



Research on algorithms for autonomous navigation of ships

Agnieszka Lazarowska¹

Received: 14 December 2018 / Accepted: 11 June 2019 / Published online: 10 July 2019
© The Author(s) 2019

Abstract

The paper reports the most recent results of a research project dedicated to the development of a path planning module, constituting a part of an intelligent control system for ships—a Guidance, Navigation, and Control (GNC) system. The paper describes the assumptions, carried out research tasks, and preliminary simulation and real experiments results. The main contribution is the development and validation of new algorithms for path planning and collision avoidance of ships. Two algorithms are introduced and tested—one using a heuristic approach and the other one applying a deterministic method. The input data to the algorithms are real navigational data registered from the navigational equipment of a ship *Horzont II*. Simulations were performed in the MATLAB environment, while for real experiments a group of mobile platforms and an Indoor Positioning System were used. The outcome of the research will affect the development of new solutions in the area of autonomous navigation of ships, what will contribute to the achievement of safer and more efficient shipping.

Keywords Autonomous navigation · Collision avoidance · Decision support · Safe trajectory

1 Introduction

Safety of navigation is one of the most important issues in maritime transport. According to Allianz Global Corporate & Specialty SE (2018), about 90% of global trade is transported by ships, while in the years 2008–2017 collisions caused 57 total losses of a ship and 34 casualties. The possibility of the collisions reduction is perceived in the development of unmanned ships technology. The technology

✉ Agnieszka Lazarowska
a.lazarowska@we.umg.edu.pl

¹ Department of Ship Automation, Gdynia Maritime University, 81-87 Morska St. 81-225 Gdynia, Poland

is developing, Kongsberg Gruppen introduced the concept of a full-electric 120 teu autonomous ship called Yara Birkeland (Kongsberg Maritime AS 2018), while in China the Wanshan Marine Test Field, a testing site for unmanned ships, is being constructed (CCS 2018).

In order to outline the background and significance of the ship's trajectory planning methods presented in this paper, a review of the recent literature dedicated to path planning and collision avoidance of a ship has been carried out.

In recent years (2010–2018), many new approaches have been introduced. The proposed methods can generally be classified into one of the two groups: deterministic or stochastic approaches.

The classical representative of the stochastic methods is the evolutionary algorithm (EA), which became very popular in application to ship's path planning. Recent approaches utilizing this algorithm were introduced, e.g., by Tam and Bucknall (2010) and Szlapczynski and Szlapczynska (2012). The main limitations of an algorithm based upon evolutionary computations might be their relatively long computational time (even hundreds of seconds) and problems with repeatability of solution for the same input data. Other methods classified to stochastic group are the swarm-based approaches such as the Particle Swarm Optimization (PSO) (Chen and Huang 2012).

One of the most promising and very popular deterministic optimization method used for path planning is the potential field approach, utilized in Montiel et al. (2015) (Bacterial Potential Field - BPF), Naeem et al. (2016) (Artificial Potential Field - APF) and Xue et al. (2011) (Potential Field method - PFM). Another proposal is an application of the fast marching method (FMM) (Song et al. 2017) and Liu and Bucknall (2015). A different deterministic approach was proposed by Tam and Bucknall (2013) (Cooperative path planning algorithm - CPP).

Another very popular subgroup constitute the graph-search algorithms, presented in Candeloro et al. (2017) (Voronoi diagram - VD), Naeem et al. (2012) (A*), and Lee et al. (2015) (Energy efficient A* - EEA*).

Other recent approaches include application of artificial neural networks (ANN) (Simsir et al. 2014), fuzzy logic (FL) (Mohamed-Seghir 2014), and differential games (DG) (Lisowski 2016). All of the above-mentioned most recent proposals are compared in Table 1.

The analysis of these approaches leads to the conclusion, that the development of an effective path planning algorithm for dynamic environments, applicable in near-real time systems, constitutes an open research problem. All of the above-mentioned approaches have some limitations and there still exists a space for some improvement with regard to run time, optimality of solution, and constraints consideration. A big challenge is also the validation of a trajectory planning approach; therefore, most of the existing methods were validated by simulations with the use of simple scenarios without the usage of real navigational data. This was the motivation to carry out the presented research.

The project presented in this paper fits in the recent trends, dealing with the development of algorithms for autonomous navigation of ships. The aim of the presented research is the development of new, effective algorithms for calculation of a safe, optimal trajectory for a ship in a collision situation at sea. The method has to be

Table 1 Comparison of different ship’s trajectory planning methods

Method	Static obs.	Dynamic obs.	Fitness function	Run time	Repeat-ability	Year
FMM	yes	no	multiple	< 1 s	yes	2017
VD	yes	1 obs.	length	<= 0.6 s	yes	2017
APF	yes	3 obs.	length	–	yes	2016
DG	no	yes	time	a few s	yes	2016
BPF	yes	1 obs.	length	< 10 s	yes	2015
EEA*	yes	no	energy	ms	yes	2015
FMM	yes	yes	length	0.1 s	yes	2015
ANN	no	2 obs.	risk	–	yes	2014
FL	no	yes	risk	–	yes	2014
CPP	no	5 obs.	$\alpha = 30^\circ$	7 s	yes	2013
PSO	yes	1 obs.	multiple	–	–	2012
EA	yes	yes	multiple	<=60 s	–	2012
A*	yes	1 obs.	length	–	yes	2012
PFM	yes	yes	length	–	yes	2011
EA	no	4 obs.	multiple	200–800 s	no	2010

characterized by low run time, repeatability of solution for every run of the algorithm with the same input data, taking into account of both static and dynamic obstacles and the International Regulations for Preventing Collisions at Sea (COLREGs).

2 Intelligent safe ship control system

The general structure of a modern safe ship control system, called the Guidance, Navigation, and Control (GNC) system (Fossen 2011), is shown in Fig. 1. The GNC system consists of three main subsystems: the Guidance System, performing path planning; the Control System, executing motion control; and the Navigation System, measuring motion parameters (ship’s positions, courses, and speeds). The main component of the Guidance System is called the Trajectory Generator (TG), which is an

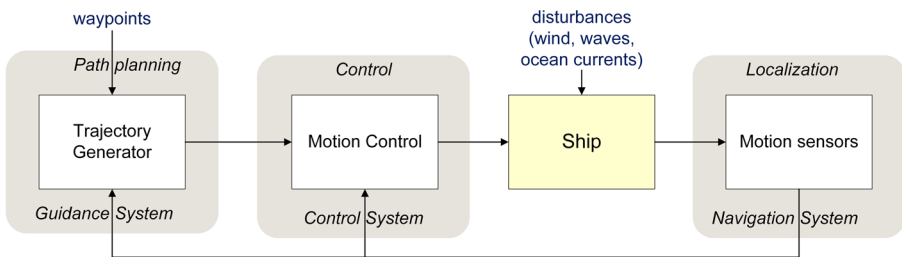


Fig. 1 Guidance, Navigation, and Control system

advanced optimization algorithm for calculation of a safe, optimal trajectory for a ship.

3 Ship's trajectory planning algorithms

3.1 Assumptions

Ship's trajectory planning is a complex optimization task with various restrictions and requirements to be considered. The following assumptions were defined in the process of solving the ship's trajectory planning problem applied in this research:

- availability of navigational data describing the current situation at sea;
- the International Regulations for Preventing Collisions at Sea (COLREGs) compliance of the calculated trajectory;
- static (lands, shallows) and dynamic (target ships - TSs) obstacles taken into account;
- safe distance (D_S) taken into account;
- trajectory calculated between the current own ship (OS) position and the defined final waypoint of the trajectory;
- TSs maintain their motion parameters;
- weather conditions (visibility) taken into account.

Input data to the algorithm, marked in Fig. 2, include:

- an OS course (ψ)
- an OS speed (V),

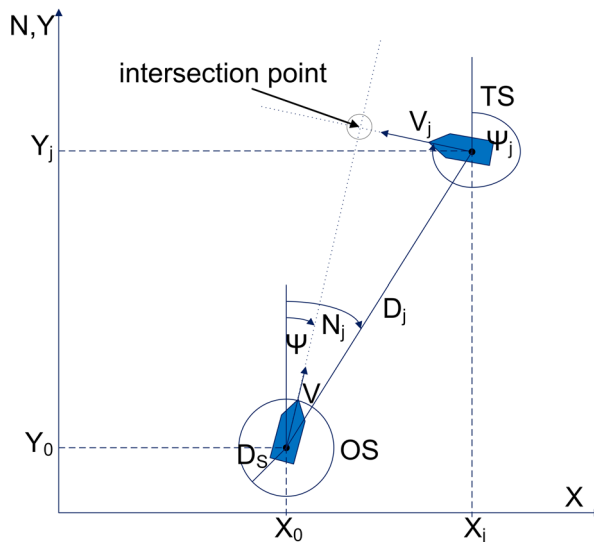


Fig. 2 Description of navigational situation

- TSs courses (Ψ_j),
- TSs speeds (V_j),
- TSs bearings (N_j)
- TSs distances from an OS (D_j),
- information concerning position of static constraints (lands, islands, buoys, fairways, canals, shallows).

These data are registered with the use of navigational equipment such as a radar with the Automatic Radar Plotting Aid (ARPA), the Automatic Identification System (AIS), the Electronic Chart Display and Information System (ECDIS), a gyrocompass, a speed log, an echo sounder, the Global Positioning System (GPS), or the Differential Global Positioning System (DGPS).

The COLREGs fulfilment by the algorithm means calculation of adequately large course change maneuvers (rule 8b) and a maneuver on the relevant side of the ship (rules 13, 14, & 15). Rule 17 of COLREGs defining the action to be taken by stand-on vessel is also taken into account by the algorithm.

A safe distance is ensured by a proper size of a ship domain around TSs. Weather conditions such as visibility (poor or good) are also taken into account by a proper size of a TS domain. It is assumed that the TSs maintain their motion parameters (course and speed). Therefore, the solution is recalculated if changes in their motion parameters and/or new TSs are detected. Two different algorithms, a deterministic and a heuristic one, have been applied for solving the ship's trajectory planning problem.

A starting point of a trajectory is defined as the current position of an own ship. An ending point of a trajectory is defined as a next waypoint of an own ship global path. For simulation tests, it was defined as a position of an own ship 9 nautical miles ahead of its current placement.

Dynamic obstacles (target ships) are considered by taking into account the hexagon shape of the ship's domain around the current positions of target ships. COLREGs are considered by the application of a proper shape and size of the ship's domain. An extension of the ship's domain in the bow direction causes in a crossing situation (rule 15 of COLREGs) passage of an own ship behind the stern of the target ship. Enlargement of the ship's domain to the starboard side will force in the head-on situation (rule 14 of COLREGs) changing the course of an own ship to the starboard side.

3.2 Ant colony optimization algorithm

ACO belongs to the Swarm Intelligence (SI) methods. Bonabeau et al. (1999) defined SI as any attempt to build an algorithm inspired by the collective behavior of the colony of insects or other animal communities. A colony of insects is composed of a number of relatively simple interacting individuals, which can achieve a lot by their interaction. The main features of a colony of insects, used for problem solving, are:

- self-organization,
- flexibility,
- robustness.

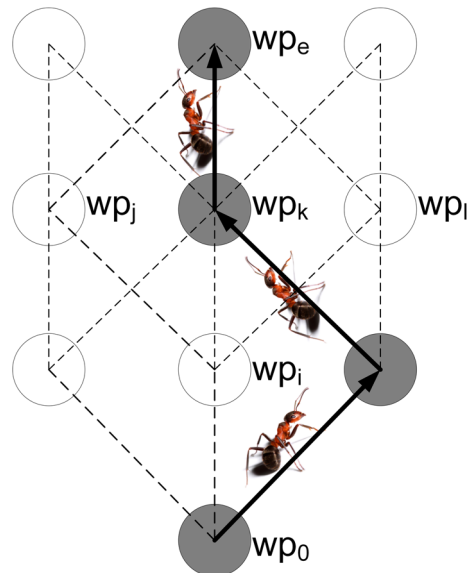
Flexibility of a colony helps it to adapt to changing environments, while robustness allows it to operate even if some individuals do not perform their tasks. Colonies of insects mastered the tasks such as foraging, building, or expanding their nest, by effective division of work among individuals. The knowledge gained during observation of the colonies behaviors allows for its usage in the engineering and computer science.

The Ant Colony Optimization (ACO) algorithm is a heuristic method inspired by the ant colony foraging behavior. ACO was introduced by Dorigo and Stutzle (2004). Its first application was solving the combinatorial optimization problem called the Travelling Salesman Problem. Ants, when searching for food, use a special mechanism to find the shortest path between the food source and their nest. They communicate with each other and with the environment by depositing a chemical substance on the ground, called a pheromone. The pheromone trail is used by the ants to transfer information to other individuals in the colony. This mechanism is called stigmergy.

This trail-laying and trail-following behavior, observed in colonies of real ants, is applied to strengthen parts of good solutions in the ACO algorithm. Artificial ants put the virtual pheromone trail on their paths, what enables to keep good solutions in the memory, so that they can be used to obtain better solutions in the future. In order to avoid the algorithm's convergence to a local minimum, the pheromone evaporation is also applied.

In ACO approach, a graph composed of possible OS positions is constructed, taking into account all of the obstacles. After that ACO calculations are carried out. They include:

Fig. 3 An exemplary construction graph



- ACO data initialization,
- solution construction,
- pheromone trail update.

ACO data initialization includes the definition of parameters such as α and β coefficients, initial pheromone trail amount at each of the possible waypoints, pheromone evaporation coefficient, number of ants, maximum number of steps to be made by an ant, and number of iterations.

At the solution construction stage, every ant constructs its path from the current OS position to the next waypoint, as shown in Fig. 3. The ant stops its search, when it is located in the final waypoint or it reaches the maximum number of steps.

At every step, an ant chooses the next OS position (vertex on the graph) with the use of the action choice rule, which works similarly to the roulette wheel selection used in Evolutionary Algorithms. The probability of choosing the next vertex is defined by Eq. 1. It depends on the pheromone trail amount deposited ($\tau_{wp_j}(t)$) on the neighboring vertex and a heuristic information called visibility ($\eta_{wp_{ij}}$), which is expressed as the inverse of the distance between the current vertex (i) and the neighboring vertex (j). When the path constructed by the ant is shorter than the shortest path found so far, then it becomes the shortest path.

$$P_{wp_{ij}}^{ant}(t) = \frac{[\tau_{wp_j}(t)]^\alpha \cdot [\eta_{wp_{ij}}]^\beta}{\sum_{l \in wp_i^{ant}} [\tau_{wp_l}(t)]^\alpha \cdot [\eta_{wp_{il}}]^\beta} \tag{1}$$

The pheromone trail update procedure, defined by Eq. 2, is composed of two stages:

- pheromone evaporation,
- pheromone deposit.

$$\tau_{wp_j}(t + 1) = (1 - \rho) \cdot \tau_{wp_j}(t) + \sum_{ant=1}^m \Delta\tau_{wp_j}^{ant}(t) \tag{2}$$

In the pheromone evaporation, the pheromone trail amount is reduced for all vertices by a defined value. In the pheromone deposit, a certain value of the pheromone trail is added to all vertices belonging to the paths constructed by ants at the solution construction stage.

After that the shortest path found by the ants in the current iteration is saved. The fitness function applied for evaluation of the ant’s paths is defined by Eq. 3. After the maximum number of iterations is reached or the maximum run time is achieved, the shortest path is returned as a final solution.

$$I = \sum_{i=1}^{M-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \rightarrow \min \tag{3}$$

3.3 Trajectory base algorithm

The Trajectory Base Algorithm (TBA) is a deterministic approach. Its operation is based upon searching a base of trajectories, which constitute candidate solutions to the problem, in order to find the best solution for the regarded collision situation. Trajectories generated with the use of an exemplary rule are shown in Fig. 4.

TBA evaluates trajectories stored in the database one after another until it finds a trajectory solving the considered navigational situation. The trajectory evaluation process is based upon the division of the trajectory into a number of sections. After that, for every section, the algorithm checks whether the instantaneous positions of an own ship and target ships do not cause a collision. When a collision is detected, the trajectory is rejected and the next trajectory is collected from the database for evaluation.

Trajectories in the database are sorted according to increasing value of their fitness function. The best trajectory is the trajectory with the minimal value of the fitness function, which is defined (in the same way as in ACO) as the length of a trajectory. This approach causes a significant reduction of the run time of the algorithm, because not all of trajectories are evaluated, but only as many as it is needed to find the best solution. COLREGs compliance of the trajectory is ensured by a proper shape and size of the TS domain. When the algorithm finds the best trajectory not exceeding the constraints, it stops the selection process and the OS course at every line segment of the trajectory is calculated. After that the solution is presented to the user in a graphical and numerical form.

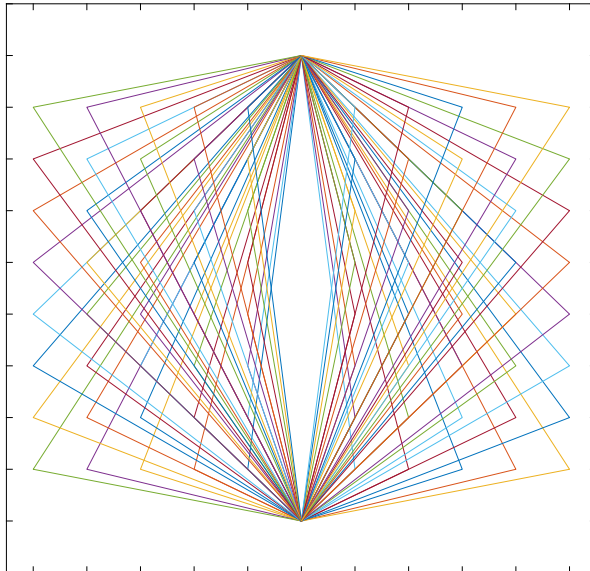


Fig. 4 Exemplary set of candidate trajectories



Fig. 5 Research/Training ship M/V Horyzont II

4 Simulation studies

4.1 Real data registration

The real navigational data describing the current situation at sea were registered during the Horyzont II voyages in 2018. Horyzont II, shown in Fig. 5, is a Research/Training ship owned by Gdynia Maritime University. The functions of Horyzont II include the following: conducting research, conducting training of students, and transporting equipment of Polish Academy of Sciences to the Polish scientific bases on Spitsbergen. In Table 2, the main technical parameters of the vessel are listed. Figure 6 shows the bridge interior of the ship, while in Fig. 7 the installation for data registration is presented.

Input data to the algorithm are registered from ARPA with the use of the NMEA standard, which is a serial asynchronous data transmission protocol used for communication between marine electronic equipment and external devices. The standard

Table 2 Technical specification of M/V Research/Training Ship Horyzont II

Parameter	Value
Length	56.34 m
Breadth	11.36 m
Designed draft	3.90 m (5.33 m together with the keel)
Deadweight	288 t
Gross tonnage	1321 BRT
Speed	12 knots
Main engine power	1280 kW
Controllable-pitch propeller (CPP)	CP 65 WARTSILA , D = 2.1 m
Bow thruster	STT 10 LK SCHOTTEL - power: 125 kW
Build year and place	2000, Gdańsk
IMO number	9231925



Fig. 6 The bridge interior of the Research/Training ship M/V Horizont II

defines the data frame structure such as one start bit, eight data bits, no parity bit and one stop bit, and the transmission speed of 4800 bits per second. It also defines the structures of transmitted sentences. The sentences needed for the ship's trajectory calculation are marked as OSD (Own Ship Data) and TTM (Tracked Target Message). The standard repetition time of OSD sentence is 1 second and of TTM sentence is 10 seconds. The OSD and TTM sentences structures are given in Fig. 8, while exemplary OSD and TTM sentences registered on board Horizont II are shown in Fig. 9.

4.2 Simulation tests

The algorithms were implemented in the MATLAB environment. Simulation studies were carried out with the use of a PC with an Intel Core 2 Duo E7500 2.93 GHz processor, 4GB RAM, 64-bit Windows 7 Professional. The parameters of ACO used in simulations were: $\tau_0 = 1$, $\rho = 0.1$, $\alpha = 1$, $\beta = 2$, iterations = 20 and ant_number = 10.

Two exemplary test cases have been presented in the paper. In Table 3, input data for test case 1 are listed. The data constitute a real navigational situation registered with the use of a Furuno radar with ARPA on board the ship Horizont II. Column 1



Fig. 7 The computer for data registration connected to the Furuno radar with ARPA

```

$RAOSD,a.a,A,c.c,X,x.x,Y,y.y,z.z,N*hh<CR><LF>
RA - talker identifier: RA - radar
OSD - sentence formatter: OSD - own ship data
a.a - heading, degrees true
A - heading status: a - valid, V - invalid
c.c - vessel course, degrees true
X - course reference: W - water-referenced
x.x - vessel speed
Y - speed reference: W - water-referenced
y.y - vessel set, degrees true
z.z - vessel drift (speed)
N - speed units: N - knots

$SRATTM,aa,b.b,c.c,T,t.t,x.x,T,y.y,z.z,X,n,Y,R,hhmss.ss,M*hh<CR><LF>
RA - talker identifier: RA - radar
TTM - sentence formatter: TTM - tracked target message
aa - target number: 00 to 99
b.b - target distance from own ship
c.c - bearing from own ship
T - degrees true
t.t - target speed
x.x - target course
y.y - DCPA - distance at closest point of approach
z.z - TCPA - time to the closest point of approach [min]
X - speed, distance units
n - user data (e.g. target name)
Y - target status: L - lost, Q - query (target in the process of acquisition), T - tracking
R - reference target (used to determine own ship speed)
hhmss.ss - time of data (UTC) M - type of acquisition: a - automatic, M - manual,
R - reported (eg.AIS)

```

Fig. 8 OSD and TTM sentences structure

includes numbers of the ships as marked in Fig. 10. Own ship is marked with 0, and target ships with consecutive numbers. Test case 1 is an encounter situation with four target ships. In the following columns of Table 3 position and motion parameters of the ships are given, such as the courses in degrees, speeds in knots, distances from an own ship in nautical miles and bearing relative to true north in degrees, what means that the direction toward the geographic North is used as a reference point.

Solutions for test case 1 are presented graphically in Fig. 10. The blue line presents the trajectory calculated by both algorithms (ACO and TBA). The areas around the target ship's positions present the target ship's domains. The ship's domains marked by dashed lines present target ship's positions at the moments, when an own ship is placed at the course alteration points. The ship's domains marked by solid lines show the target ship's positions, when an own ship is placed at the final waypoint. Analysis of the graphical results for test case 1 confirms that the determined own ship trajectory is a safe trajectory, which allows the own ship to pass from the start waypoint to the final waypoint avoiding all of the target ships. The determined own ship trajectory is compliant with rule 8b of COLREGs, because the calculated own

```

14-Jul-2018 06:40:04
$RAOSD,132.2,A,132.2,W,10.8,W,,,N*7F
$RATTM,45,5.15,169.3,T,16.32,311.9,R,3.13,15.1,N,,Q,,,M*13
$RATTM,41,1.42,345.5,T,3.50,137.8,R,0.66,21.6,N,,T,,,M*28
$RATTM,44,3.41,109.2,T,10.90,315.6,R,1.52,16.9,N,,Q,,,M*1B
$RATTM,42,1.57,108.8,T,5.02,145.7,R,0.94,-15.0,N,,T,,,M*03

14-Jul-2018 11:22:03
$RAOSD,161.4,A,161.4,W,11.6,W,,,N*70
$RATTM,70,8.29,169.3,T,16.74,340.4,R,1.28,29.4,N,,T,,,M*19
$RATTM,71,3.68,238.5,T,17.45,350.6,R,3.41,4.8,N,,T,,,M*29
$RATTM,72,2.51,249.6,T,17.67,353.2,R,2.44,2.0,N,,T,,,M*29
$RATTM,73,3.92,181.9,T,4.18,101.5,R,3.87,-9.6,N,,T,,,M*38
$RATTM,74,4.93,211.2,T,17.20,352.4,R,3.09,13.4,N,,T,,,M*16
$RATTM,75,5.21,213.4,T,17.30,354.8,R,3.25,14.1,N,,T,,,M*1C
$RATTM,76,6.78,205.9,T,17.36,352.8,R,3.71,19.6,N,,T,,,M*11
$RATTM,77,8.94,201.5,T,17.71,350.5,R,4.60,26.0,N,,T,,,M*15
$RATTM,79,2.20,206.5,T,14.38,35.2,R,0.33,9.1,N,,Q,,,M*1B
$RATTM,80,2.04,190.5,T,15.41,311.6,R,1.76,4.1,N,,Q,,,M*24
$RATTM,81,1.86,176.1,T,22.01,291.1,R,1.68,2.1,N,,Q,,,M*27
$RATTM,82,6.00,180.5,T,12.66,342.8,R,1.82,27.1,N,,Q,,,M*17
$RATTM,50,1.71,335.9,T,6.96,274.6,R,1.50,-7.1,N,,T,,,M*3C
$RATTM,64,5.16,258.6,T,8.09,314.7,R,4.28,-21.4,N,,T,,,M*0E
$RATTM,65,2.93,14.0,T,22.25,341.8,R,1.56,-6.7,N,,T,,,M*3A
$RATTM,66,2.20,67.6,T,19.68,341.6,R,2.20,-0.5,N,,T,,,M*3A
$RATTM,67,1.94,10.3,T,11.10,339.1,R,1.00,-9.0,N,,T,,,M*30
$RATTM,68,4.52,15.5,T,10.76,341.7,R,2.51,-21.0,N,,T,,,M*06

```

Fig. 9 Exemplary OSD and TTM sentences registered on board Horizont II

ship maneuvers are large enough to be readily apparent to other vessels. It fulfills also rule 13 of COLREGs dealing with the overtaking situation.

Numerical results for test case 1 are shown in Table 4. The results include the run time of the algorithms in seconds, the distance of the calculated trajectories in nautical miles, and the course alterations in degrees, needed to be executed by an own ship. Both algorithms returned the same solution for test case 1, composed of two course changes, by 14 degrees to the starboard side and after that by 25 degrees to the port side. The difference was in the run time of the algorithms. TBA returned a solution in 0.2 s, while ACO needed about 10 s.

Test case 2 dealt with an encounter situation with eighteen target ships, also registered on board the ship Horizont II in July 2018 during its XLI voyage to Spitsbergen.

Table 3 Input data for test case 1

Ship	Ψ [°]	V [kn]	D [nm]	N [°]
0	132.2	10.8	—	—
1	311.9	16.32	5.15	169.3
2	137.8	3.5	1.42	345.5
3	315.6	10.9	3.41	109.2
4	145.7	5.02	1.57	108.8

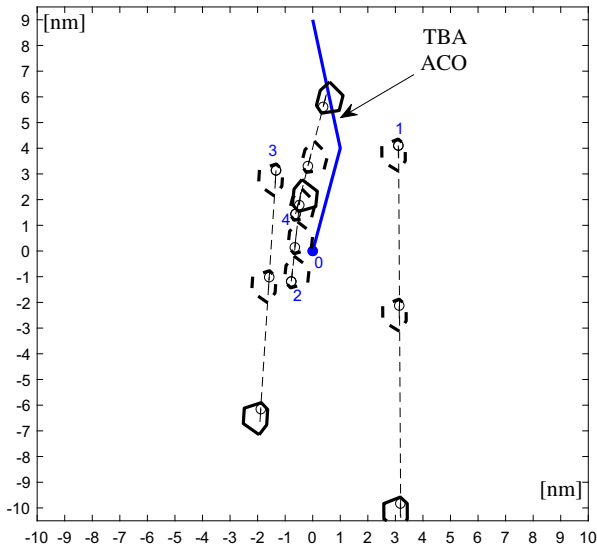


Fig. 10 Solution of test case 1

Input data for this test case are listed in Table 5. The graphical presentation of own ship safe trajectories calculated for test case 2 by ACO and TBA are shown in Fig. 11. The graphical results confirm that the own ship trajectory is a safe trajectory, enabling it to pass to the final waypoint without colliding with any of the target ships. The own ship trajectory is also compliant with rule 8b of COLREGs, because all of the determined course changes are large enough. It also fulfills rule 15 of COLREGs, because the own ship gives way to the vessel number 4.

Numerical results for test case 2 are given in Table 6. TBA returned a shorter trajectory than ACO by 1.5 nautical miles in about 1 s, composed of three course changes, by 18 degrees to the port side, after that by 18 degrees to the starboard side and finally by 19 degrees also to the starboard side. The run time of ACO was about 30 s and the trajectory was composed of two course change maneuvers, by 21 degrees to the port side and after that by 48 degrees to the starboard side.

Analysis of the results enable to state the following remarks:

- both algorithms return safe own ship trajectories,
- both algorithms return solutions compliant with COLREGs,
- for some case, both algorithms return the same trajectory, while for other TBA return shorter trajectory that ACO,

Table 4 Results for test case 1

Algorithm	Run time [s]	Distance [nm]	$\Delta\Psi$ [°]
TBA	0.2	9.22	14, 25
ACO	about 10	9.22	14, 25

Table 5 Input data for test case 2

Ship	Ψ [°]	V [kn]	D [nm]	N [°]
0	161.4	11.6	—	—
1	340.4	16.74	8.29	169.3
2	350.6	17.45	3.68	238.5
3	353.2	17.67	2.51	249.6
4	101.5	4.18	3.92	181.9
5	352.4	17.2	4.93	211.2
6	354.8	17.3	5.21	213.4
7	352.8	17.36	6.78	205.9
8	350.5	17.71	8.94	201.5
9	35.2	14.38	2.2	206.5
10	311.6	15.41	2.04	190.5
11	291.1	22.01	1.86	176.1
12	342.8	12.66	6	180.5
13	274.6	6.96	1.71	335.9
14	314.7	8.09	5.16	258.6
15	341.8	22.25	2.93	14
16	341.6	19.68	2.2	67.6
17	339.1	11.1	1.94	10.3
18	341.7	10.76	4.52	15.5

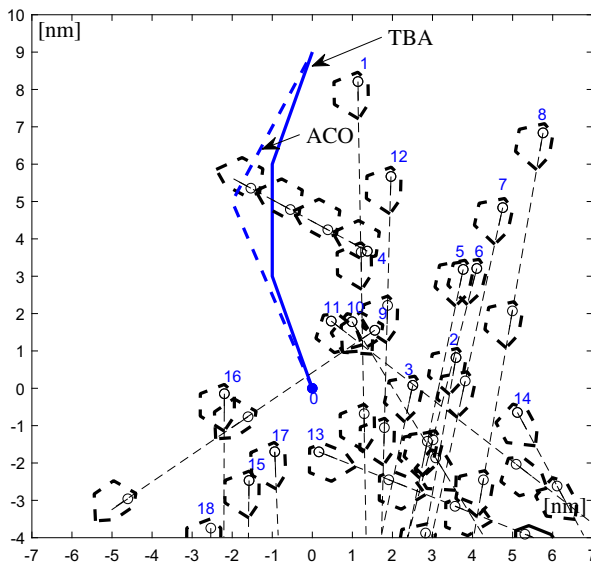
**Fig. 11** Solution of test case 2

Table 6 Results for test case 2

Algorithm	Run time [s]	Distance [nm]	$\Delta\Psi$ [°]
TBA	1.03	9.32	18, 18, 19
ACO	about 30	9.86	21, 48

- TBA returns a solution in much shorter time (in near-real time) compared with ACO.

5 Experimental studies

Preliminary experimental studies of the algorithms were conducted with the use of a system composed of a group of mobile platforms and an Indoor Positioning System (IPS) for localization of the moving objects. The IPS (Pozyx Labs BVBA 2018) is composed of tags placed on mobile platforms, providing information about their position and direction of movement, and four nodes (anchors) with known positions. The tag contains an ultra-wideband transceiver and an inertial measurement unit DWM1000 from DecaWave for measuring the orientation of an object, including accelerometers, gyroscopes, and magnetometers. The mobile platforms DFRobot Pirate-4WD (Zhiwei Robotics Corp 2018) with the dimensions of $200 \times 170 \times 105$ mm (length \times width \times height) and a speed of up to 0.9 m/s, equipped with 4 DC motors and characterized by differential-drive steering, were used in the experiments. The results of an exemplary experimental test for situation with two static obstacles in the environment are shown in Fig. 12. The mobile platform is marked as 0x6973,

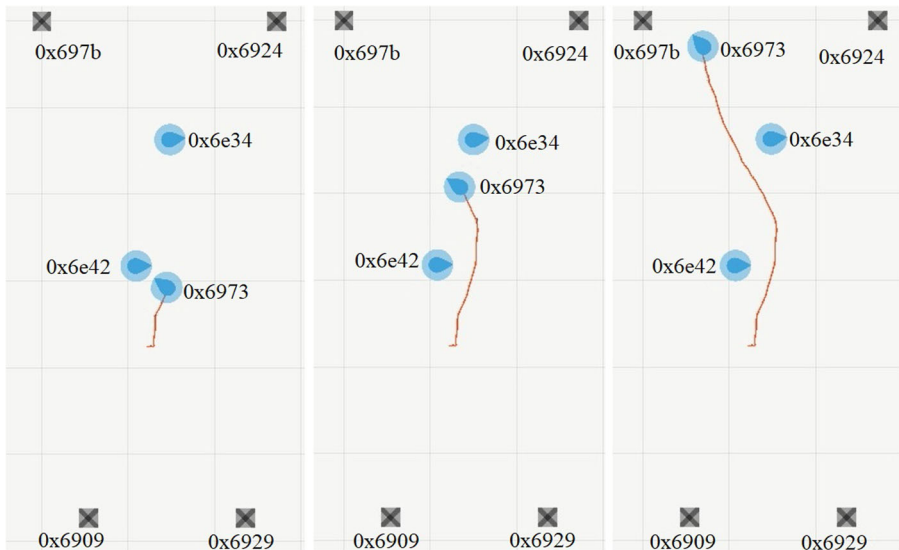


Fig. 12 Results of experimental test for an encounter situation with two static obstacles registered with the use of an Indoor Positioning System

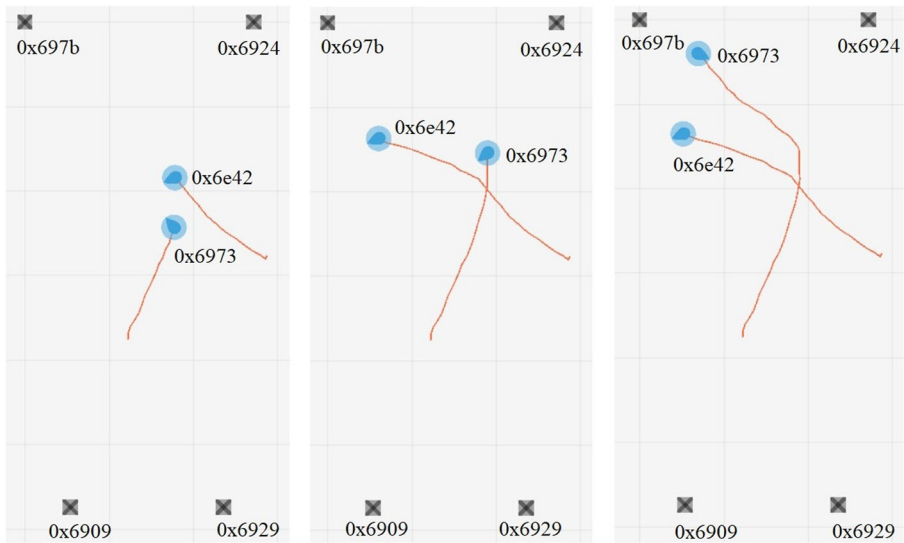


Fig. 13 Results of experimental test for an encounter situation with one dynamic obstacle registered with the use of an Indoor Positioning System

static obstacles are marked as $0\times 6e42$ and $0\times 6e34$. Objects marked by $0\times 697b$, 0×6924 , 0×6929 , and 0×6909 are the anchors of the IPS. The results of an exemplary experimental test for situation with one dynamic obstacle, marked as $0\times 6e42$, are shown in Fig. 13. The graphical presentation of results confirm that the trajectory calculated by the algorithm is a safe trajectory and that its execution will not cause a collision with the obstacles.

6 Conclusions

The paper deals with the problem of ship's safe trajectory planning in a collision situation at sea. The main goal of the presented research was to develop and test new path planning algorithms for ships, what will contribute to achieve safer shipping and progress in autonomous navigation.

Presented results lead to the following conclusions:

- the TBA and ACO algorithms are capable of finding a ship's safe trajectory in collision situations at sea, what was confirmed by simulation studies with the use of real navigational situations registered on board the ship *Horzont II*,
- the run time of both algorithms does not exceed 1 minute, therefore the algorithms are suitable for use in commercial systems.

Further works will include more extensive test with the use of mobile platforms and tests of the algorithm in real conditions on board the ship *Horzont II* during voyages.

Funding information The author received financial support from The Nippon Foundation in Japan and the International Association of Maritime Universities (IAMU) .

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Allianz Global Corporate & Specialty SE (2018) Safety and shipping review. <https://www.allianz.com/en-GB/press/news/studies/180718-agcs-safety-and-shipping-review-2018.html>. Accessed 4 December 2018
- Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: From natural to artificial systems. Oxford University Press, Oxford
- Candeloro M, Lekkas AM, Sørensen AJ (2017) A voronoi-diagram-based dynamic path-planning system for underactuated marine vessels. *Control Eng Pract* 61:41–54. <https://doi.org/10.1016/j.conengprac.2017.01.007>
- CCS ZhuhaiMunicipalGovernment, Wuhan University of Technology & Zhuhai Yunzhou Smart Co (2018) Wanshan marine test field. <https://www.marineinsight.com/shipping-news/construction-worlds-largest-unmanned-marine-testing-site-starts/>, Accessed 4 December 2018
- Chen LJ, Huang LW (2012). In: Zhang Y (ed) Ship collision avoidance path planning by pso based on maneuvering equation. Springer Publishing Company, Incorporated, Berlin, pp 675–682
- Dorigo M, Stutzle T (2004) Ant colony optimization. The MIT Press
- Fossen TI (2011) Handbook of marine craft hydrodynamics and motion control, Wiley, New York
- Kongsberg Maritime AS (2018) Autonomous ship project. <https://www.km.kongsberg.com>. Accessed 4 December 2018
- Lee T, Kim H, Chung H, Bang Y, Myung H (2015) Energy efficient path planning for a marine surface vehicle considering heading angle. *Ocean Eng* 107:118–131. <https://doi.org/10.1016/j.oceaneng.2015.07.030>
- Lisowski J (2016) Analysis of methods of determining the safe ship trajectory. *TransNav - the International Journal on Marine Navigation and Safety of Sea Transportation* 10:223–228. <https://doi.org/10.12716/1001.10.02.05>
- Liu Y, Bucknall R (2015) Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. *Ocean Eng* 97:126–144. <https://doi.org/10.1016/j.oceaneng.2015.01.008>
- Mohamed-Seghir M (2014) The branch-and-bound method, genetic algorithm, and dynamic programming to determine a safe ship trajectory in fuzzy environment. *Proc Comput Sci* 35:348–357. <https://doi.org/10.1016/j.procs.2014.08.115>, knowledge-Based and Intelligent Information and Engineering Systems 18th Annual Conference, KES-2014 Gdynia, Poland, September 2014 Proceedings
- Montiel O, Orozco-Rosas U, Sepulveda R (2015) Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles. *Expert Systems with Applications* 42:5177–5191. <https://doi.org/10.1016/j.eswa.2015.02.033>
- Naeem W, Irwin G, Yang A (2012) Colregs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics* 22:669–678. <https://doi.org/10.1016/j.mechatronics.2011.09.012>
- Naeem W, Henrique SC, Hu L (2016) A reactive colregs-compliant navigation strategy for autonomous maritime navigation. *IFAC-PapersOnLine* 49(23):207–213. <https://doi.org/10.1016/j.ifacol.2016.10.344>, 10th IFAC Conference on Control Applications in Marine Systems CAMS 2016
- Pozyx Labs BVBA (2018) Pozyx labs indoor positioning system. <https://www.pozyx.io>. Accessed 4 December 2018
- Simsir U, Amasyal MF, Bal M, Çelebi UB, Ertugrul S (2014) Decision support system for collision avoidance of vessels. *Appl Soft Comput* 25:369–378. <https://doi.org/10.1016/j.asoc.2014.08.067>
- Song R, Liu Y, Bucknall R (2017) A multi-layered fast marching method for unmanned surface vehicle path planning in a time-variant maritime environment. *Ocean Eng* 129:301–317. <https://doi.org/10.1016/j.oceaneng.2016.11.009>

- Szlapczynski R, Szlapczynska J (2012) On evolutionary computing in multi-ship trajectory planning. *Appl Intell* 37(2):155–174
- Tam C, Bucknall R (2010) Path-planning algorithm for ships in close-range encounters. *J Mar Sci Technol* 15:395–407. <https://doi.org/10.1007/s00773-010-0094-x>
- Tam C, Bucknall R (2013) Cooperative path planning algorithm for marine surface vessels. *Ocean Eng* 57:25–33. <https://doi.org/10.1016/j.oceaneng.2012.09.003>
- Xue Y, Clelland D, Lee B, Han D (2011) Automatic simulation of ship navigation. *Ocean Eng* 38:2290–2305. <https://doi.org/10.1016/j.oceaneng.2011.10.011>
- Zhiwei Robotics Corp (2018) Dfrobot pirate-4wd mobile platform. <https://www.dfrobot.com>. Accessed 4 December 2018

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.