Research on 3D global path planning technology for UUV based on fusion algorithm

Zhiyuan Hu¹, Zheng Wang¹ and Yang Yin¹*

¹ School of Electrical Engineering, Naval University of Engineering, Wuhan, Hubei, 430033, China
*Corresponding author’s e-mail: ssyymmjj@263.net, reeyan@163.com

Abstract. Aiming at the global path planning problem of unmanned underwater vehicle (UUV) in three-dimensional environment, a fusion algorithm named PACO based on particle swarm optimization (PSO) and ant colony optimization (ACO) is proposed. In order to meet the needs of three-dimensional path planning, the grid method is used to model the actual marine environmental data. Besides, PSO and ACO are optimized respectively. Taking the path length as the measurement index, the simulation is realized by using MATLAB. The simulation results show that the initial path search ability of the fusion algorithm is significantly improved, and the effectiveness of the algorithm is verified.

1. Introduction
With the development of technology, UUVs are widely used in the marine field. UUVs have the features of small size, strong manoeuvre ability, high degree of intelligence, good stealth performance and low operational risk [1]. Path planning can ensure UUVs to find an optimal path from the starting point to the target point in the work space with obstacles. According to the awareness of the environment, it can be divided into local path planning and global path planning [2]. Among them, global path planning is suitable for the known marine environment. As one of the control technologies, it provides an important support for the safety and reliability of UUVs [3].

Grid model is the most effective and widely used method in 3D path planning [4]. Based on grid model, many algorithms are generated, such as A* algorithm [5], simulated annealing algorithm, genetic algorithm, particle swarm optimization (PSO) [6] and ant colony optimization (ACO) [7]. With the development of bionic algorithm, many new algorithms were created for path planning, such as fish swarm algorithm, wolf swarm algorithm [8] and glow-worm swarm optimization algorithm.

Document [10] combined ACO with octree mode. The search space was optimized. The convergence speed was improved at the same time. Document [11] combined genetic algorithm and simulated annealing algorithm. A better path can be found within the same number of iteration. Document [12] combined genetic algorithm and ACO. At the beginning of the algorithm, genetic algorithm was used to optimize the initial pheromone. At the end of the algorithm, ACO was used to search for an optimal path. In document [13], alarm pheromone was introduced into ACO. Algorithm named alarm pheromone-assisted ant colony system (AP-ACS) was proposed. The alarm pheromone alerts the ants to infeasible areas, thus to save invalid efforts and improves the search efficiency.

In this paper, the grid method is used to model. Considering the shortcomings of ACO, PSO and ACO are optimized and integrated. Simulation results show that the proposed fusion algorithm named PACO can achieve fast global path planning.
2. Marine environment acquisition and process

2.1. Seabed terrain dataset acquisition

Obtaining real seabed terrain data can ensure the effectiveness of global path planning. At present, there are many open source databases can be used. Among them, the General Bathymetric Chart of the Oceans (GEBCO) contains 290 million seawater depth detection data, which can meet the need of path planning research.

The map plug-in provided by GEBCO is convenient for users to operate. The sampling matrix can be obtained by selecting the corresponding sea area on the map plug-in. As showing in Figure 1, the red window indicates the selected sea area. After process by MATLAB, the 3D seafloor environment map is drawn in Figure 2.

2.2. Grid method of environment modeling

The grid model is suitable for 3D marine environment modelling. Along the main navigation direction of UUV (assumed to be $X$ direction), the marine environment between the starting point and the target point is equidistant segmented, showed in Figure 3. $N$ planes $\Pi_i$ ($i = 1, 2, 3, ..., N$) perpendicular to $X$ axis and parallel to $YOZ$ plane are formed. The starting point is located in the first plane $\Pi_1$, and the target point is located in the last plane $\Pi_N$. For any plane $\Pi_i$, it is divided equally along $Y$ axis and $Z$ axis, so that the plane is divided into $M \times L$ grids.

Through the grid method, the 3D environment can be discretized into several 2D grid planes. Then, the path planning problem can be transformed into a hierarchical search method to find feasible path points in the 2D grid plane. Finally, the path points are connected to form the final path.

Figure 1. The plug-in provided by GEBCO.
Figure 2. 3D seafloor environment map.

Figure 3. Sketch map of the grid method.
Figure 4. Limitations of the search grid.
2.3. Limitations of search grid
The complexity of the search process will be increased, if all grid nodes are traversed in the process of route points finding. Besides, considering the actual manoeuvre ability of UUV, some grid nodes lose practical significance and should be regarded as infeasible points.

Limitations of search grid can be divided into vertical distance limitation $y_{\text{max}}$ and horizontal distance limitation $z_{\text{max}}$. As showed in Figure 4, when the path point $P_i$ located in the plane $\Pi_i$ searches the next path point $P_{i+1}$ in the plane $\Pi_{i+1}$, it only needs to search the grid nodes in the search frame composed by $y_{\text{max}}$ and $z_{\text{max}}$.

3. Design of fusion algorithm: PACO

3.1. Theoretical basis of fusion algorithm
ACO simulates the foraging behaviour of ants in nature, with the characteristics of strong adaptability, robustness and distributed computing [12]. However, due to the uniform distribution of initial pheromone, the initial search stage of ACO will be blindness. The convergence speed will be reduced at the same time. PSO assumes that each particle represents a possible solution, with the characteristics of simple operation and fast convergence. Therefore, we can consider the combination of PSO and ACO. A fusion algorithm named PACO is proposed.

At the beginning of PACO, PSO is used to find several feasible paths. The initial pheromone distribution of the corresponding path points is increased. Then ACO is used for global path planning. The optimized initial pheromone can reduce the blindness of ACO in initial search stage, thus to improve the convergence speed.

3.2. Flow of fusion algorithm
According to the theory of fusion algorithm, the specific flow is as follows. The corresponding flow chart of PACO is shown in Figure 5.
**Step 1.** The database is downloading from GEBCO. And then, 3D seafloor environment map is drawn.

**Step 2.** Set the information of starting point and the target point. The number of planes $N$, the number of dividing grids $M$ and $L$ is set at the same time.

**Step 3.** Set the relevant parameters of PSO. The position and velocity information are initialized.

**Step 4.** PSO is used for path planning. The fitness value is used to update the personal and the global solutions. When the number of iterations is satisfied, several feasible paths are obtained.

**Step 5.** The optimized initial pheromone is formed by increasing the initial pheromone of feasible path points.

**Step 6.** Set the relevant parameters of ACO. The optimized initial pheromone is transmitted to ACO.

**Step 7.** The fitness values of feasible paths are calculated. The local pheromone and global pheromone are updated.

**Step 8.** Determine whether the maximum number of iterations of ACO is reached. When the number of iterations is satisfied, the simulation results are saved and visualized.

4. Design of PSO

4.1. Design of traditional PSO

Each particle in PSO has position and velocity, which are updated by personal and global optimal solutions. Finally, particles will gather near the optimal path.

After the grid model, each particle can be regarded as a feasible path. Due to the hierarchical search method, the coordinate information of the main navigation direction ($X$ axis) can be ignored. Therefore, the position information only contains the $Y$ axis and $Z$ axis coordinate information. Similarly, the velocity information only contains the $Y$ axis and $Z$ axis velocity information.

The starting point and the target point are determined. So the information corresponding to the points of each particle doesn’t need to be updated. The position information should be fixed, and the speed information is constant to zero.

Assuming that, the starting point coordinate is $({y_{\text{start}}}, {z_{\text{start}}})$, and the target point coordinate is $({y_{\text{end}}}, {z_{\text{end}}})$. When the number of iteration is $p$, particle $m$ has the matrices of position and velocity shown in equation (1) and (2). We can find that, in the iteration $p$, the position coordinate of the particle $j$ in the plane $\Pi_i$ is $({y_{\text{start}}}, {z_{\text{start}}}, {y_{\text{end}}}, {z_{\text{end}}})$. The velocity and position of particles are updated based on the information from the previous iteration. Equations are as follows.

$$X^p = \begin{bmatrix} X^p_1 \\ X^p_2 \\ \vdots \\ X^p_m \\ \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} y_{\text{start}} \\ z_{\text{start}} \\ y_{1,2} \\ z_{1,2} \\ \vdots \\ y_{i,2} \\ z_{i,2} \\ \vdots \\ y_{1,N-1} \\ z_{1,N-1} \\ y_{\text{end}} \\ z_{\text{end}} \end{bmatrix} \\ \begin{bmatrix} y_{\text{start}} \\ z_{\text{start}} \\ y_{2,2} \\ z_{2,2} \\ \vdots \\ y_{2,N-1} \\ z_{2,N-1} \\ \vdots \\ y_{\text{end}} \\ z_{\text{end}} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} y_{\text{start}} \\ z_{\text{start}} \\ y_{m,2} \\ z_{m,2} \\ \vdots \\ y_{m,N-1} \\ z_{m,N-1} \\ \vdots \\ y_{\text{end}} \\ z_{\text{end}} \end{bmatrix} \end{bmatrix}$$

$$V^p = \begin{bmatrix} V^p_1 \\ V^p_2 \\ \vdots \\ V^p_m \end{bmatrix} = \begin{bmatrix} 0 & 0 & V^p_{1,2} & \cdots & V^p_{1,N-1} & 0 & 0 \\ 0 & 0 & V^p_{2,2} & \cdots & V^p_{2,N-1} & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & V^p_{m,2} & \cdots & V^p_{m,N-1} & 0 & 0 \end{bmatrix}$$

The velocity and position of particles are updated based on the information from the previous iteration. Equations are as follows.

$$V_{p+1} = wV^p + c_1r_1(P^p_p - X^p_p) + c_2r_2(G^p_p - X^p_p)$$
\[ X_{i}^{p+1} = X_{i}^{p} + V_{i}^{p+1} \] (4)

where, \( w \) is the inertia factor. The global search ability will be improved if the value of \( w \) is increased. \( c_{1} \) and \( c_{2} \) are learning acceleration factors. \( r_{1} \) and \( r_{2} \) are random numbers between 0 and 1. \( P_{p}^{i} \) is the personal optimal solution of particle \( i \), and \( G_{p}^{i} \) is the global optimal solution when iteration number is \( p \).

After the velocity and position of particles are updated, the personal and global optimal solutions are updated according to the fitness function \( F(x) \).

\[
P_{i}^{p+1} = \begin{cases} P_{i}^{p} & \text{if } F(P_{i}^{p}) \text{ is better than } F(X_{i}^{p}) \\ X_{i}^{p+1} & \text{if } F(X_{i}^{p}) \text{ is better than } F(P_{i}^{p}) \end{cases}
\] (5)

\[
G_{p}^{p+1} = \begin{cases} G_{p}^{i} & \text{if } F(G_{p}^{i}) \text{ is better than } F(P_{i}^{p+1}) \\ P_{i}^{p+1} & \text{if } F(P_{i}^{p+1}) \text{ is better than } F(G_{p}^{i}) \end{cases}
\] (6)

4.2. Optimization of PSO
Considering the manoeuvre ability of UUV and the actual marine environment, the maximum velocity of particles is set to \( V_{\text{max}} \). The relationship between \( V_{\text{max}}, y_{\text{max}} \) and \( z_{\text{max}} \) is as follows.

\[ V_{\text{max}} \leq \min \left( y_{\text{max}}, z_{\text{max}} \right) \] (7)

By the optimization of PSO, the search of infeasible path points can be reduced. The effectiveness of the search path can improve the convergence speed of PSO.

5. Design of ACO
5.1. Design of traditional ACO
ACO create a group of artificial ants. Ants will release pheromones along the path, and their concentrations will volatilize with time. The change of pheromone concentration guides ants to find the optimal path. With the increase of the number of iterations, the pheromone on the optimized path is constantly enhanced, and the artificial ants will gather near the path to realize the global path planning [14]. The design of ACO in 3D environment is as follows.

5.1.1. Design of heuristic. The convergence, stability and optimization of ACO are largely depending on the heuristic value. The heuristic value \( Q_{t+1} \) from node \( t \) to node \( t+1 \) is set as follows.

\[
Q_{t+1} = \frac{w_{1}}{D_{t+1}} \cdot \frac{w_{2}}{D_{t+1, T}} \cdot S_{t+1}^{N} \cdot M
\] (8)

where, \( D_{t+1} \) is the distance between two nodes, which can be calculated by:

\[
D_{t+1} = \sqrt{(i_{t} - i_{t+1})^{2} + (j_{t} - j_{t+1})^{2} + (k_{t} - k_{t+1})^{2}}
\] (9)

\( D_{t+1, T} \) is the distance between node \( t \) and the target point, which can be calculated by:

\[
D_{t+1, T} = \sqrt{(i_{T} - i_{t+1})^{2} + (j_{T} - j_{t+1})^{2} + (k_{T} - k_{t+1})^{2}}
\] (10)

\( S_{t+1} \) is the ratio of the number of feasible nodes \( N_{t+1} \) and total number of nodes \( N \), which can be calculated by:
\[ S_{i+1} = \frac{N_{i+1}}{N} \quad (11) \]

\[ M = \frac{1}{|k_i - k_{i+1}| + 1} \quad (12) \]

\[ w_1, w_2 \text{ and } w_3 \text{ are the corresponding weighted parameters. Through experience, } w_1 \text{ takes 1; } w_2 \text{ takes the number of the abscissa between the current node and the target point. } w_3 \text{ takes a positive number to indicate that the next node selection should consider its feasibility.} \]

5.1.2. Design of selective transition probability. In ACO, ants select the next node according to the selection transition probability. Roulette method is used for selection. The selection transition probability is closely related to the pheromone and heuristic value of nodes. The transition probability formula from node \( t \) to node \( t+1 \) is set as follows:

\[
P_{t,t+1} = \begin{cases} 
\left[ \frac{\tau_{t,t+1}^\alpha Q_{t,t+1}^\beta}{\sum \left[ \tau_{t,t+1}^\alpha Q_{t,t+1}^\beta \right]} \right]^{\text{feasible point}}, \\
0 & \text{, infeasible point} 
\end{cases} \quad (13) 
\]

where, \( \tau_{t+1} \) is the value of pheromone of node \( t+1 \), \( Q_{t,t+1} \) is the value of heuristic from node \( t \) to node \( t+1 \). \( \alpha \) is the importance factor of pheromone and \( \beta \) is the importance factor of heuristic.

Because the transition probabilities between nodes are similar, if the roulette method is used to select the next node, the node with better transition probability may not be selected. Therefore, the selection of transition probability is updated as followed:

\[
J = \begin{cases} 
\arg \max \left[ \left( \frac{\tau_{t,t+1}^\alpha Q_{t,t+1}^\beta}{\sum \left[ \tau_{t,t+1}^\alpha Q_{t,t+1}^\beta \right]} \right) \right], \text{rand} \geq 0.5 \\
equation(13), \text{roulette method}, \text{rand}<0.5 
\end{cases} \quad (14) 
\]

5.1.3. Design of pheromone update. ACO includes local pheromone update and global pheromone update. Both local update and global update ensure the successful searching ability of ACO.

The equation of local pheromone update is as followed.

\[
\tau_i = (1 - \gamma) \tau_i \quad (15) 
\]

where, \( \tau_i \) is the pheromone value of node \( t \), \( \gamma \) is the attenuation coefficient of local pheromone.

The equation of global pheromone update is as followed.

\[
\tau_i = (1 - \rho) \tau_i + \rho \Delta \tau_i \quad \Delta \tau_i = \frac{K}{\min(\text{Distance}(m))} \quad (16) 
\]

where, Distance\((m)\) is the path length of ant \( m \). \( \rho \) is the update coefficient of global pheromone and \( K \) is a constant coefficient.

5.1.4. Design of fitness value. The fitness value is used for evaluating the path search result. For the shortest path problem, the formula of fitness value is as followed.

\[
f = \text{Distance(path)} \quad (17) 
\]

where, Distance\((\text{path})\) is the total length of the path.
5.2. Optimization of ACO

5.2.1. Distribution of initial pheromone. The initial pheromone of the path points corresponding to the feasible path searched by PSO is doubled. While the initial pheromones of the other path points remain unchanged.

5.2.2. Optimization of global pheromone update coefficient. Considering that the initial pheromone distribution has been optimized in PACO, $\rho$ should be selected as a smaller number in the early stage. In the middle stage of the algorithm, $\rho$ should be a larger value to enhance the global search ability. In the late stage, in order to improve the performance, the smaller value of $\rho$ should be selected. Assuming that the maximum number of iterations is $T_{\text{max}}$, the equation is as follows:

$$
\rho = \begin{cases} 
0.2 & \text{if } T \in [1, 0.25T_{\text{max}}) \\
0.4 & \text{if } T \in [0.25T_{\text{max}}, 0.75T_{\text{max}}) \\
0.1 & \text{if } T \in [0.75T_{\text{max}}, T_{\text{max}}]
\end{cases}
$$

(18)

6. Simulation results and analysis

6.1. Simulation environment
The dataset (122.85°E ~ 123.10°E, 24.15°N ~ 24.40°N) was downloaded from GEBCO, which is sampled every 1KM along the longitude and latitude direction. The matrix size is 30*30, so the 3D environment is divided into 30 planes along the longitude direction.

The simulation platform is equipped with Windows 7 system, Intel Core i5 processor and 8GB memory. The algorithm is programmed by MATLAB 2017A.

6.2. Simulation results
Parameters relevant to PACO are set as showing in Table 1.

| Sort       | Symbol | Parameter setting | Meaning                                      |
|------------|--------|-------------------|----------------------------------------------|
| Grid model parameters | $N$ | 30 | Grid number along $X$ axis |
|            | $M$ | 30 | Grid number along $Y$ axis |
|            | $L$ | 20 | Grid number along $Z$ axis |
|            | $y_{\text{max}}$ | 2 | The maximum vertical distance limitation |
|            | $z_{\text{max}}$ | 2 | The maximum horizontal distance limitation |
|            | $S$ | (1,4,16) | Starting point coordinates |
|            | $T$ | (30,22,16) | Target point coordinates |
| PSO parameters | $m$ | 300 | Number of particles |
|            | $w$ | 0.6 | The inertia factor |
|            | $c_1$ | 1 | Learning acceleration factor |
|            | $c_2$ | 1 | Learning acceleration factor |
|            | $V_{\text{max}}$ | 2 | The maximum velocity of particles |
|            | $I$-PSO | 100 | The maximum iteration of PSO |
| ACO parameters | $Pop$ | 10 | Number of ants |
|            | $\alpha$ | 1 | The importance factor of pheromone |
|            | $\beta$ | 1 | The importance factor of heuristic |
|            | $K$ | 100 | Constant coefficient |
|            | $V_{\text{max}}$ | 2 | The maximum velocity of particles |
Simulation is carried out after parameter setting. The pre-processing result after PSO is shown in Figure 6. The simulation results of ACO and PACO are shown in the Figure 7 and 8. ACO achieves the final fitness value of 37.6935 km in 212 iterations, and PACO achieves the final fitness value of 36.8350 km in 153 iterations.

![Preprocessing results after PSO](image1)

![Path planning results](image2)

![Variation trend of fitness value](image3)

![Fitness value record](image4)

In order to fully prove the effectiveness of the algorithm, simulation has been done for 10 times continuously. Simulation results were recorded in Figure 9 and Table 2.

| Name | The final fitness value (km) | Minimum number of iterations | Time consuming (s) |
|------|-----------------------------|------------------------------|-------------------|
| ACO  | 37.1922                     | 225.4                        | 9.8554            |
| PACO | 36.7039                     | 171.8                        | 9.7862            |

6.3. Analysis of simulation results

It can be seen from Figure 6 that some feasible paths have been found through the PSO, which can provide reference for the initial pheromone distribution.
Figure 7 shows that both ACO and PACO can realize global path planning of UUV. However, through the comparison shows in Figure 8, it can be seen that PACO has faster convergence speed than ACO at the initial iteration stage. Finally, PACO can also obtain better path results at a smaller number of iterations.

From Figure 9, we can find that PACO can get better fitness value than ACO in each time of simulations. The average values recorded in Table 2 shown that time consuming between ACO and PACO is almost the same. But PACO can get the optimized fitness value with less number of iterations.

7. Conclusion
This paper combined the advantages of PSO and ACO, a fusion algorithm named PACO is designed. Based on the limitations of search grid, PSO and ACO are optimized. The real ocean environment dataset is downloaded from GEBCO. After grid modelling, the algorithm is used for 3D global path planning. From the simulation results, the effectiveness of PACO is verified.

References
[1] Robert D Holzer. (2019) Unmanned Maritime Systems: An Update on Core Technologies. Sea Technology 60:11.
[2] TSAI-SHONGLEE, JIANN-JONECHEN. (2016) A Novel Path Planning Method for Space Environment. Journal of Residuals Science & Technology, 13:3.
[3] Pengfei Chen, Yamin Huang, Eleonora Papadimitriou, Junmin Mou, Pieter van Gelder. (2020) Global path planning for autonomous ship: A hybrid approach of Fast Marching Square and velocity obstacles methods. Ocean Engineering, 214.
[4] Yu Zhen-zhong, Yan Ji-hong, Zhao Jie. (2011) Mobile robot path planning based on improved artificial potential field method. Journal of Harbin Institute of Technology, 43(1):50-55.
[5] Tang Bijun,Hirota Kaoru, Wu Xiangdong, Dai Yaping, Jia Zhiyang. (2021) Path Planning Based on Improved Hybrid A* Algorithm, 25(1).
[6] Xiao-huan Liu,Degan Zhang, Jie Zhang, Ting Zhang, Hao Li Zhu. (2021) A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm. Cluster Computing.
[7] Sara Perez-Carabaza, Eva Besada-Portas, Jose A. Lopez-Orozco,Jesus M. de la Cruz. (2018) Ant colony optimization for multi-UAV minimum time search in uncertain domains. Applied Soft Computing, 62.
[8] Shubham Gupta, Kusum Deep. (2018) Random walk grey wolf optimizer for constrained engineering optimization problems. Computational Intelligence, 34(4).
[9] Prashant Pandey, Anupam Shukla, Ritu Tiwari. (2018) Three-dimensional path planning for unmanned aerial vehicles using glowworm swarm optimization algorithm. International Journal of System Assurance Engineering and Management, 9(4).
[10] Guang-Lei Z, He-Ming J. (2012) Global path planning of AUV based on improved ant colony optimization algorithm. Automation and Logistics (ICAL), IEEE International Conference on. IEEE, 2012.
[11] Wei, Xianmin. (2013) Efficient Robot Path Planning Based on Simulated Annealing Artificial Neural Networks. International Journal of Advancements in Computing Technology, 5(4).
[12] Pan Xin,Wu XUshehong, Hou Xinguo, Feng Yuan. (2017) Global path planning based on genetic hybrid algorithm for AUV. Journal of Huazhong University of Science and Technology. 45(05):45-49.
[13] Ma Y N , Gong Y J , Xiao C F , et al. (2018) Path Planning for Autonomous Underwater Vehicles: An Ant Colony Algorithm Incorporating Alarm Pheromone. IEEE Transactions on Vehicular Technology, 2018, 1:1-1.
[14] Dorigo M, Di Caro G, Gambardella L M. (1996) Ant algorithms for discrete optimization. Artificial life, 137:72.