Expert Systems With Applications 71 (2017) 469–478

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A new deterministic approach in a decision support system for ship's trajectory planning

Agnieszka Lazarowska

Gdynia Maritime University, Morska Str. 81-87, 81-225 Gdynia, Poland

ARTICLE INFO

Article history:
Received 29 January 2016
Revised 2 November 2016
Accepted 2 November 2016
Available online 11 November 2016

Keywords:
Collision avoidance
Decision support
Knowledge-based system
Motion planning
Path planning
Ship navigation

ABSTRACT

The paper introduces a Decision Support System for ships, developed to solve a problem of collision avoidance with static and dynamic obstacles. The system maps the decision making capability of a human (navigation) expert to solve the path planning problem for a ship in a complex navigation environment. It can be further developed to provide automatic control of a ship. It utilizes a new, fast and effective, deterministic method, called the Trajectory Base Algorithm, to calculate a safe, optimal path for a ship. The system structure, a detailed explanation of a new method, followed by results of simulation tests are all presented in the paper. The results proof a successful application of the method to solve a path planning problem for ships with the consideration of both static and dynamic obstacles in the environment, marine traffic regulations and dynamic properties of a ship, what makes this approach applicable in commercial systems. The approach can also be adapted for application in mobile robots path planning. The experimental results and ability of the system to achieve a new functionality of full autonomy show significance of this contribution to the development of Expert and Intelligent Systems domain. The author believes that autonomous systems constitute the future of Expert and Intelligent Systems.

1. Introduction

A Decision Support System (DSS) is a computer-based system that aids the user in decision making. DSSs are applied in business and management, in industrial processes and control of complex systems. A DSS uses techniques from operations research, information science and artificial intelligence implemented in the form of a computer program to help a human decision maker in making appropriate choices. The greatest advantage of DSSs is the ability to help the user in making an adequate judgment and a proper decision, especially in complex and stressful situations. DSSs are particularly helpful in processes with big data to analyse, because of their ability to integrate various sources of information. Based upon that, a DSS makes a precise and optimal decision.

Planning a safe path for a ship constitutes a complex process, in which precision and optimality are of vital importance. The navigation DSS integrates various sources of information about the situation and the surrounding environment and based upon that gives advice about collision avoidance actions that should be taken. Occurrence of a huge amount of data makes this task even more complicated and difficult to perform. Modern ships are equipped with a large number of different navigational aids. Such equipment include a radar with an Automatic Radar Plotting Aid (ARPA), an Automatic Identification System (AIS) and an Electronic Chart Display and Information System (ECDIS), also gyrocompasses, speed logs, echo sounders, GPSs or DGPSs, wind speed and direction sensors and others. Access to a vast amount of data can impede rather than ease the decision making process. That is why there is a need for the development of a DSS for ships.

In this paper a new proposal of a navigation DSS along with a new path planning method for ships is introduced. The presented system is an advisory system, which mimics the human (navigation) expert decision-making capability. It is dedicated to solve complicated situations. It consists of a computer collecting and analysing input data and based upon that making a complex decision. To sum up, it is characterized by the features that allow to classify it as an intelligent DSS. Furthermore, it can be developed into an autonomous system, which automatically controls a ship or an Unmanned Surface Vehicle (USV), by joining it with an automatic pilot (automatic motion control system). The approach can also be adopted to application in other dynamic environments, where a moving object path planning problem has to be solved, for example in navigation of mobile robots. The proposal of a new path planning method for a moving object (a ship) in a dynamic environment (a marine environment), as mentioned above, affects the development of autonomous systems and is believed by the

E-mail address: a.lazarowska@we.am.gdynia.pl

http://dx.doi.org/10.1016/j.eswa.2016.11.005
0957-4174/

author to constitute a valuable contribution into Expert and Intelligent Systems.

2. Related works

Navigation DSSs for ships, called ship guidance systems, can be divided into two general approaches, based on the methods they utilize for the calculation of a collision free path. These are the systems based on deterministic and heuristic methods. In general the deterministic approaches are characterized by the convergence to the optimal solution and low computational time, but they might be incapable of solving complex situations. On the other hand stochastic-based algorithms might deal better with more complex environments, but due to the presence of random variables convergence of solution cannot be guaranteed and they are more time-consuming.

Path planning and collision avoidance methods for ships have been revised in Statheros, Howells, and Maier (2008); Tam, Bucknall, and Greig (2009) and Campbell, Naeem, and Irwin (2012). Existing approaches utilize the Dynamic Games theory (Lisowski, 2014a), the Dynamic Programming (Lisowski, 2014b), the Time-Optimum Control (Zak, 2004),the Maze Routing Algorithm (Szlapczynski, 2006), the Fast Marching Method (Liu & Bucknall, 2015), the Genetic Algorithm (Kuczkowski & Smierzchalski, 2013; Szlapczynski & Szlapczynska, 2012; Tsou, Kao, & Su, 2010), the Swarm Intelligence (Tsou & Hsueh, 2010), the Branch and Bound Method (Mohamed-Seghir, 2012) and the A* Algorithm (Naeem, Irwin, & Yang, 2012).

Based upon the review of the most actual literature in the topic, the most up-to-date and according to the author the most promising solutions were presented below for comparison with the authors contribution presented in this paper.

The most recent contributions to path planning and collision avoidance of ships are the Cooperative Path Planning algorithm (Tam & Bucknall, 2013), the Artificial Potential Field (Xue, Clelland, Lee, & Han, 2011) and the Fuzzy Logic based method (Perera, Carvalho, & Soares, 2010), the Evolutionary Algorithms (Tam & Bucknall, 2010) and the Ant Colony Optimization (Lazarowska, 2015).

The Cooperative Path Planning (CPP) algorithm is a deterministic path planning method, in which a risk of collision is first calculated for every target ship. Every ship is assigned a priority based on the International Regulations for Preventing Collisions at Sea (COLREGs) and maneuverability. In a second step of the algorithm, evasive maneuvers are calculated in order to eliminate the risk of collision. The CPP algorithm is characterized by reproducibility of solution for the same input data, consideration of COLREGs and dynamic properties of a ship, and low computational time - around a few seconds. The features that might restrain its application are omission of the static obstacles and limitation of the course change maneuver to one value of 30°.

In the Artificial Potential Field (APF) method a ship is pulled toward the destination point by the attractive force and pulled away from an obstacle by the repulsive force. Similarly to the previously mentioned method, the ships taking part in the current situation are assigned a priority according to COLREGs and the collision risk (distance between the ships). The APF method is capable of handling both static and dynamic obstacles, considers COLREGs and dynamic properties of a ship, but computational time and reproducibility of results were not raised in the paper.

In the Fuzzy Logic (FL) based approach a Bayesian network is applied for the determination of a sequence of collision avoidance actions to be taken by a ship. The collision avoidance decisions are first calculated with the use of a fuzzy logic based system, incorporating the COLREGs and the collision risk assessment in the form of if-then rules. The main features of the FL method are the COLREGs compliance of a solution and reproducibility of results. The approach does not provide the possibility to consider avoiding static obstacles. Taking into account dynamic properties of a ship and the computational time are not mentioned by the authors.

In the Evolutionary Algorithm (EA) based approach population of navigation paths is initialized with the use of a pseudo-number generator. Then, evolution is realized by selection, variation and evaluation processes. The stopping criterion is the saturation of the population fitness. The method takes into account dynamic properties of a ship. What might reduce applicability of this approach is negligence of static obstacles, its stochastic nature, the possibility that a solution may not be compliant with rule 8b of COLREGs and relatively high computational time - over 200 s.

The Ant Colony Optimization (ACO) based algorithm utilizes a method inspired by nature, by a collective behaviour of ants. A group of agents - artificial ants search through the solution space in order to find a safe and optimal path for a ship. The method is capable of taking into account both static and dynamic obstacles, and dynamic properties of a ship. Due to the application of additional mechanisms it returns a smooth path and a solution is reproducible. Its limitation is medium computational time - from several to tens of seconds.

Table 1
Comparison between different existing most promising path planning methods for ships and the TBA.

	CPP	APF	FL	EA	ACO	TBA
Static obstacles	no	yes	no	no	yes	yes
Dynamic obstacles	yes	yes	yes	yes	yes	yes
Complex environment	med	med	low	med	high	high
COLREGs	med	med	med	med	high	high
Dynamic properties	yes	yes	no	yes	yes	yes
Operator's preferences	no	no	no	pos	pos	yes
Computational time	low	?	?	high	med	v.low
Reproducibility	yes	?	yes	no	yes	yes

Table 1 shows a comparison of the most recent path planning approaches for ships. The methods were evaluated according to the fulfilment of eight requirements/features: consideration of static and dynamic obstacles, problem solving capability in complex environments, COLREGs compliance of a solution, consideration of dynamic properties of a ship, operator's preferences (possibility of the system operator to choose the path optimization objectives: path length, transition time, path smoothness, distance from obstacles), computational time and reproducibility of a solution. *Pos* abbreviation in the table means the possiblity to apply the above mentioned feature in the algorithm. Some requirements are evaluated straightforward based upon fulfilment (*yes* in the table) or failure in fulfilment (*no* in the table) of a defined criterion. For other features a degree of fulfilment is specified (*low* fulfilment, *medium* (*med* abbreviation in the table) fulfilment or *high* fulfilment). The computational time has its own scale of evaluation, where the time can be *very low* (milliseconds) (*v.low* abbreviation in the table), *low* (seconds), *medium* (several or tens of seconds) or *high* (hundreds of seconds). In the last column of the table a method proposed in this paper is evaluated for comparison with existing approaches.

The purpose of the research presented here was the development of a method applicable in a navigation DSS for ships, which will eliminate all of the limitations of the existing approaches and by achieving that will become applicable in commercial systems.

3. DSS general description

A DSS for safe ship path planning in a collision situation at sea has to fulfill requirements such as:

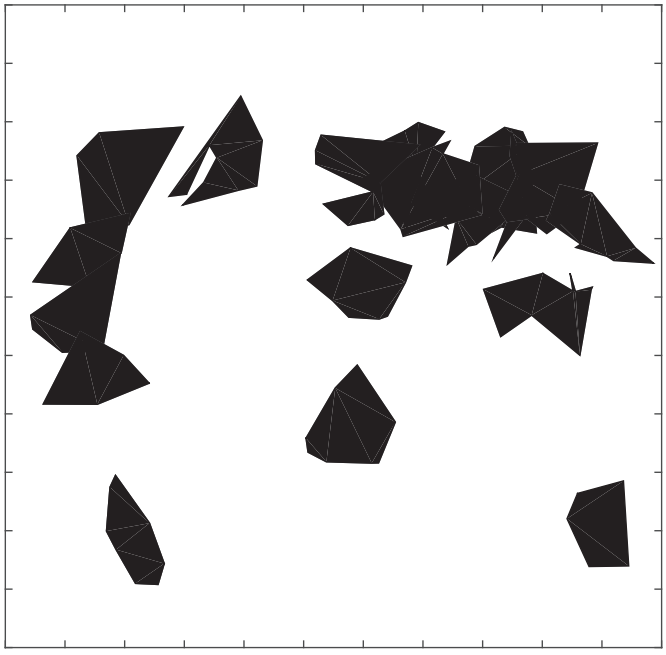

Fig. 1. Exemplary static obstacles modeled as polygons.

- ability to avoid static and dynamic obstacles in the environment,
- International Regulations for Preventing Collisions at Sea (COLREGs) compliance of a solution (proper maneuvers greater than the specified minimal value),
- near real time operation,
- consideration of dynamic properties of a ship in a solution found.

The system applicable in commercial solutions has to be reliable and has to react quickly for changes in the environment - changes of motion parameters of target ships, so it has to return a proper solution in at most a few seconds every time it detects changes in the environment.

In the ship's safe trajectory planning problem solved by the DSS, the navigation environment consists of static and dynamic obstacles. Static obstacles such as lands or shallows are modeled as convex and concave polygons, as shown in Fig. 1. Dynamic obstacles (target ships) are modeled with the use of a ship domain. The ship domain constitutes a safety area around a ship. It ensures a safe distance between the ships during collision avoidance maneuvers.

In the current version of the DSS a hexagon domain shown in Fig. 2 is used for calculations, but other shapes (a circle, an ellipse or a parabola) and sizes of a domain can be easily adopted, according to the user's preferences.

The following assumptions were made during the problem definition process:

- target ships (TSs) are assumed to maintain their motion parameters,
- a kinematic model of the ship motion is used to describe the process (Eq. (1)), although dynamic properties of an own ship (OS) are taken into account by the consideration of the time of maneuver,
- the safe ship control process is regarded as a collision avoidance maneuver (or a sequence of maneuvers) and return to a given final point of a trajectory (especially important for restricted waters).

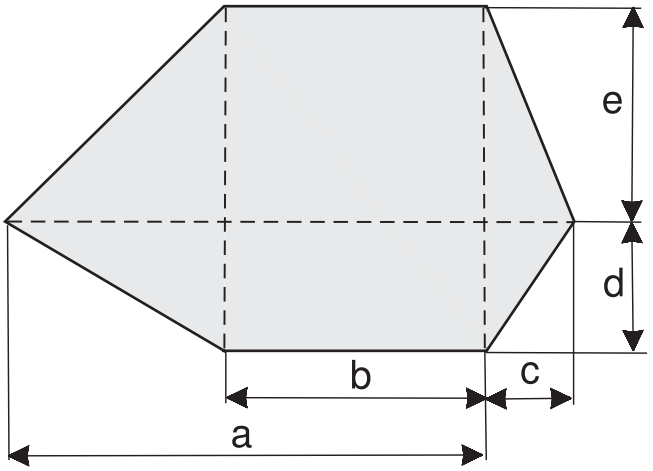

Fig. 2. Exemplary dynamic obstacle - a ship domain modeled as a hexagon.

$$\begin{aligned} \dot{x}_1 &= V \cdot \sin \Psi(t) \\ \dot{x}_2 &= V \cdot \cos \Psi(t) \\ \dot{x}_{2j+1} &= V_j \cdot \sin \Psi_j(t) \\ \dot{x}_{2j+2} &= V_j \cdot \cos \Psi_j(t) \end{aligned} \tag{1}$$

In the kinematic model of the ship motion, presented by Eq. (1), V is the speed of an OS, Ψ is the course of an OS, V_j is the speed of the jth TS, Ψ_j is the course of the jth TS, j = 1, ..., n, where n is the number of TSs in the environment, $x_1 = x$, $x_2 = y$, $x_{2j+1} = x_j$ and $x_{2j+2} = y_j$, where x and y is the longitude and latitude of the ship position.

Dynamic properties of an OS are taken into account by considering the time needed for a ship to execute the calculated maneuver - course change. The time of maneuver depends on the current rudder angle α, the speed of a ship V and the loading condition L, and is included in the evaluation process of every candidate solution - every possible trajectory.

4. DSS architecture

The system proposed in this paper constitutes an advisory system, because it is designed in order to advise the navigator in a collision situation at sea. Its task is to propose solutions of collision situations. The navigator can later use this advice or not.

The system is composed of the following modules:

- Data Input Module
- Database Module
- Trajectory Base Algorithm Module
- Solution Output Module.

The system is called the Trajectory Base Algorithm Decision Support System (TBA DSS), because it uses a database of trajectories constituting possible solutions to the problem. A general diagram of the TBA DSS is shown in Fig. 3.

The Data Input Module is an interface for data reception. Received data describe the current navigation situation. These data include the following information:

- an OS course and speed,
- TSs courses, speeds, bearings and distances from an OS,
- static obstacles (lands, islands, buoys, fairways, canals, shallows, etc.),
- visibility/weather conditions, which determine the size of a ship domain.

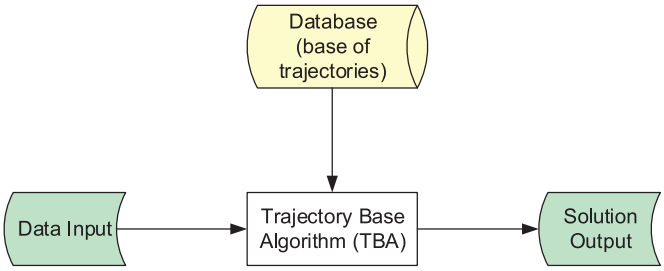

Fig. 3. A general diagram of the Trajectory Base Algorithm Decision Support System.

These data are received from the navigational equipment such as a radar with an ARPA, an AIS, an ECDIS, a GPS, an echo sounder, a gyrocompass, a log with the use of the NMEA standard.

The Solution Output Module is an interface for transmission of the output data to a display in order to present them to the user. The output data constitute:

- a graphical presentation of a safe ship trajectory composed of consecutive waypoints,
- the OS course calculated at every line segment of the trajectory,
- the length of the trajectory,
- the time taken for the ship to reach the final waypoint.

The main part of the system is the Trajectory Base Algorithm (TBA), which constitutes an engine of the decision support process. The TBA is a deterministic algorithm. Its operation principle can be explained as a process of searching a database to find the best solution of the regarded collision situation. The best solution means the solution with the minimal value of the fitness function in case of function minimization and the maximum value in case of maximization. The fitness function can optimize one criterion (the shortest path, the shortest time of arrival at the final waypoint, the smoothes path) or can consider multiple criteria. The Database Module is a base of trajectories, which constitute candidate solutions to the problem. During problem solving they are evaluated by the algorithm and the best one - the one with the lowest value of the fitness function is chosen as the final solution to the problem for the specified input data.

For comparison, in the approach based on Genetic Algorithms a random set of initial trajectories (solutions) is generated in a defined solution space. These trajectories are then modified and evaluated in order to find the final best solution. In this work a set of trajectories (solutions) is stored in the database and during problem solving is searched through in order to find the final best solution of the considered case.

Many approaches assume that the solution space is a continuous one. This assumption is of course correct and accurate, but it can lead to the impossibility to obtain a solution in a reasonable period of time.

That is why in the approach presented in this paper it is assumed that the continuous solution space is transformed into a discrete solution space, where sets of trajectories based on different rules are generated and saved in the database.

In the current version the database contains 6261 trajectories generated based upon 34 rules. It should also be mentioned that the base of trajectories can be easily developed by addition of exemplary solutions received by other methods, for example neural networks.

To present the idea of the trajectories generation process, an exemplary rule for generation of 35 trajectories is presented above as Algorithm 1. The generated trajectories are shown in Fig. 4.

Algorithm 1 Rule for generation of exemplary trajectories.

```
for j:=2 to 8 do
   for k:=1 to 5 do
      trajectory(i)=[(x,y), (x+k,j) (x,y+j) (x,y+ye)];
      i:=i+1;
   end for
end for
```

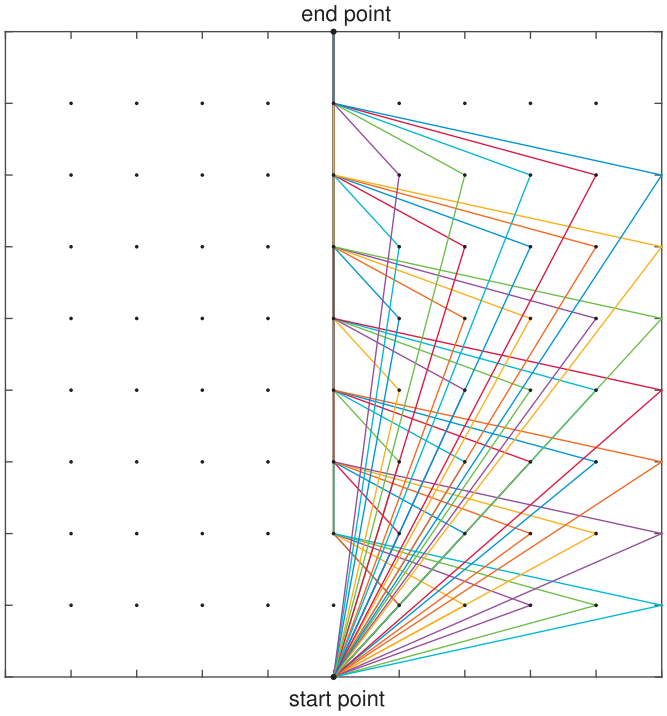

Fig. 4. Trajectories generated with the use of an exemplary rule.

5. The trajectory base algorithm

The first action of the TBA is the calculation of the relative course, speed and bearing of every TS. After that the first trajectory is retrieved from the database for evaluation.

The evaluation process begins with the division of a trajectory into a number of sections. In this way, the evaluation of a trajectory is divided into a number of steps. Later, for every step the algorithm checks whether the instantaneous positions of an OS and every TS do not collide. If the collision is detected, the trajectory is rejected and the next trajectory is collected from the database and evaluated in the same way. During this evaluation process the instantaneous position of an OS is calculated with the consideration of its dynamic properties described by the time of maneuver.

$$trajectory\_fitness = \sum_{i=1}^{k-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \rightarrow min \quad (2)$$

It should be mentioned here that trajectories in the database are sorted according to their fitness function value from the best to the worst. The best trajectory is the trajectory with the minimal value of the fitness function. The fitness function is defined as the length of a trajectory (Eq. (2)), so it uses only one criterion to evaluate solutions. The length of a trajectory is calculated as a sum of the lengths of the k−1 line segments, where k is the number of waypoints forming the trajectory. The length of a line segment connecting two waypoints is calculated using the waypoints coordinates (x and y).

The algorithm evaluates trajectories one after another, but it does not evaluate all of trajectories in the considered set as it is performed by for example the Genetic Algorithm based approach. In the actual version it evaluates trajectories sorted from the shortest to the longest one by one, but only to the moment, when the currently checked trajectory is feasible. A trajectory is feasible when it is safe and realizable with the consideration of dynamic properties of a ship.

When the algorithm finds the best trajectory not exceeding the constraints, it stops the selection process and passes the solution to the Solution Output Module for visualization. An OS course at every line segment of the trajectory is calculated and is transmitted to the Solution Output Module along with the length of the trajectory and the time of the OS passage to the final waypoint. After that, the Solution Output Module presents the solution to the user in a numerical and graphical form.

This approach is different e.g. from the method utilizing Genetic Algorithms presented in Szlapczynski and Szlapczynska (2012), where a multi-objective fitness function is used to evaluate trajectories. In the Genetic Algorithm based approach trajectory fitness is calculated based upon the length of the trajectory, fulfillment of COLREGs and violation of static and dynamic constraints. In the TBA every solution is evaluated with the use of a fitness function at the stage, when the database of trajectories is developed. Later, the TBA algorithm evaluates whether a candidate trajectory fulfills COLREGs and does not exceed static and dynamic constraints. COLREGs compliance is ensured by a proper shape and size of the TS domain.

The above described approach causes a significant reduction of the computational time, because the algorithm does not evaluate all of trajectories, but only as many as it is needed to find the best solution. The received solution constitutes the final best solution, because the trajectories have already been evaluated and sorted from the best to the worst and are retrieved from the database for feasibility evaluation in the same way - from the best to the worst.

This approach has a really significant advantage, because many trajectories can be stored in a database, but not all of them have to be evaluated during problem solving. That is especially important in the near-real time systems, such as the navigation DSS presented in this paper, where the computational time is a parameter of a vital importance.

This method has also another advantage, solutions can be sorted in the database in many different ways, using many different fitness functions to evaluate them. The user can choose his preferences after the system start-up, whether he wants to receive the shortest trajectory, the trajectory with the shortest time of passage, the trajectory with the smallest turning angles or maybe a compromise between the different criteria.

A flowchart of the TBA algorithm is presented in Fig. 5.

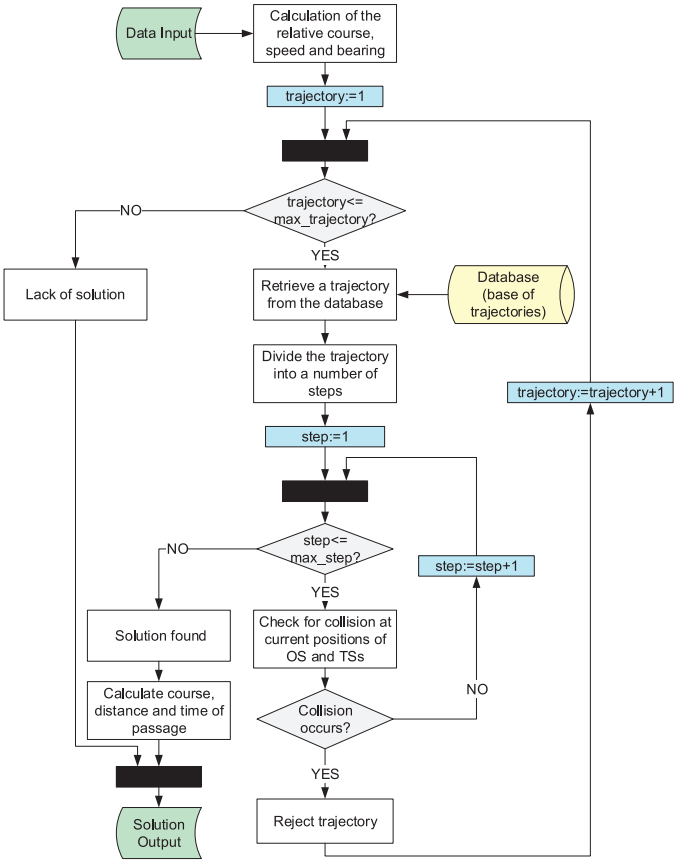

Fig. 5. A flowchart of the Trajectory Base Algorithm.

Table 2
Navigational data of case 1.

Ship	Course [°]	Speed [kn]	Bearing [°]	Distance [nm]
0	130	17.0	–	–
1	44	9.0	162	4.5
2	43	20.0	182	3.5
3	45	12.5	176	7.0

Table 3
Results of case 1.

Method	Length of trajectory [nm]	OS course [°]	Computational time [s]
ACO	10.48	175, 103, 85, 130	47–60
TBA	9.51	175, 130, 119	1.2

6. Results

In order to show advantages of this approach compared to other methods and systems, the results were compared with the results received with the use of a system utilizing a heuristic method - the Ant Colony Optimization (ACO) based approach introduced in Lazarowska (2015).The following parameters of the ACO-based ship's safe trajectory planning algorithm were used for calculations: $\tau_0 = 1$, $\rho = 0.1$, $\alpha = 1$, $\beta = 2$, iterations = 20 and ant_number = 10. The dimensions of the TS domain (Fig. 2) used in the simulation tests are: a = 1.3 nm, b = 0.6 nm, c = d = 0.5 nm and e = 0.6 nm.

Both methods were implemented in the MATLAB programming language. Calculations were carried out with the use of a PC with an Intel Core i5 M450 2.27 GHz processor, 2GB RAM, 32-bit Windows 7 Professional.

Out of many different test cases, three representative cases were chosen for the presentation in this paper.

Case 1 concerns an encounter between an OS and 3 TSs. Initial configuration for case 1 is shown in Fig. 6. Navigational data describing this scenario are listed in Table 2. Figs. 7–10 present instantaneous positions of OS and TSs. A comparison of OS trajectories calculated by the ACO-based algorithm and the TBA is shown in Fig. 11. Numerical results are compared in Table 3. The TBA solution significantly outperforms the ACO-based algorithm result. The difference in terms of the length of the trajectory is almost 1 nm (0.97 nm), but the computational time difference is even more meaningful. The TBA computational time (1.2 s) is about 50 times shorter than that achieved with the use of the ACO-based method (47–60 s). The advantage in terms of the TBA computational time

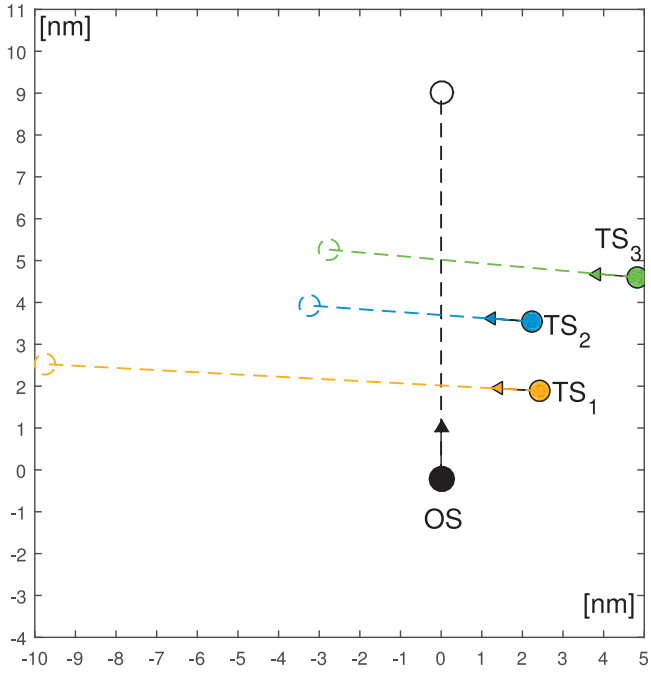

Fig. 6. Initial configuration of case 1 using TBA - OS at wp_0.

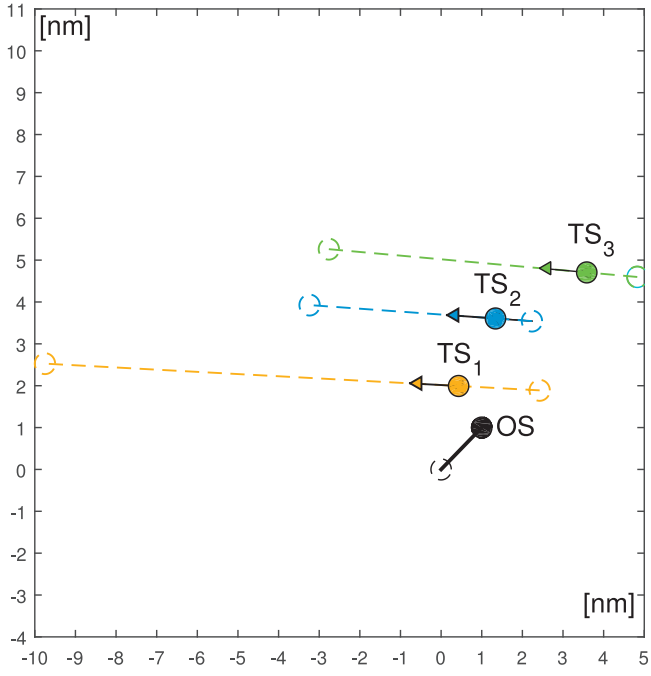

Fig. 7. Graphical solution of case 1 using TBA - OS at wp_1.

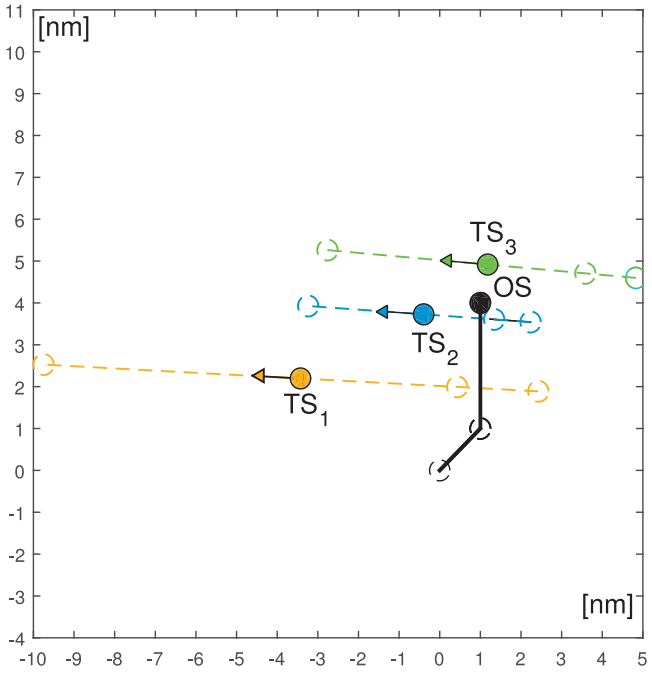

Fig. 8. Graphical solution of case 1 using TBA - OS at wp_2.

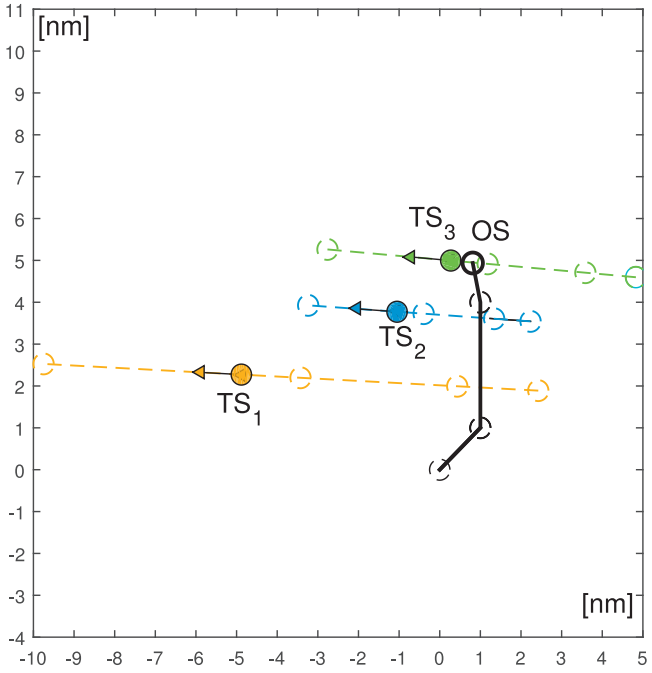

Fig. 9. Graphical solution of case 1 using TBA - OS after wp_2.

is its constant value for every run of the calculations. For the ACO-based algorithm the time is different for every run of calculations due to its probabilistic nature. The computational time presented in Table 3 was received for 100 runs of calculations.

Case 2 presents an OS encounter situation with 5 TSs. Initial configuration of this scenario is shown in Fig. 12. Table 4 includes navigational data of this case. In Figs. 13–15 temporary positions of all ships are presented. Graphical solutions returned by both algorithms (ACO and TBA) are compared in Fig. 16, while numerical results are listed in Table 5. Results of case 2 lead to the same conclusions as in case 1. The TBA trajectory is shorter by 1.08 nm compared to that of the ACO-based method and is composed of

Table 4
Navigational data of case 2.

Ship	Course [°]	Speed [kn]	Bearing [°]	Distance [nm]
0	0	12.0	–	–
1	270	7.5	45	6.0
2	180	15.0	2	8.0
3	225	16.0	25	9.0
4	275	7.5	35	8.0
5	115	5.0	345	10.0

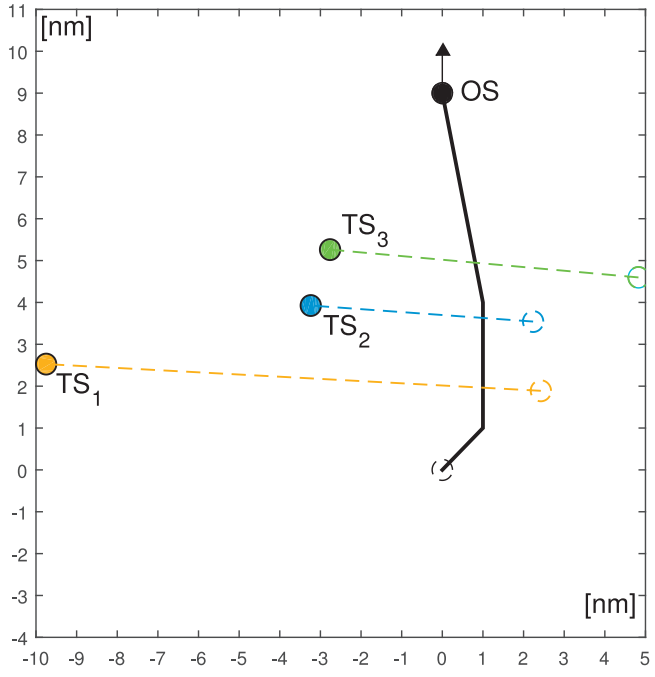

Fig. 10. Graphical solution of case 1 using TBA - OS at final wp.

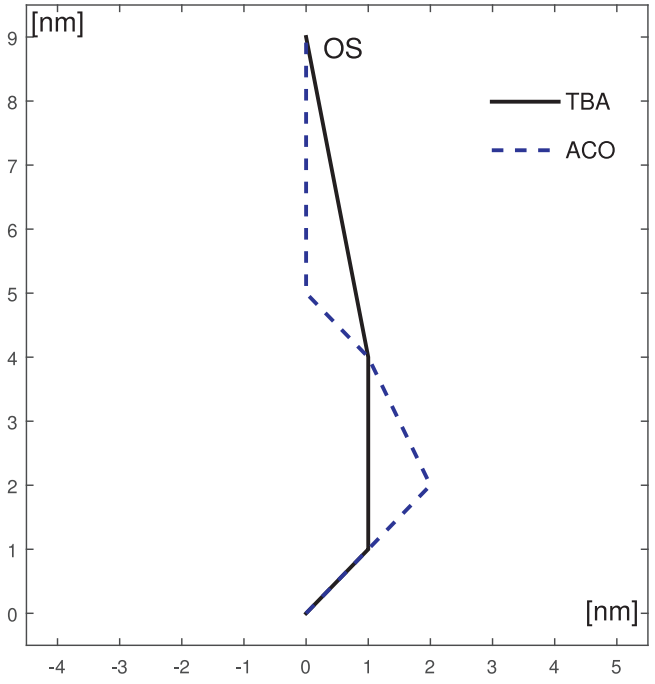

Fig. 11. Comparison of trajectories calculated by TBA and ACO for case 1.

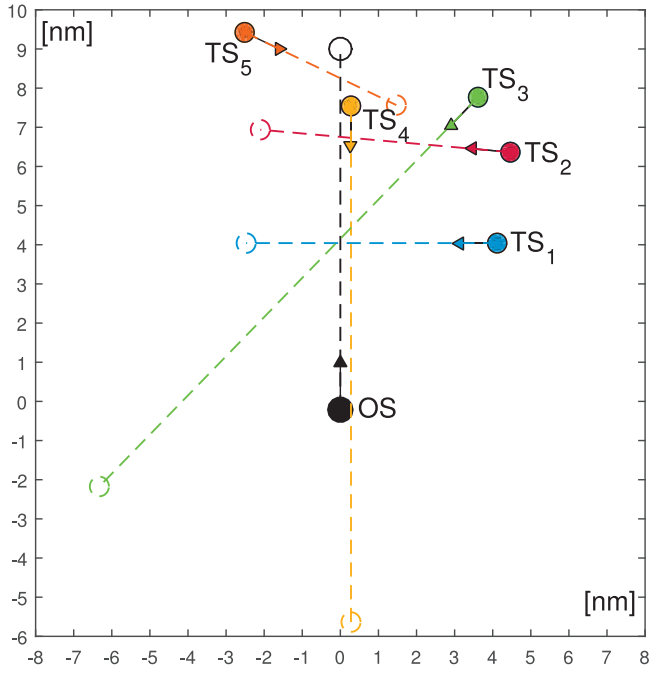

Fig. 12. Initial configuration of case 2 using TBA - OS at wp_0.

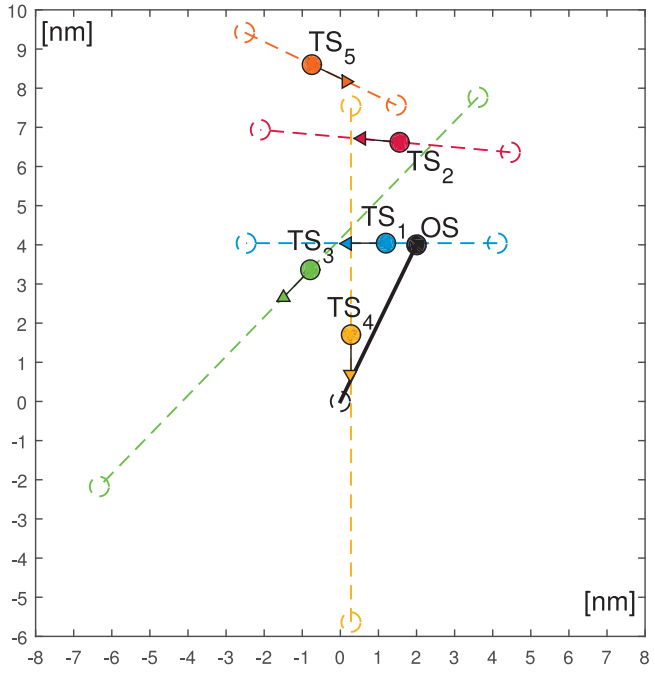

Fig. 13. Graphical solution of case 2 using TBA - OS at wp_1.

3 course changes, while the ACO-based algorithm trajectory consists of 4 maneuvers. The ACO-based method computational time is for the worst run almost 100 times longer than that of the TBA (1.77 s).

Case 3 is introduced in order to present the algorithm's capability to solve situations with the occurrence of static obstacles in the environment. This scenario includes 1 static obstacle - an island modeled in the form of a convex polygon and 3 TSs. Navigational data of this test case are listed in Table 6. Its initial configuration is shown in Fig. 17, while Figs. 18–20 present instantaneous positions of TSs, when OS is at consecutive waypoints. Comparison of the ACO-based algorithm and the TBA trajectories is presented in a numerical form in Table 7 and graphically in Fig. 21. Both trajectories are feasible, but the TBA solution is shorter by 0.38 nm and is composed of fewer waypoints (3 as compared with 4 in the ACO). Its computational time (0.76 s) is almost 80 times shorter than that of the ACO-based algorithm. Results of the ACO-based algorithm and the TBA for all cases are compared in Tables 8 and 9.

7. Discussion

The new method presented in this paper was tested on a number of test cases to check its ability of solving the path planning

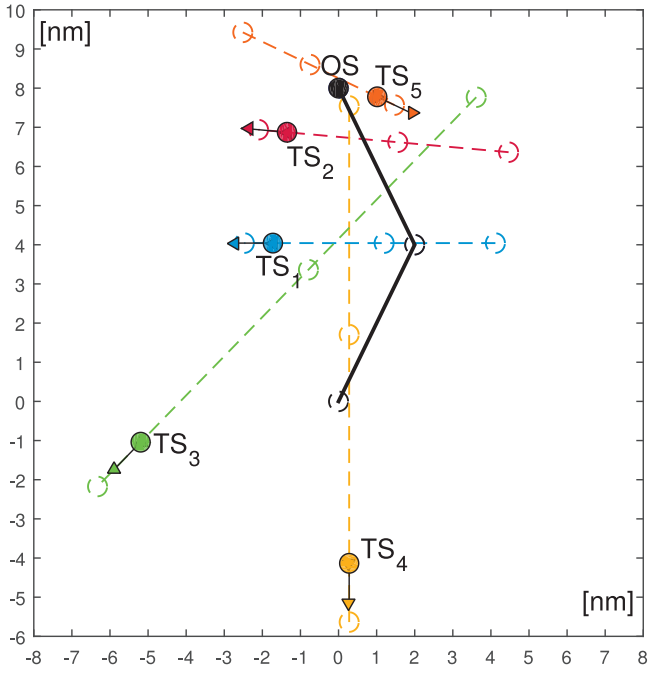

Fig. 14. Graphical solution of case 2 using TBA - OS at wp_2.

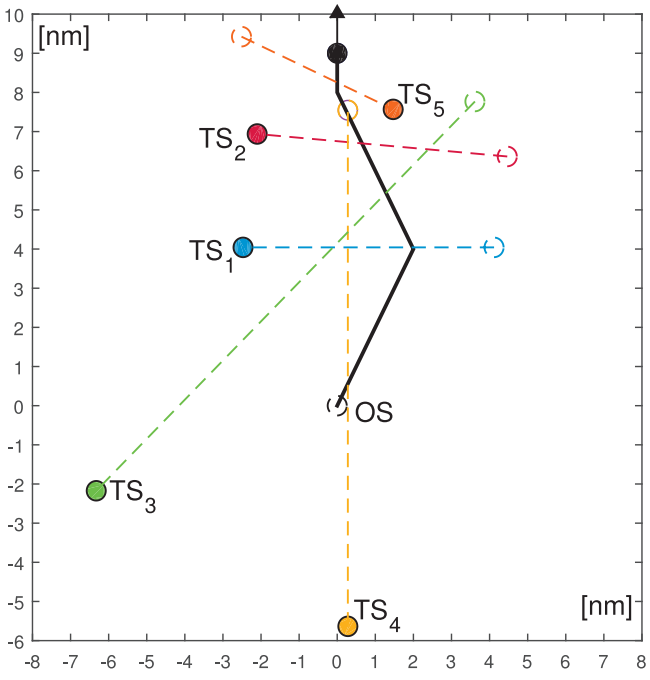

Fig. 15. Graphical solution of case 2 using TBA - OS at final wp.

problem for ships. The results of these tests were presented in the previous section. Numerical and graphical results prove the problem solving capability of the method. It is possible to solve both simple and complex navigational situations with both static and dynamic obstacles in the environment with the use of the proposed algorithm.

Solutions calculated by the proposed algorithm are practical. A safe ship path is compliant with COLREGs and is composed of a small number of line segments and waypoints. The algorithm convergence to the same solution for the same input data is assured by the deterministic nature of the method. The computational time is satisfactory, even for more complex environments it does not ex-

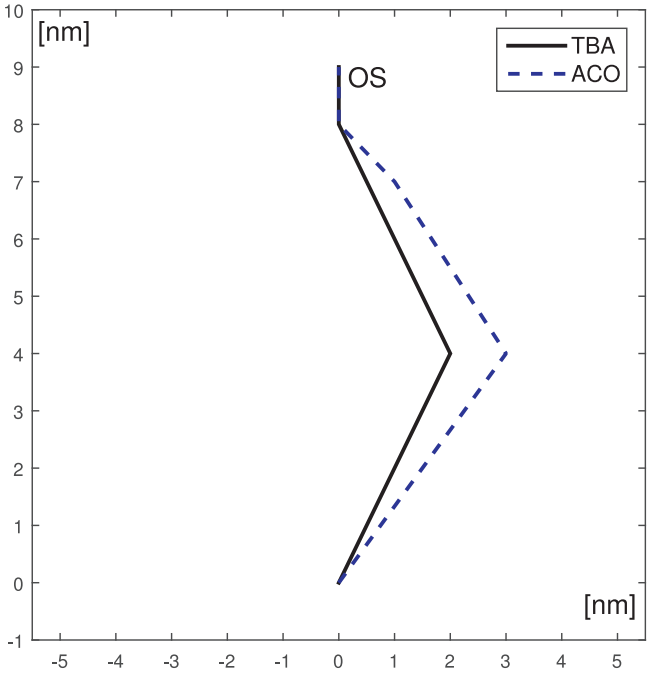

Fig. 16. Comparison of trajectories calculated by TBA and ACO for case 2.

Table 5
Results of case 2.

Method	Length of trajectory [nm]	OS course [°]	Computational time [s]
ACO	11.02	37, 326, 315, 0	100–170
TBA	9.94	27, 333, 0	1.77

Table 6
Navigational data of case 3.

Ship	Course [°]	Speed [kn]	Bearing [°]	Distance [nm]
0	0	10.0	–	–
1	246	15.0	41	6.0
2	287	4.0	25	7.0
3	191	16.0	2	4.6

Table 7
Results of case 3.

Method	Length of trajectory [nm]	OS course [°]	Computational time [s]
ACO	9.87	18, 333, 27, 333	about 60
TBA	9.49	45, 0, 352	0.76

Table 8
Comparison of the length of trajectory received by ACO and TBA.

Method	Length [nm]		
	Case 1	Case 2	Case 3
ACO	10.48	11.02	9.97
TBA	9.51	9.94	9.49

Table 9
Comparison of the computational time achieved for ACO and TBA.

Method	Computational time [s]		
	Case 1	Case 2	Case 3
ACO	47–60	100–170	60
TBA	1.2	1.77	0.76

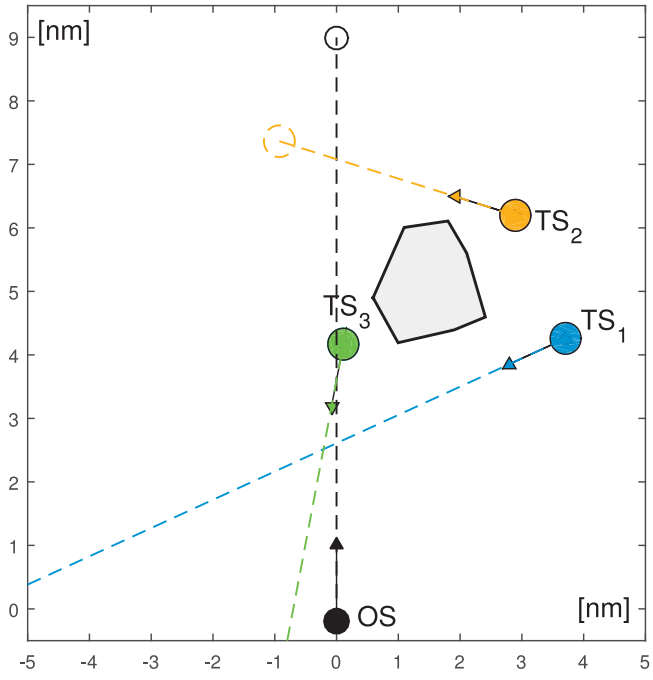

Fig. 17. Initial configuration of case 3 using TBA - OS at wp_0.

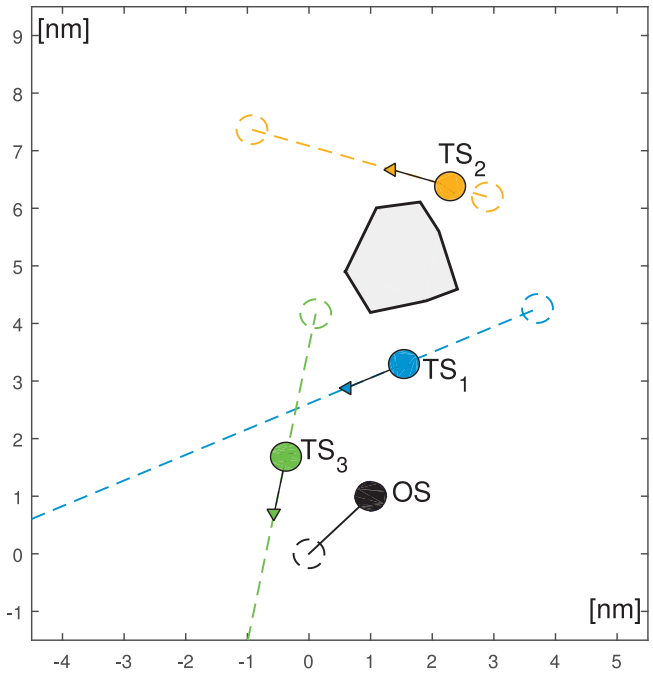

Fig. 18. Graphical solution of case 3 using TBA - OS at wp_1.

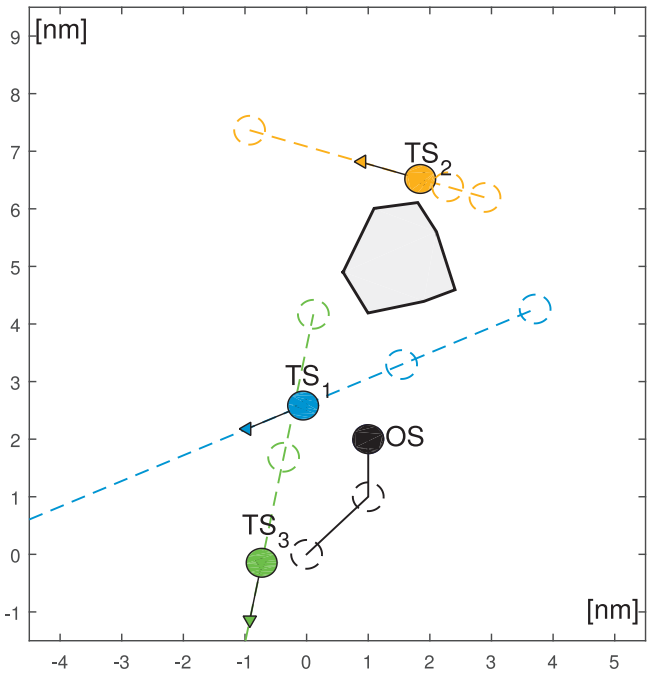

Fig. 19. Graphical solution of case 3 using TBA - OS at wp_2.

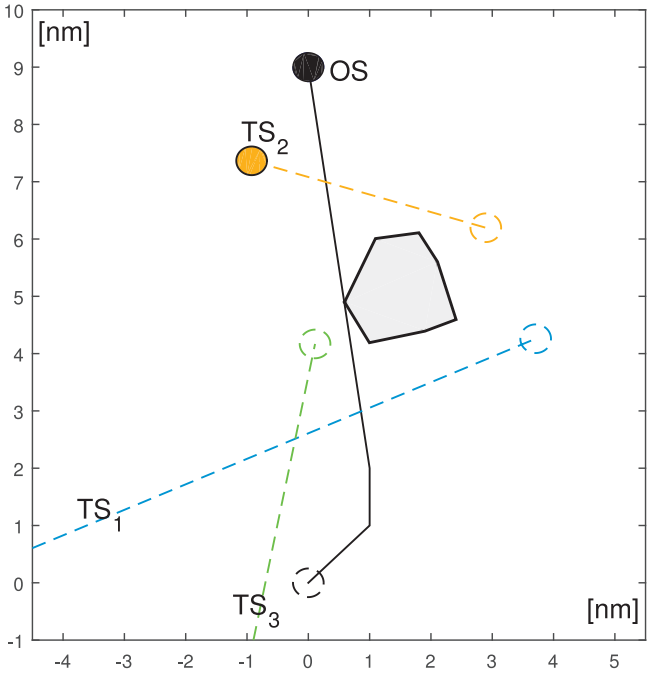

Fig. 20. Graphical solution of case 3 using TBA - OS at final wp.

ceed 2 s. It is characterized by near-real time operation - the requirement of the algorithm applicability in commercial solutions. The computational time is identical for every run of calculations for the same input data, what for a heuristic approach such as ACO is not possible to be achieved.

The algorithm performance was also compared with a heuristic method previously developed by the author of the paper - the ACO-based algorithm. The results of this comparison are shown in Tables 8 and 9. The results clearly show a significant advantage of the TBA over ACO both in regard to the solution optimality (path length) and the computational time.

The TBA was also compared with other existing methods, described in Section 2, based upon their theoretical description. The results of this comparative analysis are shown in the last column of Table 1. The new method proved to eliminate the imperfections of the existing methods such as disregarding the static obstacles and COLREGs (especially rule 8b), impossibility to solve complex situations, big time consumption and problems with convergence.

To summarize, the author thinks that the most significant advantages of the new method as compared to other approaches are:

- its simplicity,
- ability to handle static obstacles and complex dynamic environments,

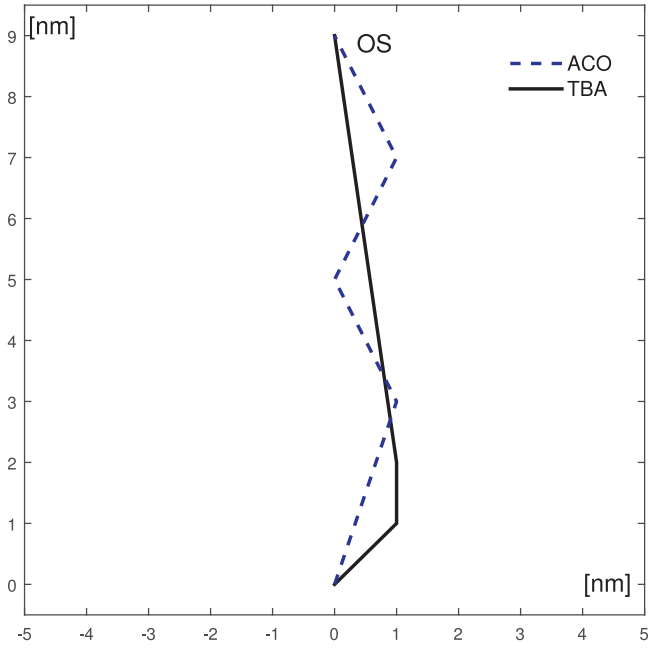

Fig. 21. Comparison of trajectories calculated by TBA and ACO for case 3.

- very short computational time,
- repeatability of the solution for every run of the same case,
- ability to extend the database in a very easy way,
- possibility to consider different preferences of the user, various weather conditions and safe distances by modification of the TS domain shape and size and change of the fitness function.

The limitations of the TBA method are:

- a predefined base of candidate solutions, sorted according to their fitness function value, which have to be searched through, what for a very complex situation can cause the computational time to grow up,
- the encountered ships motion parameters are assumed not to change during problem solving.

8. Conclusions and future works

The paper is related to the problem of planning a safe, optimal path for a ship in a complex environment, where static and dynamic obstacles occur. This is an up-to-date and important issue. A new deterministic method is proposed in the paper for finding a solution to this problem. The new method called the Trajectory Base Algorithm is introduced.

It utilizes the idea of a database of trajectories development, which is later searched through in order to find the best solution for a considered situation. Performed simulation tests proof a successful application of the proposed method. The results of the new algorithm were compared with the solutions received by one of the heuristic methods - the Ant Colony Optimization based approach. The TBA achieves better performance with regard to the optimality of solutions (shorter lengths of trajectories and smaller turning angles) and the computational time. The algorithm is applicable in commercial DSSs due to its simplicity, repeatability of solutions for every run of calculations, COLREGs compliance of solutions, consideration of static and dynamic obstacles, dynamic properties of a ship and weather conditions.

The author believes that this new approach significantly contributes to the development of intelligent DSSs applied in the field of a moving object control. The presented results prove the possibility of a system development, capable of solving a complex collision avoidance task, what opens up opportunities of an autonomous system development - an USV or an autonomous mobile robot, what shows the direction of development in the field of expert and intelligent systems.

The future research directions concerning the presented proposal are real-life experiments with the use of physical models of ships and after that in real environment on-board a ship. Another future work direction is the development of an autonomous system for an USV by the implementation of the path planning module in the motion control system of an USV. Application of the proposed algorithm to the problem of the mobile robot navigation is also a possible direction of development.

References

Campbell, S., Naeem, W., & Irwin, G. (2012). A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control, 36*(2), 267–283. http://dx.doi.org/10.1016/j.arcontrol.2012.09.008.

Kuczkowski, L., & Smierzchalski, R. (2013). Comparison of single and multi-population evolutionary algorithm for path planning in navigation situation. *Solid State Phenomena, 210*, 166–177. 10.4028/www.scientific.net/SSP.210.166

Lazarowska, A. (2015). Ship's trajectory planning for collision avoidance at sea based on ant colony optimisation. *Journal of Navigation, 68*, 291–307. doi:10.1017/S0373463314000708.

Lisowski, J. (2014a). Comparison of dynamic games in application to safe ship control. *Polish Maritime Research, 21*, 3–12. doi:10.2478/pomr-2014-0024.

Lisowski, J. (2014b). Computational intelligence methods of a safe ship control. *Procedia Computer Science, 35*, 634–643. doi:10.1016/j.procs.2014.08.145.

Liu, Y., & Bucknall, R. (2015). Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. *Ocean Engineering, 97*, 126–144. doi:10.1016/j.oceaneng.2015.01.008.

Mohamed-Seghir, M. (2012). The branch-and-bound method and genetic algorithm in avoidance of ships collisions in fuzzy environment. *Polish Maritime Research, 19*, 45–49. doi:10.2478/v10012-012-0022-6.

Naeem, W., Irwin, G., & Yang, A. (2012). Colregs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics, 22*, 669–678. doi:10.1016/j.mechatronics.2011.09.012.

Perera, L., Carvalho, J., & Soares, C. G. (2010). Bayesian network based sequential collision avoidance action execution for an ocean navigational system. *{IFAC} Proceedings Volumes, 43*, 266–271. http://dx.doi.org/10.3182/20100915-3-DE-3008.00046.

Statheros, T., Howells, G., & Maier, K. M. (2008). Autonomous ship collision avoidance navigation concepts, technologies and techniques. *Journal of Navigation, 61*(1), 129–142. doi:10.1017/S037346330700447X.

Szlapczynski, R. (2006). A new method of ship routing on raster grids, with turn penalties and collision avoidance. *Journal of Navigation, 59*, 27–42. doi:10.1017/S0373463305003528.

Szlapczynski, R., & Szlapczynska, J. (2012). Customized crossover in evolutionary sets of safe ship trajectories. *International Journal of Applied Mathematics and Computer Science, 22*(4), 999–1009. doi:10.2478/v10006-012-0074-x.

Tam, C., & Bucknall, R. (2010). Path-planning algorithm for ships in close-range encounters. *Journal of Marine Science and Technology, 15*, 395–407. doi:10.1007/s00773-010-0094-x.

Tam, C., & Bucknall, R. (2013). Cooperative path planning algorithm for marine surface vessels. *Ocean Engineering, 57*, 25–33. doi:10.1016/j.oceaneng.2012.09.003.

Tam, C., Bucknall, R., & Greig, A. (2009). Review of collision avoidance and path planning methods for ships in close range encounters. *Journal of Navigation, 62*(3), 455–476. doi:10.1017/S0373463308005134.

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

Tsou, M., Kao, S., & Su, C. (2010). Decision support from genetic algorithms for ship collision avoidance route planning and alerts. *Journal of Navigation, 63*, 167–182. doi:10.1017/S037346330999021X.

Xue, Y., Clelland, D., Lee, B., & Han, D. (2011). Automatic simulation of ship navigation. *Ocean Engineering, 38*, 2290–2305. doi:10.1016/j.oceaneng.2011.10.011.

Zak, B. (2004). The problems of collision avoidance at sea in the formulation of complex motion principles. *International Journal of Applied Mathematics and Computer Science, 14*(4), 503–514.