Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

An Intelligent Two and Three Dimensional Path Planning, Based on a Metaheuristic Method

B. Mahdipour, S. H. Zahiri [*], I. Behravan

Department of Electrical Engineering, Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran.

Article Info

Article History:
Received 25 May 2024
Reviewed 23 July 2024
Revised 27 August 2024
Accepted 28 August 2024

Keywords:
Particle Swarm Optimization (PSO)
Path planning
Autonomous Underwater Vehicle (AUV)
Unmanned Surface Vehicle (USV)

[*]Corresponding Author's Email Address: *hzahiri@birjand.ac.ir*

Abstract

Background and Objectives: Path planning is one of the most important topics related to the navigation of all kinds of moving vehicles such as airplanes, surface and subsurface vessels, cars, etc. Undoubtedly, in the process of making these tools more intelligent, detecting and crossing obstacles without encountering them by taking the shortest path is one of the most important goals of researchers. Significant success in this field can lead to significant progress in the use of these tools in a variety of applications such as industrial, military, transportation, commercial, etc. In this paper, a metaheuristic-based approach with the introduction of new fitness functions is presented for the problem of path planning for various types of surface and subsurface moving vehicles.

Methods: The proposed approach for path planning in this research is based on the metaheuristic methods, which makes use of a novel fitness function. Particle Swarm Optimization (PSO) is the metaheuristic method leveraged in this research but other types of metaheuristic methods can also be used in the proposed architecture for path planning.

Results: The efficiency of the proposed method, is tested on two synthetic environments for finding the best path between the predefined origin and destination for both surface and subsurface unmanned intelligent vessels. In both cases, the proposed method was able to find the best path or the closest answer to it.

Conclusion: In this paper, an efficient method for the path planning problem is presented. The proposed method is designed using Particle Swarm Optimization (PSO). In the proposed method, several effective fitness function have been defined so that the best path or one of the closest answers can be obtained by utilized metaheuristic algorithm. The results of implementing the proposed method on real and simulated geographic data show its good performance. Also, the obtained quantitative results (time elapsed, success rate, path cost, standard deviation) have been compared with other similar methods. In all of these measurements, the proposed algorithm outperforms other methods or is comparable to them.

Introduction

Intelligent surface and subsurface unmanned vehicles, are divided into two categories, unmanned surface vehicles (USV) and autonomous underwater vehicles (AUV). Intelligent subsurface vessels are actually unmanned submarines that are used to perform various missions. Exploration of special objects, mapping, inspection and troubleshooting of oil or gas pipes in the deep oceans, military operations such as exploration and neutralization of sea mines, etc. These floats have a rechargeable constant voltage source and all kinds of sensors to perform different missions. These sensors are used to check the surrounding environment and measure parameters such as water salinity, pollution spread in water, navigation, depth measurement, etc. The reason for using the word automatic (intelligent) in the naming of these vessels is the absence of human intervention or any other external factor in guiding and controlling them during the mission.

They are controlled and guided by a programmed processor installed on each of them. Path planning is one of the most challenging research topics related to intelligent subsurface vessels, which has attracted the attention of many researchers. Since these vessels are unmanned and even a foreign operator is not used to guide them, determining their movement path from the origin to the destination is very vital and important. It is possible to determine the path of the float from the origin to the destination in a way before the start of the movement, and how to do this optimally has been the subject of many researches so far. While moving towards the destination, the vessel must have maneuverability to avoid collision with moving obstacles, such as other vessels or underwater creatures. In simpler words, the vessel must be able to leave the predetermined course in an emergency and then return to the course again.

Therefore, finding a suitable Path for the float to move from the origin to the destination is an important challenge in the future. However, the main goal in this research is path planning for the fleet of subsurface vessels. In many missions, for example military missions, a group of vessels must move towards the designated destination in cooperation with each other. The need to move the fleet with a specific arrangement towards the destination makes the path planning process more complicated. Because each vessel in the group must, in addition to moving towards the destination, maintain its position relative to other vessels in the considered formation. This problem becomes more complicated when due to the existence of numerous moving obstacles in the underwater space, the vessels must change their position to avoid collision with these obstacles and at the same time think about maintaining the group arrangement. For this purpose, communication between the vessels and also monitoring the surrounding environment at the moment is of high importance. Therefore, two general goals can be considered for this part of the project:

a. Designing the path planning module to determine the movement path of vessels from the origin to the designated destination. This module should be able to determine a reasonable path from the origin to the destination according to the obstacles in the underwater space.
b. Modular design to maintain the arrangement of floats while moving. At every moment of movement, this module determines the position of each vessel relative to other vessels, as well as the vessel that plays the role of the leader. Also, this module should be flexible so that when a vessel encounters an obstacle, it can change its course and return to its designated position in the group while maintaining the overall arrangement of the fleet.

Many studies have been reported on different methods of path planning for subsurface vessels. A new approach that is leveraged in research and has attracted the attention of researchers due to its ease of implementation, is the use of metaheuristic methods and biological algorithms. For further investigation, the researches related to this field are mentioned below.

In the problem of path planning in smart unmanned surface and subsurface vessels, we are faced with problems such as vessels colliding with each other, high energy consumption due to long and bad route determination, failure to maintain the relative distance of the vessels to each other, failure to maintain the overall shape of the fleet arrangement, large amount of calculations , high response time in route determination, limitations in sonar data, the unknown external disturbance, etc., which make routing scenarios more complicated. In this research a novel path planning methodology based on PSO algorithm is proposed to tackle these problems by defining new and simple fitness function.

One important issue in path planning in real environment is the unknown external disturbance, such as external waves, underwater currents and sea creatures. The proposed algorithm is designed in such a way that by saving time, it corrects the path towards not hitting the immediate obstacle. It can be considered as a factor for more reliable path planning in real environment.

Literature Review

A. An Overview 0f the Work Done in the Field of Navigation for a Subsurface Vessel

In 2001, a method based on genetic algorithm for subsurface navigation was proposed by a group of European researchers. In this method, the three-dimensional space between the origin and the destination is first gridded. That is, the first try to determine the points of the space that are between the origin and the destination. Then the genetic algorithm starts searching to find the best possible path among these points. In fact, each chromosome contains a sequence of these points that connect them together to create a path between the origin and the destination.

In this research, the objective function is designed in such a way that the best path is the path that the float consumes the least possible energy to travel. As a result, Paths with shorter length are better Paths. Also, to simulate the static obstacles in the problem space, some points are randomly considered as obstacles, and this means that every path provided by each chromosome must be free of such points. It seems that networking the problem space is not a practical solution to find the right path. However, the proposed method can be effective for short-range path planning. Also, the results presented in this research have not been compared with any other method, which makes it difficult to reach a deep insight into the proposed method [1].

In 2007, a hybrid method was presented by a researcher named Zhang. In this method, which is a combination of genetic algorithm and Octree method, first, the problem space is divided into cells with a certain size. This work, is done with the aim of detecting empty areas and also detecting areas where there are static obstacles. Then the extracted cells are classified into three groups. The first category are cells that do not have any obstacle in their range. The second category are cells that are completely filled by a barrier, and the third category are cells that are partially filled by a barrier and are so-called half-filled. In the next step, the genetic algorithm searches the three-dimensional space to find a channel between the source and the destination among the empty and semi-empty cells. As a result, each chromosome contains a sequence of empty and semi-empty cells that can be followed from the origin to the destination. It should be noted that for half-empty cells, the Octree method divides the cell space into smaller sub-cells.

This increases the speed and accuracy of the presented method. In addition to this, other changes have been made in the genetic optimization algorithm. Including changing the intersection operator in such a way that the new chromosome produced from the intersection of two parent chromosomes contains the appropriate path. For this purpose, if two parent chromosomes contain a common cell, crossover is done from the location of this cell in the chromosomes and 4 new children are created, and if the parent chromosomes do not contain a common cell, crossover is not done. Similar to the previous research, the objective function is designed in such a way that paths with shorter length have better fitness value. Furthermore, the higher the number of empty cells in the designated Path, the better Path is considered. Despite Zhang's claim about the practicality of this method, it seems that the biggest weakness of the method is its dependence on the division of space into different cells, which makes the method unsuitable for long distances. In addition, the performance of the proposed method has not been scientifically compared with any other methods [2].

In 2011, in a joint project conducted by researchers from Hong Kong Polytechnic University and University of Calgary, Canada, a hybrid optimization method was presented to solve the subsurface floating path planning problem. In this method, which was designed by combining genetic algorithm and dynamic programming optimization method, the goal was to find the shortest and smoothest possible path to reduce the energy consumption of the float. In the presented method, since each chromosome contains a possible path between the origin and the destination, performing the normal and conventional Cross over in the GA algorithm produces unreasonable and inappropriate paths. To solve this problem and also reduce the probability of getting stuck in the local optimal point, the dynamic programming method has been used.

In the utilized model, after selecting two parent chromosomes, the DP method is used to extract reasonable paths among the parent chromosomes. The cost function or in other words the objective function whose minimum point should be found, is the weighted sum of three parameters:

a. The length of the Path provided, definitely shorter paths are better due to less energy consumption.
b. Sudden change of direction which is calculated by measuring the angle of direction change. Certainly, changing directions increases energy consumption. Therefore, the paths in which there are fewer such changes of direction are considered more suitable paths.
c. Sudden change of height, which is similar to the previous two parameters. In this case, the Paths are also desirable where the sudden change of height is less. The proposed method has been tested on two artificial environments and its performance has been compared with the conventional GA method.

The obtained results show the superiority of the proposed method over the conventional method. Despite the fact that the proposed method seems to be practical, simulating only on two environments without considering fixed obstacles, causes a little doubt in the accuracy and efficiency of this method. Also, only comparing the performance of the proposed method with another algorithm is not a suitable criterion for evaluating the proposed method [3].

In 2014, two Chinese researchers named Hui and Xiaodi proposed an interesting method called HPF (Heuristic Potential Field) to solve the problem of undersea path planning. In this method, unlike most of the presented methods, path planning is done in real time by the float itself. Therefore, unlike other methods, here the conditions of the problem environment and the position of fixed and dynamic obstacles are not known. Rather, the float monitors the surrounding environment at any moment using various sensors, including sonar sensors, and if there is an obstacle in the visible range of the sensors, it will correct its course to avoid collision with the obstacle.

At the beginning of the work, a direct Path between the origin and the desired destination is determined as the main Path. Then the float starts moving on a straight path and checks the conditions of its surroundings at every moment. As soon as an obstacle is observed in the path ahead, the coordinates of the points of the obstacle that intersect with the path are determined and a new path is found using these coordinates. Therefore, at any moment the path of the float towards the destination is known and the float is moving on the specified path. As soon as an obstacle is observed by the sensors installed on the float, the Path correction module is activated and finds a new Path, and this process continues until reaching the destination. In spite of the fact that many details of the path planning method of the article in question are not mentioned and the method in question is presented in general terms, but it seems that the HPF method has the ability for hardware implementation due to the assumption of not knowing the geographical characteristics the problem environment is very close to reality. In addition, by using this method, it is also possible to navigate despite moving obstacles. However, the problem with this method is that the simplicity of this method may cause it to get stuck in the local optimal point or find a very long path. This problem can be solved by using optimization methods, especially meta-metaheuristic algorithm [4].

In 2015, researchers from Flinders University in Australia presented a dynamic method for subsurface navigation. In this method, at every moment of the Path, the path planning algorithm is executed and a new Path is found between the current point and the determined destination. Then, a controller module checks the new Path by checking the environmental conditions, including static and dynamic obstacles, as well as the direction of the water currents. In this research, similar to the previous researches, the path planning problem has been formulated as an optimization problem, and the Quantum behaved Particle Swarm Optimization (QPSO) has been used to solve it.

In this method, at first, the path of the float is determined from the starting point to the destination. Then the float starts moving and until reaching a certain point of the determined Path, the path planning module has the opportunity to find a new Path from the next point to the destination. Therefore, in certain time intervals, the Path of the float is updated from the point where it is present to the designated destination. Similar to other methods, the time to reach from the origin to the destination has been the most important criterion for the design of the objective function. The difference between the objective function considered in this method is that the direction of the water currents is also considered. It seems that in this method, path planning is done by dividing the three-dimensional space into different areas [5].

In 2016, a group of Australian researchers presented an interesting method for subsurface navigation. This method, which consists of two modules, first finds a general path between the origin and the destination using the genetic algorithm, which is similar to the previous methods introduced here, and the goal is to find the shortest possible path between the origin and the destination. Also, to find this path, the space between the origin and destination is gridded and the points that can be considered as positions are extracted. Therefore, the genetic algorithm finds the best possible sequence among the available points to find the path. Then the next module processes the sequence found by the genetic algorithm. In this module, the PSO algorithm searches the space between two consecutive points in the sequence to find a suitable path between these two points.

Therefore, first a general Path is determined and then other suitable paths are found between any two points of the general path. In the second module, the objective function is time. That is, the PSO algorithm searches the space with the aim of finding the shortest path. Although, the proposed method is an interesting method and it seems that this idea can be used, but it seems that gridding the problem space in the first stage is the main weakness of this method. The research team has also implemented their proposed method using the Imperialist Competitive Algorithm ICA (Imperialist Competitive Algorithm) and compared the results obtained from the implementation of the proposed method with GA and QPSO algorithms. The obtained results show that, on average, the ICA algorithm finds the optimal Path between the origin and the destination in a shorter period of time [6].

In one of the latest studies conducted in 2020, a method similar to the previous methods has been presented. The only difference is the utilization of the Ant Colony Optimization (ACO) for path planning. In this research, the ACO algorithm has only sought to find the best possible sequence among the available points between the origin and destination of the path. This method does not have any serious superiority over the other introduced methods, and the purpose of the project was only to show the power of the ACO algorithm in solving such problems [7].

As it is known, solving the path planning problem requires the use of optimization methods, which metaheuristic and meta-metaheuristic optimization algorithms are very suitable tools to solve this problem. Basically, these algorithms are designed and used to solve complex optimization problems with high dimensions. The high power and flexibility of these methods and of course their acceptable accuracy have made them popular tools in many researches in different scientific fields. However, other methods have been presented so far to solve this problem, which are briefly introduced in the Table 1. The most important weaknesses of these methods are the high probability of getting stuck in the local optimal point, the large number of calculations, and the weakness in facing the dynamic environment, which makes their use inappropriate for long distances.

B. A Review of the Work Done in the Field of Navigation for a Group of Subsurface Vessels

In 2014, a method based on genetic optimization algorithm for path planning a group of subsurface vessels was presented by a Chinese research team. This method is presented with the premise that the static obstacles in the 3D space are already detected (for example by the sonar sensors of a surface vessel). There is also a set of points that the subsurface vessel can pass through on the way to the destination. Using these points, genetic algorithm finds a path between the origin and destination of the desired float for each float. Therefore, each vessel has a specific origin and destination, and the desired method simultaneously finds the optimal. Similar to previous researches, in this method, the cost function is designed with the aim of finding the shortest possible path.

In order to prevent the collision of the floats, if the paths considered for two floats, which are specified by the chromosomes, are crossed, the time of each float's arrival at the desired point is checked. If two vessels reach the point of intersection at the same time, the Paths are corrected. Ignoring dynamic obstacles in the way is the problem of this method. In addition to this, the method proposed in this research has no fundamental difference with the methods related to the path planning of single vessels and only deals with the problem of their non-collision with each other. While one of the most important path planning challenges for the fleet of subsurface vessels is maintaining their formation while moving towards a specific destination [8].

In 2015, two Chinese researchers presented a method based on SOM (Self-Organizing map), BINM (Biological inspired neurodynamics model) and VS (Velocity Synthesis) algorithms for Task alignment and subsurface vessel group path planning. In this method, the generality of which is similar to the previous method, the SOM network is in charge of task alignment and path planning of each vessel. That is, at the beginning of the work, the SOM network determines a destination for each vessel from among the available destinations and predicts the Path to reach that destination. In other words, as in the previous method, in this method, the goal is not to move in a group and coherently towards a specific destination, but the mission is to go to several specific destinations and perform specific operations that the SOM decides which of the vessels will go to which destination. Criterion of task alignment by SOM is the lowest energy consumption.

Therefore, each float will move to a destination that is closer to its initial position. The BINM algorithm is intended to avoid the collision of the float with static obstacles. What differentiates this method from the previous method is the consideration of water currents in the navigation of vessels, which the VS algorithm is responsible for. However, the lack of modeling of dynamic obstacles and the lack of real-time processing are the main drawbacks of this method [9].

In 2020, a research team consisting of 7 Chinese researchers presented a hybrid method to solve the task division and pathing problem of a group of subsurface vessels. What differentiates this method from other methods is considering the point of release of the floats in the water and the point of their return to the water surface as the beginning and end points of the paths intended for them. Therefore, the energy used for the return path as well as the initial path should be calculated and included in the objective function. On the other hand, due to the fact that the number of destinations (target points) may be high and the distance between them is also long, the initial release points of each float should also be optimally determined. Therefore, a mission or task intended for a vessel is to go to the target point or points intended for that vessel, and to perform this mission, the vessel's path must be determined in an optimal way.

As a result, in the first part, the problem of task division and determination of initial points is done based on the divided tasks. This work is done using the differential evolution optimization algorithm (DE). In the second part, the Ant Colony Optimization (ACO) algorithm is implemented to determine the path corresponding to each mission. The objective function in this algorithm is designed in such a way that the intended path is traveled with the least possible energy consumption and without encountering obstacles in the path. In this method, the vessels operate independently of each other and each carries out the assigned mission. Therefore, there is no need to communicate between the vessels during the mission. On the other hand, the non-collision of the floats with each other means considering the dynamic obstacles in the problem [10]. In recent research conducted by a group of Indian researchers, a method for pathing a group of subsurface vessels using the gray wolf optimization algorithm (GWO) is presented. Similar to the previous methods, the optimization algorithm determines the optimal path of each vessel towards the destination by knowing the conditions of the three-dimensional space. In simpler words, the position of static obstacles as well as the points in the three-dimensional space are already known. Therefore, the GWO algorithm performs path planning using these points and also considering the position of obstacles. The difference between this method and the previous two methods is the group movement of floats with a specific arrangement towards the destination. Therefore, the cost function is designed in such a way that in addition to finding the shortest path, the fleet arrangement is also maintained. Despite the researchers' emphasis on maintaining the arrangement of the vessels, only the relative distance of the vessels from each other has been considered. In addition to this, another big drawback of this method is not considering dynamic obstacles [11].

Many other methods have been presented so far to solve the problem of subsurface vessel pathing, surface vessel Pathing, Quadrotor motion pathing, etc, some of which are listed in the following table. Also the mentioned papers can be classified based on methodology in Table 1 format:

Table 1: A summary of the researches related to subsurface fleet Pathing

Reference Number	Year	Utilized Method	Merits & Demerits Analysis
[12]	2015	Kalman Filter	• Planning can be adjusted according to the dynamics of the environment. • Low computation cost • Only simulation results are obtained
[13]	2020	Fuzzy Logic & Ant Colony Optimization System	• Weak optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Not suitable for fast moving AUVs • Energy optimal trajectories are not obtained with collision avoidance
[14]	2017	Deep Reinforcement Learning (DRL)	• Near optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[15]	2006	Fuzzy Logic	• Weak optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Not suitable for fast moving AUVs • Energy optimal trajectories are not obtained with collision avoidance
[16]	2019	Neural Networks	• Near optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[17]	2008	Mixed Integer Linear Programming (MILP)	• Weak optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Not suitable for fast moving AUVs • Energy optimal trajectories are not obtained with collision avoidance
[18]	2019	Genetic Algorithm	• Minimizes the time expanses • Searches the solution from a large solution space • GA requires effective memory management • DE provides time optimized path in the corridor area but untimely collisions in obstruct evaluation of some good paths • The cost of computation is high

[19]	2018	Swarm Intelligence	• Near optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[20]	2014	Hybrid Genetic Algorithm & Particle Swarm Optimization	• Weak optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Not suitable for fast moving AUVs • Energy optimal trajectories are not obtained with collision avoidance
[21]	2023	Genetic Algorithm	• Minimizes the time expanses • Searches the solution from a large solution space • GA requires effective memory management • DE provides time optimized path in the corridor area but untimely collisions in obstruct evaluation of some good paths • The cost of computation is high
[22]	2023	Genetic Algorithm	• Far optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • GA requires effective memory management • Fuzzy-PID requires effective memory management • Energy optimal trajectories are obtained with collision avoidance
[23]	2023	Fuzzy Logic	• Near optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Not suitable for fast moving AUVs
[24]	2023	A Comprehensive Review of Path Planning Algorithms	• Near optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[25]	2023	Machine Learning	• Far optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[26]	2024	Artificial Potential Field Method & Multi-Algorithm Fusion	• Near optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[27]	2024	Fuzzy Logic & Simulated Annealing	• Far optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[28]	2024	Genetic Algorithm	• Near optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[29]	2024	Machine Learning	• Weak optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Not suitable for fast moving AUVs • Energy optimal trajectories are not obtained with collision avoidance

[30]	2024	Machine Learning	• Near optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[31]	2024	A Comprehensive Review of Path Planning Algorithms	• Far optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[32]	2024	Machine Learning	• Near optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[33]	2019	Particle Swarm Optimization (PSO)	• Weak optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Not suitable for fast moving AUVs • Energy optimal trajectories are not obtained with collision avoidance
[34]	2021	Particle Swarm Optimization (PSO)	• Far optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[35]	2022	Particle Swarm Optimization (PSO)	• Near optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Not suitable for fast moving AUVs
[36]	2024	Particle Swarm Optimization (PSO)	• Near optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[37]	2023	Particle Swarm Optimization (PSO)	• Weak optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Not suitable for fast moving AUVs • Energy optimal trajectories are not obtained with collision avoidance
[38]	2024	Artificial Potential Field Method & Multi-Algorithm Fusion	• Near optimal path following control for AUVs both in two- and three-dimensional environments • Computation complexity is high • Suitable for fast moving AUVs • Energy optimal trajectories are obtained with collision avoidance
[39]	2022	Machine Learning	-

In general, similar to the Pathing mode for a single vessel in the case of subsurface fleet Pathing, we are faced with an optimization problem, but more complicated, which prompts us to design the desired cost function by considering more restrictions. These restrictions, such as preventing vessels from colliding with each other, maintaining their relative distance from each other, maintaining the general shape of the fleet arrangement, etc, make the problem more complicated. As a result, solving this complex problem requires the use of powerful optimization tools that have a high ability to effectively search the response space and escape from the local optimal point. metaheuristic and heuristic optimization algorithms are among the most widely used and also the most popular optimization algorithms that have been used many times in recent years to solve complex problems that are difficult and time-consuming to analyze with other methods. The high ability of these algorithms in finding the overall optimal point, very good convergence, high flexibility and easy implementation are among the advantages of these algorithms. In recent years, as seen, these algorithms have been used in path planning problems for all kinds of vehicles, and the results obtained show their high efficiency.

As seen, with an extensive review of the work done in Table 1, the above-mentioned contents were presented. All these papers are involved with the problem of path planning and have attempted to solve it. But what has been the main motivation for writing this paper are two very important issues, which are:

a. Calculation time required to find the best path. This problem is present in most references in such a way that it makes them out of practicality at the moment. In other words, the calculation time related to path planning in most of these researches is such that due to the type of sonar used in the vessels, practically the power of maneuvering and correcting the course is taken away from them and encountering an obstacle is inevitable.
b. Another important issue in the movement of a group of movers is the emphasis on maintaining their organization and group arrangement in most of the mentioned references. This point is somehow related to the problem mentioned in part a. That is, if a mover in a group of vessels encounters an obstacle, the time required to correct its course is often such that the possibility of returning to the previous arrangement and organization is denied.

In this research, the main motivation is to improve the two aforementioned problems, which have been tried to be solved by defining a simple fitness function.

Designing a Path Planning Module for a Group of Intelligent Subsurface Vessels

The final goal of this section is to provide a solution for the movement of the fleet of subsurface vessels towards the destination. In other words, a group of automatic subsurface vessels must move from a specific origin to a specific destination to perform a specific mission. For this purpose, two modules have been designed that operate in a hierarchical manner. The first module, which is called the path planning module, is responsible for finding a suitable Path between the origin and the destination, taking into account the presence of obstacles.

In fact, the output of this module is a set of consecutive points that determine the right path to move from the origin to the destination and are provided as input to the second module. The second module is called the path follower module. This module manages the movement of the fleet of subsurface vessels by keeping a certain formation in mind. In other words, this module determines the momentary position of each of the vessels according to the determined Path, the position of the vessel in the group and dynamic obstacles. In the following, first the Pathing module is fully described, then the functioning of the second module is explained.

Path Planning Module

Similar to many complex engineering problems, the path planning process can be turned into an optimization problem and then solved using different methods introduced to solve optimization problems. For this purpose, the first step is to design a cost function, finding its optimal point is equivalent to solving the main problem. This function is actually a kind of modeling of the desired problem space, which can be used to find a suitable answer for the problem. After designing a suitable cost function that models the problem space well, the next step is to choose a suitable method to solve the optimization problem. Various factors (such as execution speed, performance accuracy, and ease of implementation) have been considered for choosing the appropriate metaheuristic algorithm. In this paper, the PSO algorithm is used to solve the path planning problem. It is used in three-dimensional space. In the following, first the PSO algorithm is introduced, then the method of using this method in the design of the path planning module is described.

A. Particle Swarm Optimization Algorithm

The Particle Swarm Optimization (PSO) algorithm is a metaheuristic optimization algorithm inspired by the group movement of birds. In this algorithm, each particle whose position represents an answer to the problem is a member of a community that seeks to find the optimal answer collectively and inspired by the collective movement of birds. This algorithm was first proposed by Eberhart and Kennedy in 1995. The PSO algorithm uses the interaction between particles to find the optimal point. In this way, in the population created in each iteration, the particle that has the best fitness is known as the leader, and the rest of the particles tend to approach its position with the mechanism described below. Therefore, in this method, the elements cooperate to reach the optimal point. In this algorithm, after creating the initial population, each particle moves to find the optimal point in the response space. The movement of the particle is affected by two parameters:

a. The best position that the particle had from the beginning of the algorithm until the current iteration.
b. The position of the leader.

Therefore, each particle, in addition to searching the space individually, also looks at the position of the group leader and tends to approach his position. In fact, the best individual position has a function similar to the mutation operator in the genetic algorithm. In this algorithm, the movement of the particle is determined by the velocity vector. That is, after the velocity vector of the particle is determined, by adding the velocity with the current position, the new position of the particle is obtained, which means the new answer. So, as mentioned above, each particle in the PSO algorithm, in addition to its position, has a memory where it stores the best position it has had so far. It also has a velocity vector that determines its position in the next iteration. The speed of each particle is obtained from the following equation:

$$v_i^{t+1} = w.v_i^t + c_1.r_1(p_{leader}^t - y_i^t) + c_2.r_2.(p_{i_{best}}^t - y_i^t) \quad (1)$$

In this relation, v_i^{t+1} is the speed of the particle in the next iteration and v_i^t is its speed in the current iteration. p_{leader}^t is the position of the leader in the current iteration and $p_{i_{best}}^t$ is the best position that the particle had from the beginning of the algorithm execution to the t th iteration. y_i^t is the current position of the particle. The coefficients $c_1$and c_2 determine whether the particle will seek answers more individually or follow the leader of the group. As it is clear from the relationship, if c_1 is larger, the particle will seek to reach its best individual position. On the contrary, if c_1 is larger, the particle likes to approach the position of the leader, which has the best position among other particles in an iteration. c_1 is called Cognitive factor and c_2 is called Social factor. Usually, these two coefficients are chosen equal to 2. It has been found experimentally that the value of 2 for these two coefficients leads to the best performance. In (1), the constants r_1 and r_2 are two random numbers in the interval [0 1] that have been added to the relation to randomize the search for the answer space. Also, w is used to control the search process.

The largeness of this coefficient causes the search for a wider area of the response space, while its decrease prevents the scattering of particles. The value of this coefficient is high at the beginning of the algorithm execution and its value is low in the final iterations. Because in the end, it is better for the answers to converge. After the velocity of the particle is determined, its position is updated using (2):

$$y_i^{t+1} = y_i^t + v_i^{t+1} \quad (2)$$

The steps of implementing the PSO algorithm are as follows:

a. Creating a random initial population.
b. Calculating the fitness of particles and determining the leader.
c. Calculation of the speed of each particle.
d. Updating the position of particles.
e. Calculation of the best individual position of each particle.
f. If the condition is fulfilled, end the loop or repeat steps b to f otherwise.
g. Presenting the best answer found as the final answer of the algorithm.

B. Path Planning Using PSO Algorithm

In general, to solve any optimization problem using metaheuristic and heuristic algorithms, two basic points should be considered:

a. The structure of search agents, which are called particles in this algorithm.
b. Existence of a suitable objective function to calculate the fitness of particles. This objective function is clear in some cases, but in many cases, an objective function must be defined for the problem in question.

To find a suitable Path between the origin and the destination, the PSO algorithm finds a point in each run, and each point is found according to the previously found point. Therefore, in order to find the right path between the origin and the destination, the PSO algorithm must be executed in the right number so that the found points establish a suitable path between the origin and the destination. Therefore, the optimization problem is to find the next best possible point of the Path. For this purpose, we must first explain the structure of search agents (particles) and the objective function well.

B. 1. The Structure of the Search Factors

In each run, the coordinates of a point of the path must be provided as output by the PSO algorithm. This point is the best possible point according to the limitations of the problem and the considered objective function. Therefore, the particles must be three-dimensional. In other words, every time the PSO algorithm is executed, a six-dimensional vector is provided as output, which determines the location of the subsurface float. It is clear that the length, width, depth and roll and yaw angles determine the location of the float.

B. 2. Objective Function

The objective function should be designed in such a way that the next best possible point is its minimum (optimal) point. For this purpose, the limitations of the problem should be considered and based on that, a general definition of the minimum point should be provided and then the objective function should be designed. It can be said that the minimum point of this problem must have the following conditions:

a. The desired point should not be inside a three-dimensional object (obstacle). As a result, the objective function must be defined in such a way that its value is very high for points inside the range of obstacles.
b. This point should not be too close to the obstacle.
c. This point must be found on the way to the final point. In other words, this point should be located in a place that is considered a step forward and towards the destination compared to the previous point.

Therefore, in a minimization problem, we must define the objective function in such a way that if the new point is one step behind the previous point, the output of the function will be a large value so that this point is not considered a suitable point.

According to these assumptions, we consider the objective function as the weighted sum of six functions, each of which is introduced below:

Function f1: If the coordinates of a particle are inside a barrier (inside a three-dimensional object), the value of this function will be infinite, and if it is outside it, its value will be zero.

Function f2: the inverse of the distance between the point and the edge of the obstacle. In fact, adding this function is to prevent the point from getting too close to the obstacle.

Function f3: This function is defined so that if the position (coordinates) of the particle is not in the direction of reaching the final point, this particle is considered as a poor quality particle. For this purpose, the angle formed between the vector consisting of the previous point and the new point and the vector consisting of the new point and the final point must be minimized.

In the following, the said material will be further examined in the form of an example. Fig. 1 shows the concept of three functions f1, f2 and f3. To make it easier to understand the functions, this figure is designed for the two-dimensional problem. That is, it is assumed that the depth is constant and only longitude and latitude should be considered as input parameters. The goal of Pathing is the distance between the points marked with red and blue stars. The direction of movement is from the side of the red star to the blue star. In this figure, the obstacle (which is land) is marked with green color. Of course, the Path points should not fall within this range. In this image, one of the path points found by the PSO algorithm (with the mentioned settings) is drawn, and the red arrow shows the distance from the desired point to the obstacle border. In addition, the angle between the two described vectors is also indicated by the term tetta. If this angle is equal to zero, the point will be placed in the direction of the vector formed between the previous point and the final point. Definitely, from the point of view of this function, the point for which tetta is equal to zero is the optimal point. Since the found point is outside the obstacle range, the value of f1 function is zero for it. This point (particle) is the minimum point of the sum of functions f1, f2 and f3.

Function f4: This function is equal to the difference of the distance of the particle to the previous optimal point and a threshold value. The considered threshold value specifies the minimum distance between two consecutive points. In fact, this threshold value is something similar to the sampling rate.

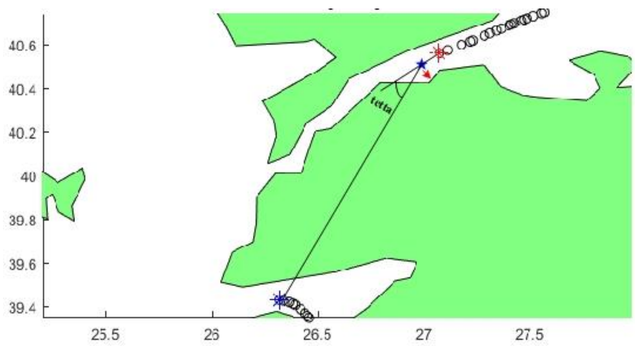


Fig. 1: Definition of functions f1, f2 and f3.

In other words, the new point must be at a minimum distance from the previous optimal point, which is determined by the threshold value entered by the user. The f4 function being zero means that the new point is at a minimal distance from the previous point.

Function f5: This function is defined to prevent the sudden change of direction of the path. In this function, the angle between the following two vectors is calculated:

a. The vector consisting of the new point and the previous point of the path (previous optimal point).
b. The vector consisting of the previous two points. If the angle between these two vectors is more than 90 degrees, the new point is not a suitable point and the f5 function value will be infinite for it. Otherwise, the f5 function value will be considered as zero.

Function f6: This function is defined to detect the optimal path leading to the destination (final point). First, a direct Path is drawn between the point under investigation and the destination. Then, among the points in this straight path, the number of points that fall within the obstacle range is considered as the value of f6 function. Certainly, the lower the f6 function value, the better the new point. Fig. 2 shows the concept of function f6 and the necessity of defining this function. Similar to the previous case, the reason for using a two-dimensional image is to better understand the conditions in which the use of the f6 function is necessary. According to this figure, it can be seen that the algorithm has found a part of the path. Now (supposedly) the next point should be selected from the points marked with red and black circles. The black circle is definitely a better choice because the value of the f6 function for it is lower than the corresponding value for the red dot. In fact, this function has been added to find the path leading to the destination and prevent getting lost.

In other words, the modeling of the problem space should be done in such a way that the value of the objective function (fitness) is lower for the blue circle. In simpler words, this point should be a better point for the algorithm, and the reason is clear.

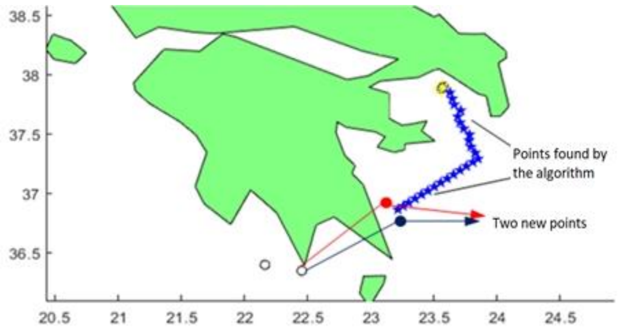


Fig. 2: The necessity of defining the f6 function in detecting the right path.

If the red circle is selected, the path to the final destination will be disturbed and the algorithm will have trouble finding the path. Because if you choose the red circle, the next points will be found in the same direction (i.e. away from the destination) and this means not finding the right path. The f6 function is intended to fix this problem and create an advantage in the blue dot over the red dot (as well as similar conditions).

Finally, the objective function is defined as the following relation:

$$objective\ function = f_1 + coeff_1 \times f_2 + coeff_2 \times f_3 + f_4 + f_5 + f_6 \quad (3)$$

The coefficients $coeff_1$ and $coeff_2$ are between zero and one and change during the execution of the process. The reason for this is the change in the importance of functions f_2 and f_3 at the beginning and end of the Pathing process. For example, at the beginning of the Pathing process, if the found point is not on the path between the previous point and the final point, there will not be much of a problem, but if it is very close to the border, the desired point is not a desirable point. On the contrary, at the end of the process, when we are close to the destination, the point must be in the direction of reaching the destination, and there is no problem if it is close to the Mazer. Therefore, at the beginning of the process, $coeff_1$ has a value close to one, and during the process, every time the PSO algorithm is executed, its value decreases, and for $coeff_2$ this procedure is considered completely opposite.

We know that the direct Path between origin and destination is the shortest Path. Therefore, if possible, the best answer for the Pathing problem is the direct Path. According to the previous explanations, we also know that the designed module seeks to find a point of the Path between the last found point and the main destination at any moment. As a result, it is possible to have a direct Path between the current origin (the last found point of the Path) and the destination. Therefore, at each stage, before the PSO algorithm is implemented, it is first checked whether it is possible to create a direct Path between the current origin and the destination or not. If such a possibility exists, the points on the direct Path between the current origin and the destination are added to the previous points and the Pathing process is terminated. Otherwise, the PSO algorithm determines the next point of the Path in the way described before. Fig. 3 shows the flowchart of the proposed method.

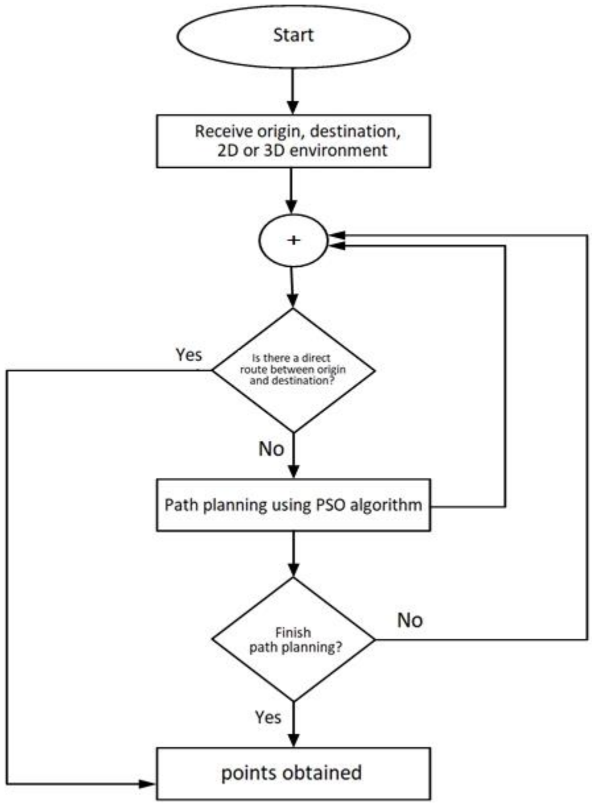


Fig. 3: Path planning module.

Similar data is necessary to evaluate the performance of the proposed module and measure its accuracy. In other words, the existence of Paths traveled by submarines or automatic subsurface vessels for comparison is inevitable. Due to the lack of access to such data and on the other hand, the existence of information related to the Paths traveled by various surface vessels on the Internet, the accuracy of the module was measured for two-dimensional space. The working method is to remove a part of the path traveled by a certain surface vessel (a ship) and find the desired path points using the designed module. Finally, the accuracy of the module is the degree of similarity between the main path traveled by the target surface float and the path completed by the designed module, which is measured using the following relationship:

$$1-\left(\frac{1}{1+e^{-dist}}-0.5\right)\times 2 \tag{4}$$

In the following, the results obtained in three different two-dimensional experiments are given. Then, in the next section, the results of using the module designed for artificial three-dimensional spaces are presented.

In this simulation, the pseudo code of PSO is shown in Table 2.

Table 2: The PSO pseudo-code

Number	the PSO pseudo-code
1	Initialize Population
2	for t=1 : maximum generation
3	for i=1 : population size
4	if $f(y_{i,d}^t(\text{t}))$ < $f(p_i^t$ (t)) then $p_i^t = y_{i,d}^t$
5	$f(p_g^t$ (t)) = min$_i$ $f(p_i^t$ (t))
6	end
7	for d=1 : dimention
8	v_i^{t+1} = w.v_i^t + $c_1.r_1$(p_{leader}^t - y_i^t) + $c_2.r_2$.($p_{i_{best}}^t$ - y_i^t)
9	$y_i^{t+1} = y_i^t + v_i^{t+1}$
10	if v_i^{t+1} > v_{max}^t then $v_i^{t+1} = v_{max}^t$
11	else if v_i^{t+1} < v_{min}^t then v_i^{t+1} = v_{min}^t
12	end
13	if y_i^{t+1} > y_{max}^t then $y_i^{t+1} = y_{max}^t$
14	else if y_i^{t+1} < y_{min}^t then y_i^{t+1} = y_{min}^t
15	end
16	end
17	end
18	end

In this pseudo-code, v_i^{t+1} is the speed of the particle in the next iteration and v_i^t is its speed in the current iteration. p_{leader}^t is the position of the leader in the current iteration and $p_{i_{best}}^t$ is the best position that the particle had from the beginning of the algorithm execution to the t th iteration. y_i^t is the current position of the particle. The coefficients $c_1$and c_2 determine whether the particle will seek answers more individually or follow the leader of the group. The constants r_1 and r_2 are two random numbers in the interval [0 1].

In this simulation, the PSO configuration parameters are as described in Table 3 below.

Table 3: The PSO configuration parameters

Parameter	Value
Maximum PSO iteration	40
PSO population size	5
C1	1.5
C2	1.1
W	At first, it is 0.9, and then it decreases linearly to the value of 0.1 with respect to iteration changes.

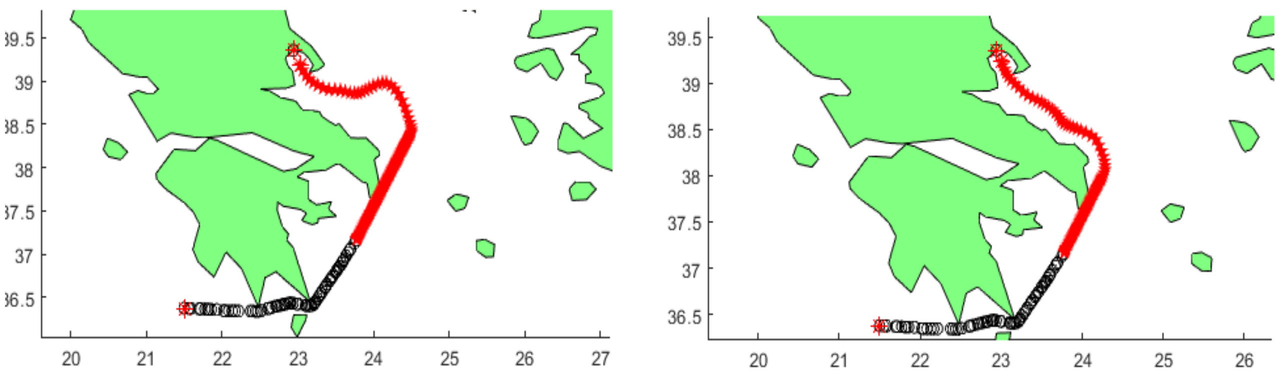
Simulation Results

A. Simulation Results on the Performance of the Module in Two-Dimensional Space

To evaluate the performance of the module, three paths were manually emptied.

A. 1. The First Path

Fig. 4 shows the main Path and the empty Path. This Path includes 192 points, from which 122 points have been removed for evaluation. In other words, 63% of it has been deleted. Fig. 5 also shows the output of the module in four executions of the Pathing module. In these images, the red five-pointed stars are the points found by the algorithm.



(a) (b)

Fig. 4: (a) & (b), the main path and the empty path, respectively.

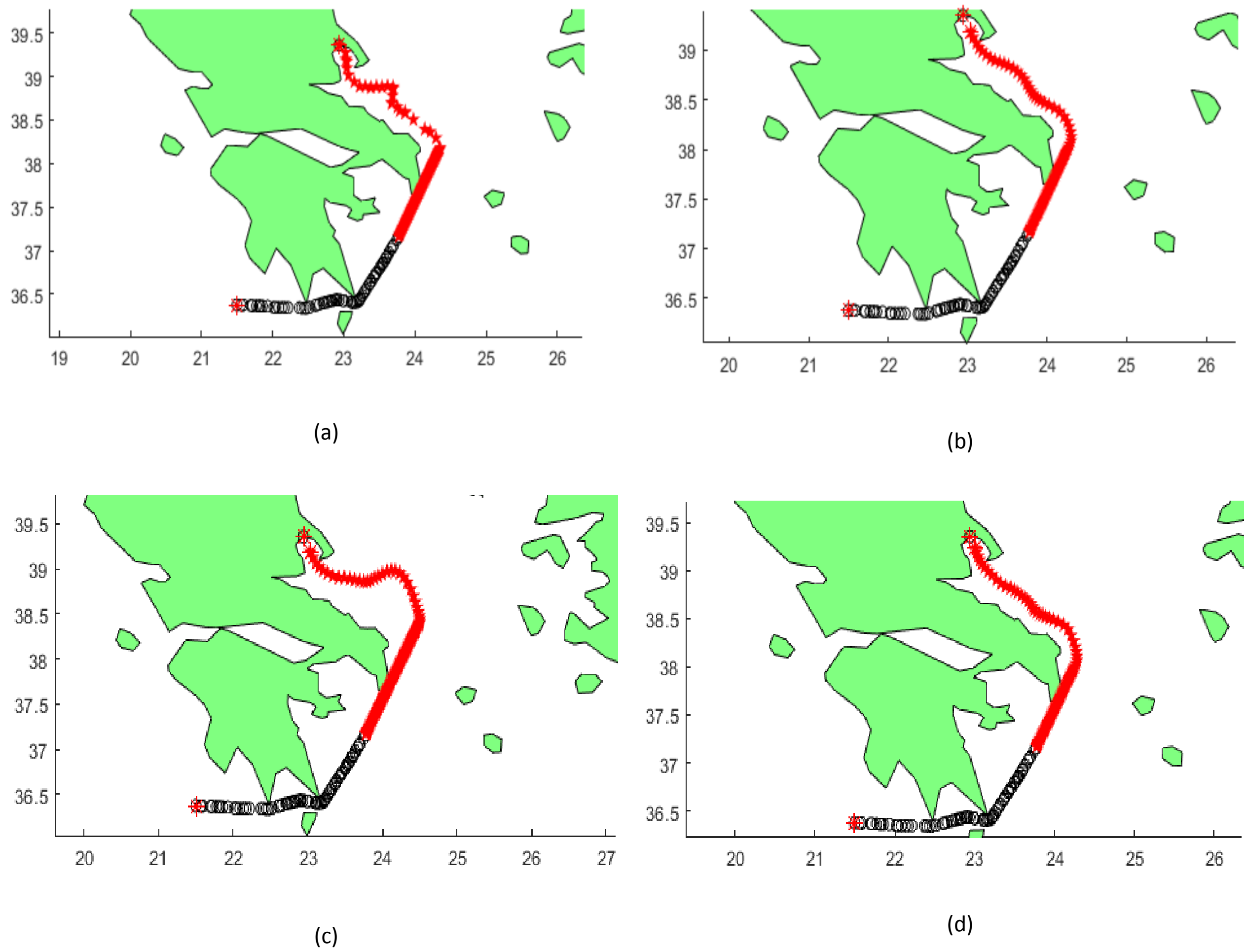


Fig. 5: Path planning module output in 4 independent experiments.

A. 2. The Second Path

95% of this Path is empty. The image of the original path and the empty path to be completed are shown in Fig. 6. Similar to the previous path, the output image of the module in four independent tests can be seen in Fig. 7.

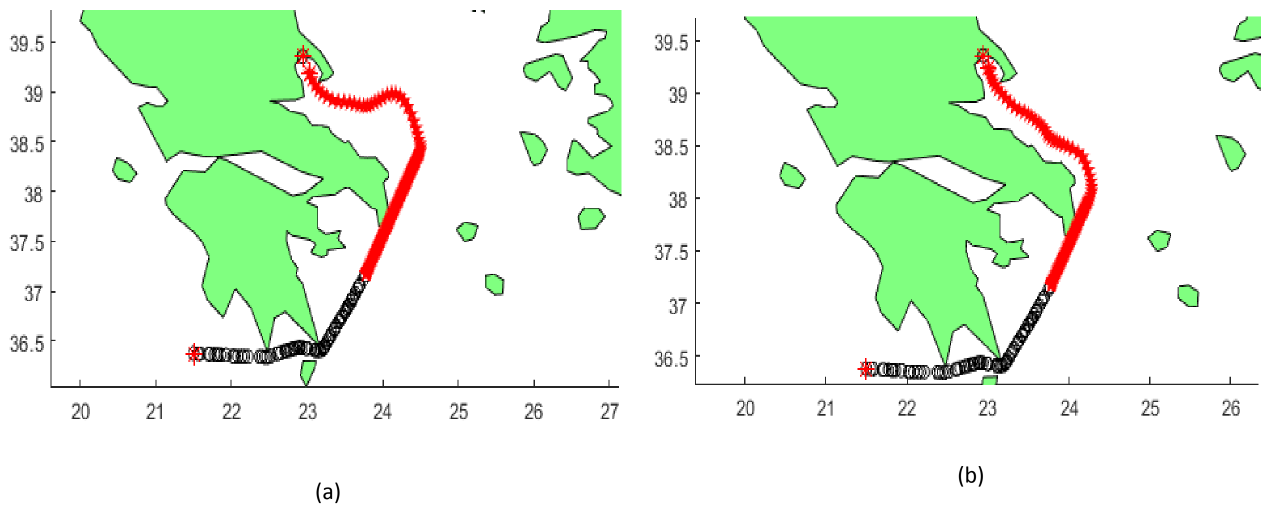
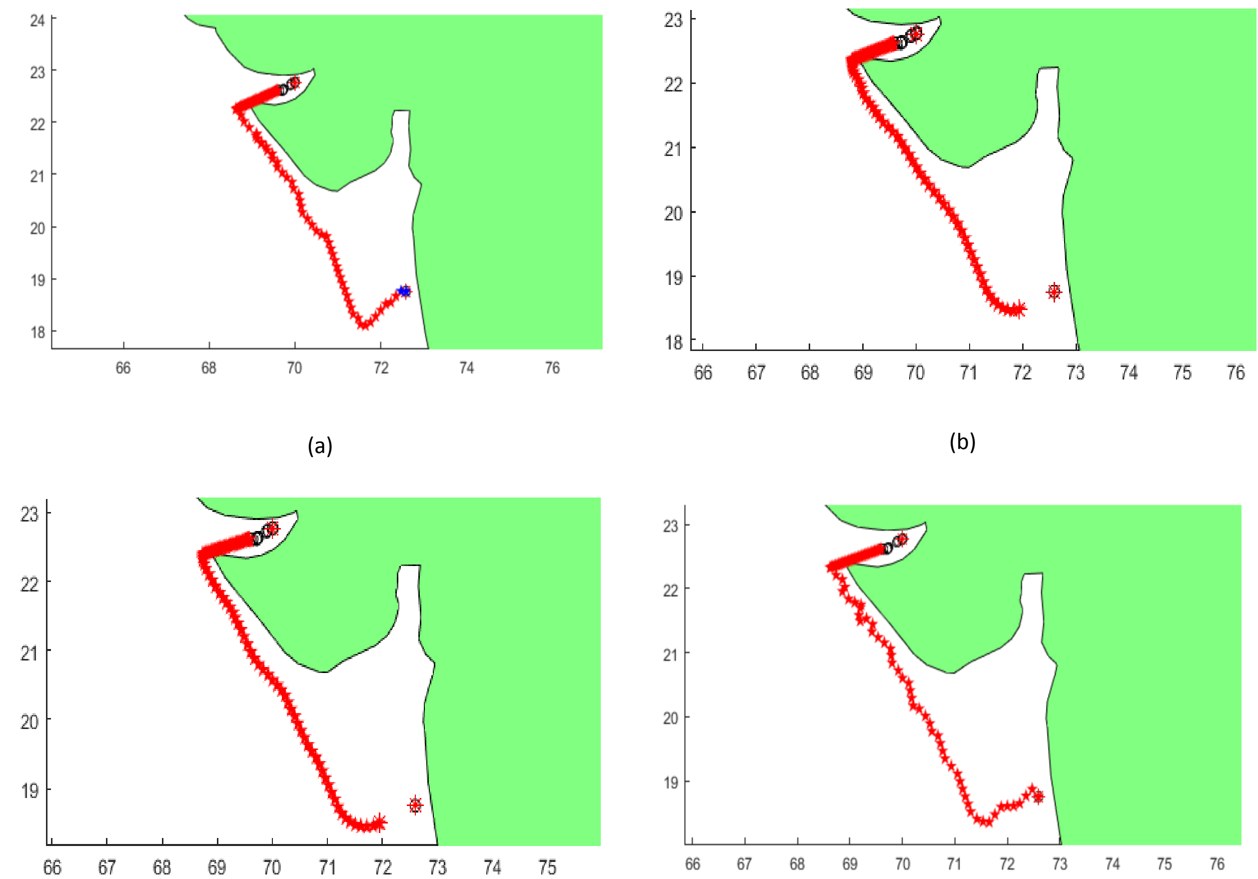


Fig. 6: (a) & (b), respectively, the main path and the empty path.



(a) (b)

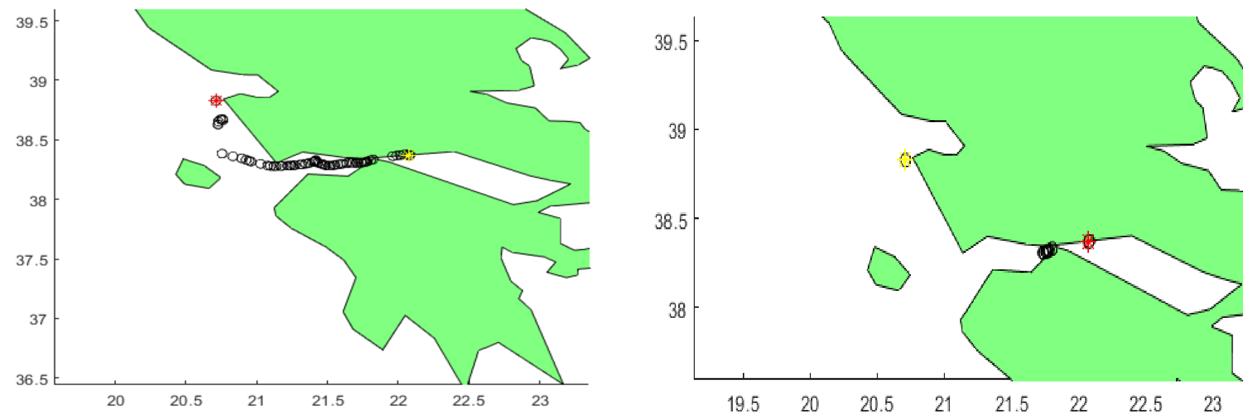
(c) (d)

Fig. 7: Path module output in 4 independent experiments.

A. 3. The third path

Fig. 8 shows the main path and the empty path, and Fig. 9 shows the output of the module in four separate tests.

For this reason, 85% of the main path is empty.



(a) (b)

Fig. 8: (a) & (b) are the main path and the empty path respectively.

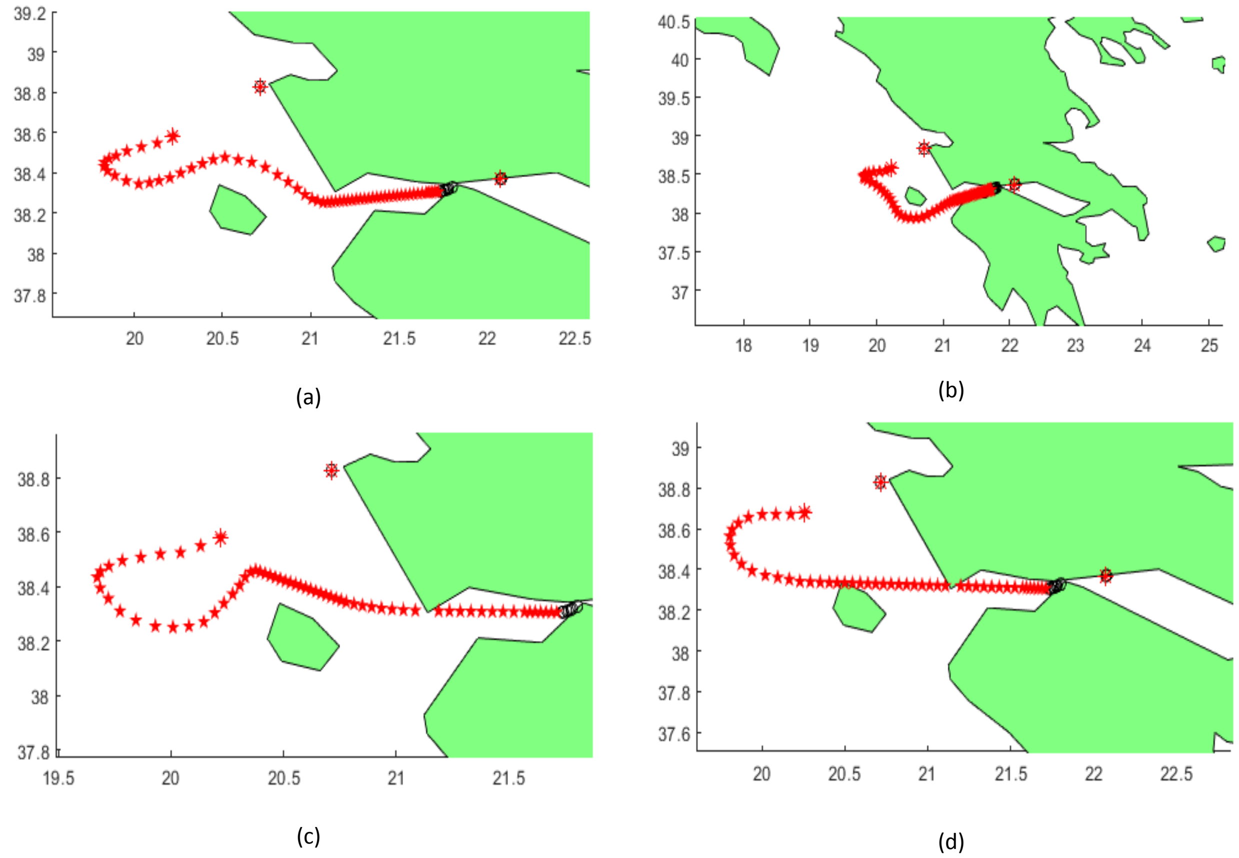


Fig. 9: Path planning module output in 4 independent experiments.

The image related to the output of the module for this test is given in Fig. 10.

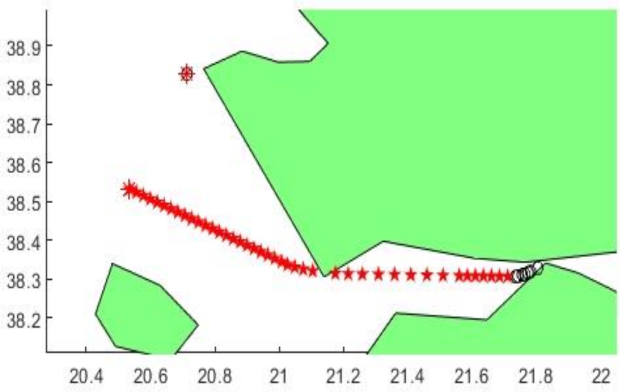


Fig. 10: The output of the module in the 12th test for the third path, for which the calculated accuracy is 83%.

According to Fig. 4 to 10, it can be seen that the proposed module provides very good performance in two-dimensional space. However, the main purpose of designing the module is to navigate in 3D space. It is easy to upgrade the designed module by adding the third dimension (which represents the depth of the sea) to the equations related to the PSO algorithm, for Pathing in the 3D space. Below is the output of the Pathing module for four experiments in different 3D spaces.

In addition to the visual detections that we had above, some quantitative parameters are presented in Table 4. In this Table, time elapsed is the time of running the proposed algorithm (in ms), success rate is the number of success in obtaining the best (or near the best) path with respect to all experiments for each scenario (12 in this paper).

Finally, standard deviation is the standard deviation value of the obtained best fitness values (for 12 experiments).

Table 4: Numerical results of experiments in two-dimensional space

Scenario	Time Elapsed (ms)	Success Rate	Standard Deviation
The first path	9.01	95%	3.2%
The second path	9.04	91%	5.1%
The third path	9.08	96%	2.7%

B. Simulation Results on the Performance of the Module in 3d Space

B. 1. The First Experiment

Fig. 11 shows the intended test environment. The scenario considered for this experiment is to move from the origin (indicated by a blue star) to the destination (indicated by a red star) in the presence of 6 obstacles (indicated as cylinders and spheres). According to this form, it is not possible to move directly between the origin and the destination. In Figures 12 to 14, the Paths found by the Pathing module are shown from three different views.

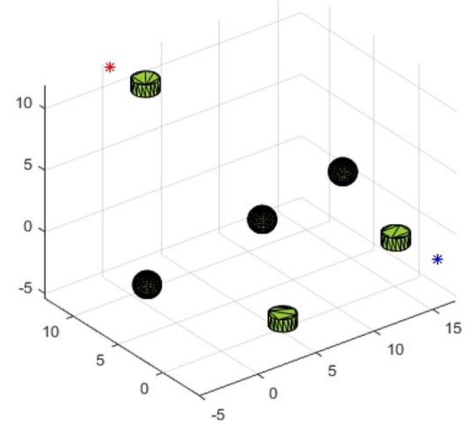


Fig. 11: 3D environment of the first experiment.

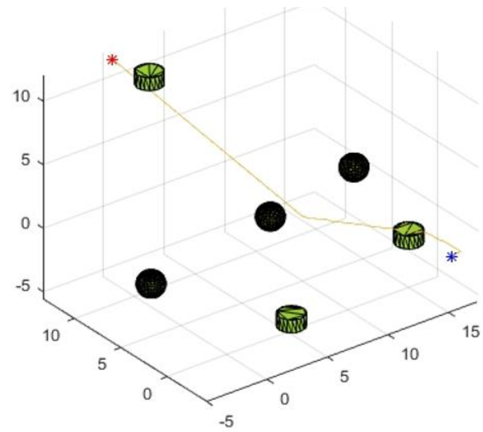


Fig. 12: The path between the origin and the destination in the first experiment from the first view.

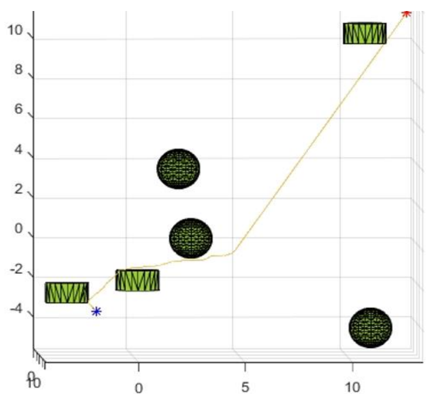


Fig. 13: The path between the origin and the destination in the first experiment from the second view.

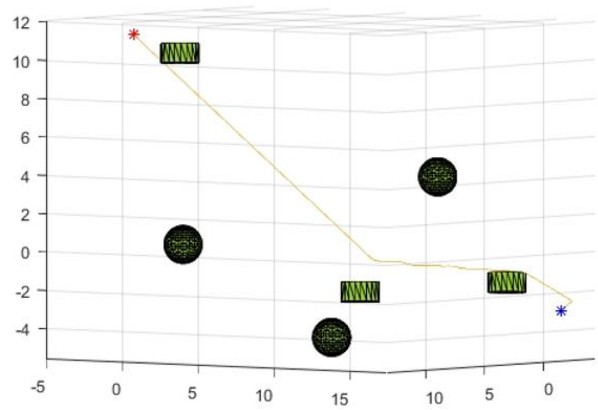


Fig. 14: The Path between the origin and the destination in the first experiment from the third view.

B. 2. The Second Experiment

In this experiment, the three-dimensional space is completely similar to the first experiment, with the difference that the place of origin and destination has been changed. Figs. 15 to 17 show the path found from three different views.

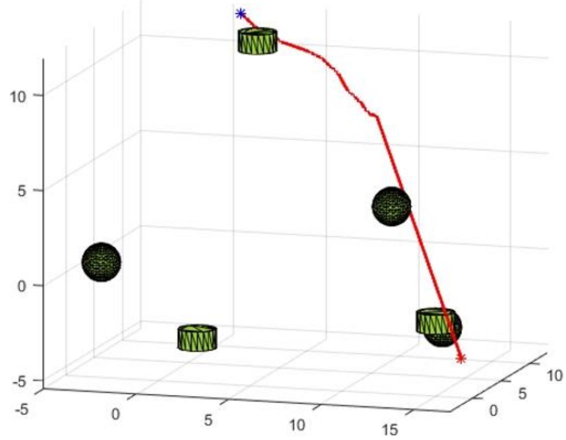


Fig. 15: The path found between the origin and the destination in the second experiment from the first view.

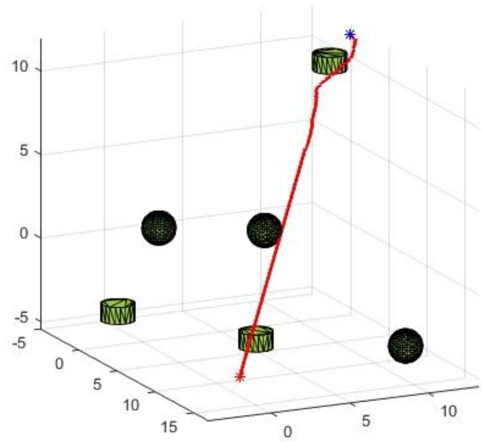


Fig. 16: The path found between the origin and the destination in the second experiment from the second view.

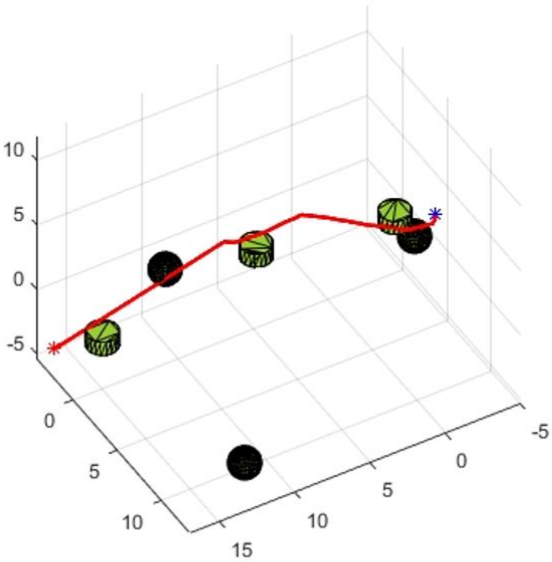


Fig. 17: The path found between the origin and the destination in the second experiment from the third view.

B. 3. The Third Experiment

The three-dimensional environment designed in this experiment is shown in Figs. 18 to 21 show the path found from three different angles.

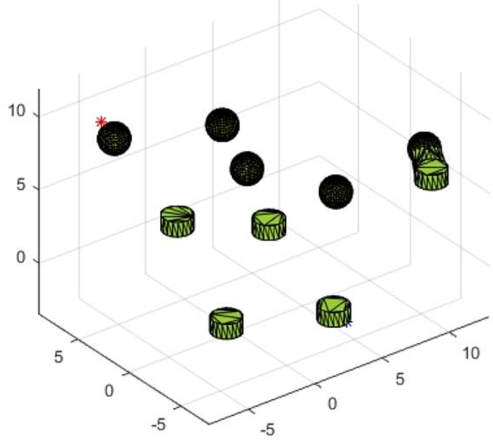
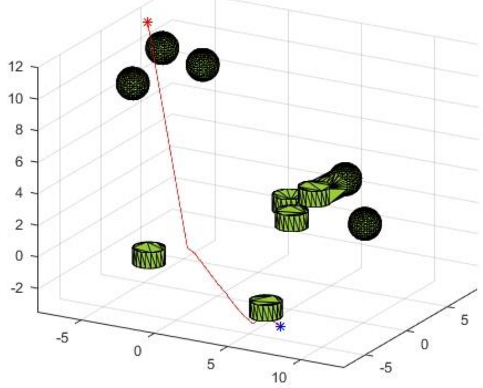


Fig. 18: 3D environment of the third experiment.



Fi. 19: The path found between the origin and the destination in the third experiment from the first view.

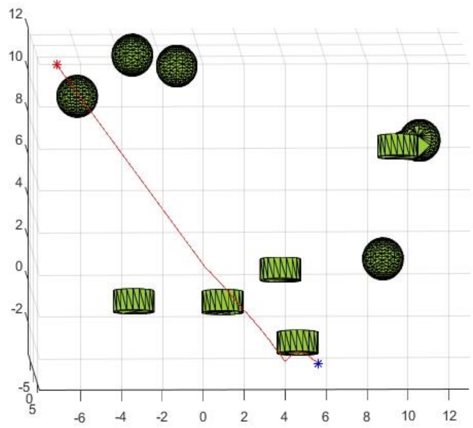


Fig. 20: The path found between the origin and the destination in the third experiment from the second view.

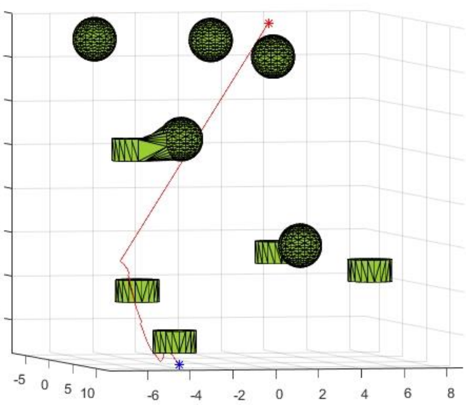


Fig. 21: The path found between the origin and the destination in the third experiment from the third view.

B. 4. The Fourth Experiment

In the fourth experiment, the environment of the problem is similar to the environment of the third experiment, with the difference that the place of origin and destination have been changed. Figs. 22 to 24 show the test output from three different views.

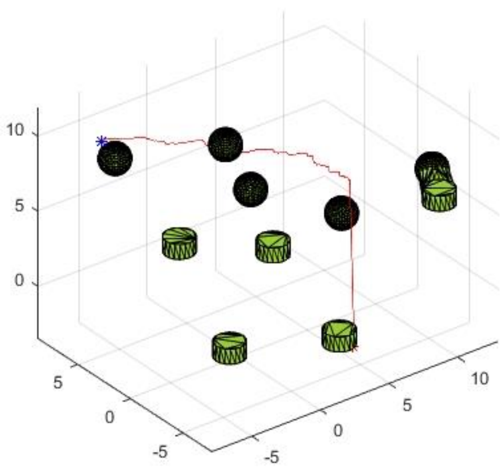


Fig. 22: The path found between the origin and the destination in the fourth experiment from the first view.

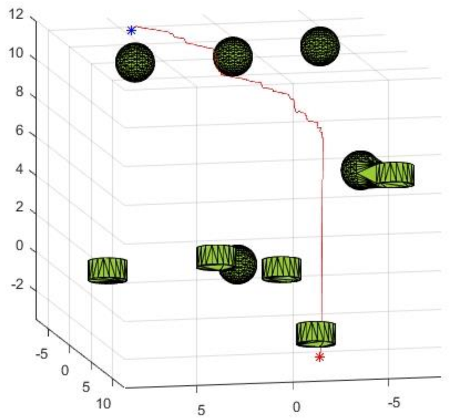


Fig. 23: The path found between the origin and the destination in the fourth experiment from the second view.

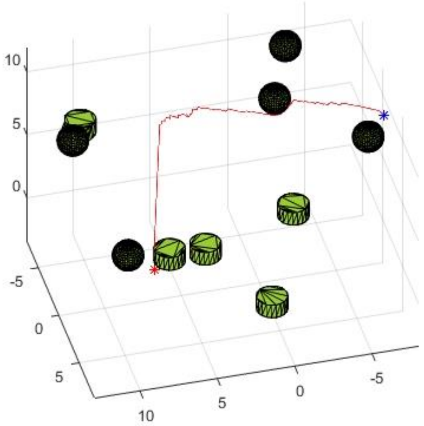


Fig. 24: The path found between the origin and the destination in the fourth experiment from the third view.

In addition to the visual detections that we had above, some quantitative parameters are presented in Table 5.

All of these parameters are defined and similar to Table 4.

Table 5: Numerical results of experiments in 3D space

Scenario	Time Elapsed (ms)	Success Rate	Standard Deviation
The first experiment	9.09	96%	6.2%
The second experiment	9.13	95%	7.5%
The third experiment	9.21	93%	8.4%
The fourth experiment	9.32	91%	10.2%

Also, regarding noise sensitivity, a numerical analysis has been done in four 3D space scenarios, in the form of Table 6. Here, we have set a tolerance of 5% in the estimation of obstacles (this tolerance can be due to sonar error, insufficient information on the map, sudden movement of obstacles, sudden underwater currents, sound noise, jamming, etc.).

We have applied this amount of tolerance randomly with a uniform distribution in each experiment (50 repetitions) to the position of the obstacles.

Table 6: Noise sensitivity analysis on the proposed Path Planning method

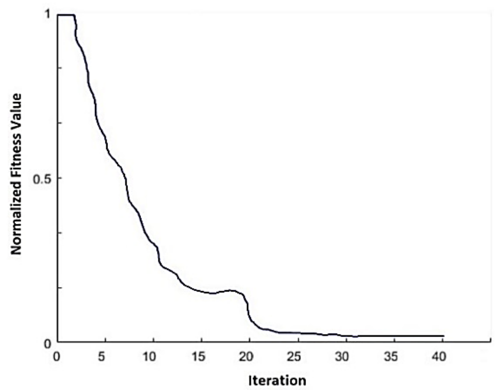
Scenario	Success Rate	Success Rate (with 5% Tolerance)
The first experiment	96%	90%
The second experiment	95%	88%
The third experiment	93%	85%
The fourth experiment	91%	82%

In the above Table 6, the impact of this tolerance on the success rate as the most important factor in the path planning performance shows that the presence of noise has been effective and has been able to reduce the success rate.

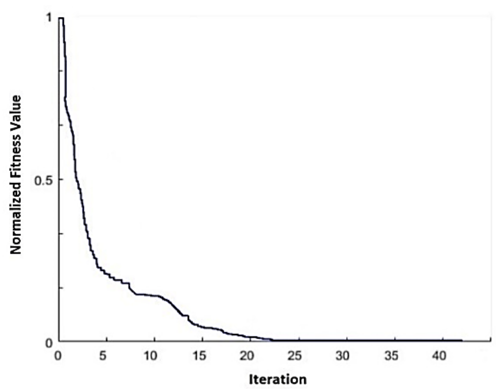
Considering that the success rate in the first experiment, is 6% on average, and in the second experiment, on average by 7%, and in the third experiment, on average by 9%, and in the fourth experiment, it has decreased by 10% on average, but still the presented method was able to find the path well.

C. The Figure of the Best Cost Function

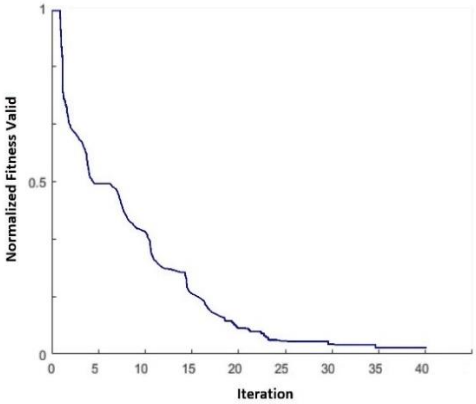
The normalized value of the fitness function is given in Fig. 25.



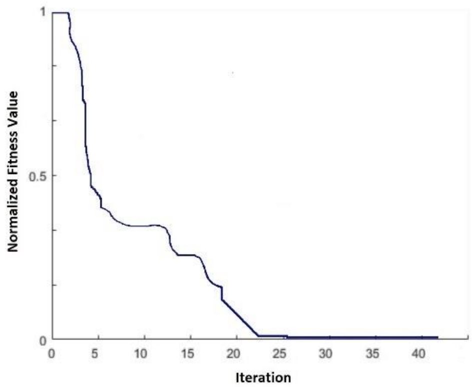
(a)



(b)



(c)



(d)

Fig. 25: The figure of the best cost function.

Path Following Module

This module manages the collective movement of subsurface vessels from origin to destination while maintaining order. In this module, after choosing a certain arrangement, the leader float and the follower floats are determined and the position of each of them is determined at each moment of the movement to the destination according to the type of arrangement. In the performed simulations, two types of arrangements are considered for the fleet of subsurface vessels:

a. Arrangement of arrowhead (^).
b. Linear arrangement.

In the first arrangement (arrowhead arrangement), the float placed at the tip of the arrow is the leader of the group and the floats placed on both sides are the follower floats. In the second arrangement (linear arrangement) the first float is the leader float and other floats are followers.

The path found by the Pathing module determines the position of the leader at each moment of the movement. Therefore, according to the position of the group leader, the position of other vessels is determined during movement.

Determining the position of the follower vessels is done according to the position of the vessel in the fleet, the position of the leader and the minimum distance from the front and side vessels.

In all the way, the main goal is to maintain the overall shape of the makeup. Therefore, at every moment of moving towards the destination, the next position of each follower vessel is calculated based on the principle of keeping the fleet formation. If the next position interferes with an obstacle, the navigation module for the desired float is activated automatically and finds the next point for the desired float. Definitely, in this situation, the overall composition of the fleet will change a little, which is inevitable.

In this way, it is considered not to encounter obstacles in this module. On the other hand, it is also necessary to mention that in the process of correcting the course of a vessel (in case of collision with an obstacle), the angular parameters related to the movement of the vessels (roll, pitch, and yaw) are also taken into account in the Pathing module to avoid collision with any obstacle. Placed. These parameters are used to model the placement of the 3D model of any subsurface vessel in the 3D space and then check whether or not the vessel collides with obstacles. In the following, pictures of group movement of floats in different tests are shown. In these tests, the fleet of subsurface vessels consists of 5 vessels that move in groups in different environments. In these pictures, subsurface vessels are marked with red (leader) and blue (follower) stars, so that in Figs. 26 to 29, the collective movement of the fleet of subsurface vessels with an arrowhead arrangement, and in Figs. 30 to 33, the movement of the group the collective fleet of subsurface vessels are shown in a linear arrangement.

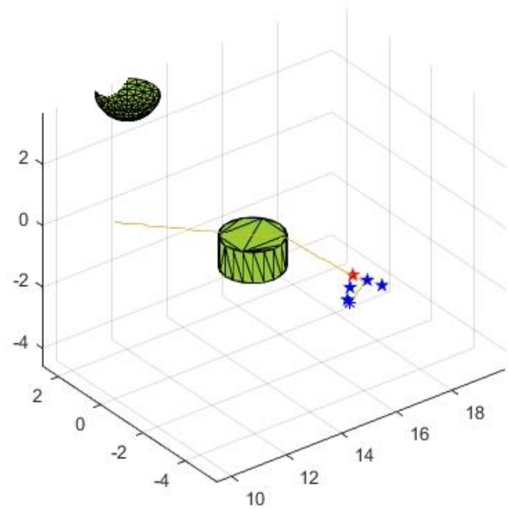


Fig. 26: Collective movement of the fleet of subsurface vessels with arrowhead formation.

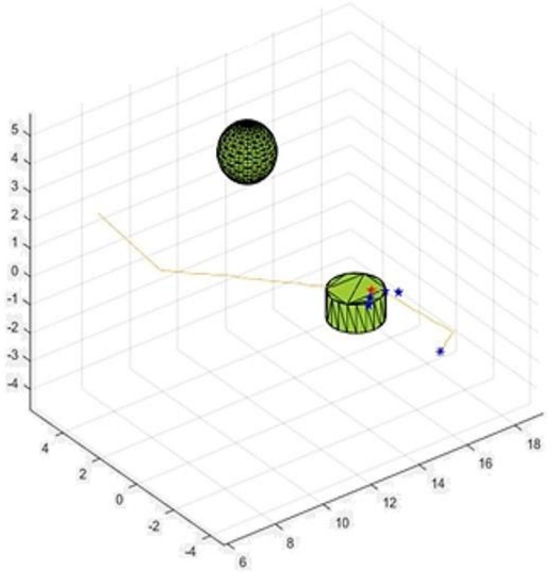


Fig. 27: Collective movement of the fleet of subsurface vessels with arrowhead formation.

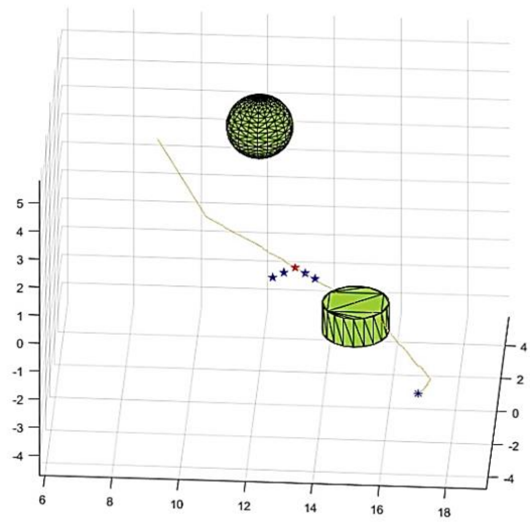


Fig. 28: Collective movement of subsurface vessels fleet with arrowhead formation.

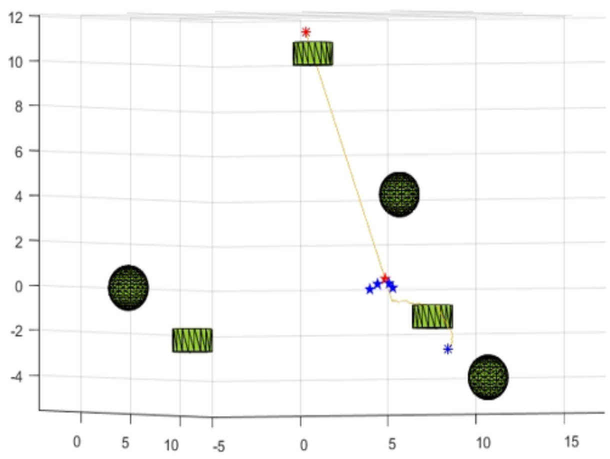


Fig. 29: Mass movement of the fleet of subsurface vessels in an arrowhead formation.

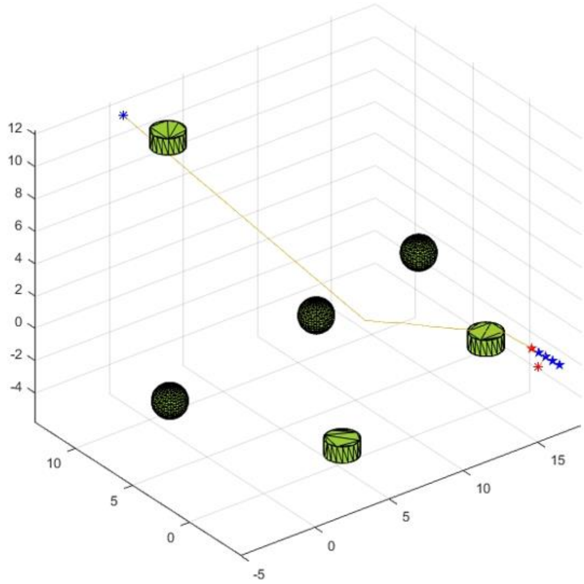


Fig. 30: Mass movement of a fleet of subsurface vessels with a linear arrangement.

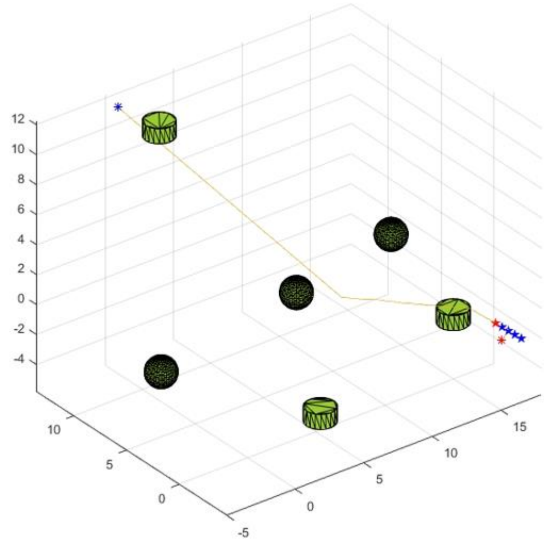


Fig. 31: Collective movement of the fleet of subsurface vessels in line formation.

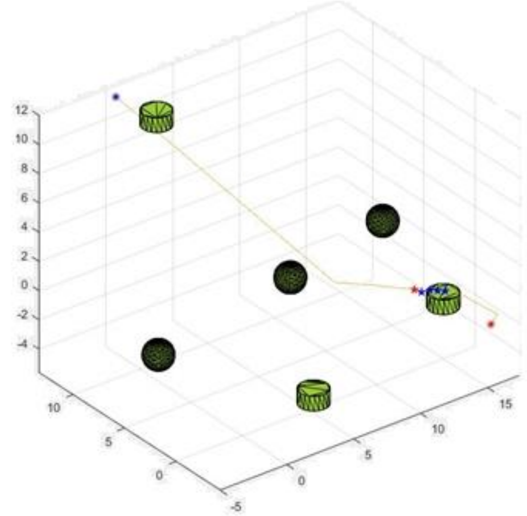


Fig. 32: Collective movement of the fleet of subsurface vessels in line formation.

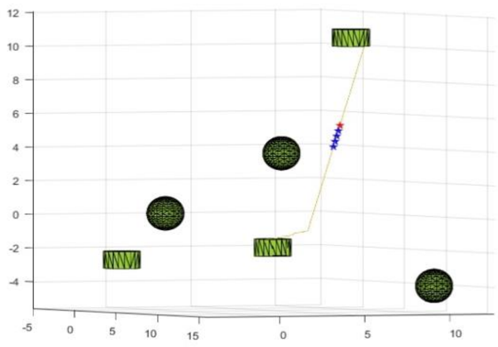


Fig. 33: Collective movement of the fleet of subsurface vessels in line formation.

By comparing the proposed method mentioned with other methods which had been utilized PSO method, it was found that our method has been able to improve the pathing speed and consequently, minimize the energy consumption of the moving group very well.

Of course, it should be noted that path changes will increase energy consumption, But naturally, in an environment that may face disturbances such as all kinds of noises, all kinds of waves and all kinds of sudden obstacles, This environment will be a random environment and because the environment is random, we must have the probability density function of the noise factors and the distribution of the types of events that cause the moving path to change and then according to these data, a theoretical research should be done in the random space to be able to provide a relatively accurate mathematical model to calculate the amount of energy consumed in that random environment.

The results of this investigation are given in the Table 7 below. The quantitative values in this table are defined for Table 4.

Table 7: Comparison with similar researches

Reference Number	Type of Path Generated	Collision/Obstacle Avoidance	Path Cost	Time of Path planning (ms)	Success Rate	Improvement Ratio (Time of Path planning)	Improvement Ratio (Success Rate)
[23]	Time optimal	Achieved	Moderate	11.53	90%	19%	05%
[33]	Time optimal	poor	High	16.34	86%	42%	09%
[34]	Energy optimal	Achieved	Low	10.29	92%	09%	02%
[35]	Time optimal	Achieved	Moderate	12.24	89%	23%	06%
[36]	Energy optimal	Achieved	Low	10.98	91%	15%	03%
[37]	Time optimal	Achieved	High	14.37	87%	35%	08%
Proposed Method	Time and energy optimal	Achieved	Low	9.32	94%	-	-

It is worth mentioning that, in this research, the maximum value of PSO repetition is considered 40, while in the mentioned papers, the maximum value of PSO repetition is considered as an average between 100 and 200. Path costs are compared as low, moderate and high. Collision and obstacle avoidance are discussed as achieved, limited and poor based on whether the algorithm focused on these issues or not.

Conclusion

In this paper, an efficient method for the path planning problem is presented. The proposed method is designed using Particle Swarm Optimization (PSO). In the proposed method, several effective fitness function have been defined so that the best path or one of the closest answers can be obtained by utilized metaheuristic algorithm. The results of implementing the proposed method on real and simulated geographic data show its fabulous performance. The achieved results are better or comparable with others method (time elapsed, success rate, Path cost, standard deviation, improvement ratio).

Of course we must pay attention to the fact that here we are facing two situations in the matter of path planning. The first mode (completely offline): In this case, the path map, static obstacles are clear, and the probability of dynamic obstacles, disturbances and noise is zero. In this case, after the algorithm is run, the path is designed and the moving will be able to move on this path. Of course, two conditions of sonar accuracy and speed must be taken into account here the second mode (online with restrictions): In this case, in the environment, it is possible to suddenly change the map and dynamic obstacles that do not have a very high speed. So that according to the times we presented in Table 4 and Table 5, immediately after the sonar detects the obstacle, the algorithm has the ability to quickly calculate and determine the next point on the path. If these two conditions are fulfilled, according to the appropriate time cost of the algorithm, it is possible for this algorithm to determine the next point on the path in an online. But if these conditions are not taken into account, the presented algorithm cannot quickly calculate and specify the next point on the path online. The limitations that can affect the method presented in this paper are often dependent on the response speed of the sensors used on moving parts to detect obstacles. It is natural that if the sensors used do not have the required speed, the time required to implement the proposed method and correct the path will not be possible and the pathing will not be successful. This issue is the most important challenge of the method presented in this paper.

Regarding the fields of future work, it is possible to mention the use of dynamic PSO methods and in general dynamic metaheuristic PSO to deal with environmental disturbances. Also, deriving an accurate mathematical model for the energy consumption based on the different paths, is considered as another important topic for further work.

Author Contributions

B. Mahdipour has searched for important articles in this field. Then, by checking the results and collecting the necessary data and simulated the proposed method in MATLAB, the implementation of the proposed method has been done. Dr. S. H. Zahiri and Dr. I. Behravan reviewed the results and made changes in the way of implementation and final editing of the work.

Acknowledgment

We sincerely thank the respected referees for their accurate review of this paper.

Conflict of Interest

The authors announce no potential conflict of interest regarding the publication of this paper. Also, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission and redundancy have been completely witnessed by the authors.

Abbreviations

SONAR	Sound and Range Navigation
PSO	Particle swarm optimization
AUV	Autonomous underwater vehicle
USV	Unmanned surface vehicle
LKF	Linearized Kalman Filter
EKF	Extended Kalman Filter
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
HPF	Heuristic Potential Field
GA	Genetic Algorithm
QPSO	Quantum behaved Particle Swarm Optimization
ICA	Imperialist Competitive Algorithm
ACO	Ant Colony Optimization
SOM	Self-Organizing map
BINM	Biological inspired neurodynamics model
VS	Velocity Synthesis
DE	Differential Evolution
GWO	Grey Wolf Optimizer

References

[1] E. Alfaro-Cid, E. W. McGookin, D. J. Murray-Smith, "Genetic algorithm optimisation of a ship navigation system," Acta Polytech., 41(4-5): 13-18, 2001.

[2] V. Kanakakis, N. Tsourveloudis, "Evolutionary path planning and navigation of autonomous underwater vehicles," in Proc. 2007 Mediterranean Conference on Control & Automation: 1-6, 2007.

[3] K. Hao, J. Zhao, Z. Li, Y. Liu, "Dynamic path planning of underwater AUV based on an adaptive genetic algorithm," Ocean Eng., 263: 11242, 2022.

[4] M. Y. Ju, S. E. Wang, J. H. Guo, "Path planning using a hybrid evolutionary algorithm based on tree structure encoding," Sci. World J., 2014: 1-8, 2014.

[5] X. Li, S. Yu, "Three-dimensional path planning for AUVs in ocean currents environment based on an improved compression factor particle swarm optimization algorithm," Ocean Eng., 280: 114610, 2023.

[6] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, Y. Xia, "survey of robot 3d path planning algorithms," J. Control Sci. Eng., 2016(7426913): 1-22, 2016.

[7] S. Singhal, S. Tanwar, Aishwarya, A. Sinha, "Ant colony optimization based routing for underwater sensor network," in Proc. 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART): 33-38, 2020.

[8] M. P. Vicmudo, E. P. Dadios, R. R. P. Vicerra, "Path planning of underwater swarm robots using genetic algorithm," 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2014.

[9] H. Tang, Y. Yin, H. Shen, "A model for vessel trajectory prediction based on long short-term memory neural network," J. Mar. Eng. Technol., 21(3): 136-145, 2022.

[10] S. Singhal, S. Tanwar, Aishwarya, A. Sinha, "Ant colony optimization based routing for underwater sensor network," in Proc. 9th International Conference System Modeling and Advancement in Research Trends (SMART), 2020.

[11] H. Qin, T. Meng, Y. Cao, "Fuzzy strategy grey wolf optimizer for complex multimodal optimization problems," Sensors, 22(17): 6420, 2022

[12] M. Dinc, C. Hajiyev, "Integration of navigation systems for autonomous underwater vehicles," J. Mar. Eng. Technol., 14(1): 32-43, 2015.

[13] M. Panda, B. Das, B. Subudhi, B. Bhusan Pati, "A comprehensive review of path planning algorithms for autonomous underwater vehicles," Int. J. Autom. Comput., 17(3): 321-352, 2020.

[14] Y. Wang, "Review on greedy algorithm," Theor. Nat. Sci., 14(1): 233-239, 2023.

[15] J. C. Kinsey, R. M. Eustice, L. L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," in Proc. IFAC Conference of Manoeuvring and Control of Marine Craft, 88: 1-12, 2016.

[16] D. Li, P. Wang, L. Du, "Path planning technologies for autonomous underwater vehicles-A review," IEEE Access, 7: 9745-9768, 2019.

[17] N. K. Yilmaz, C. Evangelinos, P. F. J. Lermusiaux, N. M. Patrikalakis, "Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming," IEEE J. Oceanic Eng., 33(4): 522-537, 2019.

[18] D. H. dos Santos, L. M. G. Goncalves, "A gain scheduling control strategy and short-term path optimization with genetic algorithm for autonomous navigation of a sailboat robot," Int. J. Adv. Rob. Syst., 6(1), 2020.

[19] M. Y. Ju, S. E. Wang, J. H. Guo, "Path planning using a hybrid evolutionary algorithm based on tree structure encoding," Sci. World J., 2014: 1-8, 2014.

[20] I. Behravan, S. H. Zahiri ,S. M. Razavi, R. Trasarti, "Clustering a big mobility dataset using an automatic swarm intelligence-based clustering method," J. Electr. Comput. Eng. Innovations, 6(2): 251-271, 2018.

[21] Y. Li, J. Zhao, Z. Chen, G. Xiong, S. Liu, "A robot path planning method based on improved genetic algorithm and improved dynamic window approach," Sustainability, 15(5): 4656, 2023.

[22] M. R. Razali, A. A. M. Faudzi, A. U. Shamsudin, S. Mohamaddan, "A hybrid controller method with genetic algorithm optimization to measure position and angular for mobile robot motion control," Front. Robot., 9, 2023.

[23] C. L. Pen, W. J. Chang, Y. H. Lin, "Fuzzy controller design approach for a ship's dynamic path based on ais data with the takagi–sugeno fuzzy observer model," J. Mar. Sci. Eng., 11(6): 1181, 2023.

[24] M. Reda, A. Onsy, A. Y. Haikal, A. Ghanbari, "Path planning algorithms in the autonomous driving system: A comprehensive review," Rob. Auton. Syst., 174: 104630, 2024.

[25] D. An, Y. Mu, Y. Wang, et al., "Intelligent path planning technologies of underwater vehicles: A Review," J. Intell. Rob. Syst., 107(22), 2023.

[26] R. Zhou, K. Zhou, L. Wang, et al., "An improved dynamic window path planning algorithm using multi-algorithm fusion," Int. J. Control Autom. Syst., 22: 1005-1020, 2024.

[27] L. Yu, Y. Cai, X. Feng, et al., "Parallel parking path planning and tracking control based on simulated annealing algorithm," Int. J. Autom. Technol., 25: 867-880, 2024.

[28] W. Qiu, D. Zhou, W. Hui, et al., "Terrain-shape-adaptive coverage path planning with traversability analysis," J. Intell. Rob. Syst. 110(41), 2024.

[29] X. Zhai, J. Tian, J. Li, "A real-time path planning algorithm for mobile robots based on safety distance matrix and adaptive weight adjustment strategy," Int. J. Control Autom. Syst., 22: 1385-1399, 2024.

[30] B. Zhang,P. Liu,W. Liu, et al., "Search-based path planning and receding horizon based trajectory generation for quadrotor motion planning," Int. J. Control Autom. Syst., 22: 631-647, 2024.

[31] M. Reda, A. Onsy, A. Y. Haikal, A. Ghanbari, "Path planning algorithms in the autonomous driving system: A comprehensive review," Rob. Auton. Syst., 174(104630), 2024.

[32] J. Yu, Z. Chen, Z. Zhao, et al., "A path planning method for unmanned surface vessels in dynamic environment," Int. J. Control Autom. Syst., 22: 1324–1336, 2024.

[33] L. Yujie, P. Yu, S. Yixin, Z. Huajun, Z. Danhong, S. Yong, "Ship path planning based on improved particle swarm optimization," in Proc. 2018 Chinese Automation Congress (CAC), 2019.

[34] X. Wang, K. Feng, G. Wang, Q. Wang, "Local path optimization method for unmanned ship based on particle swarm acceleration calculation and dynamic optimal control," Appl. Ocean Res., 110(102588), 2021.

[35] S. Blindheim, T. A. Johansen, "Particle swarm optimization for dynamic risk-aware path following for autonomous ships," IFAC-PapersOnLine, 55(31): 70-77, 2022.

[36] H. Feng, M. J. Liu, H. Y. Xu, "Multi-target path planning for unmanned surface vessel based on adaptive hybrid particle swarm optimization," J. Huazhong Univ. Sci. Technol. (Natural Science Edition), 46(6): 59-64, 2018.

[37] W. Ma, Y. Han, H. Tang, D. Ma, H. Zheng, Y. Zhang, "Ship route planning based on intelligent mapping swarm optimization," Comput. Ind. Eng., 176(108920), 2023.

[38] D. Mu, T. Li, X. Han, Y. Fan, F. Wang, "Global path planning for unmanned surface vehicles in complex maritime environments considering environmental interference," Ocean Eng., 310(2), 2024.

[39] J. Pak, J. Kim, Y. Park , H. Il Son, "Field evaluation of path planning algorithms for autonomous mobile robot in smart farms," IEEE Access, 10: 60253-60266, 2022.

Biographies

Behrouz Mahdipour received the B.Sc. degree in Electrical and Electronics Engineering from Yazd University, Iran, in 2008 and M.Sc. degree in Electrical and Electronics Engineering from Bojnord University, Iran, in 2019. Currently, He is a Ph.D. student in Electrical and Electronics Engineering at Birjand University, Iran, as well as an assistant professor and researcher at a scientific and research institute. His research interests include the path planning of all types of intelligent unmanned surface and subsurface vessels and swarm intelligence algorithms.

- Email: behrouzmahdipour@birjand.ac.ir
- ORCID: 0009-0007-2380-9733
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA

Seyed Hamid Zahiri received the B.Sc., M.Sc. and Ph.D. degrees in Electronics Engineering from Sharif University of Technology, Tehran, Tarbiat Modarres University, Tehran, and Mashhad Ferdowsi University, Mashhad, Iran, in 1993, 1995, and 2005, respectively. Currently, he is a Professor with the Department of Electronics Engineering, University of Birjand, Birjand, Iran. His research interests include pattern recognition, evolutionary algorithms, swarm intelligence algorithms, and soft computing.

- Email: hzahiri@birjand.ac.ir
- ORCID: 0000-0002-1280-8133
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA

Iman Behravan received his B.S.c in Electronics Engineering from Shahid Bahonar University of Kerman, Iran. Also, he received his M.Sc. and Ph.D. degrees from the University of Birjand, Iran. He also worked as a post-doctoral researcher at the University of Birjand under the supervision of Professor Seyed Mohamad Razavi for two years. Currently he is working as a senior data scientist and blockchain developer at Kara Group, Tehran, Iran. His research interests include big data analytics, pattern recognition, machine learning, swarm intelligence, and soft computing.

- Email: i.behravan@birjand.ac.ir
- ORCID: 0000-0003-0319-1765
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA

How to cite this paper:
B. Mahdipour, S. H. Zahiri, I. Behravan, "An intelligent two and three dimensional path planning, based on a metaheuristic method," J. Electr. Comput. Eng. Innovations, 13(1): 93-116, 2025.

DOI: 10.22061/jecei.2024.10941.751

URL: https://jecei.sru.ac.ir/article_2193.html