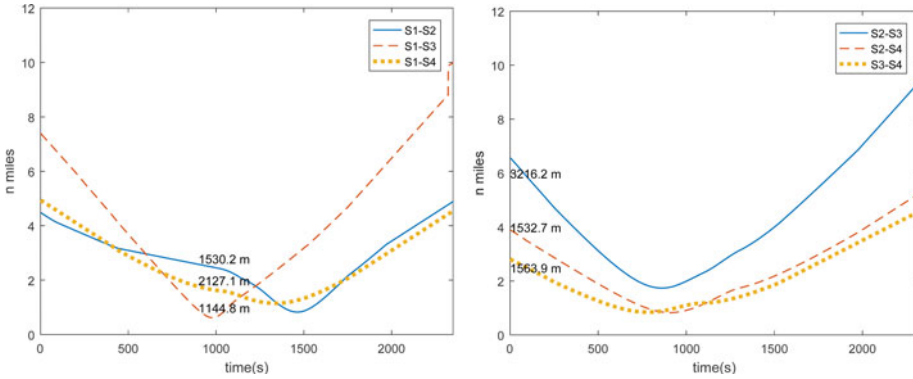


Figure 14. Relative distance between ships.

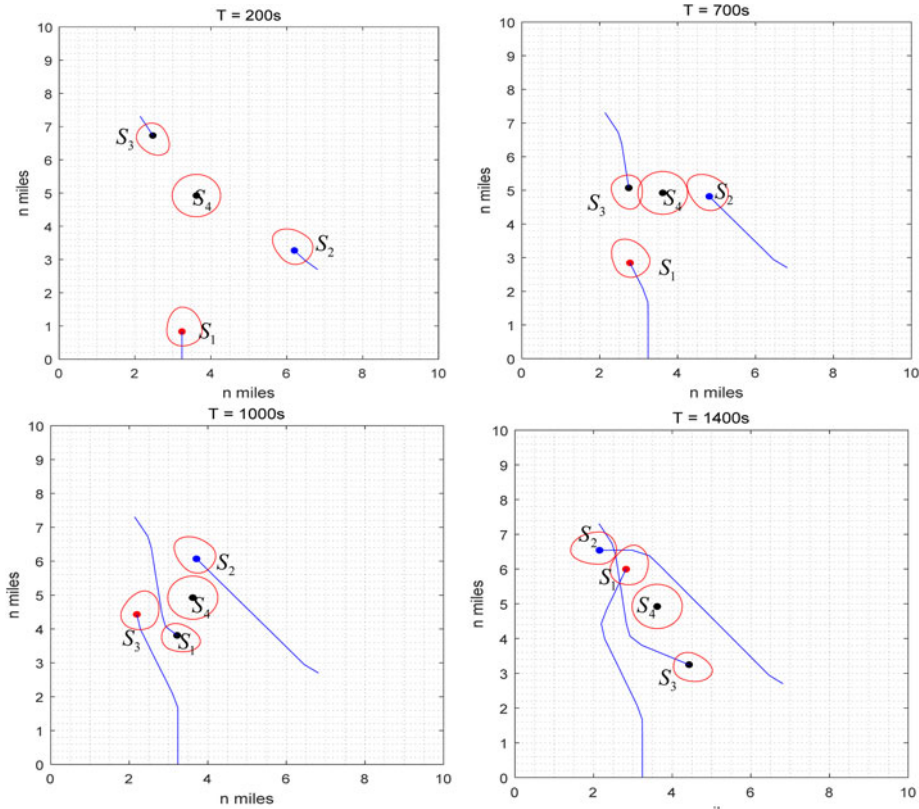


Figure 15. Trajectory of each ship at given times.

in each time cycle. Because  $S_4$  is a static obstacle, it does not need to be planned, and the corresponding value is zero.

4.4. *Computing time of the algorithm.* The calculation time of the collision avoidance method and the time of constructing the VO area are shown in Figure 16. Among them,

Table 7. The value of *ImpRS* in each cycle.

	$S_1$	$S_2$	$S_3$	$S_4$	$T_o$	$C_o$	$T_a$	$C_a$
First cycle	0.4826	1.2440	0.678	0	0.6	19.3	18.7	50.1
Second cycle	0.7257	0	0.3621	0	6.1	23.8	10.5	47.5
Third cycle	0	0	0.7651	0	3.1	22.8	12.3	58.4

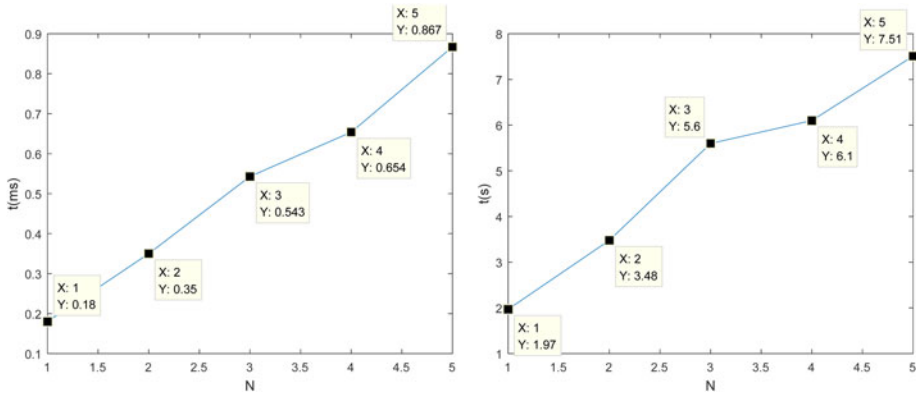


Figure 16. Construction time of VO area (left) and running time of multi-objective algorithm (right).

the horizontal ordinate  $N$  represents the number of obstacles and the running time of the program in the longitudinal coordinates. In Figure 16 (left), it can be observed that the time to construct the VO area is related to the number of obstacles. In Figure 16 (right), it is about the time of one-time collision avoidance algorithm planning for a ship. Because this paper adopts a multi-objective algorithm, assuming that the number of iterations is  $M$  and the number of population is  $P$ , the time complexity of the algorithm is generally  $O(MP)$ . From Figure 16, it can be observed that the time curve of constructing the VO zone is basically consistent with the trend of running time of the program, that is to say, the main time-consuming part of the whole algorithm is the calculation of the collision risk. At the same time, it can be observed from the figure that the running time of the algorithm increases linearly with the number of obstacles. It can be observed from the figure that the calculation time of one cycle program is 7.51 s in the case of six ships (i.e. five obstacles). Since one ship does not plan collision avoidance for the next time in each cycle, the normal time for six ships to plan each other's trajectories is 24.66 s (i.e. adding up the data in the figure). Therefore, the collision avoidance method proposed in this paper can operate normally in a general multi-ship environment.

5. CONCLUSIONS. This paper mainly studies the problem of ship collision avoidance path planning in the case of multi-ship encounters and the existence of dynamic ships and static obstacles. BCE is used to plan the entire collision avoidance trajectory; that is, the entire trajectory is considered based on four parameters:  $T_o$ ,  $C_o$ ,  $T_a$  and  $C_a$ . In this approach, although the communication times of this method are consistent with those of the DLSA algorithm in each cycle, the number of communication times will be greatly reduced from the point of view of the entire planning process. This paper combines the ship domain

with the VO method to assess potential collisions. Based on this method, a new method of calculating the risk between ships is proposed. Additionally, a new ant lion algorithm concept is added to NPC evolution to enhance the convergence of the method. Based on the steering amplitude, the reduction of risk and the time information for a collision avoidance trajectory, a new improvement function considering safety and efficiency is proposed. The improvement function is applied to select the avoidance action ship in each cycle.

Although this method can achieve good results, there are some drawbacks. For example, we do not consider the problem of velocity variation during ship turning. Moreover, we need to consider the collision avoidance planning problem with uncooperative ships, which would include trajectory prediction. In addition, this paper does not consider speed collision avoidance and the collision avoidance operation requires both a speed change and a course change, or consider multiple collision avoidance behaviour, such as multiple static obstacles environment, so as to obtain the optimal solution.

Supplementary material The supplementary material for this article can be found at <https://doi.org/10.1017/S0373463320000053>

## REFERENCES

- Altan, Y. C. (2019). Collision diameter for maritime accidents considering the drifting of vessels. *Ocean Engineering*, **187**, 106–115.
- Huang, Y., van Gelder, P. H. A. J. M. and Wen, Y. (2018). Velocity obstacle algorithms for collision prevention at sea. *Ocean Engineering*, **151**, 308–321.
- Kang, Y. T., Chen, W. J., Zhu, D. Q. and Wang, J. H. (2018). Collision avoidance path planning for ships by particle swarm optimization. *Journal of Marine Science and Technology*, **20**(6), 777–786.
- Kim, D., Hirayama, K. and Park, G. (2014). Collision avoidance in multiple-ship situations by distributed local search. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **18**(5), 839–848.
- Kim, D., Hirayama, K. and Okimoto, T. (2015). Ship collision avoidance by distributed Tabu search. *The International Journal on Marine Navigation and Safety of Sea Transportation*, **9**(1), 23–29.
- Kim, D., Hirayama, K. and Okimoto, T. (2017). Distributed stochastic search algorithm for multi-ship encounter situations. *Journal of Navigation*, **70**(4), 1–20.
- Kozynchenko, A. I. and Kozynchenko, S. A. (2018). Applying the dynamic predictive guidance to ship collision avoidance: crossing case study simulation. *Ocean Engineering*, **164**, 640–649.
- Lee, B. H., Jeon, J. D. and Oh, J. H. (2017). Velocity obstacle based local collision avoidance for a holonomic elliptic robot. *Autonomous Robots*, **41**(6), 1–17.
- Li, M., Yang, S. and Liu, X. (2016). Pareto or non-pareto: bi-criterion evolution in multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, **20**(5), 645–665.
- Li, J., Wang, H. and Zhao, W. (2019). Ship's trajectory planning based on improved multi-objective algorithm for collision avoidance. *Journal of Advanced Transportation*, Article ID 4068783, 12 pages.
- Lisowski, J. (2018). Dynamic games methods in synthesis of safe ship control algorithms. *Journal of Advanced Transportation*, **2018**, 1–8.
- Lyu, H. G. and Yin, Y. (2019). COLREGS-constrained real-time path planning for autonomous ships using modified artificial potential fields. *Journal of Navigation*, **72**(3), 588–608.
- Mercado Velasco, G. A., Borst, C., Ellerbroek, J., Van Paassen, M. M. and Mulder, M. (2015). The use of intent information in conflict detection and resolution models based on dynamic velocity obstacles. *IEEE Transactions on Intelligent Transportation Systems*, **16**(4), 2297–2302.
- Mirjalili, S. (2015). The ant lion optimizer. *Advances in Engineering Software*, **83**, 80–98.
- Ni, S., Liu, Z. J., Cai, Y. and Wang, X. (2018). Modelling of ship's trajectory planning in collision situations by hybrid genetic algorithm. *Polish Maritime Research*, **25**(03), 14–25.
- Shi, Z. Q., Junda, H., Tengli, W., Chuan, Z. and Jian, G. (2018). Optimal formation control and collision avoidance in environment with multiple rectangle obstacles. *Journal of the Franklin Institute*, **355**(15), 7626–7642.

- Song, A. L. F., Su, B. Y. R., Dong, C. Z. P. and Shen, D. W. (2018). A two-level dynamic obstacle avoidance algorithm for unmanned surface vehicles. *Ocean Engineering*, **170**, 351–360.
- Sormunen, O., Hanninen, M. and Kujala, P. (2016). Marine traffic, accidents, and underreporting in the Baltic Sea. *Zeszyty Naukowe*, **46**(118), 163–177.
- Szlapczynski, R. and Szlapczynska, J. (2012). On evolutionary computing in multi-ship trajectory planning. *Applied Intelligence*, **37**(2), 155–174.
- Szlapczynski, R. and Szlapczynska, J. (2016). An analysis of domain-based ship collision risk parameters. *Ocean Engineering*, **126**, 47–56.
- Tsou, M. C. and Hsueh, C. K. (2010). The study of ship collision avoidance route planning by ant colony algorithm. *Journal of Marine Science and Technology*, **18**(5), 746–756.
- Wang, H. B., Li, X. G. and Li, P. F. (2018). Application of real-coded genetic algorithm in ship weather routing. *Journal of Navigation*, **71**(4), 989–1010.
- Wang, N. (2013). A novel analytical framework for dynamic quaternion ship domains. *Journal of Navigation*, **66**(2), 265–281.
- Wang, T.-C. (2015). Obstacle detection and avoidance with noisy measurements using danger zone concepts. *Journal of the Chinese Institute of Engineers*, **38**(8), 1020–1025.
- Wei, Z., Zhao, K. and Wei, M. (2015). Decision-making in ship collision avoidance based on cat-swarm biological algorithm. *2015 International Conference on Computational Science and Engineering*, Advances in Computer Science Research, Atlantis Press.
- Zhang, D., Yan, X. P., Yang, Z. L. and Wall, A. (2013). Incorporation of formal safety assessment and bayesian network in navigational risk estimation of the Yangtze River. *Reliability Engineering & System Safety*, **118**(10), 93–105.