IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 16, 2020, accepted December 6, 2020, date of publication December 9, 2020, date of current version December 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3043539

Comparison of Discrete Artificial Potential Field Algorithm and Wave-Front Algorithm for Autonomous Ship Trajectory Planning

AGNIESZKA LAZAROWSKA
Department of Ship Automation, Gdynia Maritime University, 81-225 Gdynia, Poland
e-mail: a.lazarowska@we.umg.edu.pl

ABSTRACT The article presents a novel, efficient graph search algorithm for path planning. The algorithm was inspired by the potential field method and therefore it is called the Discrete Artificial Potential Field (DAPF) algorithm. Additional trajectory optimization algorithm is also applied as a path smoothing mechanism. The algorithm is intended for use in Intelligent Transportation Systems of autonomous ships. The algorithm was implemented in the MATLAB programming language and tested by extensive simulation experiments. Results show that the algorithm can generate a collision-free path in an environment with static and dynamic obstacles, achieving near-real run time. The algorithm was also compared with the state-of-the-art graph search algorithm – a wave-front algorithm. Obtained results demonstrate that DAPF achieves better results in terms of both solution quality and run time.

INDEX TERMS Autonomous ship, collision avoidance, decision support, path planning, save navigation.

I. INTRODUCTION

Collision avoidance is one of the most important issues in Intelligent Transportation Systems. Its application area covers, i.a., autonomous vehicles, military applications, aircraft and ship navigation.

The task of collision avoidance in all transportation systems is composed of tracking objects that can constitute obstacles, assessing the collision risk and determining a collision avoidance action that should be undertaken. The challenges that can be specified in this process are: the time to make a decision, the presence of multiple obstacles in the environment and changing strategies of dynamic obstacles [1].

Collision avoidance systems are used in many fields of application. In automotive industry such collision avoidance system is the adaptive cruise control (ACC). In aerospace applications a radar-based air traffic control (ATC) systems are used for collision avoidance. Traffic Alert and Collision Avoidance Systems (TCAS) help in maintaining minimum separation between aircrafts. In maritime transport radar systems with an automatic radar plotting aid (ARPA) are used to detect other vessels and a trial maneuver function helps in making a safe and efficient collision avoidance action. Other applications, where collision avoidance is of vital importance, are robot manipulators and autonomous vehicles.

Path planning algorithms are classified into off-line and on-line methods. An off-line algorithm calculates a path for a moving object in a known, static environment, before its motion begins. In an on-line approach, a path is planned during the object's movement towards a goal position and changes in the environment are included in the process. This type of planning is required for dynamic environments or unknown static environments. Near-real run time of an algorithm is particularly important for on-line path planning, where the path has to be re-planned, if changes in the environment were detected.

Therefore it is very important for the algorithm to be simplified as much as possible in order to obtain a problem-solving algorithm with a competitive performance in terms of both its run time and efficiency. The aim of the research presented in this article was to develop an algorithm capable of finding a collision-free path for environments with both static and dynamic obstacles in near real time. The presented research was focused on the development of a path planning algorithm for autonomous ships.

Along with safe-driving cars and autonomous robots, application of autonomous ships is becoming more and more

The associate editor coordinating the review of this manuscript and approving it for publication was Wajahat Ali Khan.

real nowadays. Interest in this technology increased rapidly in recent years, what is evidenced by a growing engagement of the market leaders providing solutions for the maritime industry related to positioning, navigation and automation of vessels. Kongsberg works on the development of a fully electric autonomous contained feeder Yara Birkeland [2]. The ship is planned to operate on a predefined route between three ports in southern Norway. The company declares readiness to achieve fully autonomous operation of the ship by 2022. In 2018 Rolls-Royce in cooperation with Finferries presented the operation of an autonomous ferry Falco [3]. This achievement was the effect of a project SVAN (Safer Vessel with Autonomous Navigation).

These are just the two examples of projects on autonomous ships. Many more research projects on that topic have been carried out recently, e.g. Maritime Unmanned Navigation through Intelligence in Networks (MUNIN) (2012-2015) [4], ReVolt (2013) [5], Advanced Autonomous Waterborne Applications (AAWA) (2015-2018) [6], Autosea (2015-2019) [7] and Autoferry (2016-2019) [8]. The development of autonomous ships can bring benefits such as increased safety and efficiency of maritime transport, better accessibility of potentially hazardous areas, a decline in piracy incidents in order to demand a ransom due to the lack of crew on board. Moreover, automated shipping lines can increase reliability of cargo transportation. The above mentioned arguments prove that autonomous ship navigation is a present and important topic and the development of path planning algorithms for ship's collision avoidance is an up-to-date area of research.

A. CONTRIBUTIONS

The main contributions of this article are:

- A novel graph search algorithm for path planning with a primary application to autonomous ships;
- A trajectory optimization algorithm is applied in order to achieve a shorter path;
- The proposed DAPF algorithm generates a shorter path achieving a shorter run time compared with the state-of-the-art wave-front algorithm.

B. RELATED WORKS

Among the recent path planning methods i.a. the following approaches have been reported in the literature with an application to mobile robots: an efficient double-layer ant colony optimization algorithm (DL-ACO) for autonomous robot navigation [9], a multi-objective firefly algorithm for mobile robot path planning [10], a Bacterial Potential Field method [11] and a hybrid Particle Swarm Optimization-Modified Frequency Bat (PSO-MFB) algorithm [12].

In other areas of applications an Improved Ant Colony Optimization approach for an automated guided vehicle (AGV) [13] and a reinforcement learning based grey wolf algorithm for unmanned aerial vehicles (UAVs) [14] were developed lately. A comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV path planning was presented in [15].

In marine applications recently introduced path planning methods are, i.a., an A* algorithm based approach [16], Fuzzy Sets and Game Theory methods [17], the Beam Search Algorithm (BSA) [18] and an evolutionary algorithm (EA) approach for ship collision avoidance [19]–[22], a reinforcement learning approach for maritime autonomous surface ships path planning (MASS) [23] and a trajectory planning approach for emergency ship to ship (STS) transfer operation [24]. Other promising methods for ship's trajectory planning include the Fuzzy logic (FL) based approach [25], [26], the Artificial Potential Field (APF) and the Modified Artificial Potential Field (MAPF) methods [27], [28], the Cooperative Path Planning (CPP) algorithm [29] and an approach based upon the Particle Swarm Optimization (PSO) method [30].

Recent path planning methods for ship's collision avoidance were analyzed and results of this comparison are presented in Table 1. Compared methods were listed chronologically. The main features of ship's trajectory planning algorithms include:

- static obstacles (lands, shallows)
- dynamic obstacles
- safe distance
- The International Regulations for Preventing Collisions at Sea (COLREGs) compliance
- run time
- solution repeatability
- real experiments
- different ships' perspectives.

As it can be seen in Table 1, most of the algorithms do not take static obstacles, such as lands or shallows, into account or if that is declared, do not present results of such test cases. All of compared algorithms take dynamic obstacles – target ships (TS) into account, most methods consider a few target ships (up to 4). APF presents results of encounter situations with up to 6 TS and MAPF shows solutions of test cases with more than 10 TS.

Another important feature of ship's trajectory planning algorithms is the method of ensuring a safe distance between the ships during collision avoidance maneuvers. Most of the methods apply a domain around a target ship, with its different shapes and sizes. Two methods, CPP and BSA, use DCPA (Distance at the Closest Point of Approach) and TCPA (Time to the Closest Point of Approach) measures to evaluate the collision risk and maintain a proper distance between the ships.

Most of the methods, except from BSA, apply mechanisms forcing the COLREGs compliance of solutions. It should be mentioned here that this constraint of the process can be regarded looking at various aspects and concentrating on different rules that should be obeyed. For example, rule 8b

TABLE 1. Comparison of Recent Path Planning Algorithms for Ship's Collision Avoidance.

Approach	Authors	Year	Static obstacles (lands, shallows)	Dynamic obstacles	Safe distance	COLREGs	Run time	Repeat-ability	Different perspec-tives	Simu-lations	Real experi-ments
Fuzzy logic (FL)	Perera *et al.*	2010	no	yes (up to 3)	circular domain around OS	if-then rules	-	**yes**	no	yes	no
Evolutionary algorithm (EA)	Tam & Bucknall	2010	no	yes (up to 4)	safety area (SA) around TS	COLREGs-influenced area (CA) surrounding SA	high (hundreds of seconds)	no	**yes**	yes	no
Artificial Potential Field (APF)	Xue *et al.*	2011	**yes**	**yes (up to 6)**	safe passing distance (circular domain around TS)	COLREGs compliant action for highest priority ship	-	-	**yes**	yes	no
A* algorithm	Blaich *et al.*	2012	no	yes (1 TS)	Goodwin's domain around TS	non-symmetric ship domain shape	**very low (milliseconds)**	**yes**	no	yes	**yes (results not presented)**
Cooperative Path Planning (CPP)	Tam & Bucknall	2013	no	yes (up to 4)	DCPA, TCPA	COLREGS priority, course change maneuvers of 30 deg	low (a few seconds)	**yes**	**yes**	yes	no
Evolutionary algorithm (EA)	Kuczkowski & Smierzchalski	2017	**yes**	yes (up to 2)	hexagon domain around TS	ship domain shape	low (a few seconds)	no	no	yes	no
Particle Swarm Optimization (PSO)	Kang *et al.*	2018	no	yes (1 TS)	ship domain around TS	reduced range of the particle position	medium (over a dozen of seconds)	no	yes for head-on scenario	yes	no
Modified Artificial Potential Field (MAPF)	Lyu & Yin	2019	no	**yes (up to more than 10 TS)**	ship domain around TS	applied in the virtual forces	**very low (milliseconds)**	**yes**	**yes**	yes	no
Beam Search Algorithm (BSA)	Karbowska *et al.*	2019	no	yes (up to 3)	DCPA, TCPA	does not fulfill COLREGs	**very low (milliseconds)**	**yes**	**yes**	yes	**yes**

states that maneuver should be large enough to be readily apparent for other ships. An interpretation of this rule by Cockcroft and Lameijer [31] is that in restricted visibility course alteration maneuvers should be at least 30 degrees, and preferably in the range from 60 to 90 degrees. For good visibility it is recommend to alter course by at least 10 degrees. It should also be mentioned that a sequence of small alterations should be avoided. Rules 13, 14 and 15 apply to different encounter situations types, such as overtaking, head-on and crossing. These rules define, which ship is the give-way vessel, that should alter course and which one is the stand-on vessel, that should keep its motion parameters. They also define that e.g. the give-way ship should change its course to starboard side and avoid crossing ahead of the other ship. In order to incorporate these rules, different methods are applied in the reviewed methods. FL approach applied COLREGs in the form of if-then rules. Some methods, e.g. APF or CPP determine the priorities of ships' maneuvers and maneuver is calculated for the ship with the highest priority. Different methods, e.g. A* algorithm or EA, use the specific shape of a target ship domain to force the COLREGs compliance of solutions. Other methods, e.g. PSO, restrict the solution space to force the COLREGs fulfillment. Another approach, used in EA and MAPF, is to apply an additional area or force surrounding target ships to achieve COLREGs compliant solutions, what is similar to the method of specifying a particular shape and size of a target ship domain. A general conclusion that can be stated based upon these analyzes is that most of the algorithms obey some of the COLREGs, e.g. rules 13-15 or rule 8b, but there exist limitations with the achievement of solutions fulfilling all of the COLREGs.

Repeatability of results is associated with the type of algorithms regarding division into deterministic and non-deterministic methods. Generally, deterministic approaches achieve repeatable results, while non-deterministic methods have limited ability concerning convergence to the same final solution for every run of the same test case. Carried out analysis confirms this statement, as EA and PSO return slightly different solutions for different runs of calculations using the same input data.

In the evaluation of algorithms in terms of their run time, methods were classified into one of four groups, depending on achieved value of this parameter:

- high (hundreds of seconds)
- medium (over a dozen of seconds)
- low (a few seconds)
- very low (milliseconds).

Three out of nine evaluated algorithms achieved very low run time, these are: A* algorithm, MAPF and BSA.

An important feature is the evaluation of algorithms from different ships' perspectives, what was presented for EA, APF, CPP, MAPF, BSA and partially for PSO (only for a head-on situation).

All algorithms were evaluated with the use of simulations, but only two works mention real experiments: A* algorithm, but results were not presented in the article and BSA, where real tests are evidenced in the article.

Analysis of advantages and disadvantages of different recently introduced ship's trajectory planning methods enables to state the following remarks:

- there is a need for the development of methods considering both static (lands, shallows) and dynamic (target ships) obstacles,
- more extensive tests, with multiple target ships should be performed and presented,
- all COLREGs should be taken into account in the algorithms,
- an algorithm should work in near-real time (preferably milliseconds) and should return repeatable solutions,
- an algorithm should be tested considering the same test case from different ships' perspectives,
- real experiments and tests with real navigational data are needed to further validate the approach.

The aim of this research was to develop a method addressing the above mentioned limitations, such as not taking static obstacles into account, lack of consideration of COLREGs, relatively high run time and lack of repeatability.

The rest of the article is organized as follows. Section II briefly defines the problem under consideration. Section III introduces the methods proposed to solve the problem. The Discrete Artificial Potential Field algorithm (DAPF) and the Trajectory Optimization Algorithm (TOA), applied in order to further optimize calculated path, are presented in this section. Section IV presents and discusses results of simulation experiments carried out in order to validate the proposed approach. These experiments include tests with both static and dynamic obstacles and a comparative analysis with a similar approach - the wave-front algorithm (WAFR). Finally, section V concisely summarizes the article.

II. THE PROBLEM DEFINITION

The problem to be solved is to find a path for a moving object between a start position and a goal position, avoiding collision with static and dynamic obstacles in the environment. The path, which enables to move between the start and goal positions without intersecting any of the obstacles, is called a collision-free path.

The problem stated above in a general way, is narrowed into a specific application area - marine navigation. In this application the problem is to find a safe trajectory for an autonomous ship, called an own ship, between its current position and defined goal position, avoiding collisions with static obstacles such as lands or shallows, and dynamic obstacles - encountered ships called target ships.

A. ENVIRONMENT DESCRIPTION

The environment is modeled as a grid-based two-dimensional configuration space C composed of $n \times m$ cells, where n is the number of horizontal cells (rows) and m is the number of vertical cells (columns), containing the obstacles' region C_{obs} and the free space C_{free}. It is defined as $C = C_{free} \bigcup C_{obs}$.

Static obstacles, such as lands, shallows, channels, waterways and other fixed navigational restrictions, are modeled using convex and concave polygons. A cell decomposition method is applied in order to decompose a navigational environment into free cells and cells occupied by obstacles.

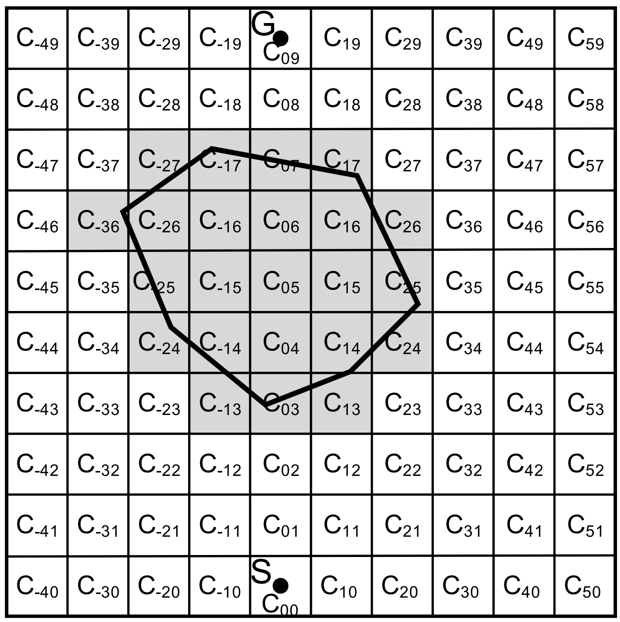

FIGURE 1. Cells assignment in DAPF algorithm.

The cell decomposition is a method of modeling the navigational environment as a two-dimensional map composed of a number of cells. In this research an approximate cell decomposition method called regular grid decomposition was applied. In this method the map of an environment is divided into a defined number of cells with equal size and cells are marked as obstacle cells and free cells. Mixed cells, partly occupied by obstacles, are marked as obstacle cells. An example of regular grid decomposition is shown in Fig. 1, where free cells are marked in white and obstacle cells are marked in grey.

Dynamic obstacles (target ships) are modeled with the use of a ship domain - an area around a target ship which is applied in order to keep a safe distance from encountered ships and static obstacles during maneuvers.

B. PROBLEM STATEMENT

The path planning problem is defined as follows:

Given a start position S and a goal position G, finding a path $p(t) : [0, t] \rightarrow C_{free}$ such that $p(0) = S$ and $p(t) = G$, where the moving object is regarded as a point mass moving at a fixed speed.

1) ASSUMPTIONS AND CONSTRAINTS

Static and dynamic obstacles constitute the constraints of the ship's trajectory planning process. Other vital constraints are the International Regulations for Preventing Collisions at Sea (COLREGs). These are the rules defining the ships' behavior during maneuvers. They define, i.e., which ship should perform a maneuver in a particular situation (a give-way ship) and which should maintain its motion parameters (stand-on ship). Rules, compared to these for road transport, are also introduced there. They define e.g. that the ship should take maneuver to the starboard side and should avoid crossing ahead of the other ship. It is also specified there, that a maneuver should be large enough to be readily apparent for other ships, that means should be at least 10 degrees for maneuvers in good visibility. The algorithm's construction has to enforce the COLREGs compliance of solutions.

The following assumptions are taken into account in the ship's trajectory planning algorithm:

- a safe trajectory is calculated to a predefined goal position of an own ship,
- an own ship is treated as a point mass moving at a fixed speed,
- target ships are assumed to maintain fixed motion parameters (course, speed)
- a safe distance between the ships during maneuvers is assured by an application of domains around target ships.

2) THE MOTION MODEL

The model of the ships' motion is the kinematic model, described by the following equations:

$$\begin{aligned} \dot{x_1} &= V \cdot \sin u(t) = V \cdot \sin \Psi(t) \\ \dot{x_2} &= V \cdot \cos u(t) = V \cdot \cos \Psi(t) \\ \dot{x_{2j+1}} &= V_j \cdot \sin \Psi_j(t) \\ \dot{x_{2j+2}} &= V_j \cdot \cos \Psi_j(t) \end{aligned} \tag{1}$$

The state variables are: $x_1 = x, x_2 = y$, $x_{2j+1} = x_j$, $x_{2j+2} = y_j$ for (j $= 1, \ldots,$ n), n is the number of tracked target ships and x and y are coordinates of the ship's position. The decision variable u is an own ship's course Ψ.

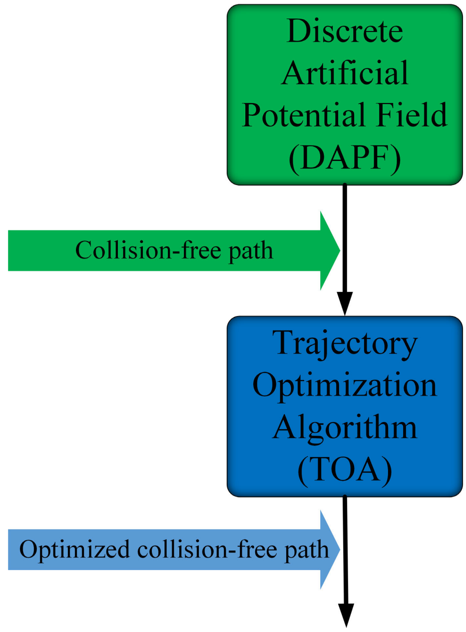

FIGURE 2. General concept of DAPF-TO algorithm.

III. THE METHOD DESCRIPTION

A method called the Discrete Artificial Potential Field with Trajectory Optimization (DAPF-TO) is proposed here for solving the above defined ship's safe trajectory planning problem. As it is shown in Fig. 2, the method is composed of two algorithms: the Discrete Artificial Potential Field (DAPF), returning a collision-free trajectory for an own ship and the Trajectory Optimization Algorithm (TOA), returning an optimized collision-free trajectory. The DAPF-TO algorithm was compared with a similar approach, the wave-front algorithm with Trajectory Optimization (WAFR-TO), which is composed of the wave-front algorithm and the Trajectory Optimization Algorithm (TOA). Results of this comparison are given in the following part of the article.

A. DAPF ALGORITHM

In order to solve the above defined problem, a method called the Discrete Artificial Potential Field (DAPF) was developed. The algorithm is inspired by the artificial potential field method (Fig. 3). It is applied as a discrete algorithm.

The following rules for the assignment of weights (potentials) to different cells were applied:

- the goal cell G has a value equal to 0;
- the cells occupied by obstacles have a value equal to infinity;
- the start cell S has a value equal to $(n - 1) \cdot k$, where n is the number of vertical cells, m is the number of horizontal cells and k is the scaling factor; in the applied version of the algorithm k is equal to 10 in order to achieve different weights for every cell in the range from 0 to about a hundred for the considered area; a different

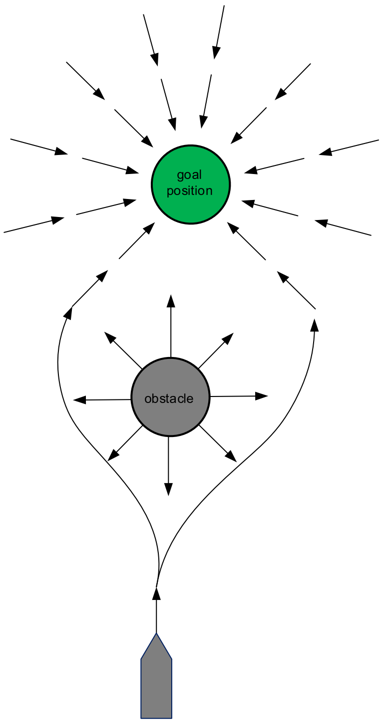

FIGURE 3. A general concept of an Artificial Potential Field.

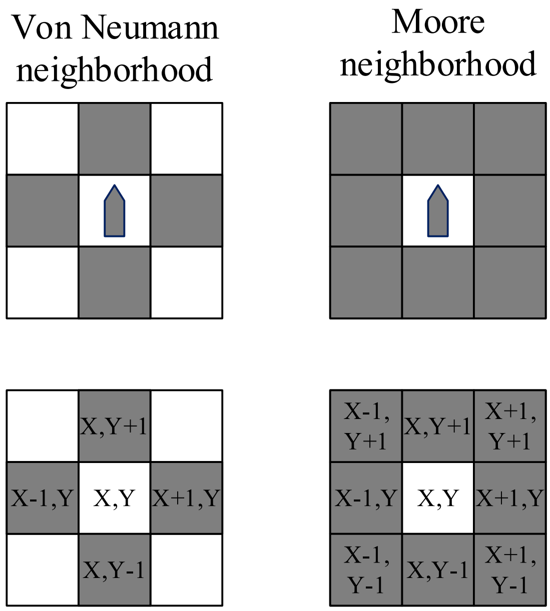

FIGURE 4. Von Neumann and Moore neighborhoods.

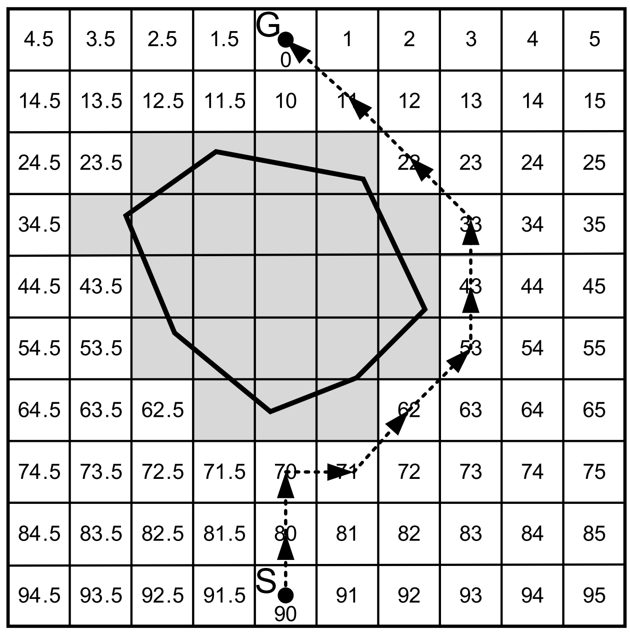

FIGURE 5. Path generated by DAPF algorithm.

assumption is admissible, but Equation 2 should also be customized then;

- other free cells c_{xy} have values according to Equation 2, where x and y are the cell coordinates;
- cells on the right side from the line segment connecting the start and goal cell have lower potentials than those on the left side in order to enforce compliance with COLREGs (rules 14 and 15).

$$\begin{array}{ll} (c_{00} - k \cdot y) & \textit{if } x = 0 \\ (c_{0y} + x) & \textit{if } x \geq 1 \\ (c_{0y} - x + 0.5) & \textit{if } x \leq -1 \end{array} \tag{2}$$

The algorithm is applied using an 8-connected grid (the Moore neighborhood). In the Moore neighborhood every cell is connected with eight adjacent cells as shown in Fig. 4. Another popular type is the Von Neumann neighborhood, with four adjacent cells connected with the central cell, as shown in Fig. 4.

In Fig. 1 cells assignment is shown for a sample environment. Fig. 5 presents weights (potentials) assignment and a path generated by DAPF for an exemplary environment.

The pseudo-code of the DAPF algorithm is given as Algorithm 1.

The general algorithm's working principle is as follows:

In every step, beginning from the start cell S, the next cell with the lowest weight is chosen from the neighboring cells, according to the Moore neighborhood. The chosen cell becomes the current cell and its weight is changed into infinity in order to prevent the algorithm from generation of loops in the path. This process continues until the goal cell G is reached.

A schematic diagram of the DAPF algorithm for ship's trajectory planning is given in Fig. 6.

The first step of the algorithm is to obtain data defining current navigational situation. These data include the course and the speed of an own ship, courses, speeds, bearings and ranges (distances from an own ship) of target ships, and

Algorithm 1 DAPF Algorithm Pseudo-Code

Input: $n, m, S, G, positions\_of\_obstacles$
Output: p

$p(0) \leftarrow S$;
while ($current\_cell \neq G$) **do**
  $pos \leftarrow min(weight[neighboring\_cells])$;
  $next\_cell \leftarrow neighboring\_cell[pos]$;
  $p \leftarrow [p; next\_cell]$;
  $weight[next\_cell] \leftarrow \infty$;
  $current\_cell \leftarrow next\_cell$;
end while
return p

positions of static obstacles. Based upon these data obtained for the current area of observation, a grid-based two dimensional configuration space composed of n × m cells is constructed and obstacles are applied on the constructed map of the environment using cell decomposition. Cells occupied by obstacles are marked as obstacle cells. The weights of occupied cells are equal to infinity. The algorithm is capable of taking into account both static and dynamic obstacles. It is assumed that the obstacles are known beforehand and new dynamic obstacles will not be detected during the solution construction process. It is also assumed that target ships do not change their motion parameter during the search for a solution. It is important to achieve a short run time in order to repeat calculations every few seconds using updated input data. Such approach enables to update the solution when a change in motion parameters of any of the target ships will be detected.

For target ships, at first dangerous dynamic obstacles are detected. A dangerous target ship is a target ship, whose course intersects with the course of an own ship. When a target ship is classified as a dangerous target, an own ship's distance from the point of trajectories intersection (Predicted Point of Collision - PPC) is calculated. Next step is the calculation of time, after which an own ship will be present at the point of trajectories intersection (PPC). After that the position of a target ship at the moment, when an own ship will be present at the PPC, is calculated, considering constant motion parameters of the ships. A target ship domain is then placed in the calculated position. After that cells that are placed inside target ship domains, are marked as cells occupied by obstacles and the weight of these cells is changed into infinity.

After that the solution construction process is applied. At first it is checked, whether the current cell is not equal to the goal cell G. Then, adjacent cells of the current cell are selected according to the Moore neighborhood. Afterwards, the next cell with the lowest weight is chosen from the adjacent cells. This cell is saved as a part of the final path. Its potential is changed into infinity in order to prevent the algorithm from getting stuck in a dead point.

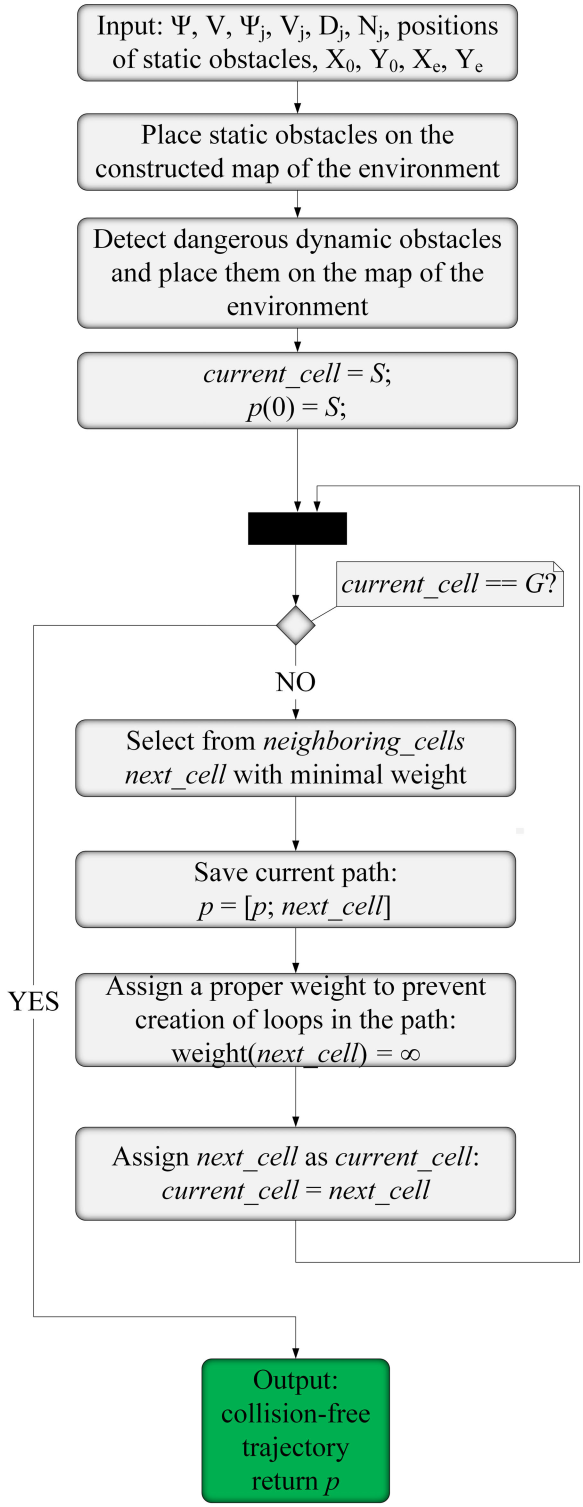

FIGURE 6. Schematic diagram of the DAPF algorithm.

The next cell becomes the current cell and the whole process repeats until the goal cell will be reached by the algorithm.

B. TRAJECTORY OPTIMIZATION

The trajectory optimization algorithm is applied in order to obtain a shorter and smoother path.

The pseudo-code of the Trajectory Optimization Algorithm (TOA) is given as Algorithm 2.

Algorithm 2 Trajectory Optimization Algorithm Pseudo-Code

Input: *path_cells*, S, G, *positions_of_obstacles*
Output: *path*

```
connect ← 1; i ← 1;
while (path_cells > min_path_cells) do
    connect nodes (i) and (i + 2) (p1 with p2);
    for (obstacle = 1; obstacle <= obstacle_num; obstacle++) do
        for (j = 1; j <= j_max; j++) do
            connect nodes (j) and (j + 1) (p3 with p4);
            if ((p1, p2) intersects (p3, p4)) then
                connect ← 0; break;
            end if
        end for
        if (connect = 0) then
            break;
        end if
    end for
    if (connect = 1) then
        connect p1 and p2
    else
        do not connect p1 and p2
        connect ← 1;
    end if
    i ← i + 1;
end while
return path
```

TOA working principle is based upon the check of intersection between two line segments – one belonging to the collision-free path and the other constituting a boundary of an obstacle.

It should be stated here that obstacles are taken into account as polygons composed of vertices (nodes) and edges. In this algorithm the main procedure is the check of intersection between two line segments, as it was stated above. One line segment is composed of a line connecting two nodes (centers of cells) constituting a part of an own ship collision-free trajectory calculated by the DAPF algorithm. These are not the two consecutive nodes of the path (i and $i+1$), but the node i and the node $i+2$. These nodes are marked as $p_1 = (x_i, y_i)$ and $p_2 = (x_{i+2}, y_{i+2})$. If such connection between p_1 and p_2 is possible, that will cause the achievement of a shorter final solution, due to the elimination of node $i+1$. Such line segment connecting nodes p_1 and p_2 is checked for intersection with any of the edges constituting boundaries of any of the obstacles. In this algorithm the nodes connecting the currently considered edge of an obstacle are marked as $p_3 = (x_j, y_j)$ and $p_4 = (x_{j+1}, y_{j+1})$. If an intersection between (p_1, p_2) and (p_3, p_4) is not detected for any of the obstacles, it means that nodes p_1 and p_2 can be connected. After that the algorithm executes the check procedure for another pair of nodes constituting a part of the DAPF collision-free trajectory. When the total number of cells is equal to the minimal number of cells (the variable *min_path_cells*) defined for the final path or more connections simplifying the path are not possible, the algorithm terminates and the DAPF-TO path is returned as an output.

A schematic diagram of the TOA is given in Fig. 7.

In Fig. 8 a path generated by DAPF-TO is presented for a sample environment.

C. WAFR ALGORITHM

The wave-front algorithm (WAFR) was chosen for a comparison with DAPF as an operation principle of these two algorithms is similar. The wave-front propagation algorithm is a method developed for path planning. Similarly as DAPF it can be applied on discrete grid maps. Its operation is inspired by the propagation of a wave-front starting from the goal cell and moving towards the start cell (Fig. 9).

The weights are assigned to cells according to the increasing value from the goal cell towards the start cell in a way simulating the wave propagation. After that a path is constructed by choosing the next cell with the lowest weight from adjacent cells until the goal cell has been reached. When more than one cell with the lowest weight exist, the first element of an array storing neighboring cells with minimal weight is chosen as the next cell.

The following rules for the assignment of weights to different cells were applied:

- the goal cell G has a value equal to 0;
- the cells occupied by obstacles have a value equal to infinity;
- the start cell S has a value equal to $(n-1)$, where n is the number of vertical cells, m is the number of horizontal cells;
- other free cells c_{xy} have values according to Equation 3, where x and y are the cell coordinates;
- the weight of every cell is not unique, some cells have the same weights.

$$\begin{array}{ll} (c_{00} - y) & if\ x = 0 \\ (c_{0y} + x) & if\ x \geq 1 \\ (c_{0y} - x) & if\ x \leq -1 \end{array} \tag{3}$$

The pseudo-code of the WAFR algorithm is given as Algorithm 3.

The operation principle of WAFR algorithm for ship's trajectory planning is similar to that of DAPF algorithm, except from the weights assignment method and the selection of a neighboring cell.

Fig. 10 shows the weights assignment for a wave-front algorithm.

In Table 2 DAPF and WAFR algorithms comparison is presented, showing their common features and differences.

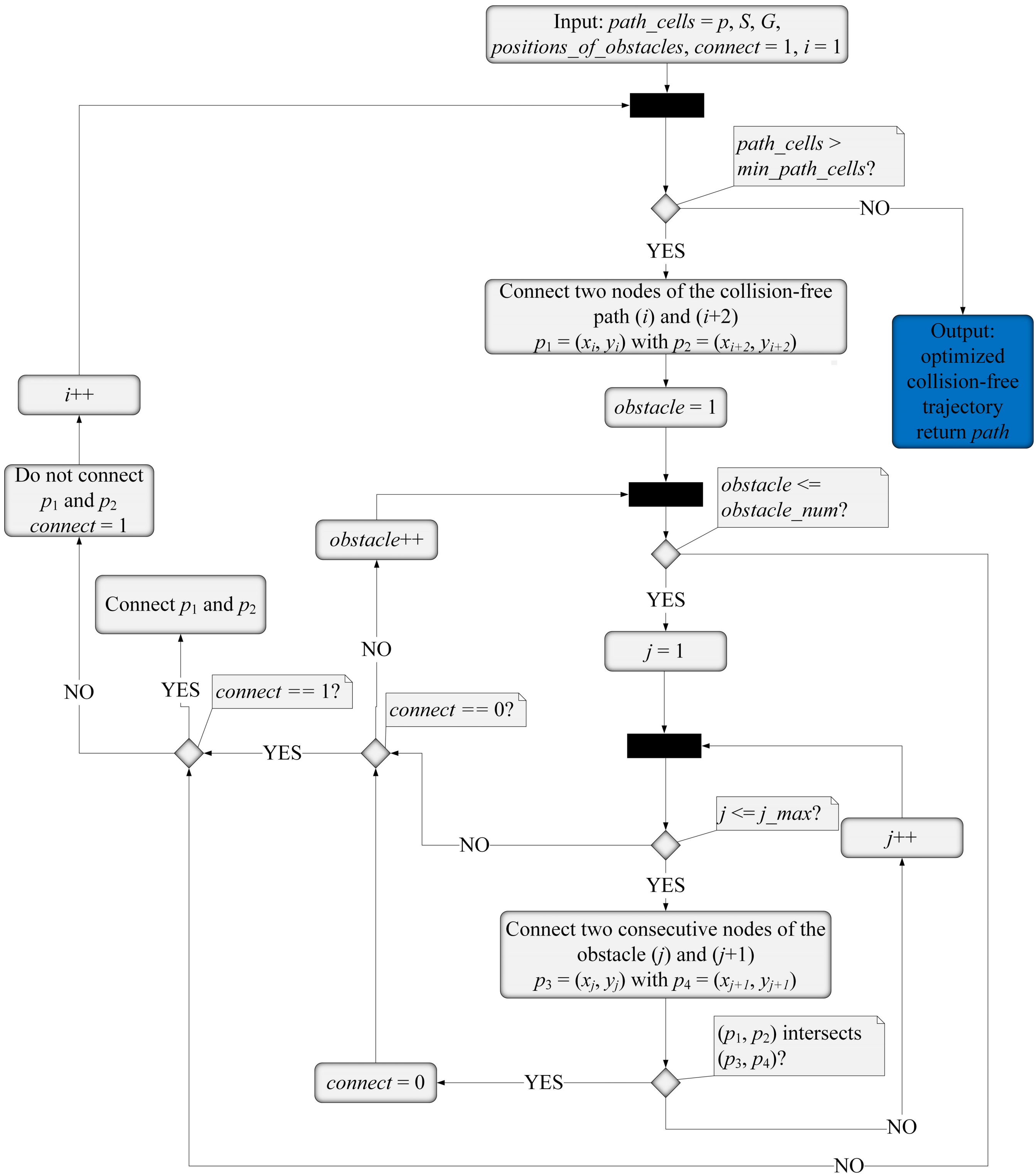

FIGURE 7. **Schematic diagram of the TOA.**

As it can be seen in Table 2, both algorithms take obstacles into account in the same manner. Cells occupied by obstacles have weights equal to infinity in both approaches. The solution construction principle is also common for the two algorithms, meaning the selection of the next cell with the lowest weight from neighboring cells until the arrival at

Algorithm 3 WAFR Algorithm Pseudo-Code

Input: $n, m, S, G, positions\_of\_obstacles$
Output: p

```
p(0) ← S;
while (current_cell ≠ G) do
    pos ← min(weight[neighboring_cells]);
    if (length[pos] > 1) then
        pos ← pos[1];
    end if
    next_cell ← neighboring_cell[pos];
    p ← [p; next_cell];
    weight(next_cell) ← ∞;
    current_cell ← next_cell;
end while
return p
```

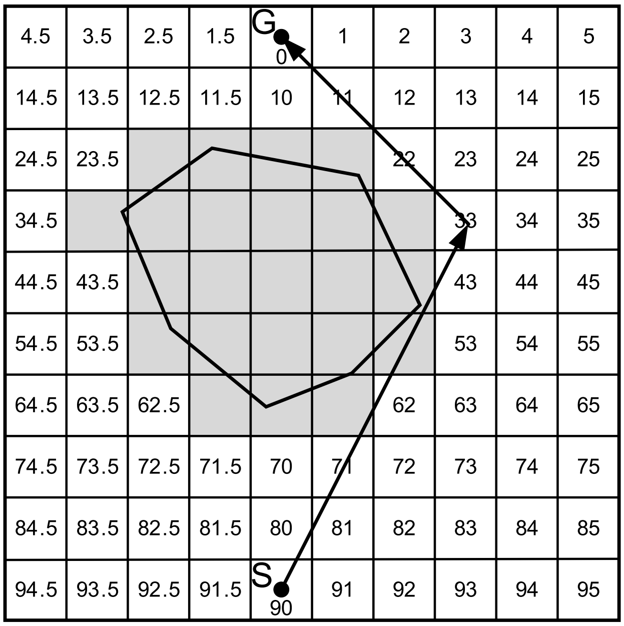

FIGURE 8. Path generated by DAPF-TO algorithm.

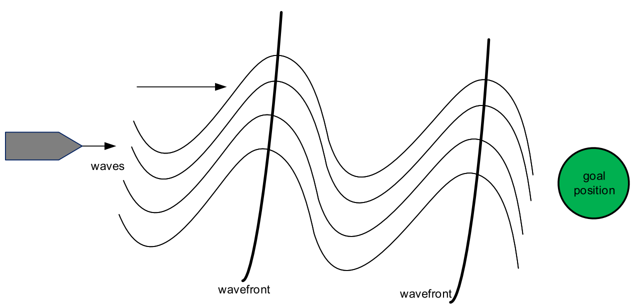

FIGURE 9. A general concept of a wave-front.

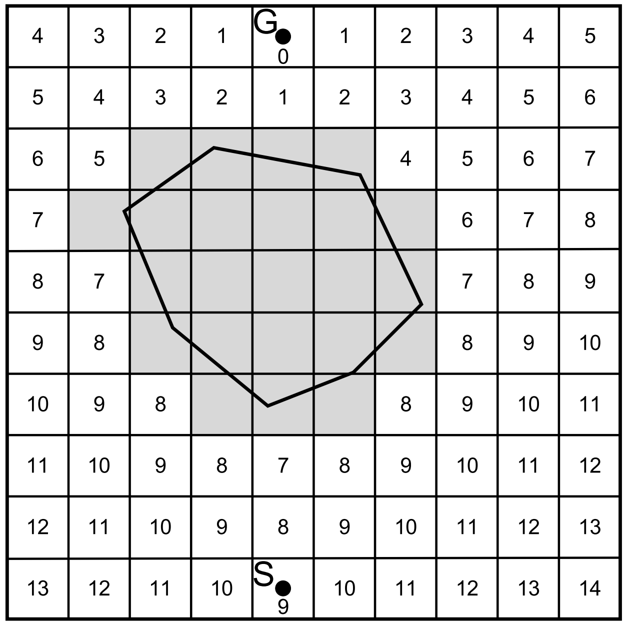

FIGURE 10. Weights assignment in WAFR algorithm.

TABLE 2. A comparison of DAPF and WAFR Algorithms Applied to Ship's Trajectory Planning.

Algorithm	DAPF	WAFR
Common features	a way of taking obstacles into account	
	solution construction principle	
	Moore neighborhood applied for determination of adjacent cells	
	cells occupied by obstacles have weights equal to infinity	
Differences	weights assignment inspired by a potential field	weights assignment inspired by a wave-front propagation
	every free cell has a different weight	some free cells have the same weights
	cells assignment forcing COLREGs fulfillment	cells assignment living possibility to choose equivalent paths without enforcement of COLREGs fulfillment

the goal position. Both algorithms apply the Moore neighborhood during the determination of neighboring cells of the current cell. The difference in the selection of the next cell results from the dissimilarity of the cells assignments process. In DAPF every free cell has a unique weigh and weights are assigned in a way aiding in the achievement of a COLREGs compliant solution. In WAFR some free cells have the same weights, what in some cases may cause a possibility for the algorithm to choose an equivalent path, not conforming to COLREGs.

IV. EXPERIMENTS AND DISCUSSION

In order to verify the algorithm's performance two types of experiments were carried out: tests with static obstacles and with dynamic obstacles in the environment.

A. EXPERIMENTS WITH STATIC OBSTACLES

In the first part of tests four different maps were chosen for the presentation in this article. The maps are a square area

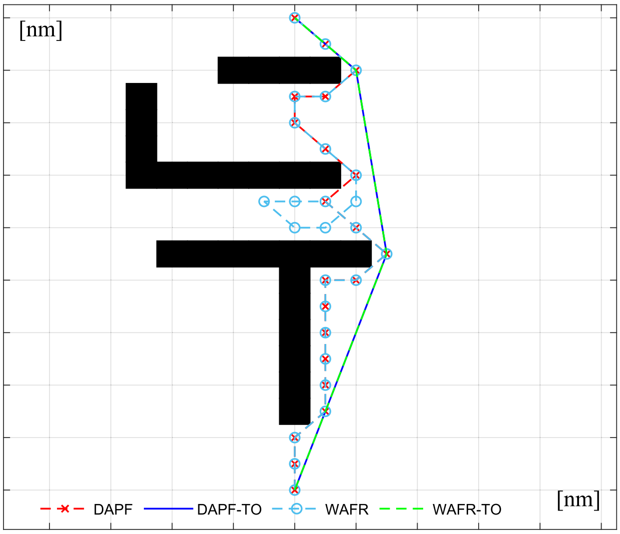

FIGURE 11. Paths generated by DAPF-TO and WAFR-TO for environments with static obstacles - test case 1.

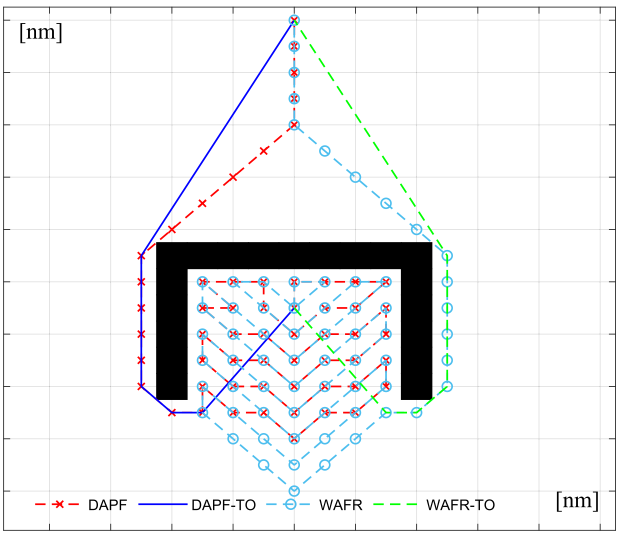

FIGURE 12. Paths generated by DAPF-TO and WAFR-TO for environments with static obstacles - test case 2.

composed of 20 × 20 cells with different number, size and shape of static obstacles, as shown in Fig. 11-14. The algorithm was applied as an 8-connected grid (Moore neighbourhood) version.

The results were evaluated in terms of the solution quality and the run time of the algorithm. The path length was used as a measure of solution quality. The results were also compared with the results of the wave-front algorithm (WAFR) [32], [33]. Pure versions of both algorithms (DAPF and WAFR) as well as version with applied trajectory optimization (DAPF-TO and WAFR-TO) were compared.

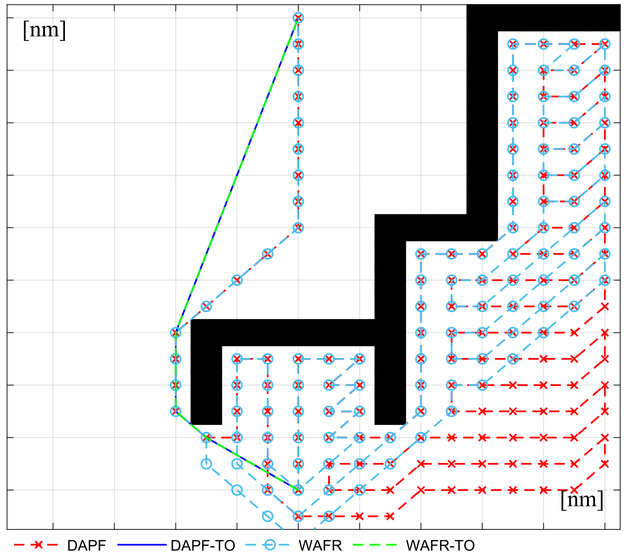

FIGURE 13. Paths generated by DAPF-TO and WAFR-TO for environments with static obstacles - test case 3.

TABLE 3. Simulation Results of DAPF and WAFR for 8-Connected Grid With Static Obstacles.

Algorithm	Test case	Path length [nm]	Optimized path length [nm]	Run time pure alg. [s]	Run time with optimization [s]
DAPF	1	**12.0711**	**9.6932**	**0.18**	**0.35**
WAFR	1	14.7782	**9.6932**	0.21	0.44
DAPF	2	**33.8848**	**11.3549**	**0.18**	**0.52**
WAFR	2	43.4411	**11.3549**	0.19	0.64
DAPF	3	82.6274	**10.3344**	**0.16**	**0.98**
WAFR	3	**71.0624**	**10.3344**	0.23	1.02
DAPF	4	**26.6924**	**24.7635**	**0.16**	**0.89**
WAFR	4	41.0416	**24.7635**	0.27	1.28

Numerical results are given in Table 1. The results include the path length of pure algorithms and algorithms with trajectory optimization and the run time of pure algorithms and algorithms with trajectory optimization applied. Run time is the average time of 100 runs of the algorithm. Calculations were performed in MATLAB R2020a on a PC with a 2.27 GHz Intel Core i5 processor and 2 GB of RAM.

It can be noticed in Fig. 11-14 that both algorithms return the same path or equivalent paths as for test case 2, when trajectory optimization is applied. Without trajectory optimization the paths returned by DAPF are shorter for three out of four test cases compared here. It can also be noticed that the run time of DAPF and DAPF-TO is shorter for all test cases than that achieved by WAFR and WAFR-TO. The results demonstrate that DAPF achieves better results in terms of both solution quality and run time.

B. EXPERIMENTS WITH DYNAMIC OBSTACLES

In the second part of experiments the DAPF-TO and WAFR-TO algorithms were tested in an environment with dynamic obstacles. Four different test cases were chosen for

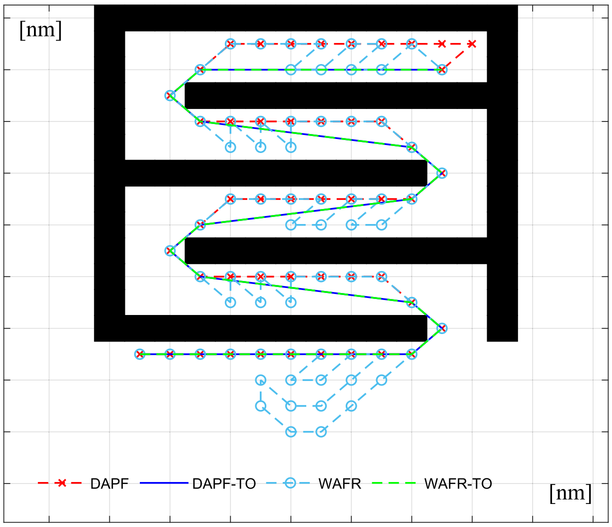

FIGURE 14. Paths generated by DAPF-TO and WAFR-TO for environments with static obstacles - test case 4.

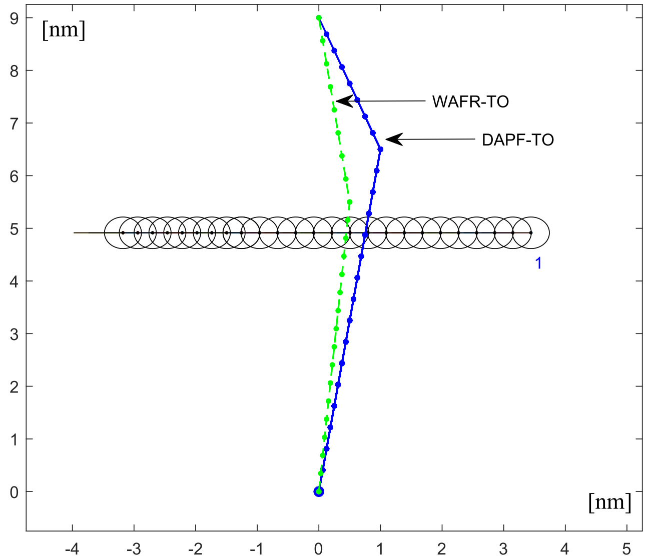

FIGURE 15. Paths generated by DAPF-TO and WAFR-TO for an environment with dynamic obstacles - test case 1 with 1 dynamic obstacle.

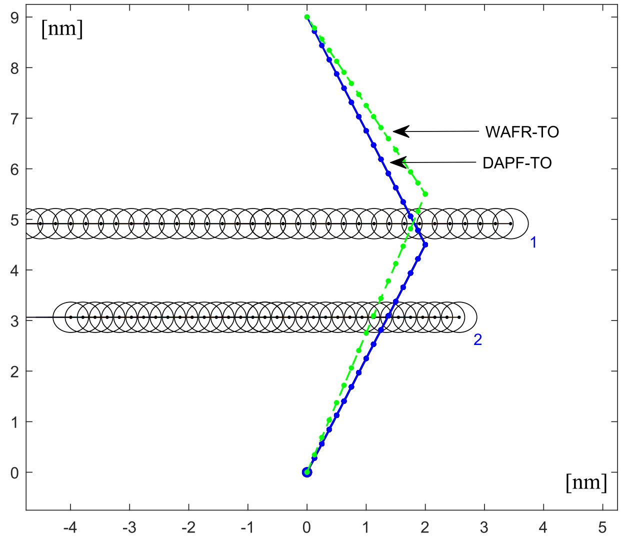

FIGURE 16. Paths generated by DAPF-TO and WAFR-TO for an environment with dynamic obstacles - test case 2 with 2 dynamic obstacles.

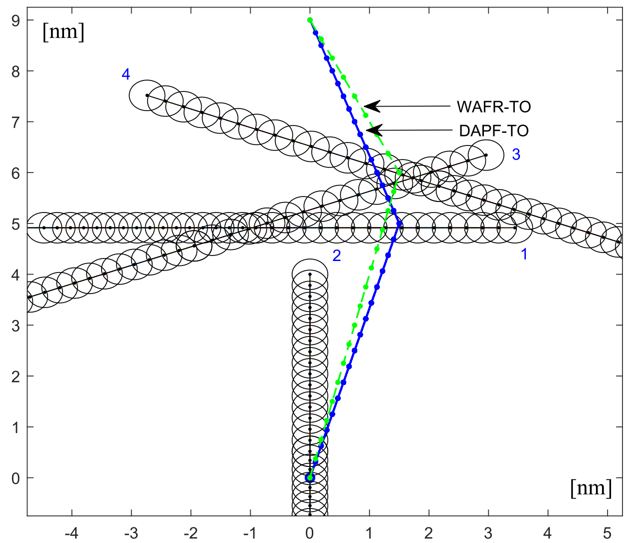

FIGURE 17. Paths generated by DAPF-TO and WAFR-TO for an environment with dynamic obstacles - test case 3 with 4 dynamic obstacles.

the presentation in the article, which are encounter situations with different number of dynamic obstacles. Obstacles are characterized by their initial position, orientation and speed.

The results shown in Fig. 15-18 demonstrate that for all cases both algorithms returned a collision-free path. Numerical results are given in Table 4. They include the path length of the algorithms with trajectory optimization in nautical miles and the run time of the algorithms with trajectory optimization in seconds. Run time is the average time of 100 runs of the algorithm.

As it can be seen in Table 4, for three out of four test cases DAPF-TO returned shorter trajectories than WAFR-TO. For all the presented test cases DAPF-TO achieved a slightly shorter run time. The results confirm conclusions resulting from the analysis of test cases with static obstacles, that DAPF-TO achieves better results in terms of both solution quality and run time for most test cases.

Advantages of the algorithm compared to other recently introduced ship's trajectory planning algorithms are: the possibility of taking into account both static and dynamic

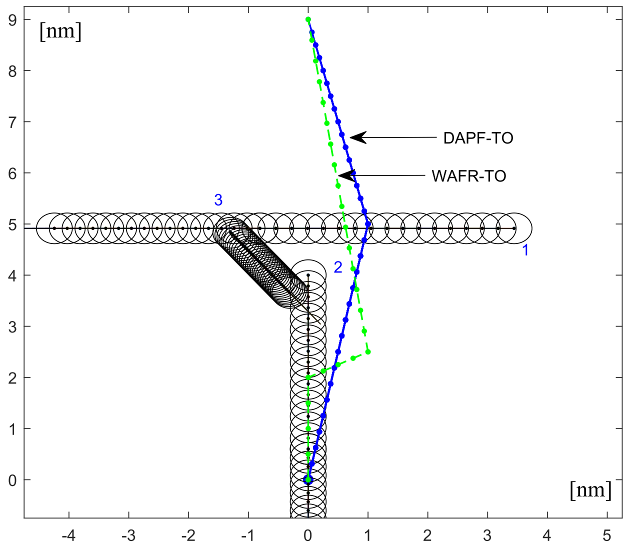

FIGURE 18. Paths generated by DAPF-TO and WAFR-TO for an environment with dynamic obstacles - test case 4 with 3 dynamic obstacles.

TABLE 4. Simulation Results of DAPF and WAFR for 8-Connected Grid With Dynamic Obstacles.

Algorithm	Test case	Number of dynamic obstacles	Optimized path length [nm]	Run time with optimization [s]
DAPF	1	1	9.2691	**0.204**
WAFR	1	1	**9.0582**	0.2101
DAPF	2	2	**9.8489**	**0.1815**
WAFR	2	2	9.8835	0.241
DAPF	3	4	**9.4922**	**0.2064**
WAFR	3	4	9.5388	0.2387
DAPF	4	3	**9.2221**	**0.1916**
WAFR	4	3	9.6945	0.2487

obstacles, achievement of COLREGs compliant solutions, repeatability of solution for every run of calculations with the same input data and very low run time (milliseconds).

V. CONCLUSION

This article introduced a new grid-based path planning algorithm for application in Intelligent Transportation Systems in maritime transport – the Discrete Artificial Potential Field (DAPF) algorithm with Trajectory Optimization (TO). The method was validated by simulation experiments in static and dynamic environments and compared with the state-of-the-art wave-front algorithm (WAFR). The results demonstrated the effectiveness of the algorithm both in terms of the solution quality and the run time. Further works will include real experiments, tests with changing strategies of dynamic obstacles and evaluation of solutions from different ships' perspectives.

REFERENCES

[1] J. Jansson and F. Gustafsson, "A framework and automotive application of collision avoidance decision making," *Automatica*, vol. 44, no. 9, pp. 2347–2351, Sep. 2008.

[2] Kongsberg Maritime. (2020). *Yara Birkeland—Autonomous Ship Project*. Accessed: Octo. 23, 2020. [Online]. Available: https://www.kongsberg.com/

[3] Rolls-Royce. (2018). *Safer Vessel with Autonomous Navigation*. Accessed: Octo. 23, 2020. [Online]. Available: https://www.rolls-royce.com/

[4] H. Ch Burmeister and J. Rãdseth. (2012). *Maritime Unmanned Navigation through Intelligence in Networks*. Accessed: Octo. 23, 2020. [Online]. Available: http://www.unmanned-ship.org/munin/

[5] H. A. Tvete and DNV GL. (2013). *The ReVolt Project*. Accessed: Oct. 23, 2020. [Online]. Available: https://www.dnvgl.com/technology-innovation/revolt/index.html

[6] Rolls-Royce. (2016). *Advanced Autonomous Waterborne Applications*. Accessed: Oct. 23, 2020. [Online]. Available: https://www.rolls-royce.com/

[7] Norwegian University of Science and Technology. (2015). *Autosea Project*. Accessed: Oct. 23, 2020. [Online]. Available: https://www.ntnu.edu/autosea

[8] Norwegian University of Science and Technology. *Autoferry Project*. Accessed: Oct. 23, 2020. [Online]. Available: https://www.ntnu.edu/autoferry

[9] H. Yang, J. Qi, Y. Miao, H. Sun, and J. Li, "A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8557–8566, Nov. 2019.

[10] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach," *Soft Comput.*, vol. 21, no. 4, pp. 949–964, Feb. 2017.

[11] O. Montiel, U. Orozco-Rosas, and R. Sepúlveda, "Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles," *Expert Syst. Appl.*, vol. 42, no. 12, pp. 5177–5191, Jul. 2015.

[12] F. H. Ajeil, I. K. Ibraheem, M. A. Sahib, and A. J. Humaidi, "Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106076.

[13] X. Wang, H. Shi, and C. Zhang, "Path planning for intelligent parking system based on improved ant colony optimization," *IEEE Access*, vol. 8, pp. 65267–65273, 2020.

[14] C. Qu, W. Gai, M. Zhong, and J. Zhang, "A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106099.

[15] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 132–141, Feb. 2013.

[16] M. Blaich, M. Rosenfelder, M. Schuster, O. Bittel, and J. Reuter, "Fast grid based collision avoidance for vessels using a search algorithm," in *Proc. 17th Int. Conf. Methods Models Autom. Robot. (MMAR)*, Aug. 2012, pp. 385–390.

[17] J. Lisowski and M. Mohamed-Seghir, "Comparison of computational intelligence methods based on fuzzy sets and game theory in the synthesis of safe ship control based on information from a radar ARPA system," *Remote Sens.*, vol. 11, no. 1, p. 82, Jan. 2019.

[18] J. Karbowska-Chilinska, J. Koszelew, K. Ostrowski, P. Kuczynski, E. Kulbiej, and P. Wolejsza, "Beam search algorithm for ship anti-collision trajectory planning," *Sensors*, vol. 19, no. 24, p. 5338, 2019.

[19] C. K. Tam and R. Bucknall, "Path-planning algorithm for ships in close-range encounters," *J. Mar. Sci. Technol.*, vol. 15, no. 4, pp. 395–407, Dec. 2010, doi: 10.1007/s00773-010-0094-x.

[20] R. Szlapczynski and J. Szlapczynska, "On evolutionary computing in multi-ship trajectory planning," *Int. J. Speech Technol.*, vol. 37, no. 2, pp. 155–174, Sep. 2012.

[21] R. Szlapczynski and J. Szlapczynska, "Customized crossover in evolutionary sets of safe ship trajectories," *Int. J. Appl. Math. Comput. Sci.*, vol. 22, no. 4, pp. 999–1009, Dec. 2012.

[22] L. Kuczkowski and R. Smierzchalski, "Path planning algorithm for ship collisions avoidance in environment with changing strategy of dynamic obstacles," in *Proc. 19th Polish Control Conf.*, Kraków, Poland, vol. 577, Jun. 2017, pp. 641–650.

[23] Ch. Wang, X. Zhang, R. Li and P. Dong, "Path planning of maritime autonomous surface ships in unknown environment with reinforcement learning," in *Proc. ICCSIP*, vol. 1006, 2018, pp. 127–137.

[24] A. Witkowska, R. Smierzchalski, and P. Wilczynski, "Trajectory planning for Service Ship during emergency STS transfer operation," in *Proc. 19th Polish Control Conf.*, Kraków, Poland, Jun. 2017, vol. 577, pp. 514–523.
[25] L. P. Perera, J. P. Carvalho, and C. G. Soares, "Fuzzy-logic based parallel collisions avoidance decision formulation for an ocean navigational system," *IFAC Proc. Volumes*, vol. 43, no. 20, pp. 260–265, Sep. 2010.
[26] L. P. Perera, J. P. Carvalho, and C. G. Soares, "Bayesian network based sequential collision avoidance action execution for an ocean navigational system," *IFAC Proc. Volumes*, vol. 43, no. 20, pp. 266–271, Sep. 2010.
[27] Y. Xue, D. Clelland, B. S. Lee, and D. Han, "Automatic simulation of ship navigation," *Ocean Eng.*, vol. 38, nos. 17–18, pp. 2290–2305, Dec. 2011.
[28] H. Lyu and Y. Yin, "COLREGS-constrained real-time path planning for autonomous ships using modified artificial potential fields," *J. Navigat.*, vol. 72, no. 3, pp. 588–608, May 2019.
[29] C. Tam and R. Bucknall, "Cooperative path planning algorithm for marine surface vessels," *Ocean Eng.*, vol. 57, pp. 25–33, Jan. 2013.
[30] Y.-T. Kang, W.-J. Chen, D.-Q. Zhu, J.-H. Wang, and Q.-M. Xie, "Collision Avoidance Path Planning For Ships By Particle Swarm Optimization," *J. Mar. Sci. Tech.-Taiw.*, vol. 26, no. 6, pp. 777–786, 2018.
[31] A. N. Cockcroft and J. N. F. Lameijer, *A Guide to Collision Avoidance Rules*. Amsterdam, The Netherlands: Elsevier, 2012.
[32] I. M. Zidane and K. Ibrahim, "Wavefront and A-Star Algorithms for Mobile Robot Path Planning," in *Proc. Int. Conf. Adv. Intell. Syst. Inf. (AISI)*, vol. 639, 2017, pp. 69–80.
[33] Ch. Tang, R. Sun, Sh. Yu, L. Chen and J. Zheng, "Autonomous indoor mobile robot exploration based on wavefront algorithm," in *Proc. 12th Int. Conf. (ICIRA)*, vol. 11744, 2019, pp. 338–348.

AGNIESZKA LAZAROWSKA received the Ph.D. degree in electrical engineering from Gdynia Maritime University, Gdynia, Poland, in 2015.

From 2009 to 2015, she was a Research Assistant with the Department of Ship Automation, Faculty of Electrical Engineering, Gdynia Maritime University, where she has been an Assistant Professor since 2015. From 2018 to 2019, she was a Research Coordinator and a Principal Investigator of the research project Research on New Obstacle Avoidance Algorithms for Ships (NOAA), founded by the International Association of Maritime Universities (IAMU) and the Nippon Foundation from Japan. Her research interests include guidance systems, autonomous navigation systems, safe ship control systems, decision support systems, path planning algorithms, deterministic and non-deterministic optimization methods, artificial intelligence, swarm intelligence, safe navigation, unmanned and autonomous ships, and other autonomous vehicles.