TRANSNAV
http://www.transnav.eu
the International Journal
on Marine Navigation
and Safety of Sea Transportation
Volume 17
Number 2
June 2023

DOI: 10.12716/1001.17.02.10

A Nature Inspired Collision Avoidance Algorithm for Ships

A. Lazarowska
Gdynia Maritime University, Gdynia, Poland

ABSTRACT: Nature inspired algorithms are regarded as a powerful tool for solving real life problems. They do not guarantee to find the globally optimal solution, but can find a suboptimal, robust solution with an acceptable computational cost. The paper introduces an approach to the development of collision avoidance algorithms for ships based on the firefly algorithm, classified to the swarm intelligence methods. Such algorithms are inspired by the swarming behaviour of animals, such as e.g. birds, fish, ants, bees, fireflies. The description of the developed algorithm is followed by the presentation of simulation results, which show, that it might be regarded as an efficient method of solving the collision avoidance problem. Such algorithm is intended for use in the Decision Support System or in the Collision Avoidance Module of the Autonomous Navigation System for Maritime Autonomous Surface Ships.

1 INTRODUCTION

1.1 *Swarm intelligence and its applications*

Swarm intelligence (SI) is a dynamically developing area of artificial intelligence, inspired by the collective behaviour of animals (ants, bees, birds, fish) or other living organisms, such as e.g. bacteria. The first approaches classified as SI methods, as can be seen on the timeline in Figure 1, started to emerge in the early 1990s. From that time many methods have been introduced. Most of them are inspired by the foraging behaviour, as Ant Colony Optimization (ACO), Artificial Bee Colony algorithm (ABC) or Wolf Search Algorithm (WSA). Sometimes breeding, such as in Cuckoo Search (CS) or Flower Pollination Algorithm (FPA), or mates finding, as in Firefly Algorithm (FA), constitutes an inspiration. Other shown algorithms include: Particle Swarm Optimization (PSO), Differential Evolution (DE), Harmony Search (HS), Honey Bee Algorithm (HBA) and Virtual Bee Algorithm (VBA), Bat Algorithm (BA), Eagle Strategy (ES) and Krill Herd Algorithm (KHA).

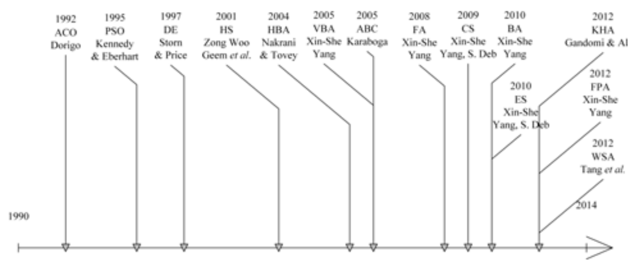

Figure 1. A timeline presenting the development of swarm intelligence and bio-inspired algorithms

Swarm intelligence algorithms were initially applied for solving combinatorial optimization problems such as traveling salesmen problem or graph colouring [1,2,3]. Over time more and more real-life applications have been developed, such as e.g. ACO in bioinformatics for DNA sequencing [4] or

GA, ACO and PSO for mobile robot path planning [5]. Other examples of SI applications include: PSO for image analysis (face detection) [6], PSO for cancer classification [7] and CS for the optimization of the steel structure design process [8]. A review of SI and bio-inspired methods can be found in [9].

Examples of applications in regard to ships include: optimization of ship course controller parameters based on ACO [10], ship safe trajectory planning in a collision situation based on ACO [11, 12] and PSO [13], optimization of the parameters of the ship dynamics model using ACO, PSO and ABC [14]. In [15] a survey of SI-based algorithms for USVs collaboration has been presented.

1.2 *Ship collision avoidance*

As can be seen from the above introduced overview of SI methods, one of their applications is the ship collision avoidance problem. This issue is one of the main tasks in the ship navigation process, where the main constraints are static, such as lands, shallows, buoys, waterways and dynamic obstacles, which change their position over time. Dynamic obstacles are called target ships, which are encountered ships, that might constitute a collision risk to an own ship. Many approaches for solving the ship collision avoidance problem were proposed since 1950s, where first methods for two-ship encounters based upon solving the triangle of velocities appeared. The latest algorithms are based upon game theory [16], fuzzy set theory [17], dynamic programming with neural constraints [18], evolutionary multi-objective optimisation [19], multi-agent reinforcement learning [20], deep reinforcement learning [21], modified artificial potential field [22] and probabilistic velocity obstacle method [23]. Safe trajectories are then fed into the ship motion control system in order to steer the ship along the determined path. Examples of recent ship motion control methods are the Linear Matrix Inequalities controller [24], a path controller with switching approach [25], the actor-critic time-delay controller [26] and the backstepping controller [27].

2 FIREFLY ALGORITHM FOR SHIP COLLISION AVOIDANCE

2.1 *Description of constructive vs population based approximate algorithms*

Optimization algorithms can be classified into one of the two groups: exact or approximate approaches. Swarm intelligence methods due to the property that they do not guarantee a global optimum, but allow to obtain a suboptimal solution at acceptable computational cost, are classified to the group of approximate algorithms. An approximate algorithm can be further classified as constructive, population-based or local search. Constructive algorithms build a solution by iteratively adding solution components, until a complete solution is achieved. A population-based algorithm starts with an initial population of individuals (candidate solutions), which is then evaluated and modified in order to find the final best solution, until the moment, when the termination condition is met. Local search algorithms start from some initial solution and iteratively apply local changes in order to improve that solution. Local changes mean the exploration of the neighbourhood of the current solution [28].

2.2 *Description of the firefly algorithm background*

Figure 2. The bioluminescence behaviour of a firefly

The Firefly Algorithm (FA) has been introduced by Xin-She Yang in 2008 [29]. As the name suggests, the algorithm is inspired by the behaviour of fireflies, which communicate with each other, look for prey and find mates using bioluminescence. Bioluminescence means the production and emission of light by living organisms, as shown in Figure 2. The main assumptions applied in the FA, inspired by the behaviour of fireflies in nature, are:

- the attractiveness of a firefly is proportional to its brightness, the value of both parameters decreases with an increasing distance between the fireflies;
- for every pair of fireflies, the less bright firefly will be moving towards the brighter one;
- if there is no brighter firefly, it will move randomly.

The movement of a firefly towards the brighter one is defined by Equation (1).

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha\left(\text{rand} - 0.5\right) \quad (1)$$

where β_0 is the attractiveness at distance $\underline{r} = 0$, α is a parameter introducing randomness, rand is a random number generator uniformly distributed in [0, 1], r_{ij} is the distance between fireflies i and j, and γ is the light absorption coefficient, which determines the variability of attractiveness. The distance between any two fireflies i and j is defined by Equation (2).

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

2.3 *The firefly algorithm for solving the ship collision avoidance*

The flowchart of a firefly algorithm applied for solving the ship collision avoidance problem is given in Figure 3. The input data for the algorithm are the values of parameters describing the collision situation at sea, such as: the course Ψ_j, the speed V_j, the bearing N_j and the distance D_j of the j-th target ship from an own ship and the current course Ψ_0 and the speed V_0 of an own ship. After that the relative courses, speeds and bearings of target ships are calculated based on the input data. In the next step dangerous target ships are determined by the algorithm. These are target ships, that intersect their courses with the course of an own ship and might pose a collision risk. Afterwards the specific parameters of a firefly algorithm are initialized, such as β_0, α and γ and an initial population of n fireflies, constituting candidate trajectories, is generated. Light intensity for every firefly in the initial population is then calculated. The light intensity of a firefly describes how good is the solution defined by the current firefly for the considered optimization problem. Next, until the termination criterion is not fulfilled, the following steps are carried out iteratively:

– calculation of movement of every firefly towards the brighter firefly according to Equation (1),
– evaluation, whether every firefly is positioned within the solution space and does not exceed constraints (static and dynamic obstacles),
– update light intensities of newly obtained solutions (fireflies),
– rank the fireflies and find the current best trajectory.

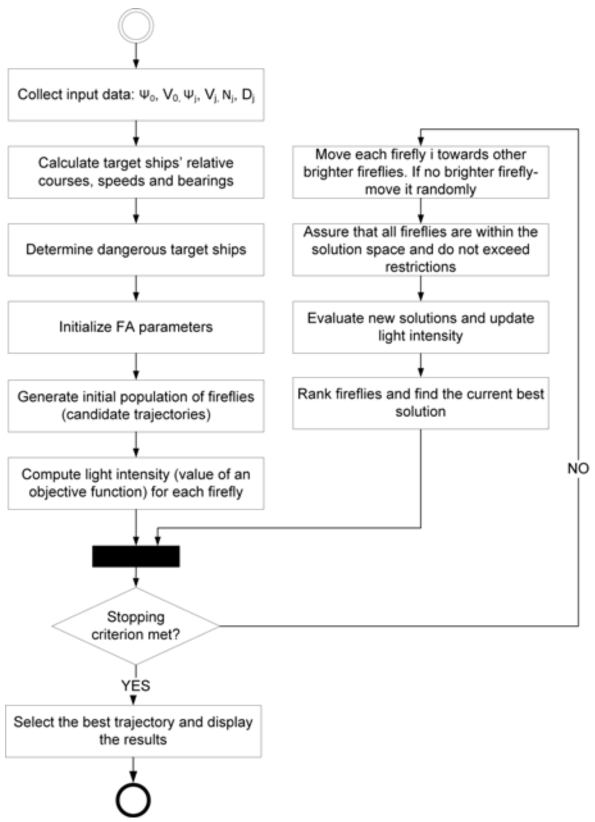

Figure 3. The flowchart of a firefly algorithm for ship collision avoidance

The termination criterion is the maximum number of iterations.

3 SIMULATION RESULTS

The firefly algorithm for ship collision avoidance was implemented in Matlab and compared with another SI algorithm, based on Ant Colony Optimization. The flowchart of the ACO algorithm for ship collision avoidance [30], used in the comparative analysis, is given in Figure 4. The applied ACO-based algorithm can be regarded as a constructive algorithm, as artificial ants construct their solutions by probabilistically choosing their next move on the graph until they reach the final waypoint. For comparison, applied firefly algorithm is a population-based algorithm that in a number of iterations changes the initial population of fireflies (candidate trajectories) in order to find the final best solution. The values of specific parameters of the algorithms used in the calculations are given in Table 1. The safe distance between the ships in the collision avoidance process is assured by the ship domain around target ships. A hexagon domain was used in the carried-out simulation tests, but other shapes might also be used, depending on the user's preferences. The dimensions of the target ship domain used in the calculations are as follows: distance towards bow: 1.05 nm, distance of amidships: 0.65 nm, distance towards starboard side: 0.65 nm, distance towards stern: 0.4 nm, distance towards port side: 0.4 nm. These are exemplary dimensions, which also can be changed according to the user's preferences. Tables 2-5 present input data of test case used as examples for the presentation in this paper. Numerical results are listed in Table 6 and a graphical presentation of the best trajectories calculated by both algorithms are given in Figures 5-8.

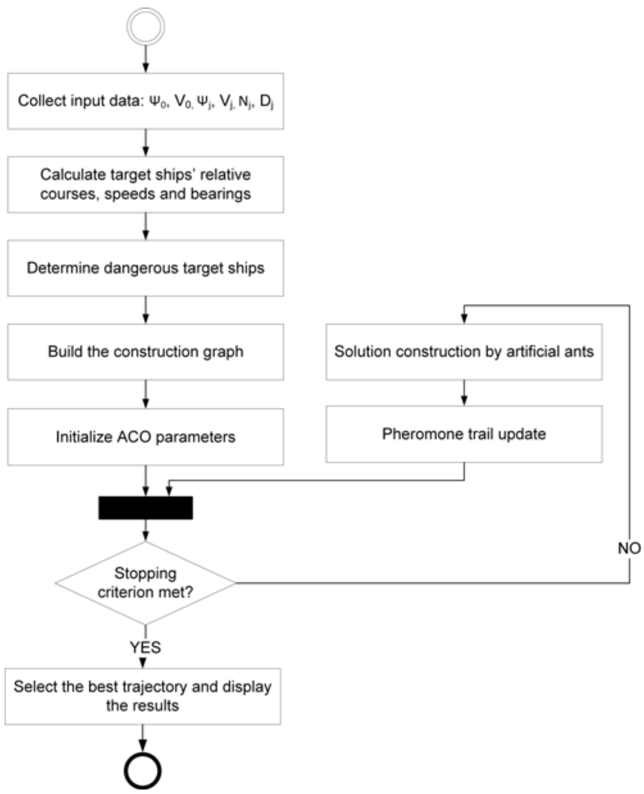

Figure 4. The flowchart of the Ant Colony Optimization-based algorithm for ship collision avoidance

Table 1. Values of ACO and FA parameters used in the calculations

Algorithm	Parameters
ACO	$\alpha = 1$, $\beta = 2$, $\rho = 0.1$, $\tau_0 = 1$, number_of_ants = 10, max_iterations = 20
FA	$\beta_0 = 1$, $\alpha = 0.4$, $\gamma = 1$, number_of_fireflies = 10, max_iterations = 50

FA-based algorithm calculated slightly shorter trajectories for all of the presented test cases. The run time of the algorithm was also shorter. For test cases 1-3 manoeuvres determined by both algorithms are similar, composed of a turn to starboard side and then return to the defined position. Interesting results might be noticed for a more complex test case 4 with 6 encountered ships. For this test case the difference between the trajectories calculated by both algorithms is more significant. ACO-based algorithm determined a trajectory composed of three course changes, with a port side turn, while FA-based algorithm calculated a starboard side manoeuvre, similarly as for the three other considered scenarios. The difference in length between the two proposed trajectories is 0.04 nautical mile, but the result of FA-based algorithm might be regarded as more practical.

Table 2. Input data for test case 1 with 2 target ships

Ship	Ψ [°]	V [kn]	N [°]	D [NM]
0	0	12	—	—
1	270	12	45	5
2	270	12	40	7

Table 3. Input data for test case 2 with 3 target ships

Ship	Ψ [°]	V [kn]	N [°]	D [NM]
0	0	12	—	—
1	190	12	5	5
2	270	12	40	7
3	270	10	55	6

Table 4. Input data for test case 3 with 4 target ships

Ship	Ψ [°]	V [kn]	N [°]	D [NM]
0	0	12	—	—
1	180	12	5	5
2	270	14	40	7
3	280	12	55	6
4	220	14	20	4

Table 5. Input data for test case 4 with 6 target ships

Ship	Ψ [°]	V [kn]	N [°]	D [NM]
0	0	14	—	—
1	105	4	340	8
2	180	12	5	5
3	270	14	40	7
4	280	12	55	6
5	220	14	20	4
6	190	8	45	5

Table 6. Results of the two algorithms for test cases 1-4

Test case	Algorithm	Distance [nm]	Ψ [°]	Run time [s]
1	FA	9.18	12,349	0.27
1	ACO	9.22	11,346	5.83
2	FA	9.21	7,352	0.69
2	ACO	9.25	9,342	5.21
3	FA	9.61	24,342	0.86
3	ACO	9.86	22,333	8.04
4	FA	9.93	28,337	3.22
4	ACO	9.97	333,18,315	7.48

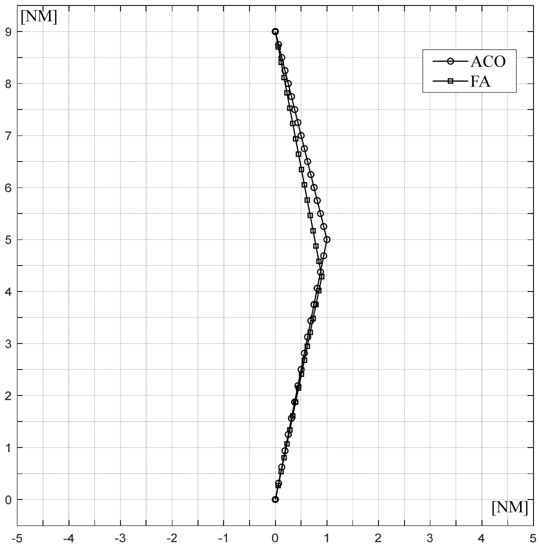

Figure 5. Comparison of safe trajectories calculated by the two algorithms for test case 1

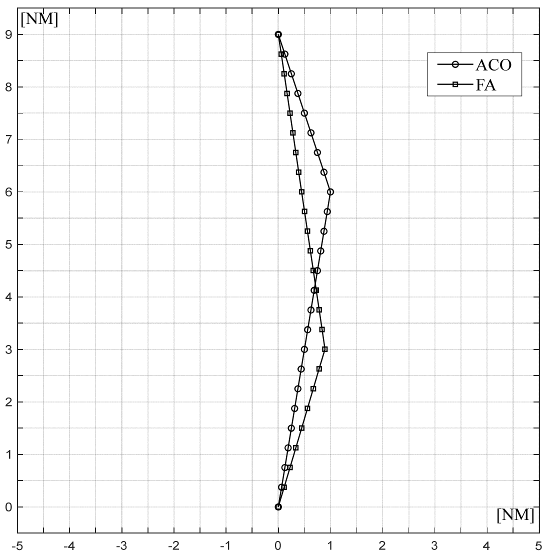

Figure 6. Comparison of safe trajectories calculated by the two algorithms for test case 2

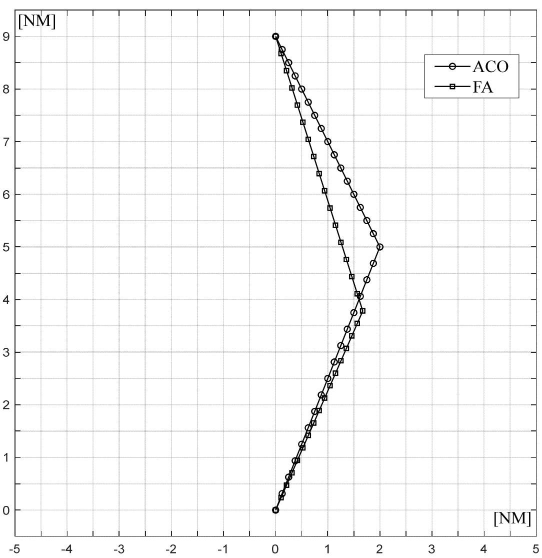

Figure 7. Comparison of safe trajectories calculated by the two algorithms for test case 3

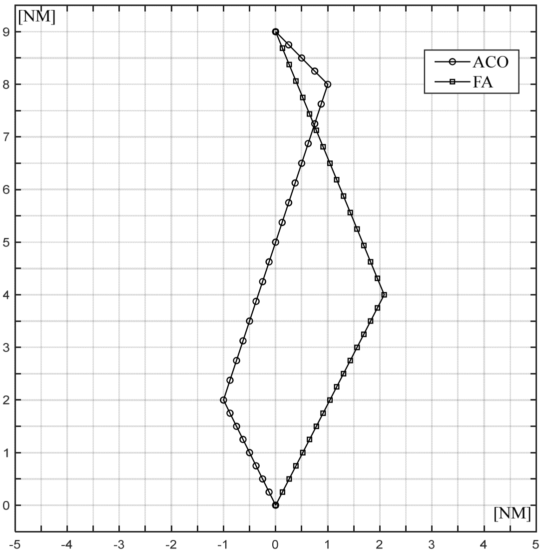

Figure 8. Comparison of safe trajectories calculated by the two algorithms for test case 4

4 CONCLUSIONS

The paper introduced an algorithm for solving the ship collision avoidance problem based one of the swarm intelligence methods – the firefly algorithm. Simulation results proved, that the firefly algorithm for ship collision avoidance is capable of solving the task in up to a few seconds, what is a reasonable amount of time for that process. Results compared with the ACO-based algorithm show, that the firefly algorithm obtains shorter trajectories within shorter calculation time. Therefore, the algorithm might constitute a competitive approach in the group of non-deterministic methods. Safe trajectory planning is a vital task in the navigation of ships. Such algorithms might be applied as a decision support tool on manned ships or as a part of an autonomous navigation system of unmanned or fully autonomous vessels. Future research direction might concern different algorithms applied for solving the safe trajectory planning problem, running in parallel, in order to finally propose the best solution for the considered collision situation. Future research should also regard evaluation of the algorithms with the use of scenarios with static obstacles and other waterway-related constraints.

REFERENCES

[1] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," Biosystems, Vol. 43 (2), pp. 73-81, 1997, doi: 10.1016/S0303-2647(97)01708-5.

[2] H. Hernández and C. Blum, "Distributed graph coloring: an approach based on the calling behavior of Japanese tree frogs," Swarm Intelligence, vol. 6, pp. 117–150, 2012, doi: 10.1007/s11721-012-0067-2.

[3] D. Karaboga and B. Gorkemli, "A combinatorial Artificial Bee Colony algorithm for traveling salesman problem," 2011 International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, Turkey, 2011, pp. 50-53, doi: 10.1109/INISTA.2011.5946125.

[4] C. Blum, M. Yabar and M. J. Blesa, "An ant colony optimization algorithm for DNA sequencing by hybridization," Computers & Operations Research, Vol. 35, No. 11, pp. 3620–3635, 2008, doi: 10.1016/j.cor.2007.03.007.

[5] T.T. Mac, C. Copot, D.T. Tran, R. De Keyser, "Heuristic approaches in robot path planning: A survey," Robotics and Autonomous Systems, Vol. 86, pp. 13-28, 2016, doi: 10.1016/j.robot.2016.08.001.

[6] M. Sugisaka and X. Fan, "An effective search method for NN-based face detection using PSO," SICE 2004 Annual Conference, Sapporo, Japan, 2004, pp. 2742-2745, vol. 3.

[7] R. Xu, G. C. Anagnostopoulos and D. C. Wunsch, "Multiclass cancer classification using semi-supervised ellipsoid art-map and particle swarm optimization with gene expression data," in The IEEE/ACM Transactions on Computational Biology and Bioinformatics, pp. 65 – 77, 2007.

[8] A. Kaveh, T. Bakhshpoori and M. Ashoory, "An efficient optimization procedure based on cuckoo search algorithm for practical design of steel structures," International Journal of Optimization in Civil Engineering, Vol. 2, pp. 1-14, 2012.

[9] X.-S. Yang, Z. Cui, R. Xiao, A. H. Gandomi and M. Karamanoglu, Eds., Swarm Intelligence and Bio-Inspired Computation: Theory and Applications (1st. ed.). Netherlands: Elsevier Science Publishers B. V., 2013.

[10] M. Tomera, "Ant colony optimization algorithm applied to ship steering control," Procedia Computer Science, Vol. 35, pp. 83-92, Proceedings of the 18th Annual Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES-2014 Gdynia, Poland, September 15-17, 2014, doi: 10.1016/j.procs.2014.08.087.

[11] M. C. Tsou and C. K. Hsueh, "The study of ship collision avoidance route planning by ant colony algorithm," Journal of Marine Science and Technology, Vol. 18 (5), pp. 746–756, 2010, doi: 10.51400/2709-6998.1929.

[12] A. Lazarowska, "Ship's trajectory planning for collision avoidance at sea based on ant colony optimisation," Journal of Navigation, Vol. 68, pp. 291–307, 2015, doi: 10.1017/S0373463314000708.

[13] C. Li-Jia and H. Li-Wen, "Ship Collision Avoidance Path Planning by PSO Based on Maneuvering Equation," in Future Wireless Networks and Information Systems, Lecture Notes in Electrical Engineering, Vol. 144, Y. Zhang, Ed., Springer, Berlin, Heidelberg, 2012, doi:10.1007/978-3-642-27326-1_87.

[14] M. Tomera, "Swarm intelligence applied to identification of nonlinear ship steering model," 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF), Gdynia, Poland, 2015, pp. 133-139, doi: 10.1109/CYBConf.2015.7175920.

[15] G. Wu, T. Xu, Y. Sun and J. Zhang, "Review of multiple unmanned surface vessels collaborative search and hunting based on swarm intelligence," International Journal of Advanced Robotic Systems, Vol. 19(2), 2022, doi:10.1177/17298806221091885.

[16] J. Lisowski, "Review of Ship Collision Avoidance Guidance Algorithms Using Remote Sensing and Game Control," Remote Sensing, Vol. 14, pp. 4928, 2022, doi:10.3390/rs14194928.

[17] M. Mohamed-Seghir, K. Kula and A. Kouzou, "Artificial Intelligence-Based Methods for Decision Support to Avoid Collisions at Sea," Electronics, Vol. 10(19), pp. 2360, 2021, doi:10.3390/electronics10192360.

[18] J. Lisowski, "Artificial Intelligence Methods in Safe Ship Control Based on Marine Environment Remote Sensing," Remote Sensing, Vol. 15, pp. 203, 2023, doi:10.3390/rs15010203.

[19] R. Szłapczyński and H. Ghaemi, "Framework of an Evolutionary Multi-Objective Optimisation Method for Planning a Safe Trajectory for a Marine Autonomous

Surface Ship," Polish Maritime Research, Vol.26 (Issue 4), pp. 69-79, 2019, doi:10.2478/pomr-2019-0068.
[20] G. Wei and W. Kuo, "COLREGs-Compliant Multi-Ship Collision Avoidance Based on Multi-Agent Reinforcement Learning Technique," Journal of Marine Science and Engineering, Vol. 10(10), pp. 1431, 2022, doi:10.3390/jmse10101431.
[21] L. Jiang, L. An, X. Zhang, C. Wang and X. Wang, "A human-like collision avoidance method for autonomous ship with attention-based deep reinforcement learning," Ocean Engineering, Vol. 264, pp. 112378, 2022, doi: 10.1016/j.oceaneng.2022.112378.
[22] Z. Zhu, H. Lyu, J. Zhang and Y. Yin, "An Efficient Ship Automatic Collision Avoidance Method Based on Modified Artificial Potential Field," Journal of Marine Science and Engineering, Vol. 10(1), pp. 3, 2022, doi: 10.3390/jmse10010003.
[23] Y. Cho, J. Han and J. Kim, "Efficient COLREG-Compliant Collision Avoidance in Multi-Ship Encounter Situations," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 3, pp. 1899-1911, March 2022, doi: 10.1109/TITS.2020.3029279.
[24] M. Rybczak and K. Podgórski, "Pareto Effect of LMI for Ship Propulsion," Applied Sciences, Vol. 11(16), pp. 7297, 2021, doi:10.3390/app11167297.
[25] M. Tomera, "Path Controller for Ships with Switching Approach," in Advanced, Contemporary Control. Advances in Intelligent Systems and Computing, Vol. 1196, A. Bartoszewicz, J. Kabziński, J. Kacprzyk, Eds., Springer, Cham, 2020, doi:10.1007/978-3-030-50936-1_126.
[26] H. N. Esfahani and R. Szlapczynski, "Robust-adaptive dynamic programming-based time-delay control of autonomous ships under stochastic disturbances using an actor-critic learning algorithm, " Journal of Marine Science and Technology, Vol. 26, pp. 1262–1279, 2021, doi:10.1007/s00773-021-00813-1.
[27] A. Witkowska and T. Rynkiewicz, "Dynamically Positioned Ship Steering Making Use of Backstepping Method and Artificial Neural Networks," Polish Maritime Research, Vol. 25(4) pp. 5-12, 2018, doi:10.2478/pomr-2018-0126.
[28] M. Dorigo and T. Stützle, Ant Colony Optimization. Cambridge, Massachusetts, London, England: The MIT Press, 2004.
[29] XS. Yang, "Firefly Algorithms for Multimodal Optimization," in Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Science, Vol. 5792, O. Watanabe, T. Zeugmann, Eds., Springer, Berlin, Heidelberg, 2009, doi:10.1007/978-3-642-04944-6_14.
[30] A. Lazarowska, "Ship's Trajectory Planning for Collision Avoidance at Sea Based on Ant Colony Optimisation," The Journal of Navigation, Vol. 68(2), pp. 291-307, 2015, doi:10.1017/S0373463314000708.