AN EFFICIENT GRAPH THEORY-BASED ALGORITHM FOR SHIP TRAJECTORY PLANNING

(DOI No: 10.3940/rina.ijme.2019.a2.530)

A Lazarowska, Gdynia Maritime University, Poland

SUMMARY

The research presented in this paper is dedicated to the development of a path planning algorithm for a moving object in a dynamic environment. The marine environment constitutes the application area. A graph theory-based path planning method for ships is introduced and supported by the results of simulation tests and comparative analysis with a heuristic Ant Colony Optimization approach. The method defines the environment with the use of a visibility graph and uses the A* algorithm to find the shortest, collision-free path. The main contribution is the development of an effective graph theory-based algorithm for path planning in an environment with static and dynamic obstacles. The computational time does not exceed a few seconds. Obtained results allow to state that the method is suitable for use in an intelligent motion control system for ships.

NOMENCLATURE

α	coefficient defining importance of τ
β	coefficient defining importance of η
η	visibility
ρ	pheromone evaporation rate
τ_0	initial pheromone trail
c_v	currently considered vertex
n_v	neighbouring vertex
f(v)	fitness function
g(v)	length of currently considered path
h(v)	Euclidean distance from current vertex v to final vertex E_v
D_{ip}	distance of OS to intersection point (nm)
D_j	distance of j-th TS from OS (nm)
E_v	final vertex
E	set of visibility graph edges
N	true north
N_j	bearing of j-th TS (º)
S_v	start vertex
V_j	speed of j-th TS (kn)
V	speed of OS (kn)
W	set of all of visibility graph vertices
X	longitude of OS position
Y	latitude of OS position
ACO	Ant Colony Optimization
COLREGs	International Regulations for Preventing Collisions at Sea
GNC	Guidance, Navigation and Control
GNSS	Global Navigation Satellite System
OS	own ship
TG	trajectory generator
TS	target ship
UAV	Unmanned Aerial Vehicle
USV	Unmanned Surface Vehicle
VGA	Visibility Graph-search Algorithm

1. INTRODUCTION

Autonomous navigation is a dynamically developing topic of research. The reason for that is the emergence in recent years of many new application areas, such as military and commercial robotics (land, underwater and flying robots or autonomous cars). The development of soft computing techniques, which can be observed over the last years, also contributes to the progress in autonomous navigation.

One of the main tasks in Autonomous Navigation Systems is path planning. The objective of path planning is to calculate a safe, optimal path for a moving object in a dynamic environment. The dynamic environment can be defined as the object's surroundings, where both static and dynamic (moving) obstacles occur. Similar approach to path planning can be applied for mobile robots as well as for other vehicles.

In the research presented in this paper, the marine environment was chosen as an application area. Therefore, the aim of the research was to develop a new, effective path planning method for a ship in a collision situation at sea.

The motivation for addressing this problem was to develop a new algorithm, working in near-real time, applicable in commercial solutions. Over the recent years, many approaches have been introduced, but these methods do not solve the problem definitely. They have some limitations, such as omitting consideration of static navigational constraints, problems with repeatability of results, applicability of the method (e.g. the trajectory has to fulfil specific rules), or with the achievement of low run time. The limitations of existing path planning methods are indicated in a more detailed way in the next section.

2. LITERATURE REVIEW

In order to outline the background and significance of the method presented in this paper, a review of the recent literature dedicated to ship's path planning and collision avoidance has been carried out. The path planning problem for a moving object in a dynamic environment can refer to mobile robots, Unmanned Aerial Vehicles (UAVs) as well as ships and Unmanned Surface Vehicles (USVs).

Table 1: Comparison of different existing ship trajectory planning methods

Method	Dynamic obstacles	Static obstacles	Fitness function	Computational time	Repeatability	Group	Authors	Year
VD	1 obs.	yes	length	<= 0.6 s	yes	graph	Candeloro *et al.*	2017
FMM	no	yes	multiple	< 1 s	yes	potential field	Song *et al.*	2017
DG	yes	no	risk	a few seconds	yes	deterministic	Lisowski	2016
BPF	1 obs.	yes	length	< 10 s	yes	potential field	Montiel *et al.*	2015
EEA*	no	yes	energy	ms	yes	graph	Lee *et al.*	2015
ANN	no	2 obs.	risk	–	yes	artificial intelligence	Simsir *et al.*	2014
FL	no	yes	risk	–	yes	artificial intelligence	Mohamed-Seghir	2014
CPP	5 obs.	no	$\alpha = 30°$	7 s	yes	deterministic	Tam & Bucknall	2013
A*	1 obs.	yes	length	–	yes	graph	Naeem *et al.*	2012
EA	yes	yes	multiple	< = 60 s	–	artificial intelligence	Szlapczynski & Szlapczynska	2012
PSO	1 obs.	yes	multiple	–	–	swarm intelligence	Chen & Huang	2012
APF	3 obs.	yes	length	–	yes	potential field	Xue *et al.*	2011
EA	4 obs.	no	multiple	200 - 800 s	no	artificial intelligence	Tam & Bucknall	2010

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. A recent review of path planning approaches for ships has been presented in (Fişkin *et al.*, 2018).

The classical representative of 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 & Bucknall, 2010) and (Szlapczynski & Szlapczynska, 2012). The main limitations of algorithms 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 the stochastic group are the swarm based approaches such as the Particle Swarm Optimization (PSO) presented by (Chen & Huang, 2012).

One of the most promising and very popular optimization method used for path planning is the Bacterial Potential Field (BPF) approach, introduced by (Montiel *et al.*, 2015) and Artificial Potential Field (APF) method proposed by (Xue *et al.*, 2011). (Song *et al.*, 2017) proposed the Fast Marching Method (FMM). A different deterministic approach, called the Cooperative Path Planning (CPP) algorithm is presented in (Tam & Bucknall, 2013).

The graph-search algorithms constitute another very popular subgroup of ship trajectory planning methods, 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).

A comparison of different ship trajectory planning methods was presented 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 concerning the run time, optimality of solution or constraints consideration. This was the motivation to carry out the presented research.

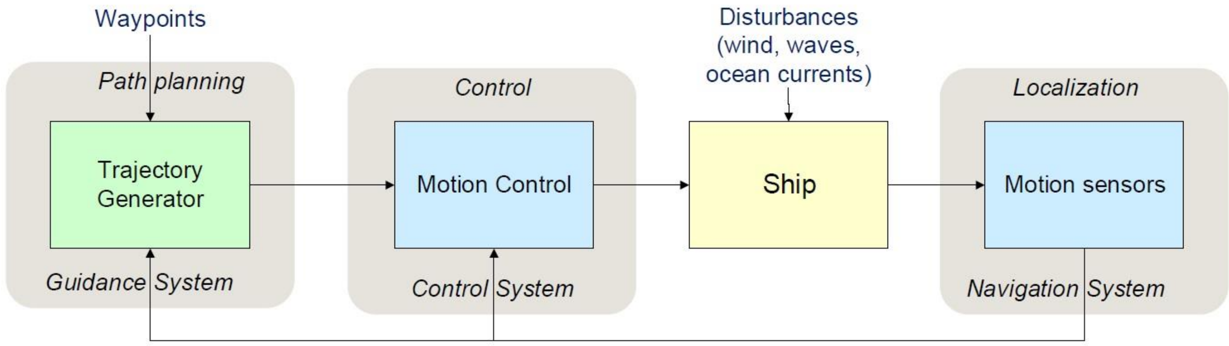

Figure 1: A general diagram of the Guidance, Navigation and Control system based upon (Fossen, 2011)

3. PATH PLANNING ALGORITHM

3.1 THE SYSTEM ARCHITECTURE

In ship control, the motion control system, called the Guidance, Navigation and Control system (GNC) is composed of three main subsystems, as shown in Figure 1. The role of the Guidance System is to calculate an optimal path (trajectory) for a ship based upon data from motion sensors (course, speed, position). The results of these computations are then fed into the Control System. The task of the Control System is to control the ship by determining appropriate control forces in order to follow the path, calculated by the Guidance System. The Navigation System determines the ship's position, course and speed with the use of the Global Navigation Satellite System (GNSS) and motion sensors such as gyrocompasses and speed logs. A similar motion system can be defined for mobile robot control. The objective of a research presented here is the development of a new path planning algorithm to be applied in the Trajectory Generator (TG) module of the Guidance System.

3.2 ASSUMPTIONS

In order to address the path planning task, the following assumptions and process constraints have been defined:

1. the motion data of all of the ships taking part in an encounter situation, which constitute the input data to the algorithm, are available;
2. the data concerning static obstacles (lands, shallows) are also available;
3. the trajectory is calculated between the predefined start and final waypoints; the start waypoint is the current ship's position received from navigational equipment, the final waypoint is the next waypoint of the ship's global path;
4. a calculated path has to enable collision avoidance with all of static and dynamic obstacles;
5. the path has to fulfil the objective defined by a fitness function (the shortest path, the smoothest path, minimal transition time);
6. the path has to be compliant with the rules specified in the International Regulations for Preventing Collisions at Sea (COLREGs) – COLREGs compliance is ensured by a proper shape and size of the target ship domain;
7. computations have to be executed in near-real time (about a few seconds) and a solution has to be repeatable for the same input data set;
8. target ships (TSs) maintain their motion parameters;
9. a kinematic model of ship's motion is applied, dynamic properties of an own ship (OS) are taken into account with the use of the time of manoeuvre parameter.

The path planning algorithm utilizes the graph theory to calculate the solution. The algorithm is composed of the following main procedures:

1. relative motion parameters calculation for every moving obstacle;
2. determination of dangerous obstacles (TSs);
3. visibility graph construction;
4. graph-search algorithm for path planning;
5. presentation of results.

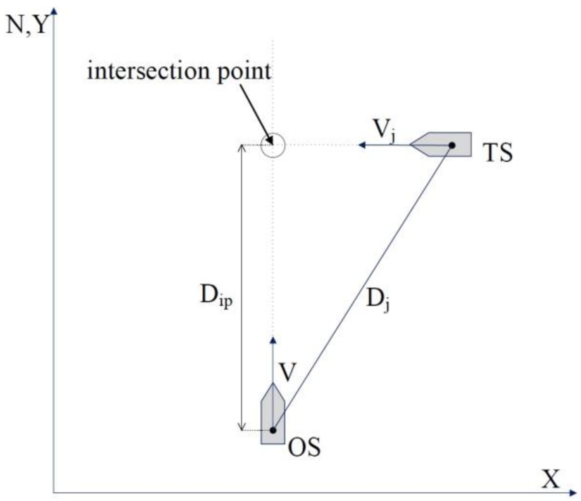

Figure 2: An intersection check between OS and TS, where D_{ip} is the distance of OS to the intersection point

3.3 DANGEROUS OBSTACLES DETERMINATION

At this stage of the algorithm, each moving obstacle (TS) is evaluated in terms of the collision risk posed for an own ship (OS). This procedure checks, whether the TS intersects its direction of movement (course) with the course of an OS, as shown in Figure 2. If the intersection exists, then the evaluated TS is marked as a dangerous obstacle and is taken into account during visibility graph construction. All of the static obstacles are considered as dangerous obstacles, so the dangerous obstacle determination procedure does not include their evaluation.

3.4 VISIBILITY GRAPH CONSTRUCTION

In path planning approaches, one of the main tasks to be solved is to define the environment of the moving objects, for which the path will be calculated. The environment can be represented as a visibility graph, a Voronoi diagram or with the use of the cell decomposition method. In the approach presented in this paper the environment representation is a visibility graph. A visibility graph is composed of vertices, which include the start and final position (waypoint) and the vertices belonging to the areas of obstacles. Static obstacles are defined as polygons. Dynamic obstacles (target ships) are defined with the use of a ship's domain term. A ship's domain is an area around a target ship that constitutes a safety margin during the process of collision avoidance. Edges of a visibility graph connect these vertices, for which the connection does not intersect the areas occupied by obstacles.

```
The Visibility Graph-search Algorithm for ship's path planning

Input W, E, S_v, E_v
for all v ∈ W connected to S_v do
        calculate h(v) = distance (v, E_v)
        f(v) = h(v)
end for
find x = min(f(v))
n_v = v(x)
path = e(S_v, n_v)
while (n_v != E_v) do
        c_v = n_v
        for all v ∈ W connected to c_v do
                calculate g(v) = path + distance (c_v, v)
                calculate h(v) = distance (v, E_v)
                f(v) = g(v) + h(v)
        end for
        find x = min(f(v))
        n_v = v(x)
        path = path + e(c_v, n_v)
end while
```

Figure 3: The pseudo code of the VGA algorithm

3.5 THE VISIBILITY GRAPH-SEARCH ALGORITHM (VGA) FOR PATH PLANNING

The Visibility Graph-search Algorithm (VGA) is a modified version of A* algorithm, adapted for use on a visibility graph. The pseudo code of the algorithm used in the presented study is shown in Figure 3, where W is the set of all of the visibility graph vertices, E is the set of visibility graph edges, c_v is the currently considered vertex and n_v is the neighbouring vertex, x is the minimal value of the fitness function determined from fitness functions values for all of the considered vertices and v(x) is the vertex with the fitness function value equal to x. The fitness function f(v) is composed of two components: g(v) and h(v). The first one g(v) is defined as the length of the currently considered path from the start waypoint (vertex) S_v to the currently considered vertex v. The second component h(v) is defined as the Euclidean distance from the current vertex v to the final waypoint (vertex) E_v.

The algorithm terminates when it reaches the final vertex (when the currently considered vertex constitutes the final one). The final step of the algorithm includes graphical and numerical presentation of the computed trajectory.

4. RESULTS OF SIMULATION TESTS

The Visibility Graph-search Algorithm (VGA) has been tested with the use of both simple (with one TS) and more complex (with up to ten TSs and static obstacles) test cases. The algorithm has been implemented in the MATLAB programming language. The solutions obtained with the use of the VGA algorithm have been compared with the results received with a heuristic method based on Ant Colony Optimization (ACO).

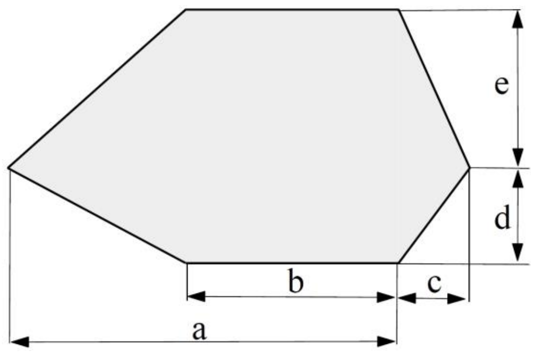

Figure 4: The TS hexagon domain

4.1 SIMULATION PARAMETERS

The dynamic constraints (TSs) were described with the use of a ship domain term. The ship domain is an area around the ship that ensures the safe distance between the ships during manoeuvres. The TS hexagon domain (Figure 4) dimensions used in the algorithms were: a = 1.0 nm, b = 0.6 nm, c = 0.4 nm, d = 0.4 nm and e = 0.6 nm. The following parameters of the ACO-based algorithm were used for calculations: $\tau_0 = 1$, $\rho = 0.1$, $\alpha = 1$, $\beta = 2$, iterations = 20 and ant_number = 10. A PC with an Intel Core i5 M450 2.27 GHz processor, 2GB RAM, 32-bit Windows 7 Professional was used to carry out the simulation tests.

4.2 SIMULATION RESULTS

Numerical results of three representative test cases have been chosen for presentation in the paper. Test case 1 represents an encounter situation between an OS and three TSs. Input data describing this scenario are listed in Table 2. In Figure 5 instantaneous positions of the ships are presented. A comparison of the paths calculated by the VGA and ACO algorithms is shown in Figure 6. Numerical results are compared in Table 3. The VGA algorithm returned a solution in 1.16 seconds, composed of two course alterations: by 20 degrees and by 37 degrees. The trajectory returned by the ACO algorithm was 0.36 nm longer than the VGA solution and the calculations lasted about 30 seconds.

Table 2: Input data of test case 1

Ship	Course [°]	Speed [kn]	Bearing [°]	Distance [nm]
0	0	12	-	-
1	270	9	45	6
2	190	11	2	4
3	90	8	315	5

Table 3: Results of test case 1

Method	Path length [nm]	OS course [°]	Run time [s]
VGA	9.5	20, 343	1.16
ACO	9.86	22, 333	about 30

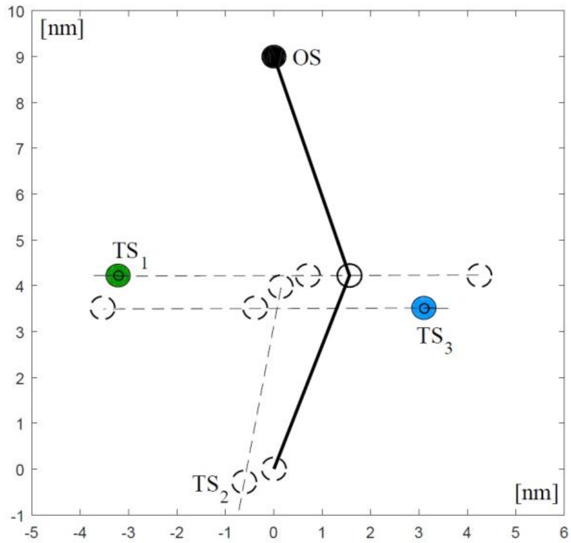

Figure 5: Solution of test case 1

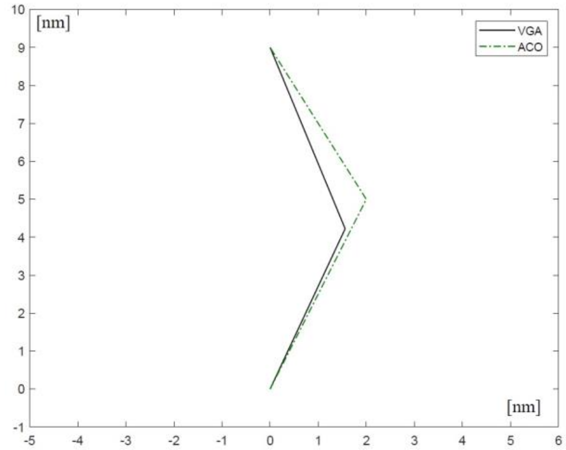

Figure 6: Comparison of VGA and ACO solutions for test case 1

Test case 2 is an encounter situation between an OS and five TSs. Input data of this test case are listed in Table 4. Figure 7 presents instantaneous positions of the ships during OS movement along the calculated path. A comparison of the solutions calculated by VGA and ACO algorithms is presented in Table 5 and Figure 8. The trajectory calculated in 1.27 seconds by the VGA algorithm consists of three course alterations: by 10 degrees, 45 degrees and 27 degrees. It is 0.05 nm shorter than the result returned by the ACO algorithm in about 60 seconds.

Table 4: Input data of test case 2

Ship	Course [°]	Speed [kn]	Bearing [°]	Distance [nm]
0	0	10	-	-
1	270	10	45	5
2	275	9	75	7
3	272	10	58	9
4	90	8	327	7
5	95	7	315	9

Table 5: Results of test case 2

Method	Path length [nm]	OS course [°]	Run time [s]
VGA	9.24	10,325,352	1.27
ACO	9.29	14, 342, 0	about 60

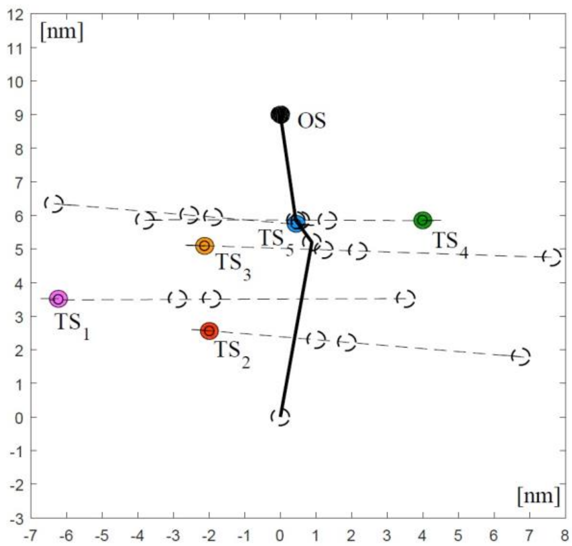

Figure 7: Solution of test case 2

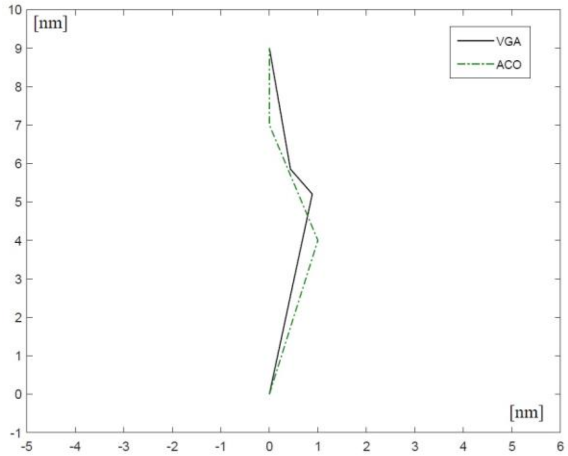

Figure 8: Comparison of VGA and ACO solutions for test case 2

Test case 3 is an encounter situation between an OS and three TSs with one static obstacle in the environment. Table 6 presents input data of this test case. In Figure 9 temporary positions of all ships are presented. Graphical solutions returned by the VGA and ACO algorithms are compared in Figure 10, while numerical results are listed in Table 7. The VGA algorithm returned a solution in 1.42 seconds. The ACO algorithm calculated a longer trajectory (by 0.81 nm) in about 60 seconds.

Table 6: Input data of test case 3

Ship	Course [°]	Speed [kn]	Bearing [°]	Distance [nm]
0	0	10	-	-
1	165	16	2	4.6
2	250	15	41	6
3	300	4	25	7

Table 7: Results of test case 3

Method	Path length [nm]	OS course [°]	Run time [s]
VGA	9.12	5, 345, 339	1.42
ACO	9.93	34, 342	about 60

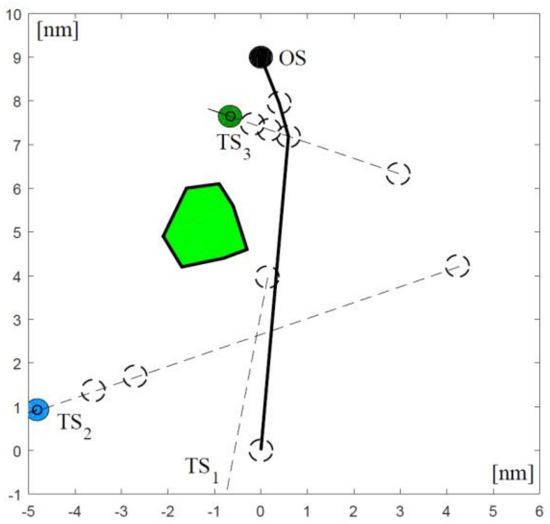

Figure 9: Solution of test case 3

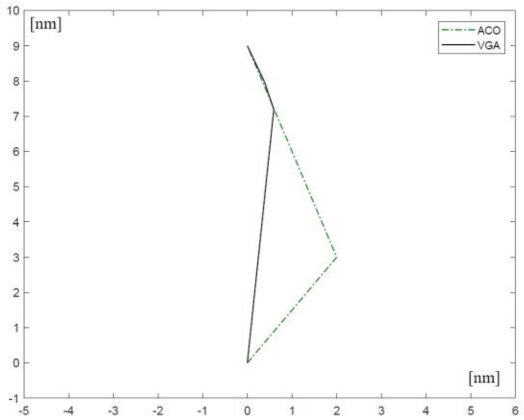

Figure 10: Comparison of VGA and ACO solutions for test case 3

4.3 DISCUSSION

The analysis of received results allows to formulate the following remarks:

1. the VGA algorithm enables calculation of a safe trajectory for a ship in a collision situation with a few static and dynamic obstacles; it is able to return a solution for both simple and more complex collision scenarios in a reasonable amount of time (a few seconds) and is therefore applicable in commercial ship motion control systems;
2. the VGA returns better solutions than ACO in terms of both the path length and run time, the comparative analysis of both algorithms demonstrates the effectiveness of the presented approach;
3. it is possible to consider a solution to the ship's path planning problem using a ranking method; the ranking method will constitute a second stage of calculations. In the first stage ACO and VGA algorithms will calculate the solution and after that with the use of a ranking method the trajectory most suited to the user's preferences will be determined and presented.

5. CONCLUSIONS

The paper introduces a new path planning method for ships. The approach utilizes a graph-search algorithm. The navigation environment is described with the use of a visibility graph. A graph-search algorithm, utilizing a modified version of A* algorithm, is applied for searching the shortest path on the graph. The method is applicable for environments with both stationary (lands, shallows) and moving (target ships) obstacles. The main advantages of the method are relatively low computational time (at most a few seconds), ability to consider static and dynamic obstacles and repeatability of solution for the same input data.

The method was also compared with a heuristic Ant Colony Optimization-based algorithm (ACO). The approach achieves better results than ACO in terms of the run time and the path length.

Summarizing, the main contribution of the paper is the presentation of a new path planning method for a moving object in a dynamic environment. The feasibility and effectiveness of the proposed method was demonstrated by the results of simulation tests. Future works planned to be carried out include tests on-board a ship.

6. REFERENCES

1. CANDELORO, M., LEKKAS, A. M. and SRENSEN, A. J. (2017) *A Voronoi-diagram-based dynamic path-planning system for underactuated marine vessels.* Control Engineering Practice, 61, 41-54.
2. CHEN, L.-J. and HUANG, L.-W. (2012) *Ship collision avoidance path planning by PSO based on maneuvering equation*. In Y. Zhang (Ed.), Future Wireless Networks and Information Systems, 675-682, Springer Publishing Company, Inc.
3. FIŞKIN, R., KIŞI, H. and NASIBOV, E. (2018) *A Research on Techniques, Models and Methods Proposed for Ship Collision Avoidance Path Planning Problem.* International Journal Maritime Engineering, 160 (A2), 187-206.
4. FOSSEN, T. I. (2011) Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons Ltd.
5. LEE, T., KIM, H., CHUNG, H., BANG, Y. and MYUNG, H. (2015). *Energy efficient path planning for a marine surface vehicle considering heading angle.* Ocean Engineering, 107 , 118-131.
6. LISOWSKI, J. (2016). *The sensitivity of state differential game vessel traffic model.* Polish Maritime Research, 31 , 14-18.
7. MOHAMED-SEGHIR, M. (2014). *The branch-and-bound method, genetic algorithm, and dynamic programming to determine a safe ship trajectory in fuzzy environment.* Procedia Computer Science, 35 , 348-357.
8. MONTIEL, O., OROZCO-ROSAS, U. and 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.
9. NAEEM, W., IRWIN, G. AND YANG, A. (2012). *COLREGs-based collision avoidance strategies for unmanned surface vehicles.* Mechatronics, 22 , 669-678.
10. SIMSIR, U., AMASYAL, M. F., BAL, M., CELEBI, U. B. and ERTUGRUL, S. (2014). *Decision support system for collision avoidance of vessels.* Applied Soft Computing, 25, 369-378.
11. SONG, R., LIU, Y. AND BUCKNALL, R. (2017). *A multi-layered fast marching method for unmanned surface vehicle path planning in a time-variant maritime environment.* Ocean Engineering, 129, 301-317.
12. SZLAPCZYNSKI, R. and SZLAPCZYNSKA, J. (2012). *On evolutionary computing in multi-ship trajectory planning*. Applied Intelligence, 37, 155-174.
13. TAM, C. and BUCKNALL, R. (2010). *Path-planning algorithm for ships in close-range encounters.* Journal of Marine Science and Technology, 15, 395-407.
14. TAM, C. and BUCKNALL, R. (2013). *Cooperative path planning algorithm for marine surface vessels.* Ocean Engineering, 57, 25-33.
15. XUE, Y., CLELLAND, D., LEE, B. and HAN, D. (2011). *Automatic simulation of ship navigation.* Ocean Engineering, 38, 2290-2305.