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

Research paper

Design and Stability Analysis of a Novel Path Planner for Autonomous Underwater Vehicles

Z. K. Pourtaheri *
Department of Mechatronics, Higher Education Complex of Bam, Bam, Iran.

Article Info

Article History:
Received 24 April 2024
Reviewed 20 June 2024
Revised 29 July 2024
Accepted 06 August 2024

Keywords:
Autonomous underwater vehicles
Path planning
Heuristic algorithms
Optimization
Stability analysis

*Corresponding Author's Email Address:
z.pourtaheri@bam.ac.ir

Abstract

Background and Objectives: According to this fact that a typical autonomous underwater vehicle consumes energy for rotating, smoothing the path in the process of path planning will be especially important. Moreover, given the inherent randomness of heuristic algorithms, stability analysis of heuristic path planners assumes paramount importance.
Methods: The novelty of this paper is to provide an optimal and smooth path for autonomous underwater vehicles in two steps by using two heuristic optimization algorithms called Inclined Planes system Optimization algorithm and genetic algorithm; after finding the optimal path by Inclined Planes system Optimization algorithm in the first step, the genetic algorithm is employed to smooth the path in the second step. Another novelty of this paper is the stability analysis of the proposed heuristic path planner according to the stochastic nature of these algorithms. In this way, a two-level factorial design is employed to attain the stability goals of this research.
Results: Utilizing a Genetic algorithm in the second step of path planning offers two advantages; it smooths the initially discovered path, which not only reduces the energy consumption of the autonomous underwater vehicle but also shortens the path length compared to the one obtained by the Inclined Planes system optimization algorithm. Moreover, stability analysis helps identify important factors and their interactions within the defined objective function.
Conclusion: This proposed hybrid method has implemented for three different maps; 36.77%, 48.77%, and 50.17% improvements in the length of the path are observed in the three supposed maps while smoothing the path helps robots to save energy. These results confirm the advantage of the proposed process for finding optimal and smooth paths for autonomous underwater vehicles. Due to the stability results, one can discover the magnitude and direction of important factors and the regression model.

Introduction

Nowadays, autonomous mobile robots have become an inseparable part of the growing world. These robots have the capability to perform difficult and sensitive tasks in high-risk environments and, therefore, have attracted a lot of attention. One example of an important type of these robots is the Autonomous Underwater Vehicle (AUV).

Path planning is one of the most important research topics in the field of Autonomous Underwater Vehicles, i.e., determining the movement path of the AUVs from the starting point to the target point to carry out a specific mission. The AUV's path from origin to destination can be pre-programmed, and extensive research has been conducted on optimizing these paths. However, during its movement, it's crucial for the AUV to avoid collisions with both fixed and moving obstacles, such as other AUVs and underwater creatures. So, it can be said that determining the appropriate path for AUV movement from origin to destination remains an important challenge.

Path planning for AUVs is a challenging problem that heavily relies on optimization techniques and so far, various methods have been presented to solve this problem. The most important weaknesses of these methods are the high probability of getting stuck into local optima, the large volume of calculations and also, the existence of deficiencies in facing the dynamic environment. These cases can make them inappropriate for long distances. But due to the ability of heuristic optimization algorithms to solve complex optimization problems with high dimensions, these algorithms are suitable candidates for solving the path planning problem. The most important advantages of heuristic algorithms are their flexibility, high compatibility, high speed and efficiency, and their global search characteristic. These algorithms can avoid local optima and converge to global optima by using special schemes. A significant body of research has been conducted in the field of heuristic optimization for AUV path planning.

For instance, genetic algorithm has been successfully applied in this area, as demonstrated in [1]. The proposed method in this study introduces a new operator to the algorithm to guarantee convergence to the global minimum value in case of multiple minima Reference [2] introduces a hierarchical approach based on genetic algorithm for path planning of AUVs. The method first divides the workspace of the AUV into obstacle-free and obstacle-filled regions. Then, it utilizes genetic algorithm to search for a path among the obstacle-free regions. Reference [3] proposes a genetic algorithm-based path planning method for AUVs. The method first discretizes the three-dimensional space between the start point and the end point into a grid. Then, it employs genetic algorithm to search for the optimal path among these grid points. Each chromosome represents a sequence of these grid points, and the path between the start and the target points is constructed by connecting these points. The objective function is designed to minimize the energy consumption of the AUV along the path. In this study, some grid points are randomly selected as obstacles to simulate static obstacles in the problem space. Therefore, the path represented by each chromosome should not include these obstacle points. References [4]-[7] provide further examples of AUV path planning in known environments with static obstacles using genetic algorithm.

Particle swarm optimization algorithm and its variant, quantum particle swarm optimization algorithm, have been widely used for path planning of autonomous underwater vehicles. A combination of particle swarm optimization algorithm with differential evolution algorithm is proposed for offline path planning of AUVs in [8]. This proposed approach reduces the computational cost while increasing the ability of the PSO algorithm to find the optimal path. An improved quantum particle swarm optimization algorithm is proposed in [9] for the path planning of AUVs. Safety, path length, and path angle are considered to define the fitness function, and a cubic spline interpolation algorithm is used to smooth the path. Reference [10] presents a method for path planning of AUVs, which consists of two parts. First, a general path between the origin and destination is determined using genetic algorithm. The goal of this part is to find the shortest possible path between the origin and destination. To find this path, the space between the origin, and destination is discretized and points that can be considered as positions are extracted. The genetic algorithm then finds the best possible sequence of points from the available points to find the path. The next part is responsible for processing the sequence found by genetic algorithm. In this module, a particle swarm optimization algorithm is used to search the space between each pair of consecutive points in the sequence to find a suitable path between these two points. Therefore, a general path is first determined, and then other suitable paths are found between each pair of points on the general path. In the second module, the particle swarm algorithm searches the space with the goal of finding the shortest path. A new path planning method for underwater environments using an enhanced quantum particle swarm optimization algorithm is introduced in [11]. This method leverages a technique called Deep Q-Network to learn and adapt its behavior. The algorithm analyzes data about the particles' positions and utilizes neural networks to choose the most suitable action from a set of five options. This approach empowers the particles to make informed decisions in various situations, leading to a significant improvement in the algorithm's ability to explore the entire search space effectively. Furthermore, the accuracy is enhanced by fine-tuning operations. Additionally, a custom fitness function is designed specifically for underwater environments. This function considers factors like path length, deflection angles, and currents, allowing the algorithm to navigate underwater environments more efficiently and locate the path with the least energy consumption.

A combination of ant colony optimization (ACO) algorithm with A* algorithm is used for path planning in [12]. The results of this research show that using this method, the AUV can successfully navigate through an area with dense obstacles. The challenge of two-dimensional autonomous path planning for AUVs operating in environments with ocean currents and obstacles is addressed in [13], and an improved Fireworks-Ant Colony Hybrid Algorithm is proposed to tackle this problem. First, a two-dimensional Lamb vortex ocean current environment scheme that incorporates randomly distributed obstacles is created. Subsequently, a mathematical model is formulated for path planning, considering factors such as navigation time, energy consumption, and total distance traveled. A multi-objective ant colony algorithm for path planning of AUV is presented in [14]. This approach goes beyond traditional, single-objective path planning by considering path length, energy consumption, and safe navigation. To address the challenge of finding optimal paths for autonomous underwater vehicles in complicated environments, an improved ant colony optimization algorithm merged with particle swarm optimization algorithm is presented in [15]. Considering the limitations faced by AUVs, such as constrained energy and visual interval, the proposed algorithm employs a modified pheromone update rule and heuristic function informed by PSO. This allows the AUV to navigate efficiently by connecting designated points while abstaining from collisions with the static obstacles. By incorporating PSO, this improved ACO algorithm overpowers the limitations of the traditional approach. A new algorithm called dynamic multi-role adaptive collaborative ant colony optimization is presented in [16] to address the limitations of slow convergence and poor diversity in the traditional ant colony algorithm. The results of applying this algorithm in robot path planning, illustrate its successfulness in solving this problem. A new approach for path planning of AUVs during dam inspections is proposed in [17]. The goal is to create safe and reliable paths that avoid obstacles while minimizing sharp turns. This method improves upon the traditional ACO algorithm by incorporating a "corner-turning heuristic function." This function helps the AUV select straighter paths, reducing turning times and improving overall efficiency. The reference [18] focuses on enhancing the underwater path-planning capabilities of AUVs by addressing limitations inherent to traditional algorithms like the ant colony algorithm and the artificial potential field algorithm. To overcome these limitations, an optimized scheme for the artificial potential field ant colony algorithm is proposed. Compared to conventional ant colony and other benchmark algorithms, the proposed algorithm achieved significant improvements: path length reductions of 1.57% and 0.63% (simple environment) and 8.92% and 3.46% (complex environment). Additionally, this algorithm demonstrated faster convergence, with iteration time reductions of approximately 28.48% and 18.05% (simple environment) and 18.53% and 9.24% (complex environment).

A novel method for planning safe paths for AUVs navigating environments filled with obstacles is presented in [19]. To address this, a hybrid approach is introduced that combines the strengths of two nature-inspired algorithms: Grey Wolf Optimization (GWO) and GA. This combined method, called Hybrid Grey Wolf Optimization, allows AUVs to find safe paths while minimizing travel distance. The proposed algorithm tackles GWO's weakness of random initialization by using GA to generate a good starting point for the search. In this research, the ideal path considers both the distance traveled and the penalties incurred from avoiding obstacles.

Many conventional heuristic algorithms struggle with two limitations: slow progress towards optimal solutions and getting stuck on suboptimal ones too early. These issues are addressed in [20] by introducing a novel hybrid heuristic algorithm. It combines the strengths of genetic algorithms, ant colony optimization, and simulated annealing. The proposed heuristic fusion incorporates a novel mutation operator inspired by ant colony optimization. This operator allows individuals from different generations to exchange information, leading to better solutions and faster convergence. Additionally, a mechanism is introduced that dynamically adjusts the probability of genetic operations, similar to simulated annealing.

To effectively navigate AUVs in intricate environments, an improved differential evolution algorithm is proposed in [21]. This approach incorporates a novel adaptive elite neighborhood learning strategy to achieve a balance between the exploitation and exploration capabilities of improved differential evolution when tackling complex problems. Additionally, a rank-guided crossover probability selection strategy is introduced to ensure effective preservation of information from elite individuals. Finally, the study explores a novel distance-greedy selection strategy, which improves population diversity while maintaining convergence accuracy. Moreover, this research introduces a new double-layer coding model for eliminating invalid path points.

Energy consumption is one of the challenges of AUVs due to the limited battery power. An AUV requires more energy to probe the coastal waters over a large path against the rough environmental situations dominant in the sea. Therefore, AUVs must supply the best detection performance with decreased search distance. An optimal and efficient path planning algorithm should be applied in AUVs [22].

Due to the unmanned nature of AUVs and the importance of saving battery power in them, the issue of the optimal path is more critical in this case. Therefore, in this paper, AUVs are specifically discussed.

According to the importance of energy consumption, in this paper, a two-step method is presented to reduce not only the path but also the energy consumption; first, the Inclined Planes system Optimization algorithm as a powerful heuristic algorithm is employed to obtain the path with optimal length and then Genetic Algorithm is utilized to smooth the obtained path in order to decrease the energy consumption of the AUV. Finally, the stability of the presented heuristic path planner is analyzed. In this part, the effect of two structural parameters of applied heuristic algorithms on the designed path planner is investigated. It's worth mentioning that stability analysis of the heuristic path planner of AUVs is addressed in this paper for the first time.

Recently, many heuristic algorithms have been introduced. Some of these methods include the Orchard algorithm [23], the Meerkat optimization algorithm [24], the Artificial Rabbits optimization algorithm [25], and the Arithmetic Optimization algorithm [26]. Evaluating the performance and capabilities of each algorithm in different applications is one of the research areas of interest. Therefore, in this paper, the IPO algorithm is used for the first time in the field of AUV path planning to evaluate its performance. This research has shown that this algorithm is suitable for the path planning problem to meet expectations.

Genetic Algorithm is employed in this paper for several reasons. Firstly, GA has a theoretical foundation for convergence and guarantees global optimality. Secondly, it has been successfully applied to robot path planning in numerous prior studies. Thirdly, it is utilized as a hybrid approach in this research. However, it is crucial to emphasize that there are numerous alternative methods that could be investigated to examine their applicability in the path planning problem. Anti-coronavirus optimization algorithm [27], Backtracking search optimization algorithm [28] and Seasons optimization algorithm [29] are Some of these methods. In addition, a large number of algorithms can be extracted from [30] because more than three hundred researches related to bio-inspired and nature-inspired algorithms are reviewed in this paper.

The rest of this paper is organized as follows: first, the employed method for stability analysis is presented. After that, a review of Inclined Planes system Optimization algorithm is described. Then the proposed combinational method for achieving the optimal and smooth path of AUVs and the stability analysis of this heuristic path planner are presented. The experimental results are reported in the next section. Finally, conclusion of the paper is explained.

Two-Level Factorial Designs

When it's necessary to investigate the joint effects of several parameters on output in the experiments, factorial designs are extensively exerted. Joint effects implicate interactions and original effects. A significant case in this field is when two levels for each of the parameters exist; this type is named 2^k factorial designs as regards every replicate owning precisely 2^k experimental runs. 2^k factorial designs are helpful when screening tests should be performed to discover significant factors.

Adjusting a first order Response Surface Model (RSM) and acquiring the estimate of factor effect are the other applications of them.

One can employ factorial designs to determine the influence of several independent factors upon one dependent variable. There are two factors in 2^2 factorial designs (*A* and *B*), and two levels are defined for each parameter. The expressions high and low are employed for these levels. *A* and *B* indicate the impact of parameters *A* and *B*, respectively. Moreover, *AB* refers to the *AB* interaction. In this scheme, + and - are applied to show high and low levels related to each factor. Table 1 shows the design matrix, which specifies four treatment combinations of 2^2 design.

Table 1: The design matrix

Run	A	B
1	-	-
2	+	-
3	-	+
4	+	+

Small letters also illustrate the four runs; small letter related to each factor indicates the high level of it and the miss of one letter specifies the low level of that factor. So, *a* betokens the situation in which the level of *A* is high and the level of *B* is low and *ab* means the levels of two parameters are high. When the levels of all parameters are low, *1* is applied.

To calculate the original effect of *A* the difference of two averages is employed; the average of two combinations where the level of *A* is high ($\bar{y}_{A^+}$) and the average of two combinations where the level of *A* is low ($\bar{y}_{A^-}$). Thus, the main effect of *A* is specified as (1).

$$A = \bar{y}_{A^+} - \bar{y}_{A^-} = \frac{ab+a}{2n} - \frac{b+1}{2n} = \frac{ab+a-b-1}{2n} \tag{1}$$

In the same way, the main effect *B* is measured in (2).

$$B = \bar{y}_{B^+} - \bar{y}_{B^-} = \frac{ab+b}{2n} - \frac{a+1}{2n} = \frac{ab+b-a-1}{2n} \tag{2}$$

The interaction effect *AB* is the average of the difference of the effect *A* at low and high levels of *B*. So, the interaction *AB* is specified as (3).

$$AB = \frac{1}{2}\left\{\frac{[ab-b]}{n} - \frac{[a-1]}{n}\right\} = \frac{ab+1-a-b}{2n} \quad (3)$$

In many experiments of 2^k designs, both the direction and magnitude of the parameter effects are studied to discover significant parameters. Comparing the magnitudes of the effects in terms of their related standard errors is an advantageous method for advising the importance of the effects. To measure the standard error of *A, B,* and *AB*, one can compute the sums of squares for effects that are specified by SS_A, SS_B, and SS_{AB} in (4)-(6), respectively.

$$SS_A = \frac{(ab+a-b-1)^2}{4n} \quad (4)$$

$$SS_B = \frac{(ab+b-a-1)^2}{4n} \quad (5)$$

$$SS_{AB} = \frac{(ab+1-a-b)^2}{4n} \quad (6)$$

The total sums of squares, i.e. SS_T is measured by using (7) where y_{ijk} is the outcome of each run and $y_{...}$ is the sum of all runs.

$$SS_T = \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{n} y_{ijk}^2 - \frac{y_{...}^2}{4n} \quad (7)$$

Finally, the error sum of squares (SS_E) is measured by using (8).

$$SS_E = SS_T - SS_A - SS_B - SS_{AB} \quad (8)$$

Considering the degrees of freedom of SS_E i.e., $4\times(n-1)$, the mean square error, MS_E, is specified as (9).

$$MS_E = \frac{SS_E}{4\times(n-1)} \quad (9)$$

Therefore, the standard error of an effect is calculated by using (10).

$$se(effect) = \sqrt{\frac{1}{n} MS_E} \quad (10)$$

Finally, for each effect estimate, two standard error limits exist as (11).

$$\begin{aligned} &A \pm 2\times se(effect) \\ &B \pm 2\times se(effect) \\ &AB \pm 2\times se(effect) \end{aligned} \quad (11)$$

Due to the considered analysis, if the interval of an effect estimate does not include zero, it is introduced a significant effect. At the end, it should be mentioned that the coefficients for the regression model are half of the corresponding factor effect estimates [31].

Inclined Planes System Optimization Algorithm (IPO)

The movement of several globular things on a frictionless ramp is the main basis of the IPO algorithm; these objects want to arrive at the lowest place on the ramp. In this algorithm, some tiny balls, as algorithm agents, probe the search space to discover the optimal point. The principal scheme of this algorithm is to attribute the height to every object according to a reference point. This height value is received from the objective function; the obtained values are an approximation of potential energy of the agents at different points, and as the balls descend, this energy is converted into kinetic energy and thus caused the balls to accelerate downwards. Therefore, the agents repeatedly move in the exploring space to discover a better answer and hence acquire an acceleration [32].

In a supposed search space with *N* agents, the position of the *i*-th agent is calculated as (12):

$$x_i = \left(x_i^1, \ldots, x_i^d, \ldots, x_i^n\right), \quad for\ \ i = 1,2,\ldots,N \quad (12)$$

in which, x_i^d is the position of *i*-th agent in the *d*-th dimension in an *n*-dimensional system. The angle between the *i*-th and *j*-th agents in dimension *d*, i.e., φ_{ij}^d is measured by (13):

$$\varphi_{ij}^d(t) = \left(\tan^{-1}\left(\frac{f_j(t) - f_i(t)}{x_i^d(t) - x_j^d(t)}\right)\right), \quad for\ d = 1,\ldots,n\ and\ i,j = 1,2,\ldots,N, i \neq j \quad (13)$$

where, $f_i(t)$ is the value of the objective function, i.e., height for the *i*-th agent at the time *t*. A certain agent wants to move toward the lowest heights on the ramp, therefore the agents with lower height values are the only agents used in calculating the acceleration. The direction and amplitude of acceleration for the *i*-th agent in dimension *d* and at the time *t*, is demonstrated in (14) where, *U*(.) means the unit step function:

$$a_i^d(t) = \sum_{j=1}^{N} U\left(f_j(t) - f_i(t)\right).\sin\left(\varphi_{ij}^d(t)\right) \quad (14)$$

Finally, (15) is employed to update the position of the balls:

$$\begin{aligned} x_i^d(t+1) = {} & k_1.rand_1.a_i^d(t).\Delta t^2 \\ & + k_2.rand_2.v_i^d(t).\Delta t + x_i^d(t) \end{aligned} \quad (15)$$

$rand_1$ and $rand_2$ are two random parameters distributed uniformly on the [0,1] interval. $v_i^d(t)$ is the velocity related to the *i*-th agent at time *t* and in dimension *d*. k_1

and k_2 are applied to control the exploring process of the algorithm. These factors are defined by using (16) and (17):

$$k_1(t)=\frac{c_1}{1+\exp\left(\left(t-shift_1\right)\times scale_1\right)} \tag{16}$$

$$k_2(t)=\frac{c_2}{1+\exp\left(\left(t-shift_2\right)\times scale_2\right)} \tag{17}$$

$v_i^d(t)$ is defined in (18), where x_{best} is placed in numerator to demonstrate the agent's desire to achieve the best position in each run:

$$v_i^d(t)=\frac{x_{best}^d(t)-x_i^d(t)}{\Delta t} \tag{18}$$

Design and Stability Investigation of Heuristic Path Planner for AUVs

This research proposes a hybrid approach that combines the strengths of IPO and Genetic GA to achieve an optimal and smooth path for Autonomous Underwater Vehicles. This two-step method leverages IPO for global optimization and GA for path refinement, ultimately leading to efficient and safe AUV navigation. In the first step, the IPO algorithm is used to find the optimal path. The objective function is one of the important issues that should be defined properly when employing heuristic algorithms for optimization. Path length is considered as objective function in this research. It is expected that this objective function will be minimized by using IPO algorithms.

The path length is considered as a criterion for measuring the quality of the path in path planning problem. Therefore, the shorter the path it is, the better the fitness function it is.

In this research, the Euclidean distance is used to calculate the path length. When the system is implemented in the real world, the unit of the obtained path length will be in terms of the actual distance between the start and the end points. An important point to note here is the path length depends on various factors, including the speed of the AUV, the run speed of the algorithm, and the sonar accuracy. For example, suppose that based on sonar accuracy, the sonar detects an obstacle ten meters away, and the speed of the AUV is one meter per second. Additionally, the algorithm requires 10 runs to reach the next point, which takes one second. Therefore, during this time, the AUV has moved one meter, and until the sonar doesn't warn, the algorithm continues on its path and begins to explore the next point.

In the path planning problem, the goal is to find a possible path from the starting point to the end point of the movement, and after optimizing the length of this route, the path should have the shortest length. While safety is a paramount concern for any path, a safe path should be both collision-free and minimize its overall length. So, one can conclude that there is a constrained optimization problem to solve. Here, the penalty function method is used to solve this constrained optimization problem. For this purpose, the objective function is defined as the summation of the path length with a penalty function in the form of (19).

$$Objective\ Function = L\times(1+\beta\times V) \tag{19}$$

where L is the path length, V is the penalty function, and β is the coefficient of the penalty function. For all points of the path, the penalty is calculated, and finally, the penalty function will be the average of all calculated values. Obviously, if a point of the path does not collide with an obstacle, the penalty for that point is zero.

After detecting the optimal path using IPO, genetic algorithm is employed to smooth the obtained path. The importance of smoothing path is due to the energy consumption of AUV when turning.

So, if the path improves in such a way that the AUV turns with less angles, as a result, less energy is consumed.

It's noteworthy that path smoothing with a genetic algorithm can offer a twofold benefit for AUVs. First, it reduces energy consumption by minimizing unnecessary rotations. Second, it shortens the overall path compared to IPO's obtained path by eliminating redundant waypoints.

The second goal of this research is using two-level factorial designs to investigate the stability of designed heuristic path planner. So, two structural parameters which are common in both algorithms, are selected to check the stability of the proposed path planner. These parameters are number of iterations and population size.

Results and Discussion

The proposed method used for designing heuristic path planner is implemented for three different maps and the results are reported as follows. The dimensions of the search agents are equal to the number of points in the cubic spline Interpolation.

It's worth mentioning the values of parameters in the IPO algorithm are considered as below:

c_1 = 0.225.

c_2 = 2.283.

$shift_1$ = 121.044.

$shift_2$ = 149.675.

$scale_1$ = 0.056.

$scale_2$ = 0.525.

A. Map 1

In this case, in the first step, IPO algorithm has reached the path with a length of 20.1796. In the next step, after applying genetic algorithm to this problem, two important consequences have obtained; firstly, the path has become smooth and secondly, the length of the path has reduced to 12.7596; That is, the path length has improved 36.77% by using the proposed method. The path found by IPO and the smooth path discovered by the combined algorithm are shown in Fig. 1 and Fig. 2.

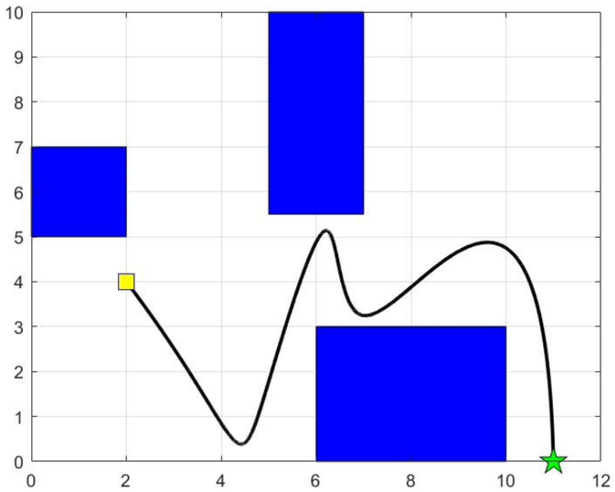

Fig. 1: Optimal path for map 1 using IPO algorithm.

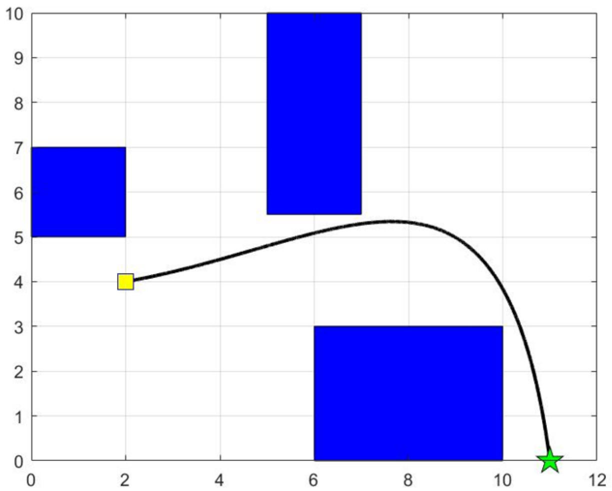

Fig. 2: Optimal and smooth path for map 1 using the hybrid algorithm of IPO-GA.

B. Map 2

In this case, at the first step, the path length obtained by IPO algorithm is 27.2011. The application of the genetic algorithm in the second step yielded two key results. First, the path was significantly smoothed. Second, the path length was reduced to 13.5540, representing a remarkable improvement of 50.17% compared to the original path. The path found by IPO and the smooth path discovered by the combined algorithm for map 2 are shown in Fig. 3 and Fig. 4.

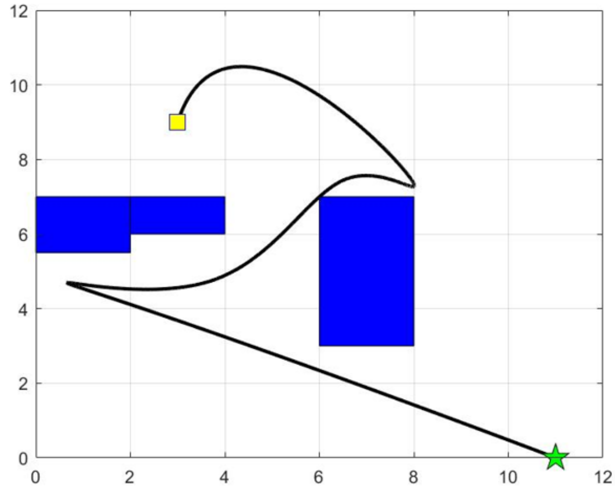

Fig. 3: Optimal path for map 2 using IPO algorithm.

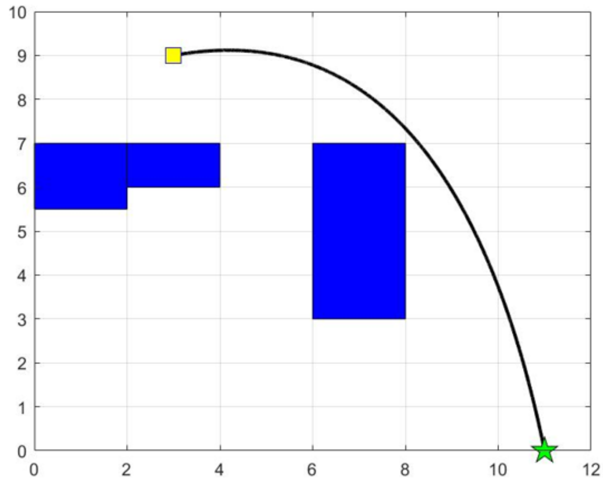

Fig. 4: Optimal and smooth path for map 2 using the hybrid algorithm of IPO-GA.

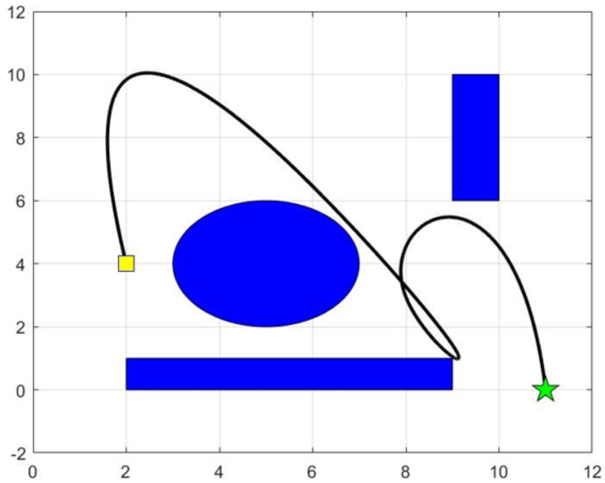

Fig. 5: Optimal path for map 3 using IPO algorithm.

C. Map 3

During the first step, the IPO algorithm was utilized to optimize path length, resulting in a path with a length of 29.4538. The final step involving the genetic algorithm yielded two significant improvements. First, the path became noticeably smoother. Second, the path length was reduced to 15.0879, representing a noteworthy improvement of 48.77% achieved through the proposed hybrid method.

The path found by IPO and the smooth path discovered by the combined algorithm for map 3 are shown in Fig. 5 and Fig. 6.

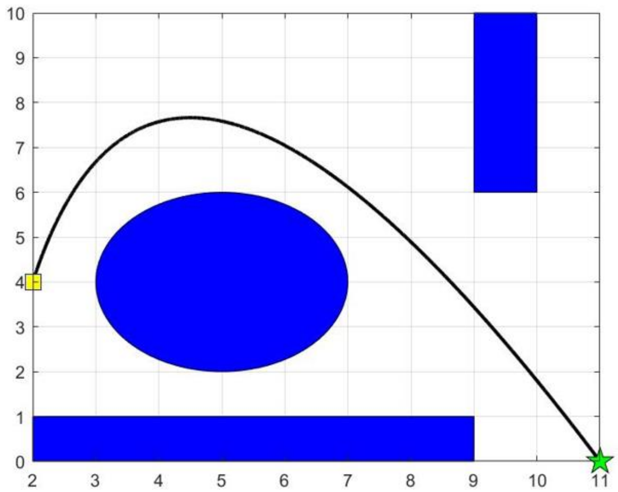

Fig. 6: Optimal and smooth path for map 3 using the hybrid algorithm of IPO-GA.

The obtained results in each step for three maps are shown in Table 2.

Table 2: The path length in each step

Map	First Step	Second Step	Improvement (%)
1	20.1796	12.7596	**36.77**
2	27.2011	13.5540	**50.17**
3	29.4538	15.0879	**48.77**

The maps used in this research are synthetic maps designed for the present study to evaluate the performance of the proposed method for AUV path planning. These maps use obstacles with different geometric shapes and variant sizes to evaluate the effectiveness of the proposed method. Therefore, a one-to-one comparison with other researches is not possible. However, computational cost can be considered as a metric for comparison with previous studies, in which case the number of iterations can be an appropriate choice. Table 3 shows the comparison results between the proposed method and several other methods in terms of number of iterations. According to this comparison, the number of iterations required by the proposed method is significantly lower than those of other methods.

Table 3: A comparative analysis of Number of iterations between the proposed method and other existing methods

Method	Number of iterations
[16]	2000
[33]	1000
[34] _PSO	2400
[34]_ACO	480
[35]	1000
Proposed Method	500

It's worth to mention that we can never accurately determine the actual operational delay of the above methods because the coding style and instructions are crucial for delays.

In fact, even if we measure the order of computations, we can never accurately determine the computational cost, as there are many important factors that affect it during implementation, such as the type of hardware, the program itself, and the programmer. If all of these factors are identical, there are still important considerations during operational implementation; even if we have a highly optimized code running on powerful hardware, there are still operational factors that can affect execution time such as sensor delay, sensor accuracy, data acquisition rate, data processing time, and data decoding; these operational considerations introduce a layer of variability that makes it difficult to precisely determine the computational cost of a method in a real-world setting.

In practice, we often rely on approximations, and metrics to get a general sense of the computational efficiency of different methods. However, it's important to recognize that these estimates may not reflect the exact execution time in a specific application or environment.

It is also important to consider that the maps, the programming style, and the methods employed in this research are all unique.

It is precisely after taking these factors into account, it appears that the number of iterations is the most logical measure and criterion that can be considered as an estimate, rather than saying that because the number of iterations is a certain value, the computational cost must also be the same.

In fact, the claim has been that since it was not possible to implement these methods, the number of iterations was simply used as a handy and documented metric for comparison. So, without a doubt the number of iterations of an algorithm may not accurately indicate its speed or efficiency, but this can still be used as a metric. Now, we can take a step further and complete the before statement: even the number of iterations reported is not entirely reliable as all heuristic methods are problem based and random methods.

This means that if a heuristic method finds a solution to a problem in an experiment with a specific number of iterations, it may take more or fewer iterations to find the solution to the same problem in a different experiment. However, in the absence of a documented metric, it seems that one of the logical metrics to use is the number of iterations. This is because the search loop is the core component of the body of all heuristic methods. In fact, all heuristic methods have a search loop that constitutes the largest part of the body of a heuristic method, including all the operators that need to be performed and the termination conditions. In fact, the search loop is typically the most computationally expensive part of a heuristic method. Therefore, the number of iterations of this part, as a common and substantial body of all heuristic algorithms, is an appropriate metric to understand how many times calculations were needed to reach the main solution.

As mentioned before, two selected factors for stability analysis are population size (*A*) and number of iterations (*B*) and two levels are assumed for each factor; 100 and 200 for population size, 500 and 700 for number of iterations in IPO. 50 and 100 for population size, 700 and 1000 for number of iterations in GA. Moreover, map 3 is used to accomplish stability analysis.

It's worth mentioning that for each treatment combination, the experiment is repeated two times in order to obtain needful data for stability investigation. Obtained data from two replicates for stability analysis of IPO are indicated in Table 4.

Table 4: Observed Data for IPO

Treatment Combination	Replicate	Path Length
1	I	27.4652
	II	33.7509
a	I	23.3083
	II	29.3106
b	I	23.8117
	II	19.7694
ab	I	28.193
	II	33.1777

Two standard error limits on the effects related to supposed fitness function (path length) is calculated by employing observed data. Obtained interval for each effect is shown in (20).

$$\begin{aligned} &A: 2.2981 \pm 5.4020 \\ &B: -2.2208 \pm 5.4020 \\ &AB: 6.5967 \pm 5.4020 \end{aligned} \tag{20}$$

Due to above intervals, it's obvious that effect *AB* in path length, optimized with IPO, is important because its interval does not include zero.

Now, the regression model for path length measure for IPO can be specified as (21). Where x_1 and $x2$ are the design factors *A* and *B*, respectively, on the coded (-1, +1) scale and β_0 is the mean of all observations of path length measure.

$$\begin{aligned} y &= \beta_0 + \beta_{12} x_1 x_2 \\ &= 27.3484 + \left(\frac{6.5967}{2}\right) x_1 x_2 \end{aligned} \tag{21}$$

The direction of each factor, which is also extracted from this analysis, is another important point; the effect *AB* in path length is positive in this path planning problem; i.e., if *AB* increases from the low to the high level, the path length measure will increase.

Obtained data from 2 replicates for stability analysis of GA are shown in Table 5.

Table 5: Observed Data for GA

Treatment Combination	Replicate	Path Length
1	I	17.2608
	II	19.9797
a	I	10.7910
	II	13.8029
b	I	13.2827
	II	13.4881
ab	I	12.9068
	II	15.8683

Two standard error limits on the effects related to the path length obtained by GA, is defined by employing observed data. The related interval for each effect is shown in (22).

$$\begin{aligned} &A: -2.6606 \pm 2.5138 \\ &B: -1.5721 \pm 2.5138 \\ &AB: 3.6627 \pm 2.5138 \end{aligned} \tag{22}$$

Due to obtained intervals, one can conclude that effect *A* and *AB* in path length, optimized by applying GA, are significant because these intervals do not include zero. Hence, the regression model for path length measure for GA can be described as (23).

$$y = \beta_0 + \beta_1 x_1 + \beta_{12} x_1 x_2 = 14.6725 + \left(\frac{-2.6606}{2}\right) x_1 + \left(\frac{3.6627}{2}\right) x_1 x_2 \quad (23)$$

The effect *A* in path length measure is negative; i.e., increasing *A* from the low to the high level will decrease the path length measure. Also, *AB* is positive in this path planning problem; i.e., increasing the level of *AB* from low to high leads the measure of path length to increase.

Conclusion

In this study, a hybrid heuristic method is developed to detect the optimal and smooth path from the starting point to the target point for AUVs. The proposed method is implemented in two steps by using IPO and GA. In the first step, the only goal is to find an optimal path for AUV by applying IPO. In the next step, GA is employed to smooth the detected path obtained in the previous step. The results of the last step are an optimal and smooth path, i.e., the GA not only smooths the path but also decreases the length of the path. This method reduces AUV energy consumption by eliminating unnecessary turns.

So, it is an efficient method for the path planning of AUVs.

The IPO-GA algorithm has applied on three different maps. In all three cases, the results confirm the efficiency of the proposed method; after using GA, the path becomes smooth and also shorter. In the best case, an improvement of 50.17% is seen in the length of the path.

After developing the proposed heuristic path planner, it's time to study the stability of this heuristic path planner.

Due to the random nature of heuristic algorithms, this part seems to be necessary. By applying the two-level factorial design, one can efficiently evaluate how each parameter (population size and number of iterations) and their potential interactions affect the objective function (path length).

This will help you identify the optimal configuration for your genetic algorithm in optimizing AUV path planning. This approach can find significant effects in each situation and also, can specify regression model related to the defined objective function.

Finally, it is necessary to emphasize that some suggestions can be made for future work in this field. One of these suggestions is using other new algorithms with a new objective function. It is also possible to use multi-objective heuristic algorithms with several objective functions.

Another important suggestion is to investigate the stability of the proposed heuristic method with other approaches and also by considering other parameters.

Author Contributions

Z. K. Pourtaheri designed the framework of the research. She designed the experiments and interpreted the results and wrote the manuscript.

Acknowledgment

This work is completely self-supporting, thereby no any financial agency's role is available.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, 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

IPO	Inclined Planes system Optimization
AUV	Autonomous Underwater Vehicle
GA	Genetic Algorithm
RSM	Response Surface Model
PSO	Particle Swarm Optimization
ACO	Ant Colony Optimization

References

[1] A. Alvarez, A. Caiti, R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," IEEE J. Oceanic Eng., 29(2): 418-429, 2004.

[2] Q. R. Zhang, "A hierarchical global path planning approach for AUV based on genetic algorithm," in Proc. 2006 International Conference on Mechatronics and Automation, 2006.

[3] W. Hong-Jian, B. Xin-Qian, Z. Jie, D. Fu-Guang, X. Guo-Qing, "A GA path planner based on domain knowledge for AUV," in Proc. Oceans '04 MTS/IEEE Techno-Ocean '04, 3: 1570-1573, 2004.

[4] J. Cao, Y. Li, S. Q. Zhao, X. S. Bi, "Genetic-algorithm-based global path planning," in Proc. 9th International Symposium on Computational Intelligence and Design, 2016.

[5] Y. Sun, R. B. Zhang, "Research on global path planning for AUV based on GA," in Proc. Mechanical Engineering and Technology: 311-318, 2012.

[6] Q. Li, X. H. Shi, Z. Q. Kang, "The application of an improved genetic algorithm in the AUV global path planning," Appl. Mech. Mate., 246: 1165-1169, 2013.

[7] S. K. Yan, F. Pan, "Research on route planning of AUV based on genetic algorithms," in Proc. IEEE International Conference on Unmanned Systems and Artificial, 2019.

[8] H. S. Lim, S. S. Fan, C. K. H. Chin, S. H. Chai, N. Bose, "Particle swarm optimization algorithms with selective differential evolution for AUV path planning," Int. J. Rob. Autom., 9(2): 94-112, 2020.

[9] L. Wang, L. L. Liu, J. Y. Qi, W. P. Peng, "Improved quantum particle swarm optimization algorithm for offline path planning in AUVs," IEEE Access, 8: 143397-143411, 2020.

[10] Z. Zeng, K. Sammut, L. Lian, F. He, A. Lammas and Y. Tang, "A comparison of optimization techniques for AUV path planning in environments with ocean currents," Rob. Auton. Syst., 82: 61-72, 2016.

[11] H. Zhang, X. Shi, "An improved quantum-behaved particle swarm optimization algorithm combined with reinforcement learning for AUV path planning," J. Rob., 2023.

[12] X. Yu, W. N. Chen, T. Gu, H. Q. Yuan, H. Zhang, J. Zhang, "ACO-A*: Ant colony optimization plus A* for 3-D traveling in environments with dense obstacles," IEEE Trans. Evol. Comput., 23(4): 617-631, 2019.

[13] Y. Ma, Z. Y. Mao, T. Wang, J. Qin, W. J. Ding, X. Meng, "Obstacle avoidance path planning of unmanned submarine vehicle in ocean current environment based on improved firework-ant colony algorithm," Comput. Electr. Eng., 87, 106773, 2020.

[14] C. L. Hu, F. Zhang, "Research on AUV global path planning based on multi-objective ant colony strategy," in Proc. Chinese Automation Congress, 2019.

[15] G. F. Che, L. J. Liu, Z. Yu, "An improved ant colony optimization algorithm based on particle swarm optimization algorithm for path planning of autonomous underwater vehicle," J. Ambient Intell. Hum. Comput., 11(8): 3349-3354, 2020.

[16] D. Zhang, X. You, S. Liu, H. Pan, "Dynamic multi-role adaptive collaborative ant colony optimization for robot path planning," IEEE Access, 8: 129958-129974, 2020.

[17] M. Ronghua, C. Xinhao, W. Zhengjia, D. Xuan, "Improved ant colony optimization for safe path planning of AUV," Heliyon, 10(7), 2024.

[18] G. Chen, D. Cheng , W. Chen, X. Yang, T. Guo , "Path planning for AUVs based on improved apf-ac algorithm," Comput. Mater. Continua, 78(3), 2024.

[19] S. P. Sahoo, B. Das, B.B. Pati, F. P. Garcia Marquez, I. Segovia Ramirez, "Hybrid path planning using a bionic-inspired optimization algorithm for autonomous underwater vehicles," J. Mar. Sci. Eng., 11(4): 761, 2023.

[20] J. Wen, J. Yang, T. Wang, "Path planning for autonomous underwater vehicles under the influence of ocean currents based on a fusion heuristic algorithm," IEEE Trans. Veh. Technol., 70(9): 8529-8544, 2021.

[21] J. Fan, L. Qu , "Innovative differential evolution algorithm with double-layer coding for autonomous underwater vehicles path planning in complex environments," Ocean Eng., 303, 2024.

[22] O. Marceau, J. M. Vanpeperstraete, "AUV optimal path for leak detection," in Proc. OCEANS–Anchorage, 2017.

[23] M. Kaveh, M. S. Mesgari, B. Saeidian, "Orchard Algorithm (OA): A new meta-heuristic algorithm for solving discrete and continuous optimization problems," Math. Comput. Simul., 208: 95-135, 2023.

[24] S. Xian, X. Feng, "Meerkat optimization algorithm: A new meta-heuristic optimization algorithm for solving constrained engineering problems," Expert Syst. Appl., 231, 2023.

[25] L. Wang, Q. Cao, Z. Zhang, S. A. Mirjalili, "Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems," Eng. Appl. Artif. Intell., 114, 2022.

[26] L. Abualigah, A. Diabat, S. A. Mirjalili, M. Abd Elaziz, "The arithmetic optimization algorithm," Comput. Meth. Appl. Mech. Eng., 376: 2021.

[27] H. Emami, "Anti-coronavirus optimization algorithm," Soft Comput., 26: 4991-5023, 2022.

[28] P. Civicioglu, "Backtracking search optimization algorithm for numerical optimization problems," Appl. Math. Comput., 219(15): 8121-8144, 2013.

[29] H. Emami, "Seasons optimization algorithm," Eng. Comput., 38: 1845-1865, 2022.

[30] M. Daniel, J. Poyatos, J. Del Ser, S. García, A. Hussain, F. Herrera, "Comprehensive taxonomies of nature-and bio-inspired optimization: Inspiration versus algorithmic behavior," Cognit. Comput., 12: 897-939, 2020.

[31] H. Myers Raymond, C. Montgomery Douglas, M. Anderson-Cook Cristine, Response surface methodology: process and product optimization using designed experiments, John Wiley & Sons, 2016.

[32] H. Mozaffari, M. H. Abdy, S. H. Zahiri, "IPO: an inclined planes system optimization algorithm," Comput. Inf., 35(1): 222-240, 2016.

[33] L. Dogan, U. Yuzgec, "Robot path planning using gray wolf optimizer," in Proc. International Conference on Advanced Technologies, Computer Engineering and Science, 2018.

[34] M. Yousif, A. Salim, W. Jummar, "A robotic path planning by using crow swarm optimization algorithm," Int. J. Math. Sci. Comput., 7(1): 20-25, 2021.

[35] X. Li, D. Wu, J. He, M. Bashir, M. Liping, "An improved method of particle swarm optimization for path planning of mobile robot," J. Control Sci. Eng., 2020: 1-12, 2020.

Biographies

Zeinab Khatoun Pourtaheri received her B.Sc. and M.Sc. degrees in Electrical Engineering from Shahid Bahonar University of Kerman in 2010 and 2012, respectively, and Ph.D. degree in Electrical Engineering from University of Birjand in 2017. She is an Assistant Professor with the Department of Mechatronic Engineering at Higher Education complex of Bam. Her major research interests include Path planning, heuristic algorithms, optimization, ensemble classification, and stability analysis of heuristic methods.

- Email: z.pourtaheri@bam.ac.ir
- ORCID: 0000-0003-0465-5488
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA

How to cite this paper:
Z. K. Pourtaheri, “Design and stability analysis of a novel path planner for autonomous underwater vehicles,” J. Electr. Comput. Eng. Innovations, 13(1): 1-12, 2025.

DOI: 10.22061/jecei.2024.10624.725

URL: https://jecei.sru.ac.ir/article_2170.html