Journal of Marine Science and Technology

Volume 26 | Issue 6 Article 3

COLLISION AVOIDANCE PATH PLANNING FOR SHIPS BY PARTICLE SWARM OPTIMIZATION

Yu-Tao Kang
College of Ocean Science and Engineering, Shanghai Maritime University, Shanghai, China., ytkang@shmtu.edu.cn

Wei-Jiong Chen
College of Ocean Science and Engineering, Shanghai Maritime University, Shanghai, China.

Da-Qi Zhu
Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Shanghai, China

Jin-Hui Wang
College of Ocean Science and Engineering, Shanghai Maritime University, Shanghai, China

Qi-Miao Xie
College of Ocean Science and Engineering, Shanghai Maritime University, Shanghai, China

COLLISION AVOIDANCE PATH PLANNING FOR SHIPS BY PARTICLE SWARM OPTIMIZATION

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 71503166) and the Natural Science Foundation of Shanghai (Grant No. 16ZR1414600). This manuscript was edited by Wallace Academic Editing.

Journal of Marine Science and Technology, Vol. 26, No. 6, pp. 777-786 (2018)
DOI: 10.6119/JMST.201812_26(6).0003

COLLISION AVOIDANCE PATH PLANNING FOR SHIPS BY PARTICLE SWARM OPTIMIZATION

Yu-Tao Kang[1], Wei-Jiong Chen[1], Da-Qi Zhu[2], Jin-Hui Wang[1], and Qi-Miao Xie[1]

Key words: collision avoidance, path planning, particle swarm optimization, ship domain.

ABSTRACT

Ship collision avoidance is a key consideration in maritime systems. Collision avoidance maneuvers depend on navigators' experience and skill levels. Because both maritime traffic densities and average ship speeds are increasing, the times available for decision-making are decreasing, which elevates the risk of human errors in the collision avoidance process. To reduce the effect of human factors and efficiently prevent collisions between ships navigating in open water with effective visibility, a particle swarm optimization (PSO) algorithm can be used to plan ship paths. An improved ship domain dynamic model can assess collision risks in close-range encounters. Several marine traffic scenarios based on standard encounter types were simulated; the proposed PSO algorithm was tested in those scenarios. This paper discusses the compatibility and consistency of the algorithm outputs as well as the execution efficiency of the algorithm.

I. INTRODUCTION

Although numerous navigational aids are available on a ship's bridge, such as an Automatic Identification System (AIS) and an Automatic Radar Plotting Aid (ARPA), a ship's collision avoidance depends mainly on the navigator's reaction and judgment. Nevertheless, with the rapid development of maritime trade, increases in both traffic density and the average cruise speed of ships have shortened the time available for making decisions during the process of collision avoidance, leading to increases in ship collisions. According to statistical analysis, 80% of ship collision incidents at sea are due to human factors. Technological enhancements, such as automated ship collision avoidance, reduce human errors because they reduce human participation. To reduce human errors, collision avoidance maneuvers have been actively researched; numerous scholars and experts have developed automated decision-making systems for ship collision avoidance. These experts have designed maneuvers and systems to assist a navigator in evaluating the danger of collisions and then generating particular maneuvers. However, path-planning functionality is limited because scant research has been published on optimal navigation paths (Tam et al., 2009).

Many studies have investigated path planning for ship collision avoidance in the past thirty years. At first, researchers mainly adopted deterministic approaches in the area of ship path planning, such as knowledge-based expert systems (Iijima and Hagiwara, 1991), analytical geometry with convex set theory (Hong et al., 1999), fuzzy set theory (Hwang et al., 2001), maze routing methods (Chang et al., 2003; Szlapcynski, 2006a), and neural networks (Liu and Shi, 2005). Path planning for collision avoidance is a multi-objective nonlinear optimization problem in a complex and dynamic environment. Path planning must balance navigational safety and economic constraints. Hence, it is unrealistic to employ a deterministic approach to solve such a problem in a real-time environment. And then, to solve the aforementioned problem, paths for collision avoidance have been supplied by heuristic approaches, such as evolutionary algorithms (Tam and Bucknall, 2010; Ming, 2016), genetic algorithms (Zeng, 2003; Cheng and Liu, 2006), ant colony algorithms (Tsou and Hsueh, 2010), and particle swarm optimization (PSO) algorithms (Chen and Huang, 2012). An optimal turn or local optimal path can be generated through the aforementioned heuristic approaches to prevent collision with other ships for an immediate encounter. However, it is difficult to calculate an optimal trajectory in a large-scale traffic scenario. Most studies have determined a navigation path without consideration of the environmental conditions and without conforming to International Regulations for Preventing Collisions at Sea (COLREGS).

This paper adopts the concept of ship domain as an assessment criterion for collision risk and uses the PSO algorithm to create a path-planning approach to optimize collision-free paths for ships in real-time navigation environments. This study is an attempt to obtain the optimal navigation path holistically by considering relevant environmental conditions and conforming to COLREGS. In this paper, we discuss the performance of the algorithm and the consistency of its results. The remainder of this paper is organized as follows. Section 2 explains the concept and method of assessing the risk of collision. Section 3 de-

Paper submitted 10/27/17; revised 03/23/18; accepted 10/22/18. Author for correspondence: Yu-Tao Kang (e-mail: ytkang@shmtu.edu.cn).
[1] College of Ocean Science and Engineering, Shanghai Maritime University, Shanghai, China.
[2] Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Shanghai, China.

scribes the proposed PSO-based path-planning algorithm. Traffic scenario simulation results are discussed in Section 4. This study is summarized in Section 5.

II. COLLISION RISK ASSESSMENT FOR SHIPS

In open waters, the distance between ships is much greater than the size of ships. Thus, all ships can be simulated as moving points in this context. The ship under direct control is denoted as the "own ship" (OS); any ship other than the OS is denoted as a "target ship" (TS). An AIS can provide static information (e.g., name, length, draft, and call sign) and accurate real-time dynamic information (e.g., course, speed, position, and relative distance) about the ship in real-time navigation. An ARPA can supply maneuvering information, closest point of approach (CPA), time to the closest point of approach, and other navigational information. Therefore, this study assumes that all ships in the simulations can obtain real-time collision avoidance information. Assessment of collision risk has two steps: the first step is to determine the type of encounter between the OS and the TS; the second step is to calculate the dimensions of the ship domain around the TS as necessary.

1. Classification of the Encounter Situation

COLREGs regarding encounters in open water with effective visibility were analyzed in terms of navigational practices; the encounter situations covered by COLREGS can be divided into three types, of which each type has its own constitutive requirements. The encounter situations are classified as follows:

(1) Overtaking: A ship shall be deemed to be overtaking when coming up on another ship from a direction more than 22.5° abaft its beam. By rule 13 of COLREGS, an overtaking encounter between two ships must meet the following three conditions:
   a. The overtaking ship is located in any direction more than 22.5° abaft the beam of the front ship.
   b. The overtaking ship is located within the visibility range of the stern light of the front ship (i.e., relative distance between two ships < 3 nm).
   c. The speed of the overtaking ship is higher than that of the front ship.

   The region used to determine the overtaking encounter is region C, as illustrated in Fig. 1. If the OS is overtaking the TS, the OS must stay out of the path of the TS.

(2) Head-on: Two ships shall be deemed to be in head-on encounter when they are meeting on reciprocal or nearly reciprocal courses in a manner that involves some risk of collision. By rule 14 of COLREGS, a head-on encounter between two ships must meet the following two conditions:
   a. Two ships are meeting on reciprocal or nearly reciprocal courses.
   b. These two ships are at risk of collision (i.e., relative distance between two ships < 6 nm, DCPA (Distance to the Closest Point of Approach) ≤ 0.5 nm, and each is within approximately 6° of the other).

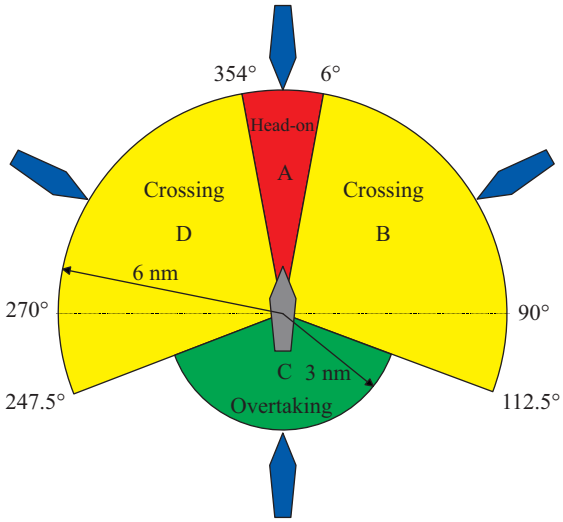

Fig. 1. Regions used to classify encounter situations.

The region used to determine the head-on encounter is region A, as shown in Fig. 1. If two ships are in a head-on encounter, each shall alter its course to starboard so that each shall pass on the port side of the other.

(3) Crossing: Two ships shall be deemed to be crossing when their paths are crossing in a manner that involves some risk of collision. By rule 15 of COLREGS, a crossing encounter between two ships must meet the following two conditions:
   a. The paths of two ships are crossing.
   b. These two ships are at risk of collision (i.e., relative distance between two ships < 6 nm and DCPA ≤ 0.5 nm).

   The region used to determine the crossing encounter is region B or D, as shown in Fig. 1. If two ships are in a crossing encounter, the ship that has the other on her starboard side must stay out of the path and shall, if the circumstances of the case admit, avoid crossing ahead of the other ship.

2. Modeling of the Ship Domain

Earlier research introduced the concept of CPA based on the principle of geometric collision avoidance to assess the risk of collision for ships. To directly demonstrate collision risks on environmental maps, most path-planning algorithms define some ship domain around an obstacle to indicate the risk of collision. The concept of a ship domain was first presented by Fujii and Kenichi, who proposed an ellipsoidal ship domain with the OS at the center in a manner suitable for restricted waters (Tam et al., 2009). Since this, numerous studies have proposed ship domains with different shapes and dimensions based on approaches such as statistical analysis (Pietrzykowski and Uriasz, 2009), functional analytic methods (Szlapcynski, 2006b; Wang et al., 2009), and artificial intelligence methods (Pietrzykowski, 2008; Wang, 2010). A ship domain can be influenced by many factors (e.g., ship type, length, course, speed, maneuverability, encounter type, and marine environment conditions), most of which change dynamically in real-time marine environments. Hence, it is unrealistic to use a constant ship domain to assess

Table 1. Shapes of the ship domain for different encounter types.

Shape	Condition
Circular	Encounter type = Overtaking and $V_{TS} \le V_{OS}$
Nothing	Encounter type = Overtaking and $V_{TS} > V_{OS}$
Half-elliptical	Encounter type = Head-on
Half-elliptical	Encounter type = Crossing
Circular	Static obstacle
Nothing	Safety

collision risks and planned paths in real-time navigation. Therefore, Tam and Bucknall (2010) proposed a dynamic ship domain around a TS that varies with ship type, speed, encounter type, marine traffic environment, and other parameters.

This study develops the Tam and Bucknall's ship domain to assess the collision risk. First, according to the type of encounter as well as the relative speed of the OS and the obstacle of concern, the shape of the ship domain is determined, as listed in Table 1. Then, the dimensions of the ship domain are calculated from the speed of the TS, the minimum safe distance between the OS and TS, and other factors related to the ship and environment conditions.

For an overtaking encounter between the OS and a TS (the speed of the TS is less than or equal to the speed of the OS) such that the state of the TS is static, the ship domain around the TS is circular. R_C is the radius of the circular safety domain, and is computed as follows:

$$R_C = \begin{cases} V_{TS} \times S_{OT} & if \quad V_{TS} \times S_{OT} > D_{\min}, \\ D_{\min} & otherwise; \end{cases} \tag{1}$$

where V_{TS} (0 when the TS is static) is the velocity vector of the TS; S_{OT} (1.0 min in the general case) is the scaling factor of the safety domain for the overtaking encounter, which is introduced to customize the dimensions of the safety domain; and $D_{\min}$ is the minimum safe distance that must be maintained between the OS and the TS (0.25 nm based on the distance covered by the TS travelling at 30 kn in 30 s). Because the speeds of most ships are not more than 30 kn, the 30-s time interval is considered sufficient for most collision avoidance maneuvers.

For head-on encounters and crossing encounters, the safety domain is half-elliptical; it is elliptical at the fore section and circular at the aft section of the TS. The dimensions of this half-elliptical area are determined by the semimajor axis and semiminor axis of the ellipse. A_E is the semimajor axis of the halfelliptical area, which can be computed as follows:

$$A_E = \begin{cases} V_{TS} \times S_{EA} + S_{SI} & if \quad V_{TS} \times S_{EA} + S_{SI} > D_{\min}, \\ D_{\min} & otherwise; \end{cases} \tag{2}$$

where S_{EA} (1.0 min) is the scaling factor of the semimajor axis, which depends on the type of encounter and S_{SI} (0.15 nm in this algorithm) is the space interval variable of the path-planning algorithm, which is half of the distance between the adjacent waypoints on the Y axis in the environmental map of the PSO algorithm. Thus, a large space interval provides a slightly enlarged safety area to offset the space delay. B_E is the semiminor axis of the half-elliptical area, as well as the radius of the semicircular area, which is calculated as follows:

$$B_E = \begin{cases} R_B + S_{SI} & if \quad R_B + S_{SI} > D_{\min}, \\ D_{\min} & otherwise; \end{cases} \tag{3}$$

where R_B is the safety area's aft-section radius without considering the space interval, which is computed as follows:

$$R_B = \begin{cases} V_{TS} \cdot S_{EB} & if \quad V_{TS} \cdot S_{EB} < D_{\max}, \\ 2R_{\max} - V_{TS} \cdot S_{EB} & otherwise; \end{cases} \tag{4}$$

where S_{EB} (1.0 min) is the scaling factor of the semiminor axis, which is similar to S_{EA}, and $R_{\max}$ is the maximum allowable radius that limits the range of the safety area on the side and stern sections, which is defined as 0.5 nm and depends on the maneuverability of the TS. At low V_{TS}, the output of R_B function increases with speed. Due to the high maneuverability and the low inertia at any low speed of the ship, the TS can easily turn in any direction. For high V_{TS}, the value of R_B decreases with increases in speed. This can account for the weak turning ability of the TS to the side and aft directions at high speeds.

Although different ships have different characteristics and maneuverability, for simplicity, all ships were assumed to have the same dynamic properties in this study. Therefore the values of the parameters used in this paper were based on educated guesses for the performance of a typical 10-t ship.

3. Simulation of the Ship Domain

According to the aforementioned analysis, ship domains for three types of encounters can be calculated using the proposed method. For an overtaking encounter, circular safety domains can be determined using MATLAB, as shown in Fig. 2. For low V_{TS} ($\le$ 0.25 nm/min), the radius of the circular safety domain is equal to the minimum safe distance. However, when $V_{TS} >$ 0.25 nm/min, the radius of the circular safety domain grows linearly with V_{TS}.

For the head-on and crossing encounters, half-elliptical safety domains are shown in Fig. 3. For low V_{TS}, A_E and B_E increase with V_{TS}; thus, the safety domain is circular. For $V_{TS} >$ 0.5 nm/min, A_E increases and B_E decreases until reaching $D_{\min}$. Therefore, the fore section of the safety domain becomes elliptical, the aft section remains circular and decreases with the radius. For high V_{TS}, A_F continues to increase according to the magnitude of V_{TS}, but B_E remains at $D_{\min}$. The minor axis of the safety domain has a radius similar to that of the aft section. Thus, the safety domain always has a continuous boundary.

The aforementioned ship domain was inspired by the work of

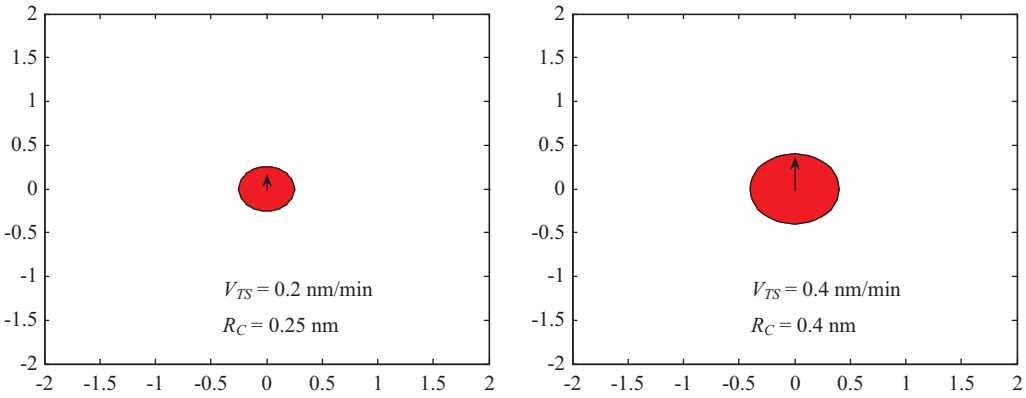

Fig. 2. Simulation of circular safety domains.

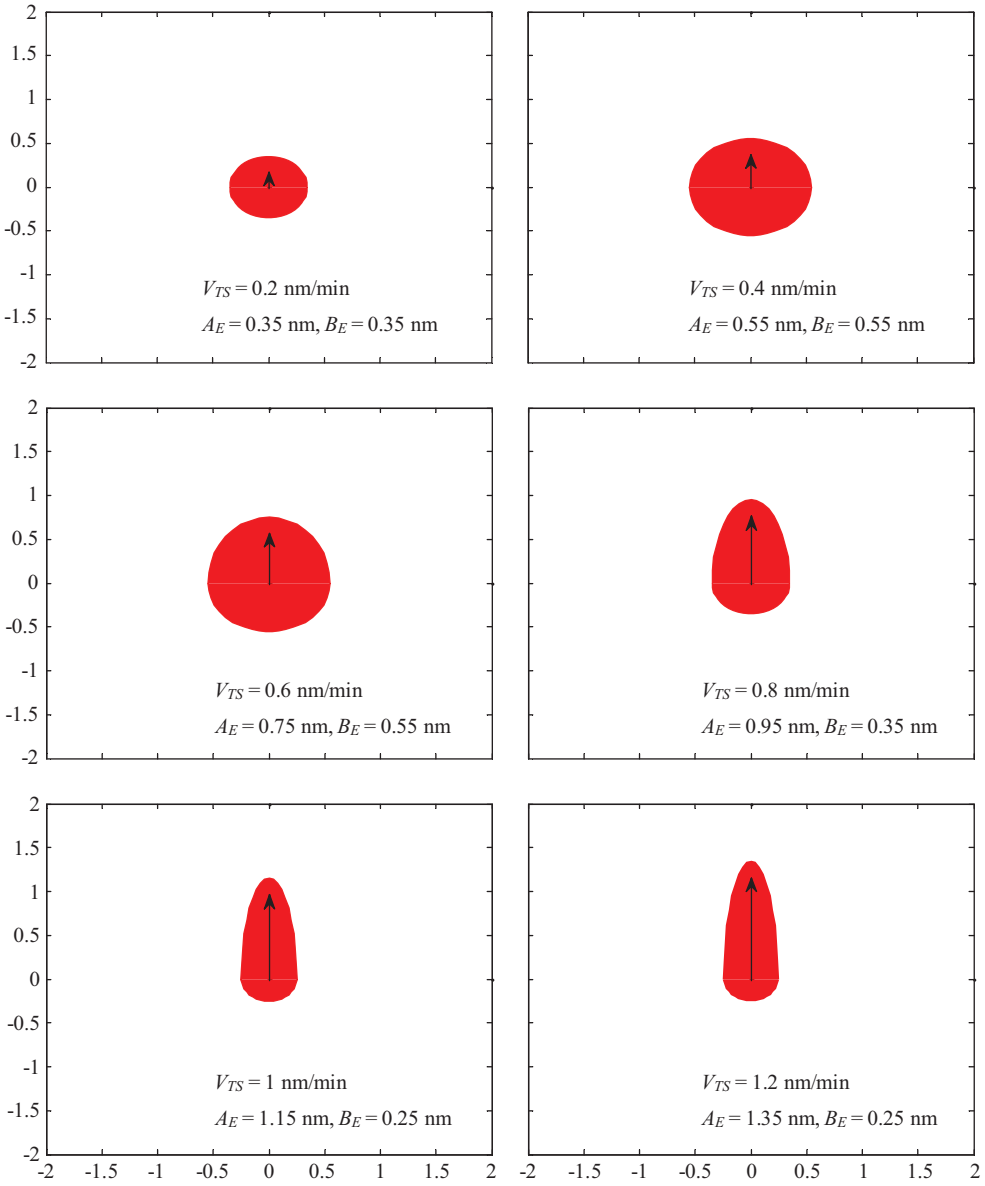

Fig. 3. Simulation of half-elliptical safety domains.

Tam and Bucknalls. For the PSO algorithm, some parameters (i.e., S_{SI} and R_{max}) have been altered so that the dimensions of the safety domains are slightly different from those proposed by Tam and Bucknalls. All the parameters can be altered to fit with the changes in ship characteristics, environmental conditions, or human factors.

III. PATH-PLANNING ALGORITHM FOR SHIPS

A two-dimensional space was adopted in this study, in which the real-time data for the OS and TS were defined. The navigation path can be discretized into a number of linear segments by division into equal segments from the start to the destination. At each segment, the OS must provide a safety domain that is defined in the aforementioned ship domain model around the TS. The velocity vector of the TS is assumed to be constant due to the instantaneous navigation path in this study. By rule 8 of COLREGS, substantial alteration of course alone may be the most effective action for sufficient sea-room to avoid a close-quarters situation provided that it is made in a short time, and does not result in another close-quarters situation. Thus, the speed of the OS was assumed to be constant. The course of the OS can be altered to avoid a collision with the TS.

1. Building the Environmental Map

Although collision risk is not explicitly defined in COLREGS, it is generally considered that no risk of collision exists if two ships are separated by 6 nm. Therefore, the domain of interest in this paper is limited to the area within a 6-nm radius of the OS. The environmental map is constructed with the initial point of the OS as the origin in the Y-axis direction, according to initial positions and velocity vectors of the OS and TS, and the destination of the OS, as shown in Fig. 4. The task of path planning is to search for a set of waypoints in the environmental map to obtain the shortest path, which enables the adjacent points and their connecting lines to avoid the TS and its safety domain. P is a set of waypoints, defined as follows:

$$P = \{S, p_1, p_2, \ldots, p_D, F\} \tag{5}$$

where S is the start point of the OS, F is the target point of the OS, and p_n is a waypoint in the path. The line SF is equally divided into $D + 1$ segments, where the perpendicular of the line SF is constructed through each equal division point. Random points (i.e., p_1 to p_2) that are selected in order on perpendiculars of the line SF compose the set P together with S and F. Due to the equidistance between waypoints on the vertical axis, the path can be uniquely determined by the vector X consisting of the horizontal coordinates of the waypoints, defined as follows:

$$X = \{x_S, x_1, x_2, \ldots x_D, x_F\} \tag{6}$$

2. Description of the Particle Swarm Optimization Algorithm

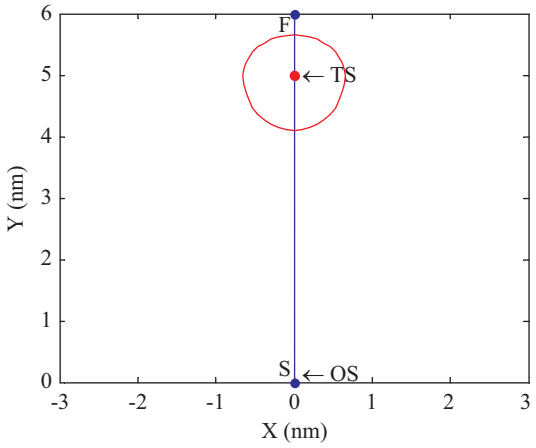

Fig. 4. Environment map.

PSO is a population-based stochastic optimization algorithm proposed by Kennedy and Eberhart in 1995 that was inspired by the social behavior of birds flocking or fish schooling. Compared with genetic algorithms, PSO is easier to implement and requires fewer parameter adjustments. When dealing with an optimization problem by using PSO, each potential solution, called a "particle," flies in the problem space looking for its optimal position (similar to the process by which a flock of birds searches for food). As time passes, each particle adjusts its position according to its own experiences as well as the experiences of neighboring particles.

In this paper, it is assumed that the dimension of the search space is D, and the number of the particles is n. The PSO system is initialized with a population of random particles, where the vector $X_i = \{x_{i1}, x_{i2}, \ldots, x_{iD}\}$ represents the position of the i^{th} particle. Moreover, each particle is assigned a randomized velocity with vector $V_i = \{v_{i1}, v_{i2}, \ldots, v_{iD}\}$. Each particle is evaluated according to its fitness, which is explained in detail later. By comparing the fitness values, the best solution for each particle is denoted p_{best}, and the best solution for the whole particle swarm is denoted g_{best}; $pb_i = \{p_{i1}, p_{i2}, \ldots, p_{iD}\}$ is the position of the best solution that has been achieved so far for the i^{th} particle. The position of the overall best solution obtained so far for the particle swarm is represented as $gb = \{g_1, g_2, \ldots, g_D\}$. At each time step, each particle updates its position according to the following formulas:

$$\begin{aligned} v_{id}(t+1) = {} & w v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) \\ & + c_2 r_2 (g_i(t) - x_{id}(t)) \end{aligned} \tag{7}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), 1 \le i \le n \quad 1 \le d \le D \tag{8}$$

where c_1 and c_2 are the acceleration constants that pull each particle toward p_b and g_b, respectively. Because of prior experience with this implementation of PSO, we set the acceleration constants equal to 2.0 for most applications. r_1 and r_2 are random functions with the range [0,1]. w is the inertia weight specified

by the user and can control the effect of the previous value of particle velocity on its current one. A large inertia weight compels particles to search new areas (global searching) whereas a smaller inertia weight compels particles to search the current area (local searching). In this paper, the inertia weight is defined as follows:

$$w = 1 - \frac{g_c}{g_{\max}} \tag{9}$$

where g_c is the current iterative time and $g_{\max}$ is the maximum iterative time.

To prevent the particles from moving beyond the problem space, the range of the particle position on each dimension is set to $[X_{\min}, X_{\max}]$, and the range of the particle velocity on each dimension is set to $[-V_{\max}, V_{\max}]$. If the position or velocity calculated using the aforementioned equations exceeds the range on that dimension, the position or the velocity on that dimension is limited to the boundary value.

3. Realization of PSO for Path Planning

Assume that the path $P = \{S, p_1, p_2, \ldots, p_D, F\}$ represents a solution for path planning, where the midpoints $p_i (i = 1, 2, \ldots, D)$ compose a particle. Then, $x_i (i = 1, 2, \ldots, D)$ is the position of the particle in the i^{th} dimension. The path planning task is to shorten the length of the path to obtain the optimized path and to avoid any potential collisions with other ships or obstacles. Therefore, the fitness function of each particle is defined as follows:

$$l = \sqrt{(x_1 - x_S)^2 + \left(\frac{y_F - y_S}{D+1}\right)^2} + \sum_{i=1}^{D-1} \sqrt{(x_{i+1} - x_i)^2 + \left(\frac{y_F - y_S}{D+1}\right)^2} \tag{10}$$

$$f = \frac{D + 2 - D_{null}}{l} \tag{11}$$

where l is the length of the path; (x_S, y_S) and (x_F, y_F) represent the coordinates of the start point and the goal point, respectively; f is the fitness of the particle; and D_{null} represents the number of invalid path segments (i.e., where the OS collides with or is in the safety domain of the TS). The higher the fitness value is, the better the solution is.

The implementation process of the PSO algorithm is described as follows:

(1) Initialize a population of particles with random positions and velocities on D dimensions in the problem space. Each particle's historic optimal position p_{best} is its initial position. Calculate the fitness value of each particle according to the aforementioned equations, and label the particle with the maximum fitness value as g_{best}.
(2) Update the velocity and position of each particle according to Eqs. (7) and (8).
(3) Calculate the fitness value of each particle according to Eqs. (10) and (11), as shown in Fig. 5.

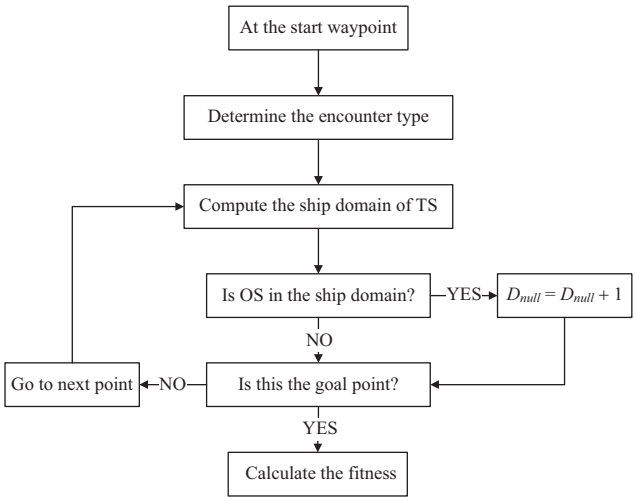

Fig. 5. Calculation process for the fitness of each particle.

(4) Compare the particle's fitness value with that of p_{best}. If its current fitness value is better than that of p_{best}, then set p_{best}'s fitness value to the current fitness value, and set p_{best}'s location to the current location in the D-dimensional space.
(5) Compare the fitness value with the value of g_{best}. If the current fitness value is better than g_{best}, then reset g_{best} to the current particle's value.
(6) Repeat step (2) until the user-defined stopping criterion has been met.

IV. SIMULATION

1. Traffic Scenarios for Simulation

The set of traffic scenarios was based on real-world incidents of two ships meeting, such that a single obstacle collided with the OS from various directions. According to the classification of the encounters and the real-time marine traffic environment, four traffic scenarios were designed to test the path-planning algorithm as listed in Table 2.

All the traffic scenarios were set up with a convergent bearing, such that the OS would collide with the TS if the OS were not to change its course. The position and velocity of the TS were measured relative to the initial position of the OS, which was set to be at the point (0,0) in the coordinate system. Scenario 1 simulated a head-on encounter, where the velocities of the OS and TS were exactly opposite. Scenario 2 simulated an overtaking encounter with the same courses and different speeds between the OS and TS. Scenario 3 simulated a crossing encounter with different courses and different speeds. Scenario 4 tested a collision between the OS and a stationary ship or static obstacle. Fig. 6 illustrates the initial states of the OS and TS for all traffic scenarios.

2. Simulation Results

The traffic scenarios were simulated in MATLAB run on an Intel core i7 processor at 3.40 GHz (8 cores) with 4 GB of RAM

Table 2. Traffic scenarios.

Scenario	OS		TS		Encounter type
	Initial position (nm)	velocity (nm/min)	Initial position (nm)	velocity (nm/min)	
1	[0,0]	[0,0.5]	[0,6]	[0,-0.5]	Head-on
2	[0,0]	[0,0.6]	[0,2]	[0,0.2]	Overtaking
3	[0,0]	[0,0.5]	[3,3]	[-0.5,0]	Crossing
4	[0,0]	[0,0.5]	[0,3]	[0,0]	Static

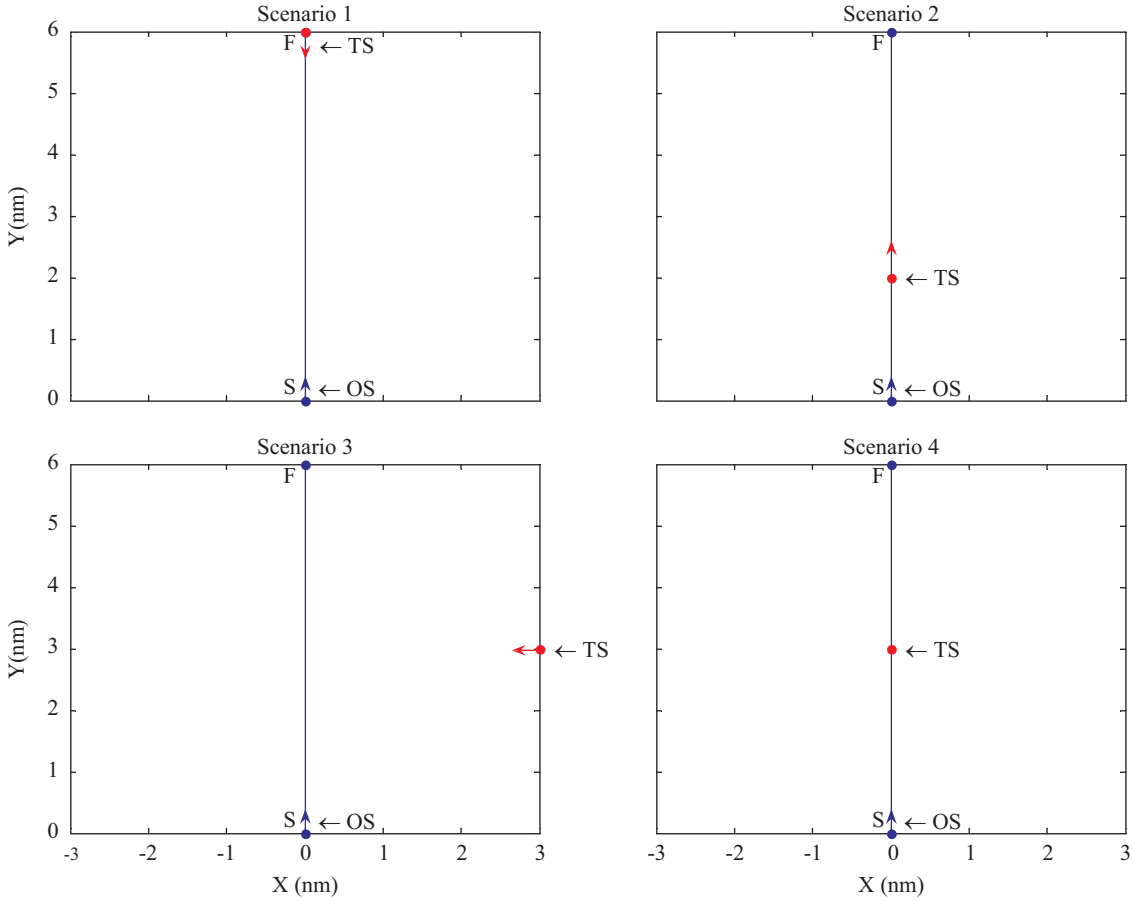

Fig. 6. Traffic scenarios.

using Windows 7. Parameters of the PSO were set as follows:

$$n = 20,\ D = 19,\ c_1 = c_2 =,\ g_{\max} = 1000$$

The simulation results of the PSO algorithm for these four traffic scenarios are presented in Fig. 7 and Table 3. To verify the output compatibility of this algorithm with the optimal path of the OS and the path of the TS, the roles of the OS and TS in scenarios 1-3 can be reversed.

Scenario 1 was constructed to test the head-on encounter, in which the OS and TS have the reverse roles under the same traffic scenario. Therefore, the path of TS can be obtained by rotating the path of the OS under the TS scenario. Both ships performed maneuvers that complied with COLREGS rule 14 (i.e., both passed port to port for a head-on encounter).

In scenario 2, the OS was overtaking the TS from the stern. Only the OS was maneuvering according to COLREGS rule 13, namely that the overtaking party should stay out of the path of the vessel being overtaken. Rule 13 does not explicitly specify on which side the ship should overtake. Hence, both starboard and larboard maneuvers can be allowed in the algorithm. However, the TS always maintains its course, to avoid confusion with the overtaking party.

Scenario 3 was constructed to evaluate the crossing encounter, and the algorithm conforms to COLREGS rule 15 for both the OS and TS. The combined paths of the two ships are shown in the results of scenario 3. COLREGS rule 15 states that any OS that has a ship approaching from its starboard side should

Table 3. Simulation results.

Scenario	Encounter type	The optimal path length (nm)	DCPA (nm)	execution time (sec)
1	Head-on	6.1394	1.2924	18.05
2	Overtaking	6.0210	0.2500	16.98
3	Crossing	6.2065	0.6500	18.13
4	Static	6.0209	0.2500	14.89

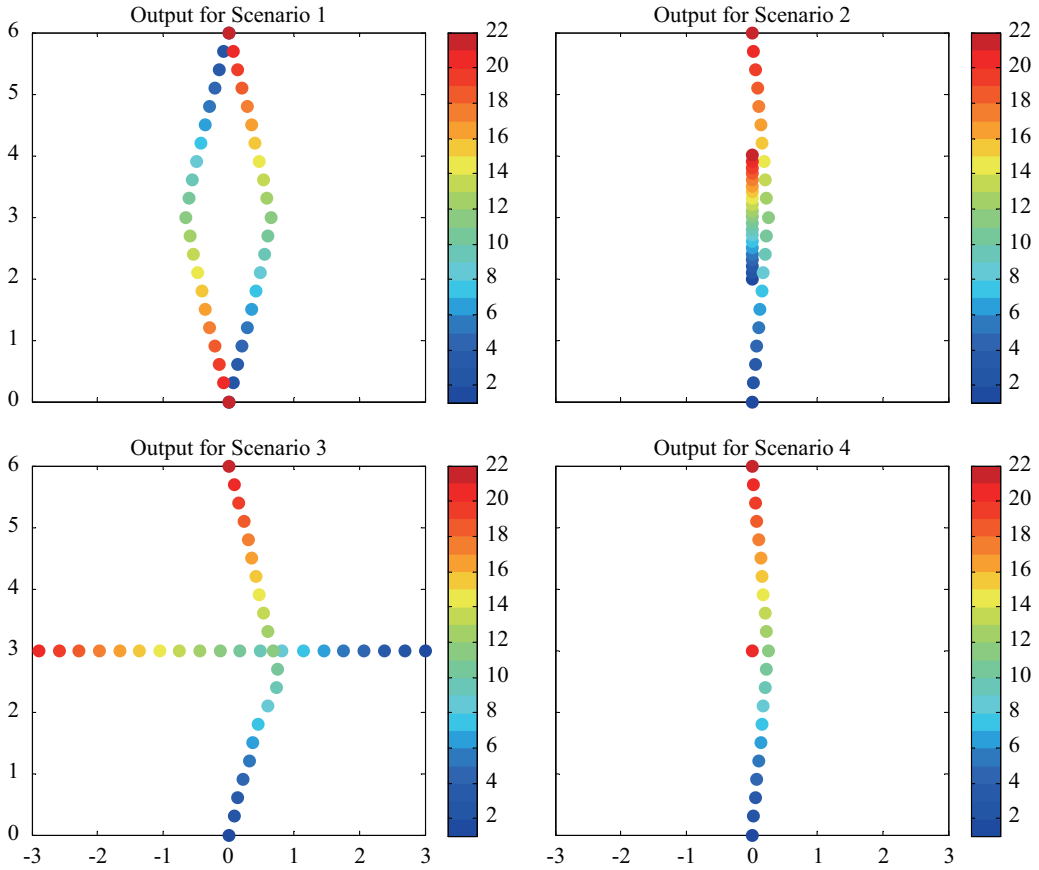

Fig. 7. Algorithm outputs for traffic scenarios.

maneuver and avoid passing ahead of the other party, when the other party (TS) is maintaining its course.

In scenario 4, the TS is in a static state as a obstacle. Thus, only the OS has an optimal path in the output for scenario 4.

The relative distances between the OS and TS for four traffic scenarios are depicted in Fig. 8. In general, the algorithm can generate the optimal path, on which the OS can stay a safe distance away from the TS. Due to the high relative velocity, scenario 1 (the head-on encounter) had a high minimum relative distance, to ensure enough space for collision avoidance. The smallest relative distance occurred in scenarios 2 and 4 (the overtaking encounter and static state) because the relative velocity was lower in those scenarios.

3. Discussion

The proposed PSO-based path-planning algorithm produced satisfactory results for dynamic and static obstacles; those results were consistent with COLREGS. In addition, considering the algorithm output compatibility from other perspectives, the output for the dynamic obstacle was produced by reversing the roles of the OS and TS under the same traffic scenario; the algorithm is suitable for both centrally managed and independently executed navigation systems.

The algorithm outputs illustrated in Fig. 7 were selected from recorded outputs for each traffic scenario; the algorithm consistently performed similar maneuvers around the same location. However, the PSO algorithm was difficult to control in terms of its output consistency due to the lack of any restricted condition in its fitness function. Moreover, the ship domain that evaluates the encounter type and collision was only used to determine whether the OS was on a collision course or supposed to yield. Therefore, this algorithm lacks consistent guidance and

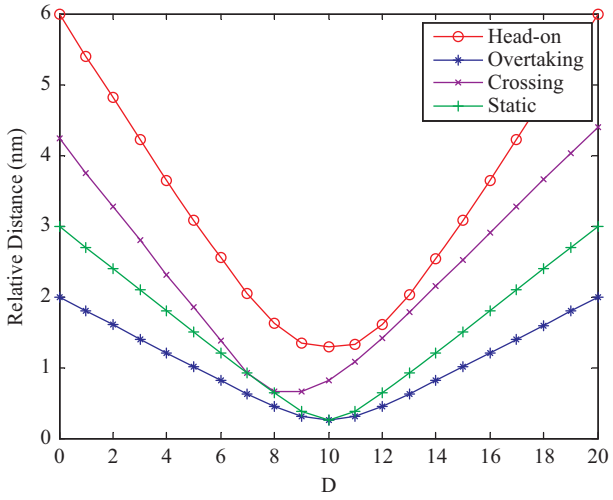

Fig. 8. Relative distance between ships in all scenarios.

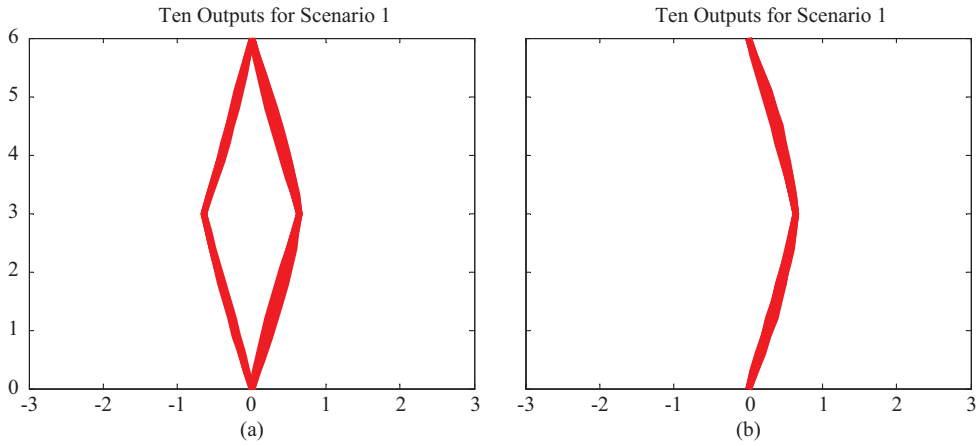

Fig. 9. Ten repeated outputs for scenario 1.

the OS can either perform a starboard or larboard maneuver to avoid a collision. Ten outputs for scenario 1 are shown in Fig. 9(a), where six navigation paths are repeated on the starboard side and four paths are on the larboard side. Thus, the range of the particle position was reduced to limit the search space on the starboard side of TS, such that the algorithm was forced to output an appropriate maneuver according to the COLREGS. This improved the consistency of the algorithm. Scenario 1 was simulated ten times with this method. The algorithm outputs are presented in Fig. 9(b), where all runs of the algorithm performed a starboard maneuver to yield on the starboard side.

With the regard to computational efficiency, MATLAB R2014a was the environment that executed the routines ten times for each scenario; the average computational times for path planning for each scenario are presented in Fig. 10. Compared with the path planning for other scenarios, the path planning for scenario 3 required more time under the same conditions, because it used a higher number of iterations in search routines. However, most runs of path planning for each scenario returned the optimal navigation path within a limited number of iterations.

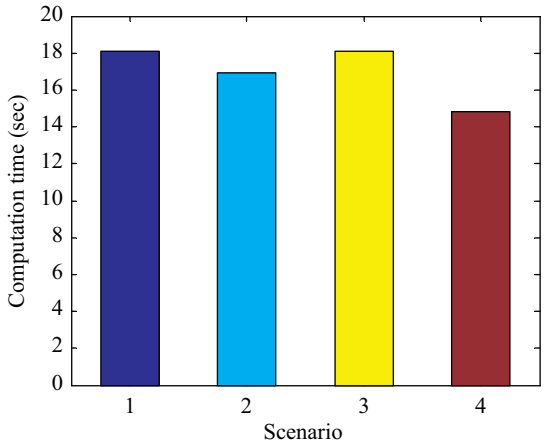

Fig. 10. Average computational time for path planning of each scenario.

V. CONCLUSION

In this paper, a novel method of fractional steps is proposed to solve the collision avoidance problem for surface ships in open water. This method considers many crucial aspects in real-time navigation that have been neglected by published studies, such as navigation path optimization in a general traffic scenario, environmental conditions, and conforming to COLREGS. The design of this method is based on PSO, and is divided into two steps: First, the ship domain model is formalized to assess the collision risk between the OS and TS; next, the PSO algorithm is adopted to obtain the shortest path from the start point to the goal point in the environmental map. Simulation results have shown that the algorithm is capable of consistently obtaining an optimized, collision-free, COLREGS-compliant, and practical navigation path for all simulated traffic scenarios.

Because ship collision avoidance requires the OS to actively maneuver but also requires the TS to execute the corresponding maneuver, the compatibility and consistency of the algorithm outputs were tested and were proven to be adequate. In addition, the computational efficiency was evaluated and verified as satisfactory. Furthermore, through parallel computation or application of high-performance computers, the computational time for path planning can be further reduced. Therefore, the proposed path-planning algorithm enables the planning of real-time navigation paths. Path planning involving multiple obstacles is not addressed in this paper, but it will be studied in future work.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 71503166) and the Natural Science Foundation of Shanghai (Grant No. 16ZR1414600). This manuscript was edited by Wallace Academic Editing.

REFERENCES

Chang, K. Y., G. E. Jan and I. Parberry (2003). A method for searching opti-mal routes with collision avoidance on raster charts. Journal of Navigation 56, 371-384.

Chen, L. J. and L. W. Huang (2012). Ship collision avoidance path planning by PSO based on maneuvering equation. Lecture Notes in Electrical Engineering 144, 675-682.

Cheng, X. D. and Z. Y. Liu (2006). Study on Ship Collision Avoidance Trajectory Optimization in Inland Waterways with Genetic Algorithm. Navigation of China 67, 38-46.

Hong, X., C. J. Harris and P. A. Wilson(1999). Autonomous ship collision free trajectory navigation and control algorithms. Proceeding of IEEE International Conference on Emerging Technologies and Factory Automation 2, 923-929.

Hwang, C. N., J. M. Yang and C. Y. Chiang (2001). The design of fuzzy collision avoidance expert system implemented by h1-autopilot. Journal of Marine Science and Technology 9, 25-37.

Iijima, Y. and H. Hagiwara (1991). Results of collision avoidance manouvre experiments using a knowledge-based autonomous piloting system. Journal of Navigation 44, 194-204.

Liu, Y. H. and C. J. Shi (2005). A fuzzy-neural inference network for ship collision avoidance. Proceeding of International Conference on Machine Learning and Cybernetics 8, 4754-4759.

Ming, C. T. (2016). Multi-target collision avoidance route planning under an ECDIS framework. Ocean Engineering 121, 268-278.

Pietrzykowski, Z. (2008). Ship's fuzzy domain-a criterion for navigational safety in narrow fairways. Journal of Navigation 61, 499-514.

Pietrzykowski, Z. and J. Uriasz (2009). The ship domain-a criterion of navigational safety assessment in an open sea area. Journal of Navigation 62, 93-108.

Szlapcynski, R. (2006a). A new method of ship routing on raster grids, with turn penalties and collision avoidance. Journal of Navigation 59, 27-42.

Szlapcynski, R. (2006b). A unified measure of collision risk derived from the concept of a ship domain. Journal of Navigation 59, 477-490.

Tam, C. K., R. Bucknall and A. Greig (2009). Review of collision avoidance and path planning methods for ships in close range encounters. Journal of Navigation 62, 455-476.

Tam, C. K. and R. Bucknall (2010). Collision risk assessment for ships. Journal of Marine Science and Technology 15, 257-270.

Tam, C. K. and R. Bucknall (2010). Path-planning algorithm for ships in close-range encounters. Journal of Marine Science and Technology 15, 395-407.

Tsou, M. C. and C. K. Hsueh (2010). The study of ship collision avoidance route planning by ant colony algorithm. Journal of Marine Science and Technology 18, 746-756.

Wang, N., X. Meng, Q. Xu and Z. Wang (2009). A unified analytical framework for ship domains. Journal of Navigation 62, 643-655.

Wang, N. (2010). An intelligent spatial collision risk based on the quaternion ship domain. Journal of Navigation 63, 733-749.

Zeng, X. M. (2003). Evolution of the safe path for ship navigation. Applied artificial intelligence 17, 87-104.