ISSN: 2788–7669 *Journal of Machine and Computing 4(3)(2024)*

Improving Autonomous Underwater Vehicle Navigation: Hybrid Swarm Intelligence for Dynamic Marine Environment Path Finding

[1]Husam Alowaidi, [2]Hemalatha P, [3]Poongothai K, [4]Sundoss ALmahadeen, [5]Prasath R and [6]Amarendra K
[1]Department of Information Technology, College of Science, University of Warith Al-Anbiyaa, Karbala, Iraq.
[2]Department of TIFAC-CORE in Cyber Security, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu, India.
[3]Department of Information Technology, M. Kumarasamy College of Engineering, Karur, Tamil Nadu, India.
[4]Department of Civil Engineering, Mutah University, Karak, Jordan.
[5]Department of Computer Science and Engineering, KCG College of Technology, Chennai, Tamil Nadu, India.
[6]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India.
[1] husam@uowa.edu.iq, [2]p_hemalatha@cb.amrita.edu, [3] sakthipoongi@gmail.com, [4] sundoss.radi@mutah.edu.jo, [5] prasathr05@gmail.com, [6] amarendra@kluniversity.in

Correspondence should be addressed to Hemalatha P : p_hemalatha@cb.amrita.edu.

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202404061
Received 22 November 2023; Revised from 02 January 2024; Accepted 06 June 2024
Available online 05 July 2024.

Abstract – Underwater research and monitoring operations rely significantly on Autonomous Underwater Vehicles (AUVs) for scientific investigations, resource management, and monitoring, and underwater infrastructure is provided maintenance levels amid other applications. Efficient navigation and preventative methods are only a couple of the numerous challenges that Path-Finding (PF) in rapidly changing and sophisticated Underwater Environments (UE) requires overcoming. Dynamic environments and real-time improvements are problems for traditional models. In order to provide superior solutions for navigating uncertain UE, this work suggests a hybrid optimization technique that combines Ant Colony Optimization (ACO) for local path selection with Particle Swarm Optimization (PSO) for global path scheduling. Runtime efficiency, accuracy, and distance focused on decrease are three metrics that demonstrate how the PSO-ACO hybrid method outperforms conventional algorithms, proving its significance for improving AUV navigation. The improvement of AUV functions in fields such as underwater research, along with others, is supported by the current research, which further assists with the invention of Autonomous Underwater Navigation Systems (AUNS). The PSO+ACO hybrid method is superior to the PSO, ACO, and GA algorithms in pathfinding with a 6.43-second execution time and 93.5% accuracy—the ACO model completed in 12.53 seconds, superior to the proposed system.

Keywords – Autonomous Underwater Vehicles, Deep Learning, Ant Colony Optimization, Genetic Algorithms, Accuracy.

I. INTRODUCTION

The investigation and tracking of the world's lowest oceans, reservoirs, and lakes have been greatly improved by the emergence of Autonomous Underwater Vehicles (AUV), which are addresses that have been employed in recent years to indicate significant technical progress [1]. These automobiles are called autonomous vehicles and usually function without requiring human involvement. These autonomous vehicles have the possibility to be employed in an enormous number of programs, which include but are not restricted to scientific research and data collecting, inspection of underwater systems, army tracking, and the field of exploring undiscovered lands according to the outermost layer of the deep sea [2]. Such AUVs are essential for improving the study of marine systems, may be beneficial to the advancement of maritime security, and are also possible remedies for developing techniques in taking advantage of the resources of the bottom of the sea.

Their knack to function autonomously and navigate across complicated and unreachable waters allows them to complete all of these objectives. The PF system plays an important role in the successful operation of AUVs, notwithstanding their great potential. Underwater environments (UE) are typically challenging to forecast that which has been an important challenge for the development of this system. In order to maximize the effectiveness of their operation, the AUVs have had to become accustomed to an operational setting that constantly evolves. Sea waves, underwater challenges, and various environments define this UE. With every aspect taken into account, AUNs are required to be adaptable and efficient to navigate underwater successfully. Although reducing Energy Consumption (EC) and improving the precision of collecting information, these techniques must be enabled to adapt successfully to developments that have not been predicted [3].

A greater number of individuals become aware of how crucial it is to use good AUNS in UE owing to how constantly changing and unpredictable these settings are. Incorporating real-time information into the UE proved to be a prevalent issue for conventional PF techniques. The probability of security risks has grown due to insufficient and erroneous navigation, for example, which occurred regularly as a consequence of all these factors [4]. These restrictions have made it more challenging to develop effective systems that can respond to real-world circumstances while maintaining secure travel. Optimizing task execution has been successful in large part due to this threshold [5].

Algorithms designed in order to deal with those problems frequently employ incorrect predictions about the external environment or utilize static pictures of the UE. However, unfortunately, these methods may not help to capture the complexities of real-world scenarios fully. While many heuristic and swarm intelligence methods like Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) have also been applied to AUNS, even such models each have limitations. PSO models, though, excel in global search but may overlook local optimality; ACO models have provided detailed local path refinement. However, they can be quite computationally intensive for larger input areas, and GA could offer a balance between exploration and exploitation, but these models require extensive tuning to achieve desired outcomes [6].

In this work, in order to address the above-discussed challenges in the field of PF in AUV and the inherent limitations of empirical models, we have proposed a PSO+ACO hybrid model for PF for AUV. The hybrid approach combines PSO+ACO to utilize PSO's broad search capability for identifying global paths and ACO's detailed focus for local path changes. The proposed solution addresses the problem of PF using the most efficient method, which is mapping the complete UE into a layout. In addition, collision prevention techniques are employed to do this. In order to execute the study, a Virtual Matrix System (VMS) was built, and the success of the method was assessed based on its capacity to decrease the range to the objective, as well as its precision and time efficiency metrics. Using tests, the analysis demonstrates that the proposed PSO+ACO hybrid model helps optimize the route. In particular, this research shows that the recommended approach is higher than individual PSO, ACO, or GA regarding PF.

The paper is structured as follows: Section 2 presents the literature of the work, Section 3 presents the methodology, Section 4 presents the evaluation of the work and Section 5 presents the conclusion.

II. LITERATURE REVIEW

Applying the Deep Deterministic Policy Gradient (DDPG) technique, researchers established a model that could improve the planning of paths for AUV. The design aimed to address the problems of driving AUVs throughout underwater tunnels and avoid anonymous risks. Their strategy employed sensor data as input, and with the support of that sensor data, researchers attempted to attain the optimum results in terms of the driving speed and the rotation angle. The AUVs have effectively steered past static and dynamic hazards by employing a structured reward function model and the artificially generated potential field technique for persistent rewards.

An article that was presented by [8] studies the subject of PF for Autonomous Surface Vehicles (ASV) within and around challenging coastlines. Researchers intended to identify a remedy to the problem by using an approach called PSO, which has been improved by Visibility Graphs (VG) to define the number of option possibilities. This approach efficiently addresses the primary constraints of early convergence that are vital to the PSO, enhancing its reliability in sending near-optimal outcomes. Also, the authors implemented pragmatic reward-based planning in order to achieve an acceptable balance between route performance and experimental trip results.

The problem that has been given to the domain of Internet of Underwater Things (IoUT) technologies was investigated by [9], which investigated the use of algorithmic methods and Reinforcement Learning (RL) methods. The researchers proposed a method that focuses on Q-learning and ACO for the optimal PF of AUV. Apart from focusing on improving the Value of Information (VoI) by developing the PF of the AUV, experts determined that significantly reducing the delay throughout the data collection process was necessary. Tests have been done that showed the success rate of the algorithm they used when compared with standard algorithms in many circumstances. In the Underwater Wireless Sensor Networks (UWSN) framework, [10] suggested a PF system that thoughtfully incorporated any possible reduction in EC. Using GA, researchers built a successful AUV route planning system that requires the advantage of reducing EC. Through experiments, they had shown that their model had faster convergence and extended the lifetime of UWSN.

[11] had employed an improved Fireworks-Ant Colony Hybrid Algorithm (FACHA) for 2D autonomous PF. They developed their model to handle path planning in environments that are greatly affected by ocean currents and problems. This model included various factors such as EC, navigation time, and distance costs to achieve this task. They successfully demonstrated the effectiveness of their proposed work through different simulation experiments.

An integrated model was proposed by [12], which combines the memory function with the artificial Jellyfish Search (JS) algorithm to improve its convergence accuracy. They formulated an objective function that considered the ocean current disturbance model. The improved algorithm demonstrated effectively optimal performance for the time cost and ocean current penalty cost along the planned paths, thereby showcasing their model's adaptability for multi-AUV movements.

The authors in [13] designed an autonomous underwater review robot, where a novel controller is fabricated to reduce steady-state errors. Hybrid swarm algorithms are applied to detect and classify Underwater images [14]. Additionally, the article emphasises the hybrid method of GA [15], which can be used for PF in UE. Followed by [16], the authors explore Federated Deep Reinforcement Learning (FDRL) for efficient pathfinding in UE.

III. METHODOLOGY

Setup A Grid System for Underwater Terrain Environment (UTE)

To simulate the complex UTE is represented as a 2D matrix, M, where each element, E_{mn}, corresponds to a specific portion of the seabed. The UE is segmented into a grid, with each grid cell representing either an obstacle or open water. Cells corresponding to obstacles are assigned a value of '0'. Conversely, cells representing navigable water are assigned a value of '1'. This study pays morphological operations, dilation and erosion to mitigate the computational load imposed by the terrain's complexity on the AUNA algorithm. These operations refine the grid, M', emphasizing essential navigational data and discarding superfluous data. Such simplification is pivotal for optimizing the AUV 's-PF capabilities.

Each grid cell within the VMS is uniquely identified by a co-ordinate pair, (u_m, v_n), facilitating accurate AUV localization. The transformation from a cell's linear index, l, to bidimensional co-ordinates is crucial for mapping and navigation systems. Given the VMS dimensions $R \times S$ (rows by columns), the formulas for calculating the row index u_m and the column index v_n from the linear index l are as follows:

Row Index (u_m)

The row index is determined by dividing the linear index by the number of columns, rounded down to the nearest whole number, EQU (1)

$$u_m = \left\lfloor \frac{l}{S} \right\rfloor \tag{1}$$

Column Index $(\boldsymbol{v_n})$

The column index is calculated as the remainder of the linear index divided by the number of columns, EQU (2)

$$v_n = l \bmod S \tag{2}$$

These EQU (1) and EQU (2) ensure that each cell in the grid can be precisely located and referenced during the simulation of AUV path-finding, enhancing the accuracy of navigation algorithms. The objective function, denoted as F, measures the viability and optimality of navigational paths through the underwater grid environment. It includes path length, EC, hindrance avoidance, and environmental adaptability.

Path Length $(\boldsymbol{L})$

The length of the path, L, is calculated as the sum of the distances between consecutive nodes (grid cells) along the path, expressed as EQU (3)

$$L = \sum_{i=1}^{n-1} d(P_i, P_{i+1}) \tag{3}$$

where $d(P_i, P_{i+1})$ represents the distance between repeated points P_i and P_{i+1} on the path, and n is the total number of points.

Energy Consumption $(\boldsymbol{E})$

EC, E is modelled as a function of path length and vehicle-specific parameters, EQU (4)

$$E = \alpha \cdot L + \beta \cdot \sum_{i=1}^{n} e(P_i) \tag{4}$$

where $'\alpha'$ represents the energy cost per unit distance, β is a co-efficient accounting for the energy cost due to sensor and system functions, and $e(P_i)$ encapsulates the energy overhead at a point P_i, including communications with environmental factors.

Obstacle Avoidance ($\boldsymbol{O}$)

Obstacle avoidance, O, ensures the AUV steers clear of hazards, preserving its integrity and mission continuity. It can be incorporated into the objective function through drawback terms associated with proximity to known obstacles, EQU (5)

$$O = \gamma \cdot \sum_{i=1}^{n} \frac{1}{d(P_i, \text{Obs}) + \epsilon} \tag{5}$$

where Obs represents the location of problems, $d(P_i, Obs)$ is the distance from the point P_i to the nearest obstacle, γ is a weighting factor, and ϵ is a small constant to prevent division by '0'.

Environmental Adaptability ($\boldsymbol{A}$)

Adaptability to environmental conditions, A, reflects the AUV's ability to navigate efficiently through dynamic underwater currents and varying terrains, potentially optimizing EC and reducing transit time, EQU (6)

$$A = \delta \cdot \sum_{i=1}^{n} a(P_i) \tag{6}$$

where $a(P_i)$ evaluates the adaptability of the path at point P_i in response to environmental conditions, and δ is a weighting factor that balances adaptability with other path-finding objectives. The objective function $F(P)$ is then expressed by considering the above factors as EQU (7)

$$F(P) = \lambda_1 \cdot L(P) + \lambda_2 \cdot E(P) + \lambda_3 \cdot O(P) + \lambda_4 \cdot A(P) \tag{7}$$

Where P denotes a specific path and $\lambda_1, \lambda_2, \lambda_3$, and λ_4 are weighting co-efficient.

Optimization Strategy

To handle the complexities associated with the UTE for AUV, this work introduces a hybrid model that combines the strengths of PSO+ACO. The dual-phase optimization process is directed by this work objective function F, to minimize the combined criteria of path length, EC, problem avoidance, and environmental adaptability. The optimization target can be formally stated as EQU (8).

$$P^* = \text{Arg}\,\underset{P}{\text{Min}} F(P) \tag{8}$$

Here, P^* denotes the optimal path that minimizes the objective function $'F'$.

PSO for Global PF

PSO utilizes the collective intelligence behaviour inherent within swarms [16-20]. They adeptly explore and exploit the search space to identify efficient routes within the mentioned constraints. The following section describes the PSO-based global path planning.

PSO and Path Representation

PSO simulates social behaviour patterns observed in nature, such as birds flocking, to search for optimal solutions in a multidimensional space. Each particle in the swarm represents a potential path $'P'$, defined by a sequence of route points. (x_j, y_j). This route points chart the AUV's proposed route from its starting point to its destination.

Velocity and Position Update Rules

The core mechanism driving PSO's search capability lies in the iterative update of each particle's velocity and position, governed by the following rules:

Velocity Update

The velocity of a particle is adjusted based on its previous velocity, the distance from its current position to its personal best position, and the distance to the swarm's global best position. Mathematically, the velocity update for the $'i'$ particle is given by EQU (9).

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot \text{rand}() \cdot \left(\text{pbest}_i - x_i(t) \right) + c_2 \cdot \text{Rand}() \cdot (\text{gbest} - x_i(t)) \tag{9}$$

Position Update

The position of a particle is updated by adding its velocity to its current position, facilitating the exploration of new potential paths, EQU (10).

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{10}$$

Where w is the inertia weight, controlling the impact of the previous velocity on the current velocity. c_1 and c_2 are the cognitive and social coefficients, respectively, guiding the particle towards its personal best and the global best positions; $rand()$ and $Rand()$ are random functions generating values between 0 and 1, *pbest* $_i$ is the personal best position of the i th particle, and *gbest* is the global best position found by the swarm.

Adapting the PSO for underwater PF involves modifying its parameters, namely, the inertia weight (w), cognitive co-efficient (c_1), and social co-efficient (c_2)-to maintain the balance between exploration (searching new areas) and exploitation (focusing on promising areas):

Inertia Weight ($\boldsymbol{w}$)

Adjusting w helps control the trade-off between global and local search abilities. A higher w promotes exploration, while a lower $'w'$ enhances exploitation.

Cognitive and Social Co-efficients ($\boldsymbol{c_1}$ and $\boldsymbol{c_2}$)

Fine-tuning c_1 and c_2 dictates the tendency of particles to navigate towards their personal and global best, respectively. Balancing these coefficients is crucial for effective search behaviour in the context of underwater problems and mission objectives.

Static Problem Avoidance in PSO-Based Global Path Planning

The UE is conceptualized as a 2-D grid or a continuum, where obstacles are precisely located based on their coordinates. Each obstacle denoted as Obs_i, is considered by its location (x_{obs_i}, y_{obs_i}) and potentially its size or radius r_{obs_i} to represent its physical extent accurately. Within the PSO framework, each particle symbolizes a potential navigational path P, constructed from a sequence of co-ordinates (x_j, y_j). These route points chart the course the AUV is to navigate from its starting position towards its goal. The dynamics of each particle, including its position and velocity, evolve over iterations according to PSO's optimization rules, guiding the swarm towards optimal paths.

The essence of PF, $F(P)$, is quantified by a fitness function that integrates a critical obstacle avoidance component. This component employs a repulsive potential field concept around obstacles to penalize paths that either intersect with obstacles or traverse too closely to them. Mathematically, the obstacle avoidance aspect of the fitness function can be articulated as EQU (11)

$$\mathbb{O}(P) = \lambda_{obs} \sum_{i=1}^{n_{obss}} \sum_{j=1}^{n_{\text{points}}} \frac{1}{d\left((x_j,y_j),Obs_i\right)^2 + \epsilon} \tag{11}$$

where:

- n_{obs} signifies the count of problems within the UE.
- n_{points} represents the number of route points defining path $'P'$.
- $d\left((x_j, y_j), Obs_i\right)$ computes the distance from the $'j'$ route points in $'P'$ to the $'i'$ problem.
- λ_{obs} is a co-efficient weighing the importance of obstacle avoidance within the total fitness function.
- $'\epsilon'$ is a minor constant to ensure the denominator never zeroes out, maintaining computational stability.

This $\mathbb{O}(P)$ effectively institutes a repulsion from problems, deterring the selection of unsafe paths. The overarching fitness function that a particle (path) P must minimize becomes EQU (12).

$$F(P) = \lambda_1 L(P) + \lambda_2 E(P) + \lambda_3 \mathbb{O}(P) \ + \lambda_4 A(P) \tag{12}$$

Here, the PSO aims to minimize $F(P)$, steering the optimization towards identifying optimal paths.

Enhanced Route Adaptability in PSO-Based Global Path Planning

The adaptability of a route to environmental changes can be significantly enhanced by dynamically adjusting the weighting factor $'\delta'$ of the adaptability term $A(P)$ in the fitness function. This adjustment is based on the level of environmental variation or the changes in the AUV's operational context. The modified adaptability term can be expressed as EQU (13).

$$A(P, t) = \delta(t) \cdot \sum_{i=1}^{n_{\text{points}}} \ a(P_i, E(t)) \tag{13}$$

where:

- $a(P_i, E(t))$ assesses the adaptability of the $'i'$ route points in path P concerning the current environmental conditions $E(t)$ at time t.
- $\delta(t)$ is a time-varying weighting factor that dynamically adjusts the importance of environmental adaptability based on real-time feedback.

The dynamic adjustment of $\delta(t)$ is modelled based on environmental unpredictability. For instance, if environmental volatility is quantified by a metric $V(t)$, $\delta(t)$ is represented as follows: EQU (14)

$$\delta(t) = \frac{1}{1+e^{-k(V(t)-V_{\text{thiresh}})}} \tag{14}$$

where:

- k is a scaling constant that determines the sensitivity of $\delta(t)$ to changes in $V(t)$.
- V_{thresh} is a threshold value for environmental unpredictability beyond which the importance of adaptability significantly increases.

Implementation in PSO

Integrating this dynamic weighting mechanism into the PSO algorithm involves recalculating $\delta(t)$ at each iteration based on current environmental data. This ensures that the fitness function, and consequently the optimization process, dynamically prioritizes path adaptability in response to changing environmental conditions:

Real-time Environmental Feedback

Continuously monitor environmental conditions $E(t)$ and calculate the variability metric $V(t)$ to adjust $\delta(t)$ accordingly.

Fitness Function Update

Given the dynamic adjustment of the weighting factor $\delta(t)$ based on environmental variability or changes, the updated composite objective (fitness) function $F(P, t)$ at time $'t'$ is expressed as EQU (15).

$$F(P, t) = \lambda_1 \cdot L(P) + \lambda_2 \cdot E(P) + \lambda_3 \cdot \mathbb{O}(P) + \delta(t) \cdot A(P, E(t)) \tag{15}$$

This updated fitness function $F(P, t)$ enables the PSO algorithm to dynamically prioritize paths that are not only efficient and safe but also highly adaptable to the current underwater environmental conditions. The complete process of the PF is presented in the following algorithm,

Algorithm 1 for PSO for AUV Global PF

Inputs:

- Grid Environment (G)
- Number of Particles (N)
- Maximum Iterations (Max_Iter)
- Inertia Weight (w)
- Cognitive Co-efficient (c_1)
- Social Co-efficient (c_2)
- Start and Goal Positions

1. **Initialize the Grid Environment**
   - Construct the grid G based on input environmental data, marking obstacles and navigable waters.
2. **Initialize Particles**

- For each particle $'i'$ in the swarm $(i = 1,2, \ldots, N)$:
- Randomly initialize the position x_i representing a potential path in $'G'$ from the start to the goal position.
  - Initialize velocity v_i randomly.
  - Set *pbest* $_i$ to its initial position.
- Initialize *gbest* based on the initial calculations of $F(P)$.

3. **Evaluate Fitness**
   - For each particle, compute $F(P)$ considering the path P represented by x_i, integrating path length ($L(P)$), EC ($E(P)$), problem avoidance ($\mathbb{O}(P)$), and adaptability ($A(P)$).
4. **Update Personal and Global Bests**
   - For Each particle, if $F(P)$ at x_i is better than $F(P)$ at pbest $_i$, update pbest $_i$ to x_i.
   - Update *gbest* if any *pbest* $_i$ offers a better fitness than the current g best.
5. **Velocity and Position Update**
   - **For Each** particle:
     - Update v_i using the formula considering w, c_1, c_2, pbest $_i$, and $gbest$.
     - Update x_i by adding v_i to the current position, ensuring the new position is valid within G and avoids obstacles.
6. **Iterate and Convergence Check**
   - Repeat Step 3 to Step 5 for *max_iter* iterations *gbest* changes minimally between iterations, indicating convergence.
7. **Output**
   - The optimal path for the AUV is represented by the *gbest* particle's position [21].

Local PF using ACO

ACO is a probabilistic technique that is designed by the inspiration of the foraging behaviour of ants. The ACO utilizes the concept of pheromones to guide the search for optimal solutions. The essential ACO components and functions are:

Initialization

The algorithm begins by initializing all paths with a small volume of pheromone to ensure that every path can be explored. The initial pheromone level on each path, or edge, in the graph, is typically set to a constant value, τ_0.

Solution Construction

Each ant in the colony constructs a solution by traversing the graph from the starting point to the destination. The choice of the next node to visit is probabilistic, heavily influenced by the amount of pheromone on the connecting edges and the heuristic desirability of the move. The probability of moving from node i to node j for ant k is given by EQU (16)

$$p_{ij}^{k} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{l \in allowed_k} \left(\tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}\right)} \tag{16}$$

Where τ_{ij} is the pheromone concentration on the edge from i to j, η_{ij} is the heuristic value associated with the edge from i to j, often related to the inverse of the distance between the nodes, α and β are parameters that control the influence of τ_{ij} and η_{ij}, respectively and $allowed_k$ is the set of nodes available for the next move for ant k.

Pheromone Update

After all ants have constructed their solutions, the pheromone levels on the paths are updated to reflect the newly acquired knowledge. This involves two main steps:

Pheromone Evaporation

To prevent the algorithm from converging too early on a suboptimal path, a certain amount of pheromone evaporates from all paths. This is modelled by EQU (17).

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} \tag{17}$$

where ρ is the pheromone evaporation rate, a parameter between 0 and 1.

Pheromone Deposition

Ants deposit pheromones on the paths they traverse based on the quality of their solution. The amount of pheromone deposited, $\Delta\tau_{ij}^{k}$, often depends on the inverse of the path length or cost found by ant k, encouraging the selection of shorter or more efficient paths in future iterations.

Convergence Check

The algorithm repeats the solution construction process and pheromone update until a stopping criterion is met, such as a maximum number of iterations or a solution quality threshold.

Enhanced Cost Function for ACO-Based Local Path Planning

For local path planning using ACO, the cost function is modified to include (i) Safety Distance Compliance ($S(P)$) and (ii) Dynamic Obstacle Avoidance ($D(P)$).

Safety Distance Compliance ($\boldsymbol{S}(\boldsymbol{P})$)

This component evaluates how well a path adheres to maintaining a predefined safe distance from obstacles in a grid-based UE.

Given that the environment is represented as a 2D matrix, where obstacles are marked with an 0 (indicating no-go zones) and navigable water with a 1 (safe zones), the safety distance compliance can be recalibrated as follows: EQU (18).

$$S(P)=\sum_{i=1}^{n_{\text{points}}} \max\left(0, d_{\text{safe}} - \min_{\forall Obs\in N(P_i)} d(P_i, \text{Obs})\right) \tag{18}$$

- d_{safe} is the predefined safety distance, translated into the number of grid cells that form the minimum buffer between the AUV and any obstacle.
- $d(P_i, Obs)$ Now, it measures the shortest grid-based distance (in terms of cells) from the i th waypoint on the path P to the nearest problem cell Obs, respecting the grid layout.
- $N(P_i)$ represents the set of neighbouring cells around the point P_i considered in the safety distance calculation, adjusted according to d_{safe} .
- This formula penalizes paths where any part of P comes within d_{safe} Grid cells of obstacles, thereby promoting routes that maintain the integrity and safety of the AUV.

In this grid context, the Euclidean distance is adapted to account for the discrete nature of the grid, potentially using the Manhattan distance metric for grid navigation.

Dynamic Obstacle Avoidance (D(P))

$D(P)$ consider both the ability to reroute around new problems dynamically and the necessity of maintaining a safe buffer zone, as defined by $S(P)$, around those obstacles. The dynamic problem avoidance capability is given by EQU (19).

$$D(P)=\sum_{i=1}^{n_{\text{points}}-1} \delta\left(P_i, P_{i+1}, O_{grid}, S(P)\right) \tag{19}$$

Here, $\delta\left(P_i, P_{i+1}, O_{\text{grid}}, S(P)\right)$ is a function assessing the adjustability of the path segment from P_i to P_{i+1}, taking into account both the presence of dynamic obstacles within the grid $\left(O_{\text{grid}}\right)$

Overall Cost Function Formulation

The overall cost function, integrating these considerations with traditional path optimization criteria, EQU (20).

$$\text{Cost}(P)=\lambda_1\cdot L(P)+\lambda_2\cdot E(P)+\lambda_3\cdot \mathbb{O}(P)+\lambda_4\cdot A(P)+\lambda_5\cdot S(P)+\lambda_6\cdot D(P) \tag{20}$$

- $L(P), E(P), \mathbb{O}(P)$, and $A(P)$ represent path length, EC, problem avoidance, and environmental adaptability, respectively.
- λ_1 to λ_6 are the weighting coefficients for each component to balance efficiency, safety, and adaptability.

The entire process of finding local paths using ACO is presented in algorithm 2.

Algorithm 2: ACO-Based Local Path Planning Algorithm for AUVs

Inputs:

- Grid Environment (G)
- Number of Ants (N)
- Maximum Iterations (Max_Iter)
- Pheromone Evaporation Rate (ρ)
- Influence Parameters (α, β)
- Initial Pheromone Level (τ_0)
- Safety Distance (d_{safe})
- Start and Goal Positions

Output Specification:

- The optimal path, P^*, from S to T that considers safety distance compliance and dynamic obstacle avoidance.

Algorithm Steps:

1 **Initialization:**
- Initialize the grid G with the current state of UE.
- Set all paths in G with an initial pheromone level τ_0.
- Place N ants at the starting location S.

2 **Path Construction:**
- **For Each** ant:
  - Construct a path from S to T by selecting moves based on transition probabilities influenced by pheromone levels and heuristic values (distance to T, safety compliance).
  - Calculate transition probability using:
- Ensure moves comply with d_{safe}, avoiding paths that breach the safety distance from obstacles.

3 **Local Pheromone Update:**
- Optionally, local pheromone updates should be applied after each move or path construction to encourage exploration.

4 **Global Pheromone Update:**
- After all ants complete their paths, update pheromones globally:
  - Apply evaporation: $\tau_{ij} = (1 - \rho) \cdot \tau_{ij}$.
  - Deposit pheromones on paths traversed by ants, with $\Delta\tau_{ij}^k$ proportional to path quality (inversely related to path cost considering $S(P)$ and (P)).

5 **Cost Function Evaluation:**
- Evaluate the cost of each path using:
  - $\text{Cost}(P) = \lambda_1 \cdot L(P) + \lambda_2 \cdot E(P) + \lambda_3 \cdot \mathbb{O}(P) + \lambda_4 \cdot A(P) + \lambda_5 \cdot S(P) + \lambda_6.$
  - $D(P)$
- Select the path with the lowest cost as the current best solution.

6 **Convergence Check:**
- Determine if a stopping criterion is met (*e.g.*, no significant improvement in path cost, maximum number of iterations reached).
- If not met, return to Step 2 with updated pheromone levels.

7 **Output the Optimal Path:**
- Return to the optimal path P^* as the solution, ensuring it adheres to safety and efficiency criteria while adapting to dynamic changes in the underwater environment.

IV. EXPERIMENTAL ANALYSIS

The simulation platform for our AUV path planning experiments was built on a system powered by an AMD Ryzen 9 3950X 16-Core Processor @ 3.5 GHz, running a Linux-based operating system. The algorithm was evaluated using Python, leveraging libraries such as NumPy for numerical computations and Matplotlib for graphical representations. The simulated underwater domain covered an area of 100×100 *km*, organized into a grid where each cell spanned 10×10 *km*, resulting in a 10×10 grid, as shown in **Fig 1**.

Adjustments to the simulation parameters included:

- **AUV's Maximum Acceleration:** Adjusted to 3 m/s^2 for the AUV's propulsion system capabilities to model varying UE.
- **Grid Resolution:** Set at N = 1 *km*.
- **Safety Margin:** Established at 2 *km*.

- **AUV's Initial and Target Coordinates:** Selected as (10, 20) and (100, 90) within the grid.

This setup tests the algorithm across the maximum possible distance within the simulated domain, presenting various navigation challenges from shallow to deep-water scenarios.

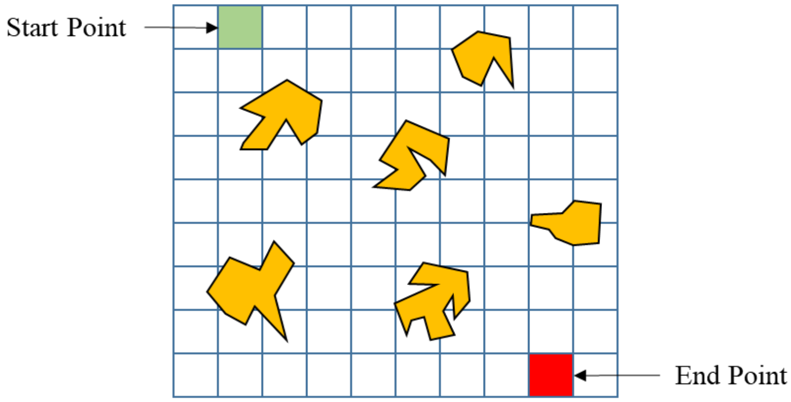

Fig 1. Simulation Environment Grid.

Three models, PSO, ACO, and GA, are compared against the PAO+ACO model. The compared model's route from the start to the endpoint is presented in **Fig 2** (a) to (d). the further analysis of the compared model's performance in terms of inflexion count, average route length, and time to reach the target accuracy are analyzed in the following sections:

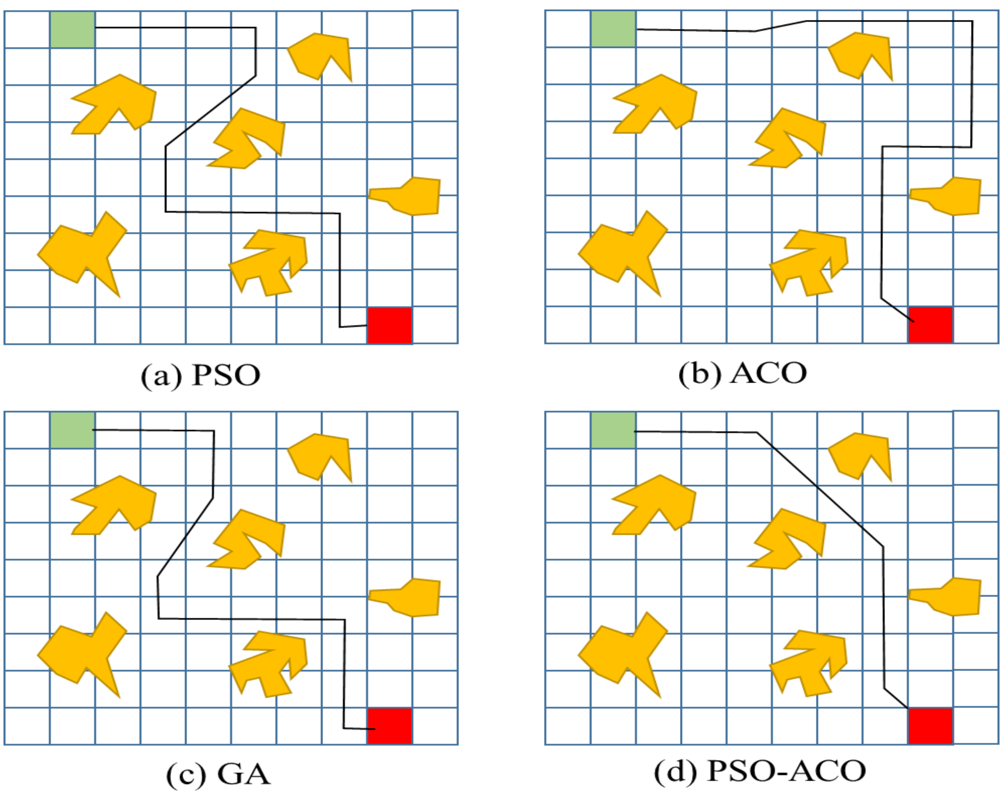

Fig 2. PF for Different Models.

The simulation results, as shown in **Fig 3**, show the effectiveness of different algorithms in terms of the following metrics:

Inflection Count:

A lower inflection count indicates a smoother path with fewer sharp turns that are best for energy efficiency and operational feasibility. The PSO+ACO hybrid algorithm outperforms PSO, ACO, and GA by reducing the inflection count to 5.

Route Lengths

The PSO+ACO hybrid algorithm performed better by achieving the shortest long route (17.69 *km*) and the shortest route (16.13 *km*).

Average Route Length

The average route length measures the overall path efficiency across simulations (Fig 3). The PSO+ACO hybrid algorithm obtained the lowest average route length (16.91 *km*) compared to PSO (24.20 *km*), ACO (23.42 *km*), and GA (22.52 *km*).

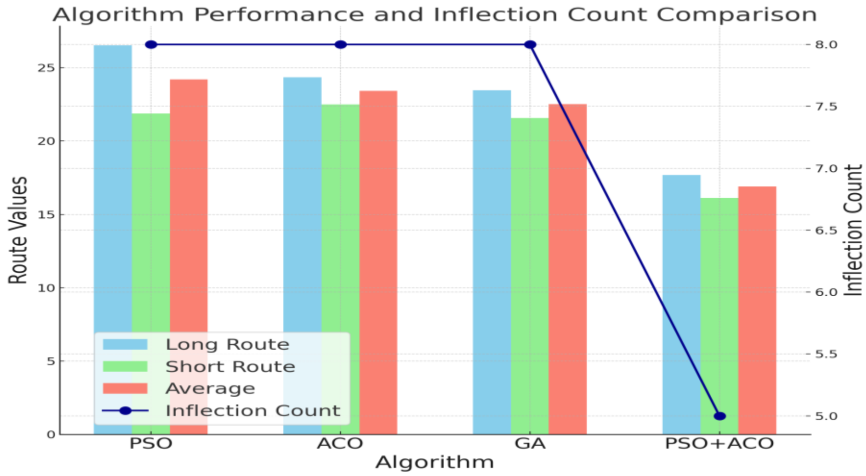

Fig 3. Average Route Length and Inflection Count.

The following simulation results, as shown in **Fig 4**, analyse the time efficiency and accuracy against other models. The proposed PSO+ACO hybrid algorithm showed better performance regarding the speed and precision of pathfinding. The model had a completion time of 6.43 *Sec*. and had effectively surpassed its counterparts. Also, the hybrid algorithm showed an accuracy rate of 93.5%, a way better performance than the PSO, ACO, and GA algorithms, which achieved 66.2%, 63.4%, and 65.7%, respectively. Among the later performing models, the ACO showed a completion time of 12.53 *Sec*. next to the proposed model.

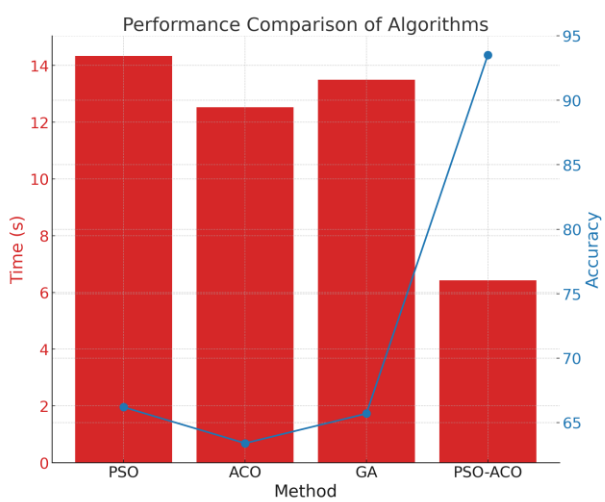

Fig 4. Accuracy and Target Reach Time.

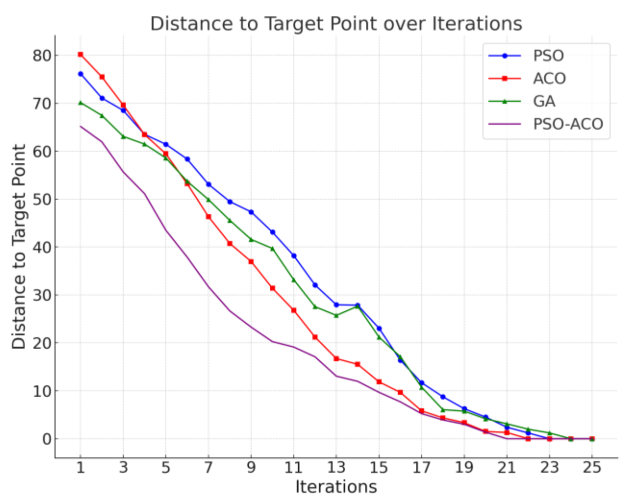

Fig 5. Target vs Iterations

The graph in **Fig 5** represents the distance to the target analysis over iterations for the compared models, such as PSO, ACO, GA, and the PSO+ACO hybrid algorithms. The proposed model showed a significant reduction in distance to the target by reaching it much ahead of other models. The model achieved reaching the target in fewer iterations than both stand-alone models. Even though it possessed a higher overall value at the outset, the ACO approach accomplished a level of accuracy similar to that of the recommended model. Closure emerges at the same distance ratio for both the PSO+ACO models over the number of iterations. Throughout the framework of this study, the value of the proposed algorithm's performance in successfully AUNS the UE and, as a result, driving the AUV to reach its location was highlighted.

V. CONCLUSION AND FUTURE WORK

The main objective of the study was to explore the idea of employing Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) in order to Path Finding (PF) for Autonomous Underwater Vehicles (AUV). Using the techniques of static problem avoidance and enhanced route adaptability, the approach utilized PSO for global route planning and ACO for local path planning. Both of these approaches were applied during the entire method. Simulations were performed by employing the proposed approach in a framework that was represented as a grid. For the aim of comparing performance comparison, the computer simulation used PSO, ACO, and GA models. The analysis was conducted on how well the models performed regarding the total time they saved, the accuracy they attained, and the distance they decreased to the objective. The results indicate that the built PSO+ACO hybrid model performed better throughout all regions based on all guidelines. These results reinforce the practical application of the framework in the route planning method for AUV in Underwater Environments (UE).

It is predicted that in further work, the prototype will be evaluated in a UE that is comparable to the real world, and it will additionally incorporate additional variables like EC and navigational safety.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding agency is associated with this research.

Competing Interests

There are no competing interests.

References

[1]. S. Watson, D. A. Duecker, and K. Groves, “Localisation of Unmanned Underwater Vehicles (UUVs) in Complex and Confined Environments: A Review,” Sensors, vol. 20, no. 21, p. 6203, Oct. 2020, doi: 10.3390/s20216203.

[2]. Sharma, R., & Sungheetha, A. (2023). Revolutionizing Underwater Exploration of Autonomous Underwater Vehicles (AUVs) and Seabed Image Processing Techniques. arXiv e-prints, arXiv-2402.

[3]. R. Kot, “Review of Collision Avoidance and Path Planning Algorithms Used in Autonomous Underwater Vehicles,” Electronics, vol. 11, no. 15, p. 2301, Jul. 2022, doi: 10.3390/electronics11152301.

[4]. L. Zhao and Y. Bai, “Unlocking the Ocean 6G: A Review of Path-Planning Techniques for Maritime Data Harvesting Assisted by Autonomous Marine Vehicles,” Journal of Marine Science and Engineering, vol. 12, no. 1, p. 126, Jan. 2024, doi: 10.3390/jmse12010126.

[5]. X. Sun, G. Wang, Y. Fan, D. Mu, and B. Qiu, “A Formation Autonomous Navigation System for Unmanned Surface Vehicles With Distributed Control Strategy,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 5, pp. 2834–2845, May 2021, doi: 10.1109/tits.2020.2976567.

[6]. M. Juříček, R. Parák, and J. Kůdela, “Evolutionary Computation Techniques for Path Planning Problems in Industrial Robotics: A State-of-the-Art Review,” Computation, vol. 11, no. 12, p. 245, Dec. 2023, doi: 10.3390/computation11120245.

[7]. Y. Sun, X. Luo, X. Ran, and G. Zhang, “A 2D Optimal Path Planning Algorithm for Autonomous Underwater Vehicle Driving in Unknown Underwater Canyons,” Journal of Marine Science and Engineering, vol. 9, no. 3, p. 252, Feb. 2021, doi: 10.3390/jmse9030252.

[8]. E. Krell, S. A. King, and L. R. Garcia Carrillo, “Autonomous Surface Vehicle energy-efficient and reward-based path planning using Particle Swarm Optimization and Visibility Graphs,” Applied Ocean Research, vol. 122, p. 103125, May 2022, doi: 10.1016/j.apor.2022.103125.

[9]. “Optimal Data Collection Path Finding for AUV in Internet of Underwater Things,” International Journal of Intelligent Engineering and Systems, vol. 17, no. 1, pp. 827–844, Feb. 2024, doi: 10.22266/ijies2024.0229.69.

[10]. Y. Cui, P. Zhu, G. Lei, P. Chen, and G. Yang, “Energy-Efficient Multiple Autonomous Underwater Vehicle Path Planning Scheme in Underwater Sensor Networks,” Electronics, vol. 12, no. 15, p. 3321, Aug. 2023, doi: 10.3390/electronics12153321.

[11]. Y. Ma, Z. Mao, T. Wang, J. Qin, W. Ding, and X. Meng, “Obstacle avoidance path planning of unmanned submarine vehicle in ocean current environment based on improved firework-ant colony algorithm,” Computers & Electrical Engineering, vol. 87, p. 106773, Oct. 2020, doi: 10.1016/j.compeleceng.2020.106773.

[12]. S. Guo, M. Chen, and W. Pang, “Path Planning for Autonomous Underwater Vehicles Based on an Improved Artificial Jellyfish Search Algorithm in Multi-Obstacle Ocean Current Environment,” IEEE Access, vol. 11, pp. 31010–31023, 2023, doi: 10.1109/access.2023.3257025.

[13]. H. Gopinath, V. Indu, and M. M. Dharmana, “Autonomous underwater inspection robot under disturbances,” 2017 International Conference on Circuit , Power and Computing Technologies (ICCPCT), Apr. 2017, doi: 10.1109/iccpct.2017.8074222.

[14]. K. Sreekala, N. N. Raj, S. Gupta, G. Anitha, A. K. Nanda, and A. Chaturvedi, “Deep convolutional neural network with Kalman filter based objected tracking and detection in underwater communications,” Wireless Networks, Mar. 2023, doi: 10.1007/s11276-023-03290-z.

[15]. S. Anil and R. Remya, “A hybrid method based on genetic algorithm, self-organised feature map, and support vector machine for better network anomaly detection,” 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Jul. 2013,

doi: 10.1109/icccnt.2013.6726604.

[16]. S. Shivkumar, J. Amudha, and A. A. Nippun Kumaar, “Federated deep reinforcement learning for mobile robot navigation,” Journal of Intelligent & Fuzzy Systems, pp. 1–16, May 2024, doi: 10.3233/jifs-219428.

[17]. A. D. Wanekar, N. Praveen Babu Mannam, and P. Rajalakshmi, “Novel Approach to Underwater Object Detection Using Sonar Sensors for Autonomous Underwater Vehicles (AUVs),” 2023 International Conference on Sustainable Technology and Engineering (i-COSTE), Dec. 2023, doi: 10.1109/i-coste60462.2023.10500768.

[18]. E E. Vidal, N. Palomeras, and M. Carreras, “Online 3D Underwater Exploration and Coverage,” 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), Nov. 2018, doi: 10.1109/auv.2018.8729736.

[19]. G. Indiveri, “Geotechnical Surveys with Cooperative Autonomous Marine Vehicles: the EC WiMUST project,” 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), Nov. 2018, doi: 10.1109/auv.2018.8729794.

[20]. S. Song and S.-C. Yu, “Underwater marking AUV using paraffin wax,” 2016 IEEE/OES Autonomous Underwater Vehicles (AUV), Nov. 2016, doi: 10.1109/auv.2016.7778716.

[21]. V. L. Narla, R. Kachhoria, M. Arun, D. Vijendra Babu, and B. M. Jos, “IoT based energy efficient multipath power control for underwater sensor network,” International Journal of System Assurance Engineering and Management, Apr. 2022, doi: 10.1007/s13198-021-01560-7