SWARM INTELLIGENCE APPROACH TO SAFE SHIP CONTROL

Agnieszka Lazarowska
Gdynia Maritime University, Poland

ABSTRACT

This paper presents an application of the Ant Colony Optimization (ACO) technique in a safe ship control system. The method developed solves the problem of path planning and collision avoidance of a ship in the open sea as well as in restricted waters. The structure of the developed safe ship control system is introduced, followed by a presentation of the applied algorithm. Results showing the problem-solving capability of the system are also included. The aim of the system developed is to increase automation of a safe ship control process. It is possible to apply the proposed method in Unmanned Surface Vehicles (USVs) control system, what will contribute to the enhancement of their autonomy.

INTRODUCTION

Technological development led to an increased marine traffic, which caused navigation to become more demanding for deck officers. A safe ship control system, containing equipment such as a log, a gyrocompass, a GPS, a radar with an Automatic Radar Plotting Aid (ARPA), an Automatic Identification System (AIS), an Electronic Chart Display and Information System (ECDIS) and an autopilot, enables realization of various methods of navigator's decision support.

Safe ship control is a complex process, because it requires a continuous analysis of large amount of information and quick decision making. Incorrect assessment of the current navigational situation based upon data from navigational devices can lead to a collision situation, often with very tragic consequences.

The newest ARPA systems enable manual and/or automatic tracking of up to 100 objects detected by a radar, and then simulating the manoeuvre planned by the navigator (trial manoeuvre function), but do not determine safe course or speed changes.

The development of modern computational intelligence methods enables the synthesis of a safe ship control system. The aim of a safe ship control system is to determine a safe course or speed manoeuvre, or a safe trajectory of the ship, taking into account the dynamic properties of the vessel at various loading conditions and at various operational speeds, manoeuvring rules of the vessel in fair weather condition and in reduced visibility at sea, limitations of the environment and the corresponding quality index of control, taking into account both the optimality and safety criterion.

Research on ships collision avoidance started in the 1960s and the first studies were concerned with a collision avoidance manoeuvre in an encounter situation between two ships. There was a necessity to consider the International Regulations for Preventing Collisions at Sea (COLREGs) in solving a collision situation, therefore in 1974 Jones [12] developed the manoeuvre diagram to provide the possibility of including the COLREGs in determining the evasive manoeuvre of a ship. In the 1970-1980s the concept of collision risk assessment with the use of a ship domain was developed by Fujii & Tanaka (1971) [9], Goodwin (1975) [10], Davis et al. (1980) [7] and Coldwell (1983) [6]. In the 1970s the idea of determining optimal collision avoidance manoeuvres with the use of the differential games theory was introduced by Olsder & Walter (1977) [17] and Kudriaszov & Lisowski (1979) [13], and considered for an encounter situation between two ships. Later Lisowski developed a differential games approach for multi-ship encounter situations [16].

In the 2000s ship path planning approaches were reported. A deterministic path planning method was introduced by Chang et al. in 2003 [5] and developed by Szłapczyński in 2006

[20]. Artificial intelligence methods for ship path planning were reported by Śmierzchalski & Michalewicz (2000) [22], Lisowski (2001) [15], Ito & Zeng (2001) [11], Tam & Bucknall (2010) [23], Tsou & Hsueh (2010) [25], Perera et al. (2011) [18], Szłapczyński (2012) [21] and others. However, all of these methods are not deprived of disadvantages. They are characterized by the limitations such as disregarding of the ship's dynamics or static obstacles and ignoring some of the COLREGs.

The paper presents an application of one of the modern global optimization methods to a safe ship control process. The idea of utilizing the Ant Colony Optimization (ACO) method in a decision support system for ships was introduced by Lazarowska in 2012 [14]. The method is inspired by a collective behaviour of ants and due to that reason it is called the Ant Colony Optimization (ACO). ACO belongs to Swarm Intelligence (SI) methods. The term SI was introduced by G. Beni and J. Wang in relation to cellular robotic systems [2]. E. Bonabeau, M. Dorigo and G. Theraulaz extended the definition of SI, defining this term as any attempt to build an algorithm inspired by the collective behaviour of the colony of insects or other animal communities [3].

The first problem solved with the use of ACO was the Traveling Salesman Problem (TSP) [8], which is a transition path optimization task. Brand et al. [4], M. Pluciński [19] and others developed ACO based algorithms for mobile robot path planning. Other applications of ACO include feature selection [1], vehicle routing problem (VRP), data network routing and ship routing [24]. In this paper ACO is used in a safe ship path planning method.

SAFE SHIP CONTROL UTILIZING ACO

The task of a safe ship control system is to determine safe controls of a ship, which is a safe course and/or speed change or a sequence of safe course and/or speed changes, called a safe ship trajectory. This is accomplished with the use of an appropriate control algorithm. The input to the algorithm is the current state of the process. The state of the process is determined with the use of information describing the actual navigational situation, which include:

- the course and speed of the own ship (OS),
- the course and speed of j-th target ship (TS),
- the bearing of the j-th target ship,
- the distance of the j-th target ship from the own ship,
- data about static obstacle (shoals, lands, buoys).

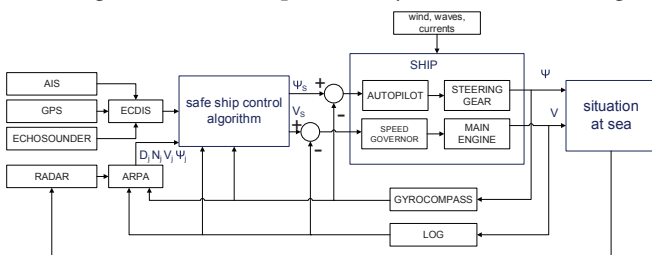A diagram of a safe ship control system is shown in Fig. 1.

Fig. 1. A block diagram of safe ship control system (Ψ – OS course, V – OS speed, Ψj – TS course, Vj – TS speed, Nj – TS bearing from OS, Dj – TS distance from OS, Ψs – safe course of OS, Vs – safe speed of OS)

In the method developed the following assumptions were made:

- to take into account static and dynamic limitations,
- to determine a safe ship trajectory to a specified endpoint,
- a kinematic model of ship motion,
- to take into account the dynamics of the OS through the application of the time of manoeuvre,
- navigational data concerning target ships from ARPA and AIS,
- to determine a safe ship trajectory in compliance with COLREGs,
- a safety area around the TS in the form of a hexagon domain,
- computational time not exceeding sixty seconds,
- reproducible results,
- not taking into account the strategies of TSs.

MODEL OF THE PROCESS

It was assumed that TS does not change its course and speed. The process of collision avoidance at sea was described with the use of a kinematic model of ship motion, where state variables and controls are defined in the following way: $x_1 = X$, $x_2 = Y$, $x_{2j+1} = X_j$, $x_{2j+2} = Y_j$, $u = Ψ$, where $j = 1, 2, ..., m$ – number of encountered TSs, X and Y are coordinates of the OS position, X_j and Y_j are coordinates of the TS position. The state equation of the safe ship control process are described by equations (1). Dynamic properties of OS are considered during the assessment of a determined trajectory in the form of the time of manoeuvre.

$$
\begin{aligned}
\dot{x}_1 &= V \times \sin u(t) = V \times \sin Ψ(t) \\
\dot{x}_2 &= V \times \cos u(t) = V \times \cos Ψ(t) \\
\dot{x}_{2j+1} &= V_j \times \sin Ψ_j(t) \\
\dot{x}_{2j+2} &= V_j \times \cos Ψ_j(t)
\end{aligned}
\tag{1}
$$

SAFE SHIP CONTROL ALGORITHM

After reception of information describing the current navigational situation, a relative course, bearing and speed of each TS is calculated. The procedure of determining dangerous objects consists of checking for each TS, whether it is a dangerous object (Fig. 3). A dangerous object is an object which intersects its course with the course of the OS. Then a division of the OS route from an actual position to an endpoint into a certain number of steps is executed and on that basis a graph of possible waypoints of the OS is built in the allowable state space (Fig. 4). The calculation of a safe ship trajectory with the use of ACO consists of three main stages: the data initialization, the solutions construction (Fig. 5) and the pheromone trail update procedure (Fig. 6). When the algorithm reaches the maximum number of iterations or the maximum computational time, the best solution is selected based upon the fitness function calculation (Fig. 7). After that the following output data are displayed: the course of the OS at every line segment of the determined trajectory, the time taken for the OS to reach the endpoint, the distance travelled and a graphical presentation of the safe ship trajectory.

A pseudo-code of the safe ship control algorithm utilizing ACO is shown in Fig. 2.

Input: Ψ, V, Ψ_j, V_j, N_j, D_j, static restrictions
1: calculate Ψ_{rj}, N_{rj}, V_{rj}
2: determine dangerous objects
3: build a graph of permissible OS waypoints
4: initialize α, β, ρ, τ_0, m, it
5:      **for** i=1, ..., it, **do**
                        6:
           if (t < t_{max}) **then**
7:                       solutions construction
8:                       pheromone update procedure
                        9:
           end-if
10:     **end-for**
11: choose the best solution
Output: Ψ_s, V_s, time, d, graphical solution

Fig. 2. A pseudo-code of the safe ship control algorithm (Ψrj – TS relative course, Nrj – TS relative bearing, Vrj – TS relative speed, α – coefficient defining the importance of τ, β – coefficient defining the importance of heuristic value, ρ – pheromone evaporation rate, $\tau 0$ – initial pheromone trail amount, m – number of ants, it – number of iterations, tmax – maximum computational time, time – time of OS passage to the endpoint, d – length of the safe trajectory

Procedure of determining dangerous objects
1: **for** j=1, ..., n, **do**
2:      **if** (intersection point==empty) **then**
3:         mark TS as safe
4:      **else**
                        5:
        if (intersection point> allowable space) **then**
6:                 mark TS as safe
                        7:
           else
8:                 calculate distance of OS to intersection point
9:                    calculate time of OS to intersection point
10:                 calculate TS position
                        11:
           end-if
12:     **end-if**
13: **end-for**

Fig. 3. Procedure of determining dangerous objects (n – number of TSs)

Procedure of building a graph of possible OS waypoints
1: division of OS route
2: waypoint generation
3: placement of TS domains
4: **for** j=1, ..., n, **do**
5:      **for** w=1, ..., w_{max}, **do**
                        6:
           if (w exceeds TS domains) **then**
7:                 mark w as exceeding constraints
                        8:
           end-if
9:      **end-for**
10: remove waypoints exceeding constraints
11: **end-for**

Fig. 4. Procedure of building a graph of possible OS waypoints (n – number of TSs, wmax – number of waypoints)

Procedure of ACO solution construction
1: **for** ant=1, ..., m, **do**
2:      **for** step=1, ..., $step_{max}$, **do**
                        3:
           if (wp_i != wp_e) **then**
4:                 action choice rule
                        5:
           end-if
6:      **end-for**
7:      calculate the length of ant's path
8:      **if** (ant's path < min path) **then**
9:         min path:= ant's path
10:     **end-if**
11: **end-for**

Fig. 5. Procedure of ACO solution construction

Procedure of pheromone trail update
1: **for** w=1, ..., w_{max}, **do**
2:      $\tau[w]:=(1-\rho)\times \tau[w]$
3: **end-for**
4: **for** ant=1, ..., m, **do**
5:      **for** w=1, ..., w_{max}, **do**
                        6:
           if (w belongs to ant's path) **then**
7:                 $\tau[w] := \tau[w] + \Delta\tau$
                        8:
           end-if
9:      **end-for**
10: **end-for**

Fig. 6. Procedure of pheromone trail update

```
Procedure of the best solution selection
1: for i=1, …, paths, do
2: division of path into k stages
3: placement of TS domains
4:      for k=1, …, k_max, do
5:          constraints placement
            6:
        if (OS exceeds constraints) then
7:              mark i as incorrect path
8:              break;
            9:
        end-if
10:     end-for
11: mark i as correct path
12: end-for
13: determine the best path from correct paths
```

Fig. 7. Procedure of the best solution selection

RESULTS OF SIMULATION TESTS

The developed algorithm for the safe ship control was implemented in the MATLAB programming language. The choice of this environment was due to the built-in plotting and dynamic simulation functions that allow easy graphical presentation of results. The algorithm was tested extensively using dozens of various navigational situations. Calculations were made using a PC with an Intel Core i5 M430 2.27 GHz, 2GB RAM processor and 32-bit Windows 7 Professional system. Results of several representative test cases are presented below in Fig. 8 to 19.

The results obtained allow to formulate the following conclusions:

- every solution constitutes a safe ship trajectory, what means that at any stage of the own ship movement along the path the static and dynamic navigational restrictions are not exceeded,
- solutions fulfil the COLREGs,
- results, despite the probabilistic nature of the applied optimization method, are the same for each repetition of calculations for the same navigational situation,
- the developed algorithm has low computational time, not exceeding one minute even for complex situations,
- the algorithm is suitable for use in a safe ship control system due to the fulfilment of all the most important criteria, that is the safety, compatibility with COLREGs, short computational time and repeatability of solution,
- solutions returned by an algorithm from the perspective of all of the vessels involved in the navigational situation do not intersect, what allows the use of such a safe ship control system by all of the vessels simultaneously.

The COLREGs compliance of solutions can be observed by analysing the graphical presentation of results. For example a trajectory calculated by the algorithm for a navigational situation presented in Fig. 16 fulfils rule 15 of COLREGs. Rule 15 states that in the crossing situation of two power-driven vessels, the vessel that has the other vessel on her starboard side shall keep out of the way and if the circumstances of the case admit, avoid crossing ahead of the other vessel. In the situation presented in Fig. 16 OS marked with 0 and TS 2 are crossing. As it can be seen in Fig. 17, OS which is the give-way ship, keeps out of the way and passes astern TS, what proofs that OS trajectory is compliant with rule 15 of COLREGs.
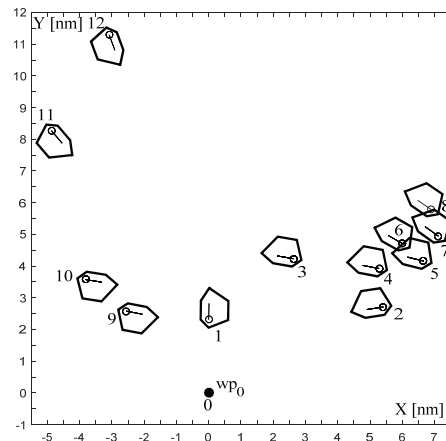

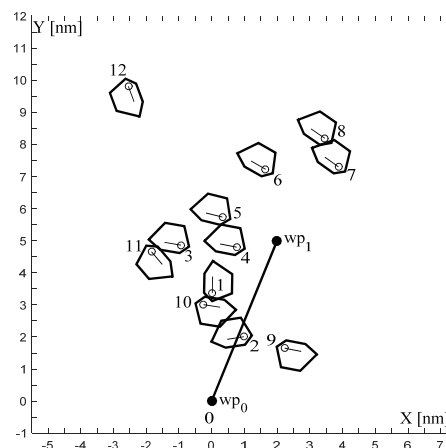*Fig. 8. Navigational situation with 12 TS - OS at wp_0*

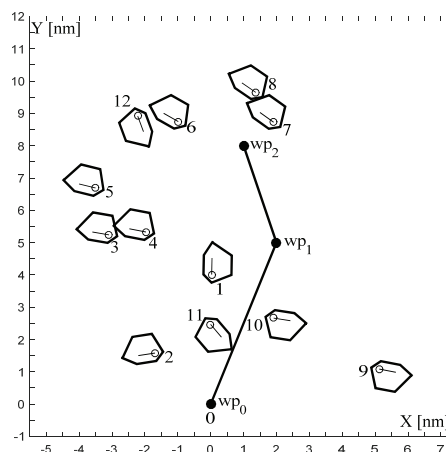*Fig. 9. Graphical solution for situation with 12 TS - OS at wp_1*

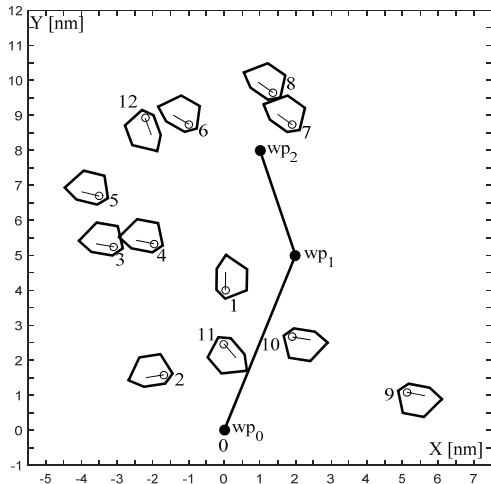*Fig. 10. Graphical solution for situation with 12 TS - OS at wp_2*

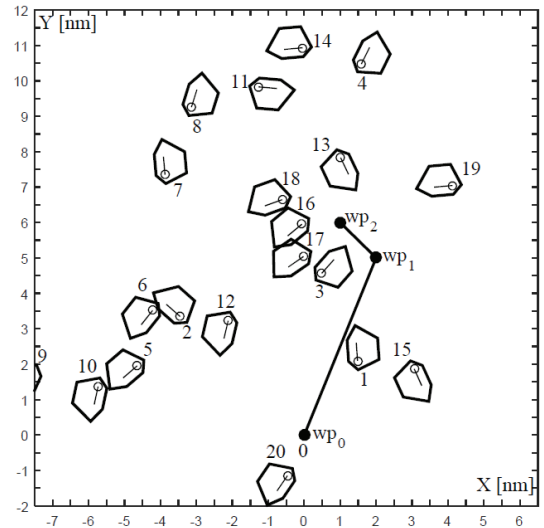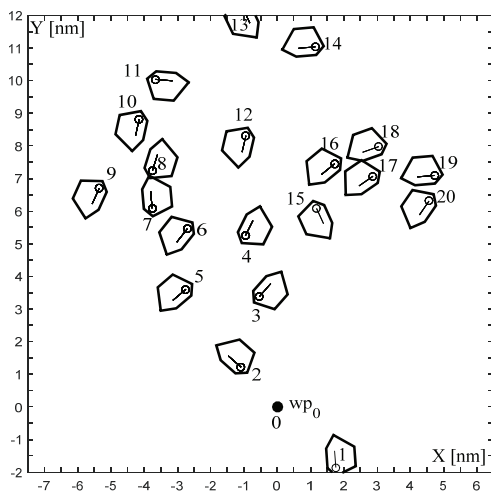Fig. 11. Graphical solution for situation with 12 TS - OS at wp_e

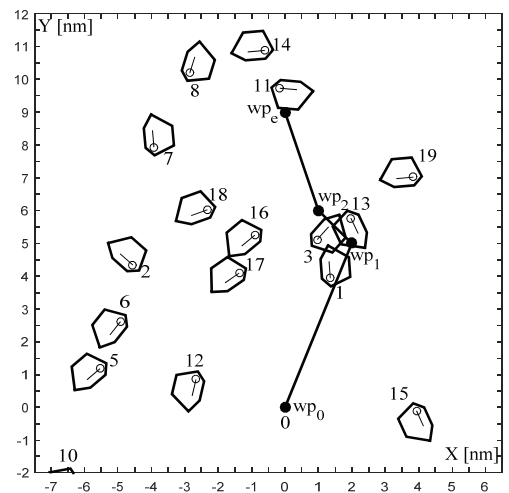Fig. 12. Navigational situation with 20 TS - OS at wp_0

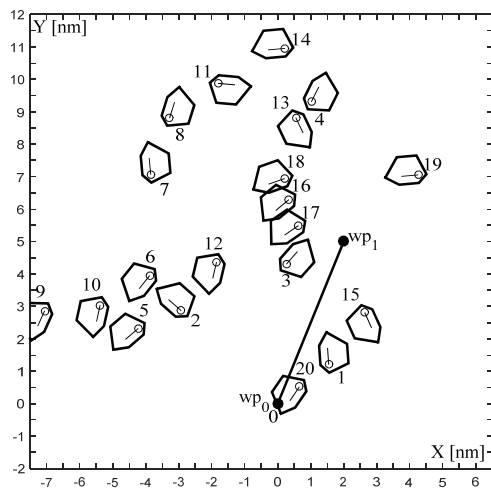Fig. 13. Graphical solution for situation with 20 TS - OS at wp_1

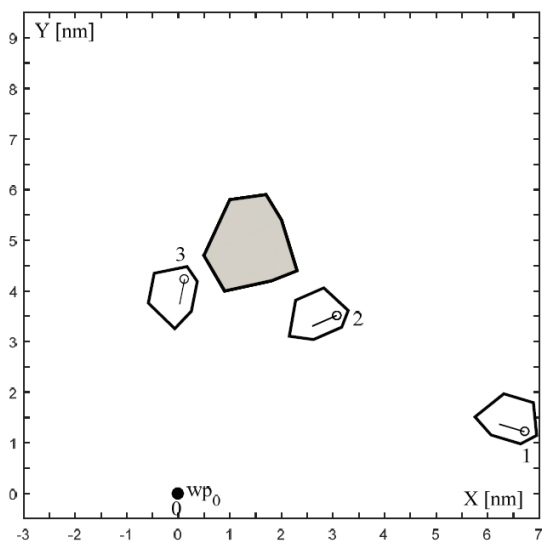Fig. 14. Graphical solution for situation with 20 TS - OS at wp_2

Fig. 15. Graphical solution for situation with 20 TS - OS at wp_e

Fig. 16. Navigational situation with an island and 3 TS - OS at wp_0

Fig. 17. Graphical solution for situation with 3 TS - OS at wp_1

Fig. 18. Graphical solution for situation with an island and 3 TS - OS at wp_2

Fig. 19. Graphical solution for situation with an island and 3 TS - OS at wp_e

CONCLUSIONS

Analysis of the safe ship control method, taking into account the condition of optimal control, and results of simulation studies of the developed algorithm allows to formulate the following concluding remarks.

Presented method of safe ship control utilizing ACO can be easily adapted for use in collision avoidance support systems in aviation and land navigation.

Developed method of safe, optimal course changes in collision situation at sea, with the presence of a great amount of target ships and taking into account the static and dynamic limitations, is the process of determining the safe, optimal path of a moving object in a dynamic environment, and the solution of this problem is widely applicable in robotics and military, including control systems of unmanned vehicles.

The synthesis of safe ship control system using stochastic global optimization method allowed to take into account the COLREGs, greater amount of dynamic and static navigational restrictions, dynamic properties of the ship, determination of a safe ship trajectory to the specified endpoint and obtain not colliding solutions from the perspective of all of the vessels involved in the collision situation.

The results of the simulation tests show that the process of reproducing the behaviour of a biological system, such as an ant colony, to resolve complex problems of maritime transport, marine environment protection and increase of safe ship control automation through an automatic determination of safe, optimal course changes in collision situation at sea, indicates a high potential of nature-inspired methods in practical industrial applications.

The developed method is suitable for solving situations in areas of heavy traffic, where most of the collisions occur.

The analysis of simulation results showed that despite the application of a stochastic optimization method, the algorithm provides reproducible solutions.

The proposed algorithm is robust to changes of target ships strategies, because it is able to continuously return solutions, based on actual data from ARPA and AIS, in a very short time.

The possibility to change the shape and size of the target ship domain enables taking into account weather conditions of good and limited visibility and also the preferences of the decision maker - the system operator.

BIBLIOGRAPHY

1. Barbosa H.J.C.: *Ant Colony Optimization - Techniques and Applications,* InTech, 2013.

2. Beni G., Wang Jing: *Theoretical problems for the realization of distributed robotic systems.* IEEE International Conference on Robotics and Automation, 1991.

3. Bonabeau E., Dorigo M., Theraulaz G.: *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, Inc., 1999.
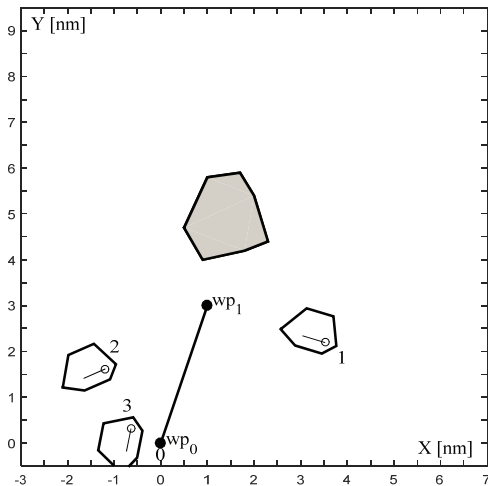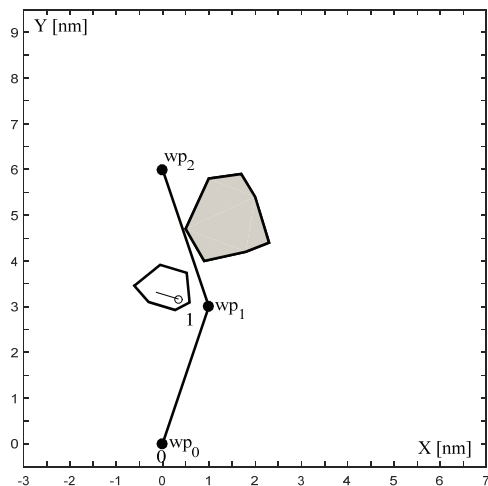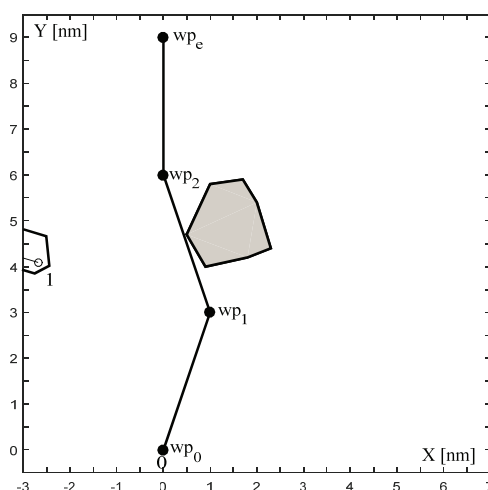
4. Brand M., Masuda M., Wehner N. , Yu Xiao-Hua: *Ant Colony Optimization Algorithm for Robot Path Planning*, International Conference On Computer Design And Applications, 2010.

5. Chang Ki-Yin, Jan G. E., Parberry I.: *A Method for Searching Optimal Routes with Collision Avoidance on Raster Charts.* The Journal of Navigation, Vol. 56, 2003, p. 371–384.

6. Coldwell T.G.: *Marine Traffic Behaviour in Restricted Waters.* The Journal of Navigation, Vol. 36, 1983, p. 430–444.

7. Davis P.V., Dove M. J., Stockel C. T.: *A Computer Simulation of Marine Traffic Using Domains and Arenas.* The Journal of Navigation, Vol. 33, 1980, p. 215–222.

8. Dorigo M., Stützle T.: *Ant Colony Optimization.* The MIT Press, 2004.

9. Fujii Y., Tanaka K.: *Traffic Capacity.* The Journal of Navigation, Vol. 24, 1971, p. 543–552.

10. Goodwin E. M.: *A Statistical Study of Ship Domains.* The Journal of Navigation, Vol. 28, 1975, p. 328–344.

11. Ito M., Zeng Xiao-Ming: *Planning a collision avoidance model for ship using genetic algorithm.* IEEE International Conference on Systems, Man, and Cybernetics, 2001.

12. Jones K. D.: *Application of a Manoeuvre Diagram to Multi-ship Encounters.* The Journal of Navigation, Vol. 27, 1974, p. 19–27.

13. Kudriaszov V., Lisowski J.: *Model of positional game applied for the synthesis of safe ship control.* Archiwum Automatyki i Telemechaniki, Vol. 24, 1979, p. 483–497.

14. Lazarowska A.: *Decision support system for collision avoidance at sea.* Polish Maritime Research, Vol. 19, Issue S1 (74), 2012, p. 19–24.

15. Lisowski J.: *Computational intelligence methods in the safe ship control process.* Polish Maritime Research, Vol. 8, Issue 1, 2001, p. 18–24.

16. Lisowski J.: *Comparison of dynamic games in application to safe ship control.* Polish Maritime Research, Vol. 21, Issue 3, 2014, p. 3–12.

17. Olsder G. J., Walter J. L.: *A differential game approach to collision avoidance of ships.* Proceedings of the 8th IFIP Conference on Optimization Techniques, Würzburg, 1977.

18. Perera L. P., Carvalho J. P., Guedes Soares C.: *Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions.* Journal of Marine Science and Technology, Vol. 16, No.1, 2011, p. 84–99.

19. Pluciński M.: *Application of the Ant Colony Algorithm for the Path Planning.* Enhanced Methods in Computer Security, Biometric and Artificial Intelligence Systems, 2005, p. 345–352.

20. Szłapczyński R.: *A New Method of Ship Routing on Raster Grids, with Turn Penalties and Collision Avoidance.* The Journal of Navigation, Vol. 59, 2006, p. 27–42.

21. Szłapczyński R.: *Evolutionary approach to ship's trajectory planning within Traffic Separation Schemes.* Polish Maritime Research, Vol. 19, Issue 1, 2012, p. 11–20.

22. Śmierzchalski R., Michalewicz Z.: *Modeling of Ship Trajectory in Collision Situations by an Evolutionary Algorithm.* Transactions on Evolutionary Computation, Vol. 4, 2000, p. 227–241.

23. Tam Ch., Bucknall R.: *Path-planning algorithm for ships in close-range encounters.* Journal of Marine Science and Technology, Vol. 15, 2010, p. 395–407.

24. Tsou M.-Ch., Cheng H.-Ch.: *An Ant Colony Algorithm for efficient ship routing.* Polish Maritime Research, Vol. 20, Issue 3, 2013, p. 28–38.

25. Tsou M.-Ch., Hsueh Ch.-K.. *The Study of Ship Collision Avoidance Route Planning by Ant Colony Algorithm.* Journal of Marine Science and Technology - TAIWAN, Vol. 18, 2010, p. 746–756.

CONTACT WITH THE AUTHOR

Agnieszka Lazarowska

Gdynia Maritime University
11/12 Narutowicza St.
80 - 233 Gdańsk
Poland