Received July 18, 2020, accepted July 29, 2020, date of publication August 3, 2020, date of current version August 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013953

Improved Quantum Particle Swarm Optimization Algorithm for Offline Path Planning in AUVs

LEI WANG, LILI LIU, JUNYAN QI, AND WEIPING PENG
School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, China

Corresponding author: Lili Liu (lili_laus@163.com)

This work was supported in part by the Educational Commission of Henan Province of China under Grant 20A520015, in part by the Key Science and Technology Project of Henan Province under Grant 192102210123, and in part by the Foundation for University Key Youth Teacher by the Henan Province under Grant 2019GGJS061.

ABSTRACT This paper proposes an offline path planning method based on the Improved Quantum Particle Swarm Optimization (IQPSO) algorithm for Autonomous Underwater Vehicles (AUVs) in the underwater environment. The spherical modelling method is adopted to represent irregular underwater obstacles as spheres with a specified radius. Then, the IQPSO algorithm is developed to solve the problem of the limitations of the convergence and optimization ability of the traditional Quantum Particle Swarm Optimization (QPSO) algorithm and to identify the best path for AUVs. In this algorithm, to satisfy the three factors of path safety, path length and angle change of the path point, the fitness function is constructed to achieve multi-objective optimization. A smooth path is designed using the cubic spline interpolation algorithm. Different scenes or the same scene with different obstacles are designed to verify the effectiveness of the algorithm. The simulation results show that compared with PSO algorithm, QPSO algorithm, EGA algorithm and DENPSO algorithm, the path generated by IQPSO algorithm in various scenes is shorter, smoother and more stable.

INDEX TERMS AUV, improved QPSO algorithm, multi-objective optimization, path planning.

I. INTRODUCTION

With the development of underwater applications, such as seabed exploration and underwater resource exploitation, Autonomous Underwater Vehicle (AUV) technology is increasingly common. In most underwater applications, AUVs are an increasingly important auxiliary tool for underwater work [1]. An AUV completes the specified work by moving within the work area. Path planning is the problem of determining the path of an AUV from the starting point to the target point [2], [3]. Path planning is recognized as a key technique of AUVs. Traditional path planning algorithms include the A* algorithm [4], the Rapidly exploring Random Tree (RRT) [5] algorithm, the D* algorithm [6], the Dijkstra algorithm [7], [8], the Ant Colony Optimization (ACO) algorithm [9]–[11] and the Artificial Potential Field (APF) algorithm [12], [13]. The A*, D*, Dijkstra and ACO algorithms are path planning methods based on graph theory. This kind of method depends on building a grid model of the environment. The optimal path is obtained by searching. However, the grid method has a strict grid size and is vulnerable to environmental restrictions. Thus, for complex large-scale underwater environments, the computing costs of these algorithms are high, and their efficiency is low. The RRT algorithm can quickly generate conflict-free paths, but it cannot guarantee optimality. The advantage of the APF method is that the planning speed is very fast, but when the attraction and repulsion forces are equal, this method easily becomes trapped in local minimum points. As a result, the AUV will be unable to move forward, resulting in the failure of the planning task. In addition, the traditional path planning algorithm does not consider such factors as path safety, path length and angle change. Small angle change values make the path shorter and smoother. When an AUV moves underwater, the path is short and smooth, which effectively saves energy. Therefore, when energy is limited and the environment is complex, the use of the traditional path planning algorithm often results in AUVs exhausting their energy and colliding with obstacles. To date, the path planning method based on the intelligent evolutionary algorithm has been the most effective and fastest method. This method can increase the convergence speed, avoid local optimal solutions and identify

The associate editor coordinating the review of this manuscript and approving it for publication was Aysegul Ucar.

feasible solutions that meet the requirements in a short time [14].

In recent years, many researchers have proposed that the path planning problem of AUVs can be regarded as an NP complete problem [15]. Therefore, path planning algorithms combined with intelligent optimization, including the Genetic Algorithm (GA) [16], the Differential Evolution (DE) algorithm, the Firefly Algorithm (FA), the Particle Swarm Optimization (PSO) algorithm [17] and the Artificial Bee Colony (ABC) algorithm have been proposed. However, in most literature on path planning, single objective optimization is only carried out for the factor of path length, while the factors of smoothness and security are not considered, resulting in unsatisfactory path planning results. Moreover, there are few path planning methods for the three-dimensional (3D) environment. Many path planning methods in the two-dimensional (2D) environment are not available for the 3D environment. The optimization performance of most intelligent optimization algorithms is unsatisfactory because the convergence rate is low and the algorithms easily fall into local optimization.

Given the limitations of traditional path planning methods and common optimization algorithms, many methods have been proposed. ACO is an evolutionary algorithm that simulates the foraging behaviour of ants [18], [19]; it was inspired by the process of ants seeking food and created by Marco Dorigo. ACO is often combined with the grid method [20]. Compared to other evolutionary algorithms, the ACO algorithm is used earlier and more widely in path planning. Duan *et al*. [21] first modelled a grid in 3D space and proposed a hybrid heuristic ACO-DE algorithm for path planning calculations. The authors suggest using the DE algorithm in the process of pheromone updating unlike the traditional ACO algorithm. Although this method can realize path planning in a 3D space, the mesh modelling of 3D space still limits the performance of the algorithm. Nazarahari *et al*. [22] proposed a multiple target multiple robots path planning algorithm based on the Enhanced Genetic Algorithm (EGA). First, using the APF algorithm generates a path from the source points to the target points as the initial path. Then, the path length, path smoothness and security are regarded as the objective function, and using the EGA algorithm, the optimal path is generated. Finally, the path is smoothed by cubic spline interpolation to obtain the final path. Although the method achieves multi-objective optimization in path planning, it is only available in the 2D grid environment. In addition, the path smoothed by cubic spline interpolation may not be a safe path. Parhi and Kundu [23] proposed using the DE method in path planning. Wu *et al.* [24] proposed a Distance Evolution Nonlinear Particle Swarm Optimization (DENPSO) algorithm, with energy as the optimization objective to avoid the influence of eddy current fields. However, the convergence of the DE and PSO algorithms is poor. Han [25] proposed a COSPS method, which can be used to determine key obstacles without considering non-key obstacles, thus improving the efficiency of path planning. However, synthetic factors such as the path length and smoothness are not considered in this method. Lucas *et al.* [26] used the Non-dominated Sorting Genetic Algorithm II (NSGA-II) for the multi-objective path planning of underwater gliders. Lin *et al.* [27] used a convolutional neural network to design an online path planning algorithm to improve the autonomy and intelligence of obstacle avoidance. Neural networks require a parameter training process. Thus, the algorithm complexity is high, and its application in underwater environments is limited. Silva Junior *et al.* [28] applied the Q-learning method to the path planning of sailing robots. In addition to the problem of the algorithm's complexity, this method is only available for the 2D water surface.

To solve the abovementioned problems, this paper designs a 3D offline path planning algorithm based on the Improved Quantum Particle Swarm Optimization (IQPSO) algorithm. Offline means non-real-time, that is, planning the route for an AUV prior to its operation. This method does not require rasterized modelling of the underwater environment. The expansion-contraction factor is improved to prevent particles from falling into local optimization. To improve the convergence speed of the QPSO algorithm, the average best position of the particle is modified from the arithmetic average to the weighted average, and the weight depends on the fitness value of the particle. In addition, the corresponding fitness functions are designed for the three factors of path length, angle change and path safety to achieve the multi-objective optimization of underwater AUV path planning.

To summarize, the main contributions of this paper are as follows:

1) The convergence and optimization ability of the traditional QPSO algorithm are improved.
2) A reasonable fitness function is designed for the 3D underwater environment. Path safety, path length and angle change are considered to achieve multi-objective optimization of path planning. The cubic spline interpolation algorithm is used to smooth the path.
3) Simulation tools are used to verify the effectiveness of the algorithm, which is compared with other algorithms to prove its superiority.

Section II defines AUV kinematic model and obstacle model. Section III gives a detailed description of the proposed methods. Simulation results are elaborated and analysed in the section IV. Conclusion are summarized in the section V.

II. RELEVANT MODEL

A. AUV KINEMATIC MODEL

The Cartesian workspace is shown in Fig. 1. This workspace contains two coordinate systems: the inertial coordinate system based on global positioning and the body-fixed coordinate system. Both of these systems obey the right-hand rule [29].

In 3D space, an AUV has 6 Degrees of Freedom (DOFs), including translational DOFs and rotational DOFs on the x, y and z-axes. To better analyse the motion model of an

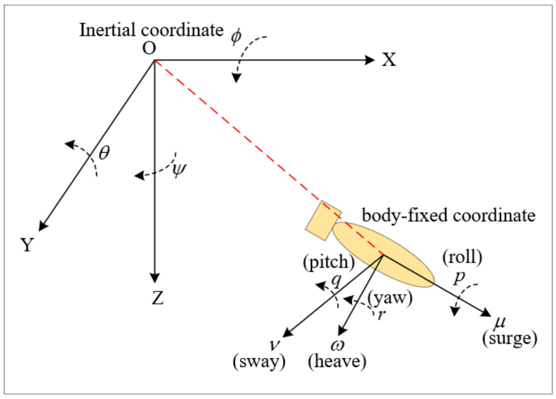

FIGURE 1. Workspace coordinate system.

underwater AUV, the AUV motion parameters are defined as follows:

1) The source point: $q_0\ (x_0, y_0, z_0)$.
2) The target point: $q_{n+1}\ (x_{n+1}, y_{n+1}, z_{n+1})$.
3) Position and attitude vectors: $\eta = [x, y, z, \phi, \theta, \psi]^T$. x, y, z represent the position of the AUV, and ϕ, θ, ψ represent the Euler angle of the AUV in the coordinate system.
4) The translational and rotational velocities parameters are $V = [\mu, \nu, \omega, p, q, r]^T$, where μ, ν, ω represent the translational velocity (the wave, swing and heave components, respectively), and p, q, r represent the rotational velocity (the roll rate, pitch rate and yaw rate, respectively).

Therefore, the motion model of the AUV is given as

$$\eta = \begin{bmatrix} J_1 & \\ & J_2 \end{bmatrix} V \tag{1}$$

where J_1 and J_2 are expressed as follows:

In the 3D underwater working environment, the AUV is regarded as a point O. The path of points from the source (q_0) to the target (q_{n+1}) can be represented by $X = \{q_i | 1 \le i \le n\}$, which is a discrete set of points. In general, AUVs do not easily roll in the underwater environment, so it is assumed that $\theta = 0°$. In this paper, the Euler angle components of the AUV are considered to only be in the XOY plane and the XOZ plane, as follows:

$$\psi_i = \tan^{-1}\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right) \tag{4}$$

$$\phi_i = \tan^{-1}\left(\frac{z_i - z_{i-1}}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}\right) \tag{5}$$

The underwater environment model is an important part of underwater path planning because the energy consumption of underwater AUVs is mainly influenced by underwater eddies and obstacles. When an AUV passes through an ocean vortex, it easily yaws due to the external force of the vortex field. The AUV also needs to expend more energy to arrive at the destination. It is assumed that the working velocity of the AUV is a constant V, and the velocity will change under the influence of the eddy current field in the underwater environment. The final velocity component is expressed as follows:

$$\begin{cases} \mu = |V| \cos\phi \cos\theta + \mu_c \\ \upsilon = |V| \sin\phi \cos\theta + \nu_c \\ \omega = |V| \sin\theta + \omega_c \end{cases} \tag{6}$$

where (μ_c, ν_c, ω_c) is the component of the flow velocity.

B. OBSTACLE MODEL

Active sonar technology can be used to detect underwater obstacles. In this paper, it is assumed that the location of underwater obstacles is detected by sensors carrying active sonar. Irregular obstacles are expressed as spheres with specified spherical centres and radii based on the spherical modelling method. The real obstacle is limited inside the sphere. To avoid collision between the AUV and obstacles after circular modelling in a 2D environment, d_0 is used as the safe distance, as shown in Fig. 2. Similarly, in the 3D environment, when the path of the AUV is a tangent line of the sphere, this strategy ensures that the AUV does not collide with obstacles and improves the safety of the path.

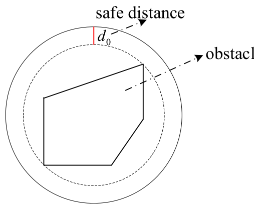

FIGURE 2. 2D obstacle model diagram.

III. THE PROPOSED PATH PLANNING ALGORITHM

A. PARTICLE SWARM OPTIMIZATION

Inspired by the foraging behaviour of birds, the PSO algorithm was proposed by Eberhart and Kennedy [30] in 1995. The algorithm sets a flock of birds to search for food randomly in a designated area, and only one piece of food will be found in the area. All birds only know the distance from their current position to the food; they do not know the specific location of the food. Therefore, individuals search by the individual closest to the food in the flock.

The PSO algorithm uses particles to simulate the above individual birds, and each particle can be regarded as a search individual in the N-dimensional search space. The current position of the particle is a candidate solution of the corresponding optimization problem, and the flight process of the particle is the search process of the individual. The flight velocity of particles can be dynamically adjusted according to the historical optimal position of particles and the historical optimal position of the population. Particles have only two properties: velocity $V_i = (v_{i1}, v_{i2}, \ldots, v_{im})$, which represents how fast they move, and position $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$, which represents the direction in which they move. The optimal solution of individual search for each particle is called individual extremum, and the optimal individual extremum in the particle swarm is regarded as the current global optimal solution. By constantly updating the speed and location, the optimal solution satisfying the termination condition is finally obtained. The PSO algorithm needs to update the position and velocity of the particles simultaneously in the iterative process. The iterative formulas of the velocity and position are as follows:

$$V_i(k+1) = \omega V_i(k) + C_1 rand(0,1)(P_i(k) - X_i(k)) + C_2 rand(0,1)(G_i(k) - X_i(k)) \quad (7)$$

$$X_i(k+1) = X_i(k) + V_i(k+1) \quad (8)$$

where ω is the inertia weighting factor; C_1 and C_2 are acceleration constants: C_1 is the individual learning factor of the particle, and C_2 is the global learning factor of the particle; $P_i(k)$ represents the individual extremum; and $G_i(k)$ represents the global optimal solution.

B. QUANTUM PARTICLE SWARM OPTIMIZATION ALGORITHM

In PSO, the particle velocity is always limited. The search space in the search process is a limited area that does not cover the entire feasible space. However, the particles have the characteristics of an aggregation state in the quantum space; thus, the algorithm can implement search in the entire feasible solution space. According to the basic convergence properties of the particle swarm and inspired by the basic theory of quantum physics, Sun *et al.* [31] proposed the QPSO algorithm. Particle updating does not require a velocity vector; therefore, this algorithm improves the search speed of the PSO algorithm, reduces the number of parameters and has a simple form.

Assuming that $X_i(k), p_i(k), y_i(k), C(k)$ and $\hat{y}(k)$ represent the current position, attractor position, individual best position, average best position and global best position, respectively, of the particle swarm when the number of iterations is k, the particle update equation of the QPSO algorithm is as follows:

In this equation, α is the expansion-contraction factor that ranges from 0 to 1. It is either a constant, or it gradually decreases as the number of iterations increases, and it is the only parameter in the QPSO algorithm other than the population size and the number of iterations. $\mu_{i,j}(k)$ and $\varphi_{i,j}(k)$ are uniformly distributed random variables that range from 0 to 1, $i = 1, 2, \ldots, N$. N is the number of particles. $j = 1, 2, \ldots, D$, and D is the particle dimension.

C. IMPROVED QUANTUM PARTICLE SWARM OPTIMIZATION ALGORITHM

In complex problems, because of the excessive number of particles, the QPSO algorithm has a slow convergence speed and easily falls into local optima. To solve these problems, an improved QPSO algorithm is proposed.

In Eq. (11), due to the continuous decline of α, the particles converge prematurely, and the fitness value is locally rather than globally optimal, which is not ideal in the iterative process. Therefore, this paper proposes a self-adaptive α method. If the fitness difference between the current particle and the global optimal particle is less than the threshold $\gamma = 0.01$, the value of α decreases as the number of iterations increases. Otherwise, α is a random number in the range of $(\alpha_{\min}, \alpha_{\max})$. In this paper, $\alpha_{\min} = 0$ and $\alpha_{\max} = 1$. When the fitness value of the particle does not reach convergence, adding a random disturbance to α can prevent the particle from falling into the local optimum. The value of α is as follows:

$$\alpha = \begin{cases} (\alpha_{\max} - \alpha_{\min}) \dfrac{Maxn - k}{Maxn} + \alpha_{\min}, \ldots fitness(X(k)) \\ \quad - fitness(\hat{y}) < \gamma, \\ rand(\alpha_{\min}, \alpha_{\max}), \ldots fitness(X(k)) - fitness(\hat{y}) \geq \gamma. \end{cases} \quad (14)$$

where *Maxn* is the maximum number of iterations.

According to Eq. (11), the average best position also plays an important role in particle updating, and the direction of particle updating tends to the average best position. In this paper, the average best position is calculated by combining the fitness of the particle to increase the convergence speed of the QPSO algorithm and to make the particle update to a higher fitness value. The larger the particle fitness is, the stronger the particle fitness is. Thus, the larger the weight a_i of $y_i(k)$ is, the closer $C(k)$ is to the optimal value. Therefore, the improved average best position $C(k)$ is updated as

$$C(k) = \sum_{i=1}^{N} a_i y_i(k) \quad (15)$$

where a_i can be calculated as

$$a_i = \frac{fitness(X_i(k))}{\sum_{i=1}^{N} fitness(X_i(k))} \quad (16)$$

$$J_1 = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\phi\cos\theta \end{bmatrix} \quad (2)$$

$$J_2 = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \quad (3)$$

Each path point of the particle must meet two conditions: it must be within the boundary of the 3D environment, and it must avoid the obstacle. The conditions can be expressed as

$$\begin{cases} X_{\min}^{x} \leq X_{ij}^{x} \leq X_{\max}^{x} \\ X_{\min}^{y} \leq X_{ij}^{y} \leq X_{\max}^{y} \\ X_{\min}^{z} \leq X_{ij}^{z} \leq X_{\max}^{z} \end{cases}, d\left(X_{ij}, obs_k\right) > d_0 \tag{17}$$

By Eq. (17), the safety and effectiveness of the initial particle generation path are guaranteed.

D. IQPSO FITNESS FUNCTION

The fitness functions in some studies only consider the path length and path safety or only the path length. The angle change is not considered, which causes the path planning task to fail and causes energy to be consumed by the directional adjustment of the AUV. In this paper, the proposed algorithm obtains a short and safe path, and the fitness function of the IQPSO algorithm is given as follows:

$$fitness\,(X) = S_1\,(X) \cdot (c_1 D\,(X) + c_2 S_2\,(X)) \tag{18}$$

where $X = \{q_1, q_2, \ldots, q_n\}$ represents a series of discrete points between the source point and the target point. c_1 and c_2 are constants, and $c_1 + c_2 = 1$. $S_1\,(X)$ is the safety function, $D\,(X)$ is the path length function and $S_2\,(X)$ is the angle change function. The larger the particle fitness value is, the higher the particle fitness is. By iteration, the fitness values gradually converge. At the same time, the obtained particles are output as the path points. The path formed by these path points is the optimal path.

1) Safety function.

The safety of the path can be calculated using the following formula:

$$S_1\,(X) = \begin{cases} 1, \ldots d\left(q_i q_{i+1}, obs_j\right) > d_0 + r_{obs}^{j} \\ 0, \ldots else. \end{cases} \tag{19}$$

where $d\left(q_i q_{i+1}, obs_j\right)$ represents the distance between $q_i q_{i+1}$ (the adjacent path point vector) and obs_j (the centre of the obstacle). r_{obs}^{j} represents the radius of the obstacles. The distance between the line $q_i q_{i+1}$ and the centre of any obstacle is greater than the radius of the obstacle. In other words, when the line between two adjacent path points does not pass through the obstacle, the value of $S_1\,(X)$ is 1, which represents the path safety. Otherwise, the path is in danger of colliding with obstacles.

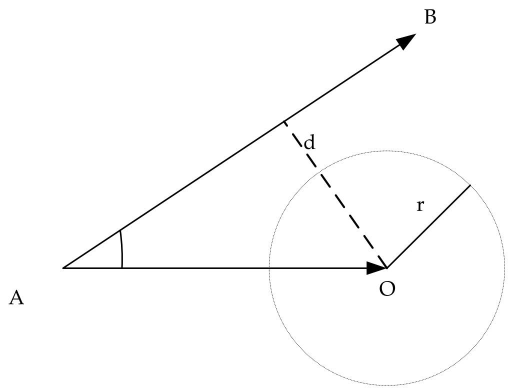

FIGURE 3. Schematic diagram of distance solution.

As shown in Fig. 3, it is assumed that in a 3D environment, the coordinates of points A, B and O and the radius of sphere O are known. The distance from the centre of the sphere to the vector AB can be obtained as

$$d = \frac{\| \overrightarrow{AB} \times \overrightarrow{AO} \|}{\| \overrightarrow{AB} \|} \tag{20}$$

Therefore, as above, $d\left(q_i q_{i+1}, obs_j\right)$ can be expressed as follows:

$$d\left(q_i q_{i+1}, obs_j\right) = \frac{\| (q_{i+1} - q_i) \times \left(obs_j - q_i\right) \|}{\| (q_{i+1} - q_i) \|} \tag{21}$$

2) Path length function.

The path length function is given as

$$D\,(X) = \frac{d\,(q_0, q_{n+1})}{\sum_{i=0}^{n} d\,(q_i, q_{i+1})} \tag{22}$$

$$y_i\,(k+1) = \begin{cases} X_i(0), \ldots k = 0, \\ y_i\,(k)\,, \ldots k > 0, & fitness(y_i(k)) \geq fitness(X_i(k+1)), \\ X_i(k+1), \ldots k > 0, & fitness(y_i(k)) < fitness(X_i(k+1)). \end{cases} \tag{9}$$

$$\hat{y}\,(k) = \begin{cases} \max\left\{y_1(k), y_2(k), \ldots y_N(k)\right\}, \ldots k > 0, \\ \max\left\{X_1(0), X_2(0), \ldots X_N(0)\right\}, \ldots k = 0. \end{cases} \tag{10}$$

$$\mathrm{X}_{i,j}\,(k+1) = \begin{cases} p_{i,j}\,(k) + \alpha |C\,(k) - X_{i,j}\,(k)\,| \ln \frac{1}{\mu_{i,j}(k)}, \ldots \mu_{i,j}\,(k) \geq 0.5 \\ p_{i,j}\,(k) - \alpha |C\,(k) - X_{i,j}\,(k)\,| \ln \frac{1}{\mu_{i,j}(k)}, \ldots \mu_{i,j}\,(k) < 0.5. \end{cases} \tag{11}$$

$$p_{i,j}\,(k) = \varphi_{i,j}\,(k)\, y_{i,j}\,(k) + \left(1 - \varphi_{i,j}\,(k)\right) \hat{y}_{i,j}\,(k) \tag{12}$$

$$C\,(k) = \frac{1}{N} \sum_{i=1}^{N} y_i\,(k) \tag{13}$$

where $d\left(q_0, q_{n+1}\right)$ represents the distance from the beginning to the end, and $\sum_{i=0}^{n} d\left(q_i, q_{i+1}\right)$ is the path length. $D(X)$ is in the range of (0,1]. The shorter the path length is, the larger the value of $D(X)$ is.

3) Angle change function.

The angle change function is given as

$$S_2(X) = \frac{\sum_{i=1}^{n} U_i}{n} \quad (23)$$

$$U_i = \begin{cases} 1, \ldots \theta_i \leq \frac{\pi}{2}, \\ 0, \ldots \text{else.} \end{cases}, (i = 1, 2, \ldots, n) \quad (24)$$

$$\cos\left(q_i q_{i-1}, q_{i+1} q_i\right) = \frac{\left(q_i - q_{i-1}\right) \times \left(q_{i+1} - q_i\right)}{d\left(q_i, q_{i-1}\right) \times d\left(q_{i+1}, q_i\right)}$$
$$\theta_i = \arccos(q_i q_{i-1}, q_{i+1} q_i) \quad (25)$$

where θ_i represents the angle change between two adjacent paths and is in the range of $[0, \pi]$. By Eq. (25), $\cos\left(q_i q_{i-1}, q_{i+1} q_i\right)$ can be calculated, and θ_i can be obtained at the same time. If $\theta_i > \frac{\pi}{2}$, the path length may be extended. Otherwise, the path length is shorter. The smaller the angle changes are, the larger the value of $S_2(X)$ is, and the shorter and smoother the path is.

E. CUBIC SPLINE INTERPOLATION

According to the above IQPSO algorithm, a set of discrete points from the starting point to the end point can be obtained. The path obtained by connecting these points satisfies C0-continuity, but not C1-continuity. This paper uses the cubic spline interpolation [32], [33] to smooth the path. Cubic spline interpolation is a sectional interpolation method. By using a series of points based on cubic polynomial interpolation, the interval forms a smooth curve. Using the cubic spline interpolation method, the mobile robot path fitting curve is smoother, and the robot is assured to have good dynamic characteristics when it abruptly stops or turns. Compared with straight line and arc fitting, this method is more prominent.

If there are n nodes on the interval $[a, b]$, $a = x_1 < \ldots < x_n = b$. The corresponding functional values of these nodes are $f(x_i) = f_i\,(i = 1, 2, \ldots, n)$. In the interval $[x_j, x_{j+1}]\,(j = 1, 2, \ldots, n-1)$, there is a function $g(x)$ expressed as $g(x) = a_j x^3 + b_j x^2 + c_j x + d_j$. Cubic spline interpolation requires the following conditions to be met:

(1) $g(x_j) = f_j$;
(2) $g(x)$ is continuous;
(3) The first derivative of $g(x)$ is continuous;
(4) The second derivative of $g(x)$ is continuous. g(x) can be expressed as follows:

$$g(x) = \frac{(x_{j-1} - x)^3}{6(\Delta x_j)} M_j + \frac{(x - x_j)^3}{6(\Delta x_j)} M_{j+1} + A_j x + B_j,$$
$$\Delta x_j = x_{j+1} - x_j. \quad (26)$$

where

$$A_j = \frac{f_{j+1} - f_j}{\Delta x_j} - \frac{M_{j+1} - M_j}{6} \cdot \Delta x_j \quad (27)$$

$$B_j = f_{j+1} - \frac{M_{j+1}}{6} \cdot \left(\Delta x_j\right)^2 - \left(\frac{f_{j+1} - f_j}{x_{j+!} - x_j} - \frac{M_{j+1} - M_j}{6} \cdot \left(\left(\Delta x_j\right)\right)\right) x_{j+!} \quad (28)$$

In Eq. (27) and (28), the values of M_j and M_{j+1} can be obtained by the following formula:

$$\begin{bmatrix} 2 & 1 & & & \\ \mu_2 & 2 & 1-\mu_2 & & \\ & \cdots & \cdots & \cdots & \\ & & & 2 & 1-\mu_{n-1} \\ & & & 1 & 2 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \cdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} \beta_1 \\ d_2 \\ \cdots \\ d_{n-1} \\ \beta_n \end{bmatrix} \quad (29)$$

where

$$\beta_1 = \frac{6}{\Delta x_1}\left(\frac{f_2 - f_1}{\Delta x_1} - f_1'\right) \quad (30)$$

$$\beta_n = \frac{6}{\Delta x_n}\left(\frac{f_n - f_{n-1}}{\Delta x_{n-1}}{}_1 - f_n'\right) \quad (31)$$

$$\mu_j = \frac{\Delta x_{j-1}}{\Delta x_{j-1} + \Delta x_j} \quad (32)$$

$$d_j = 6\left(\frac{f_{j+1} - f_j}{\Delta x_j} - \frac{f_j - f_{j-1}}{\Delta x_{j-1}}\right)\frac{1}{\Delta x_{j-1} + \Delta x_j} \quad (33)$$

Therefore, the interpolation function of each segment of the interval can be obtained by solving Eq. (29).

F. IQPSO PATH PLANNING ALGORITHM

The path planning steps based on the IQPSO algorithm are as follows:

Step 1: Initialize the number of particles N, the dimension D and the maximum number of iterations.

Step 2: Initialize the population and obtain N groups of initialization paths $X_i = \left\{q_i^1, q_i^2, \ldots, q_i^N\right\}$. All particles use the fitness equation to calculate the fitness value.

Step 3: Calculate attractor position p and the weighted average optimal extremum C according to the fitness value. Update the particle position.

Step 4: Calculate the particle fitness value according to the fitness function, and obtain the individual optimal value y and the global optimal value $\hat{y}$.

Step 5: Repeat steps 3 to 5 until the maximum number of iterations is reached.

Step 6: Output the global optimal solution.

Step 7: Take the optimal solution of the output as the path points. The path points are processed using cubic spline interpolation to generate smooth paths.

Fig. 4 shows the flow chart of the IQPSO path planning algorithm.

IV. SIMULATION RESULTS

To verify the effectiveness and feasibility of the proposed algorithm for underwater AUV path planning, in this section,

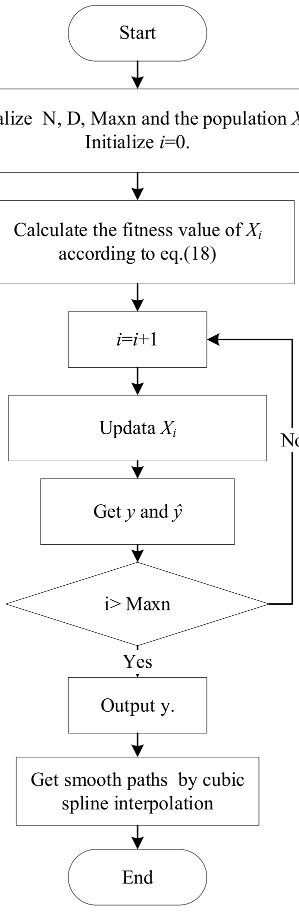
Start
Initialize N, D, Maxn and the population X_i. Initialize i=0.
Calculate the fitness value of X_i according to eq.(18)
i=i+1
Updata X_i
Get y and $\hat{y}$
i> Maxn
No
Yes
Output y.
Get smooth paths by cubic spline interpolation
End

FIGURE 4. Flow chart of path planning based on the IQPSO algorithm.

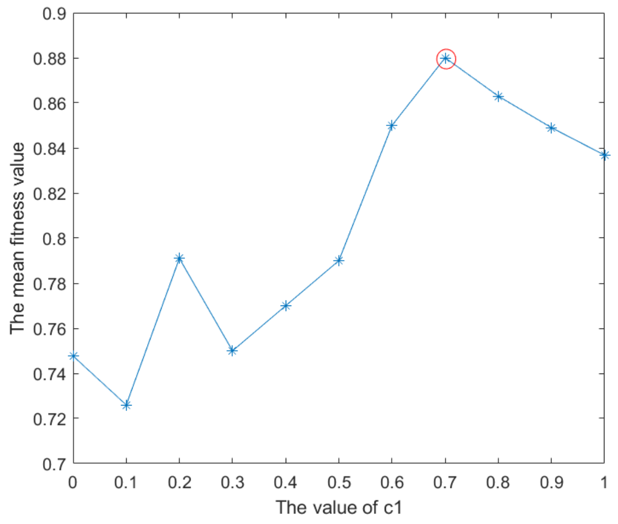

FIGURE 5. Change of the mean fitness value when c1 increases.

TABLE 1. Details of the obstacles in scenario 1.

Obstacle number	Spherical centre coordinate	Radius/(m)
1	(2.2, 1.9,2.3)	0.9
2	(4.0,4.5,5.2)	1.0
3	(7.2,8.5,8.3.)	1.8

TABLE 2. Details of the obstacles in scenario 2.

Obstacle number	Spherical centre coordinate	Radius/(m)
1	(29.5,32.5,27.5)	7.4
2	(7.0,12.0,16.0)	1.0
3	(18.3,21.9,29.8)	5.4
4	(29.8,37.5,42.2)	3.8

the algorithm is implemented and compared with the PSO algorithm, the QPSO algorithm, the EGA and the DENPSO algorithm by simulation.

A. EXPERIMENTAL PARAMETERS

In this paper, the PSO, DENPSO, traditional QPSO and IQPSO algorithms and the EGA are applied to the path planning of an AUV. By comparing these algorithms, the performance of the proposed IQPSO algorithm is verified. To determine the values of c1 and c2 in the fitness function of the algorithm proposed in this paper, 10 different environments are set. By changing the values of c1 and c2 in each environment, the values of c1 and c2 can be obtained by finding the best mean fitness value. In the experiment, c1 is set to increase linearly and c2 is set to decrease linearly in [0,1]. For both, the change step is 0.1, and the constraint condition c1+c2=1 is met. From Fig. 5, it can be seen that the value of c1 is 0.7, and the value of c2 is 0.3. In this situation, the mean fitness value equals to 0.88, which is the best.

According to the following settings, in the above three experimental scenarios, the path planning ability and algorithm stability of the PSO, DENPSO, traditional QPSO and IQPSO algorithms and the EGA were compared and analysed.

B. ALGORITHMS VALIDITY

1) SPECIFIC SCENARIOS WITH DIFFERENT SCALES AND NUMBERS OF OBSTACLES

By using the simulation tool, three random 3D experimental scenarios are established. In the three scenarios, there are different numbers of obstacles and obstacles with different radii. The size of every experimental scenario is also different. The size of scenario 1 is 10 m*10 m*10 m. The size of scenario 2 is 50 m*50 m*50 m. The size of scenario 3 is 100 m*100 m*100 m. These scenarios represent small, medium and large scale underwater areas, respectively. By doing so, the robustness of the algorithm is verified. Details of the spherical centre and radius of the obstacles in each scenario are shown in Tables 1 to 3. The experimental scenarios are shown in Fig. 6.

In this section, the PSO, DENPSO, traditional QPSO and IQPSO algorithms and the EGA are tested for 200 times each in every scenario. The minimum, maximum, mean and

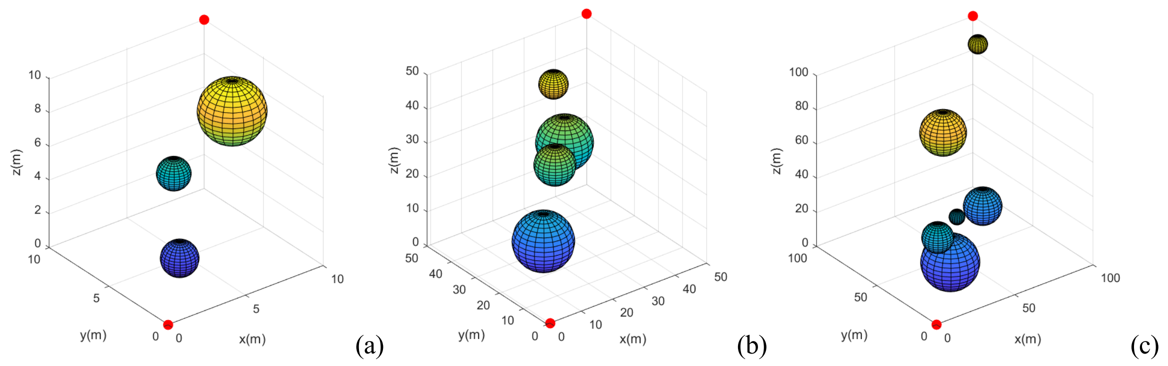

FIGURE 6. Experimental scenarios: (a) Scenario 1; (b) Scenario 2; (c) Scenario 3.

TABLE 3. Details of the obstacles in scenario 3.

Obstacle number	Spherical centre coordinate	Radius/(m)
1	(24.1,20.0,19.0)	15.1
2	(60.2,40.1,30.0)	9.9
3	(12.0,14.7,40.1)	8.2
4	(30.4,22.6,42.1)	4.1
5	(41.0,48.0,76.1)	12.0
6	(94.1,88.2,90.9)	5.0

TABLE 4. Algorithm parameters.

Name	Parameter	Parameter value
Inertia factor	ω	1
Acceleration constant	C1	1.5
Acceleration constant	C2	1.5
Number of particles	N	150
Particle dimension	D	3
Maximum number of iterations	Maxn	100
Minimum of expansion-contraction factor	$\alpha_{\min}$	0
Maximum of expansion-contraction factor	$\alpha_{\max}$	1
Threshold value	γ	0.01
Constant 1	c1	0.7
Constant 2	c2	0.3

standard deviation values of the path length, angle change and fitness value are obtained. In Table 5, the shortest path length is the minimum path length, and the longest path length is the maximum path length. The experimental results are shown in Tables 5 to 7.

As shown in Table 5, the path length of the IQPSO algorithm is the shortest compared with those of the other algorithms in terms of the minimum, maximum, mean and standard deviation values. In every scenario, the results the IQPSO algorithm are the best. In terms of the mean and standard deviation, in the experimental scenarios, as the scale of the scenarios and the number of obstacles increase, the gap between the average path length value of the IQPSO algorithm and the average lengths of the other algorithms is increasingly large. Moreover, the standard deviation is always the smallest, which proves that the path planning results of the IQPSO algorithm have the smallest degree of dispersion. These data indicate that the IQPSO algorithm has the capability of planning the shortest path.

In Table 6, as the scale of the experimental scenarios and the number of obstacles increase, the mean and standard deviation of the angle change gradually increase. In terms of the angle change values, the results of the IQPSO algorithm are optimal compared with those of the other four algorithms, regardless of the scenario or indicator. These data indicate that the IQPSO algorithm has the ability to minimize the change of the path planning angle.

Table 7 shows the fitness value result of every algorithm. In scenarios 1 and 2, the mean fitness values of all the algorithms are above 0.9. In scenario 3, the mean fitness value of the IQPSO algorithm is more than 0.9, but the mean fitness values of the other algorithms are below 0.9. In each scenario, the IQPSO algorithm has the best mean fitness value. Compared with the other algorithms, the standard deviation of the fitness value of the IQPSO algorithm is the best, except in scenario 2. In scenario 2, the standard deviation of the fitness value of the IQPSO algorithm is only lower than that of the EGA, but the values of the IQPSO algorithm and the EGA are very close. This proves that the algorithm has a better multi-objective optimization capability.

Clearly, the mean and standard deviation are used as indicators to measure these algorithms. The average reflects the central trend of a dataset. The standard deviation can reflect the dispersion degree of a dataset and the robustness. In Tables 5 and 6, the mean and standard deviation of the path length and angle change of the IQPSO algorithm are the smallest. In Table 7, the mean fitness value of the IQPSO algorithm is the largest, and the standard deviation is the smallest, indicating that this algorithm has a strong robustness in different scale scenarios.

TABLE 5. **Comparison of path length results.**

Scenario	Algorithm	Shortest path length/(m)	Longest path length/(m)	Mean path length/(m)	Standard deviation of path length/(m)
Scenario 1	PSO	18.87	20.66	19.33	0.96
	QPSO	18.76	19.98	19.21	0.64
	EGA	17.85	19.89	19.30	0.97
	DENPSO	17.55	19.45	19.23	1.12
	IQPSO	17.41	18.32	17.99	0.45
Scenario 2	PSO	90.21	101.12	96.87	5.83
	QPSO	90.01	101.20	96.05	5.35
	EGA	89.89	99.83	95.98	5.71
	DENPSO	89.33	98.42	95.23	4.48
	IQPSO	89.29	96.62	91.67	2.82
Scenario 3	PSO	189.21	201.71	197.25	4.87
	QPSO	188.33	202.11	196.77	4.82
	EGA	188.03	198.43	194.24	4.01
	DENPSO	188.42	197.56	194.03	3.32
	IQPSO	187.64	194.24	190.43	2.72

TABLE 6. **Comparison of smoothness results.**

Scenario	Algorithm	Minimum of angle change/(°)	Maximum of angle change/(°)	Mean angle change/(°)	Standard deviation of angle change/(°)
Scenario 1	PSO	23.32	35.41	32.64	4.95
	QPSO	21.43	33.82	28.70	4.01
	EGA	20.48	32.51	25.55	4.44
	DENPSO	20.52	33.12	27.25	3.92
	IQPSO	19.21	29.31	24.97	3.12
Scenario 2	PSO	75.84	85.63	81.32	3.29
	QPSO	76.32	84.63	81.99	3.04
	EGA	74.23	88.01	82.34	5.15
	DENPSO	73.21	84.22	80.96	4.45
	IQPSO	69.41	79.72	76.31	2.92
Scenario 3	PSO	145.52	172.11	153.44	7.93
	QPSO	142.33	171.98	150.01	8.06
	EGA	140.26	171.62	147.83	6.59
	DENPSO	139.82	168.91	146.87	6.92
	IQPSO	130.25	163.79	143.25	4.59

Fig. 7 shows the comparison graph of the 3D path planning results of the five algorithms under scenario 1. Among these algorithms, the path planning results from (a) to (e) are those for PSO, QPSO, EGA, DENPSO and IQPSO, respectively. Similarly, Fig. 8 shows the path planning result graph in scenario 2, and Fig. 9 shows the path planning result graph in scenario 3. As seen from the figure, all three algorithms can accomplish the path planning task from the beginning to the end. However, the algorithm results are different. In Fig. 7, the fitness values of PSO, QPSO, EGA, DENPSO and IQPSO are 0.9612, 0.9768, 0.9842, 0.9901 and 0.9943, respectively. In Fig. 8, the fitness values of PSO, QPSO, EGA, DENPSO and IQPSO are 0.9226, 0.9414, 0.9477, 0.9659 and 0.9711, respectively. In Fig. 9, the fitness values of PSO, QPSO, EGA,

TABLE 7. Comparison of fitness value results.

Scenario	Algorithm	Minimum of fitness value	Maximum of fitness value	Mean fitness value	Standard deviation of fitness value
Scenario 1	PSO	0.9817	0.9901	0.9843	0.0021
	QPSO	0.9854	0.9905	0.9884	0.0017
	EGA	0.9798	0.9898	0.9801	0.0013
	DENPSO	0.9899	0.9901	0.9872	0.0011
	IQPSO	0.9913	0.9946	0.9931	0.0009
Scenario 2	PSO	0.9292	0.9612	0.9462	0.0123
	QPSO	0.9398	0.9642	0.9453	0.0089
	EGA	0.9432	0.9582	0.9501	0.0041
	DENPSO	0.9389	0.9794	0.9596	0.0152
	IQPSO	0.9596	0.9721	0.9693	0.0064
Scenario 3	PSO	0.8772	0.8901	0.8803	0.0073
	QPSO	0.8743	0.8962	0.8824	0.0062
	EGA	0.8712	0.8991	0.8835	0.0095
	DENPSO	0.8824	0.9001	0.8943	0.0094
	IQPSO	0.8923	0.9186	0.9015	0.0057

DENPSO and IQPSO are 0.8821, 0.8952, 0.8964, 0.8981 and 0.9067, respectively. Obviously, the IQPSO results are the best. Form the following figures, it can be seen that the paths obtained by the IQPSO algorithm are smoother and shorter those obtained with the other algorithms.

2) RANDOM SCENARIOS WITH THE SAME NUMBER OF OBSTACLES

To further verify the algorithm proposed in this paper, the PSO, DENPSO, traditional QPSO and IQPSO algorithms and the EGA are compared in 10 random experiment scenarios. The size of every scenario is a random number ranging from 10 to 100. The number of obstacles in each experimental scenario is also randomly generated and ranges from 1 to 10. In each scenario, the size and position of the obstacles also are random. Therefore, the number of obstacles may not be the same in the 10 scenarios. In every randomly generated scenario, each algorithm is tested 200 times. Finally, the mean and standard deviation of the path length, angle change and fitness values of each algorithm are obtained. The results are shown in Figs. 10 to 12. The results of the 10 scenarios are sorted by the mean value of the path length (to show the data clearly in the figure).

As shown in Fig. 10(a), as the scenario number increases, the average path length increases gradually. In each scenario, the mean values of the path length of the PSO and QPSO algorithms are close and higher than those of the other three algorithms. The IQPSO algorithm has the smallest average path planning length. In Fig. 10(b), the standard deviation of the path length of the PSO algorithm is the largest in every scenario apart from scenario 2. The standard deviation of the path length of the IQPSO algorithm is the smallest and the best of any scenario.

As shown in Fig. 11(a), in terms of the overall trend, the mean value of the angle change shows an increasing trend. The angle change value is easily affected by the number and position of obstacles. In the case of a large number of obstacles, the AUV needs to bypass the obstacles, which leads to angle change increases. In every scenario, the mean angle change value of the IQPSO algorithm is the smallest. In Fig. 11(b), the standard deviation of the angle change of the IQPSO algorithm is also the smallest. In scenario 7, the gaps between the results of the IQPSO algorithm and those of the other algorithms are the largest.

The goal of the algorithm proposed in this paper is to obtain the maximum fitness value. As shown in Fig. 12(a), the mean fitness value is the largest in each scenario compared with the PSO, QPSO and DENPSO algorithms and the EGA. These data prove that the optimization ability of the IQPSO algorithm is superior to that of the other algorithms. In Fig. 12(b), the standard deviation of the fitness value of the IQPSO algorithm is the smallest, proving that the IQPSO algorithm has better stability than the other algorithms.

Overall, in terms of the path length, angle change and fitness value, the IQPSO algorithm proposed in this paper is superior to the other algorithms in both specified and random environments.

C. ALGORITHMS STABILITY

In this section, by changing the number of particles N and the particle dimension D, the influence on the fitness value and the convergence location of the PSO, traditional QPSO DENPSO and IQPSO algorithms and the EGA is analysed.

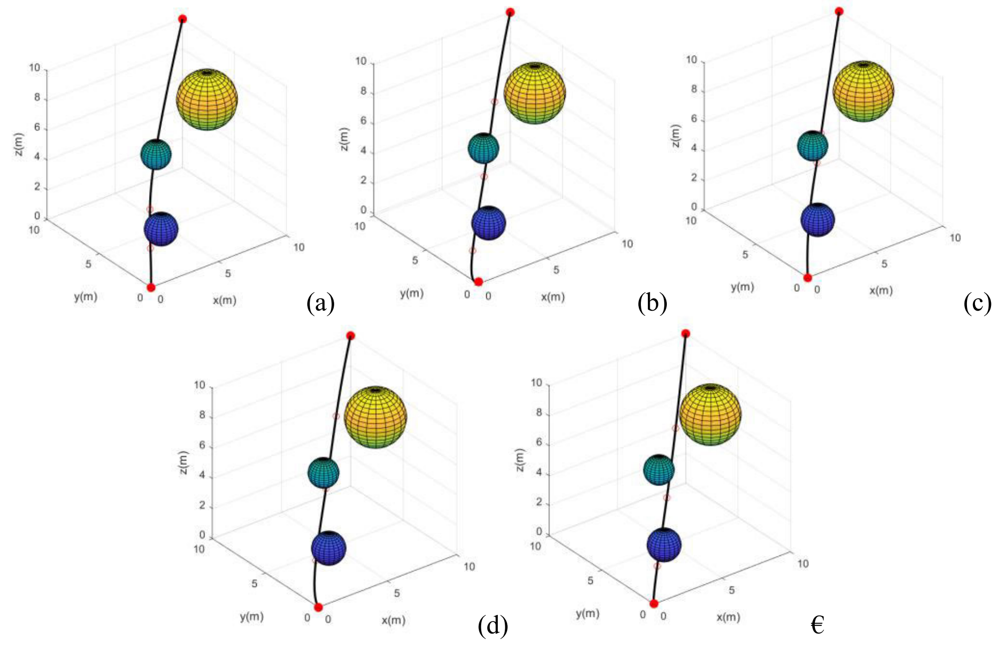

FIGURE 7. **Comparison of the 3D path planning results in scenario 1 for (a) PSO; (b) QPSO; (c) EGA; (d) DENPSO; (e) IQPSO.**

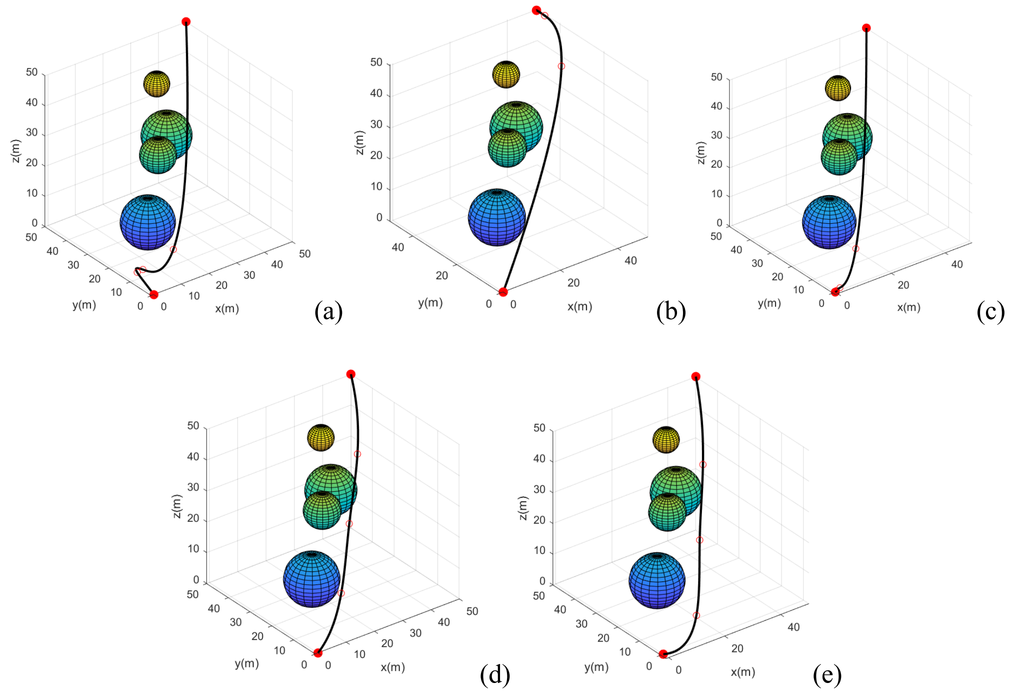

FIGURE 8. **Comparison of the 3D path planning results in scenario 2 for (a) PSO; (b) QPSO; (c) EGA; (d) DENPSO; (e) IQPSO.**

In this instance, the convergence location is the number of iterations when the algorithm begins to converge. When the number of particles is changed, the particle dimension is 3 and remains unchanged. When the dimensions of the particles are changed, the number of particles is 150, and this number remains the same. The number of particles are increased from 100 to 200 with an interval of 25. The particle dimensions are increased from 3 to 7 with an interval of 1. The experimental environment of this section is that of scenario 2 of Fig 6.

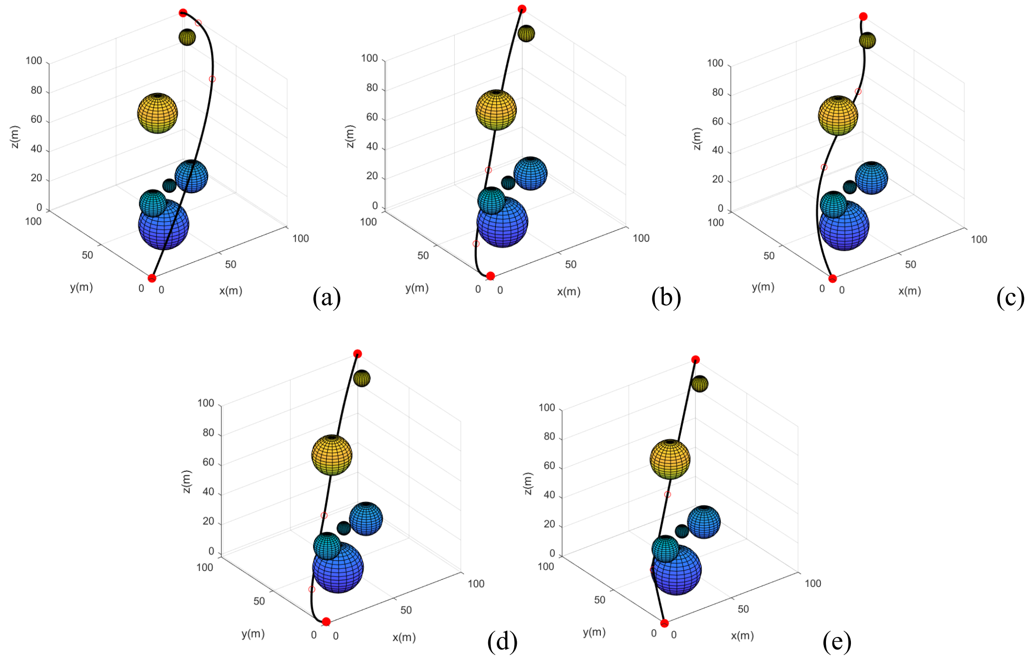

FIGURE 9. Comparison of 3D path planning results in scenario 3 for (a) PSO; (b) QPSO; (c) EGA; (d) DENPSO; ∈ IQPSO.

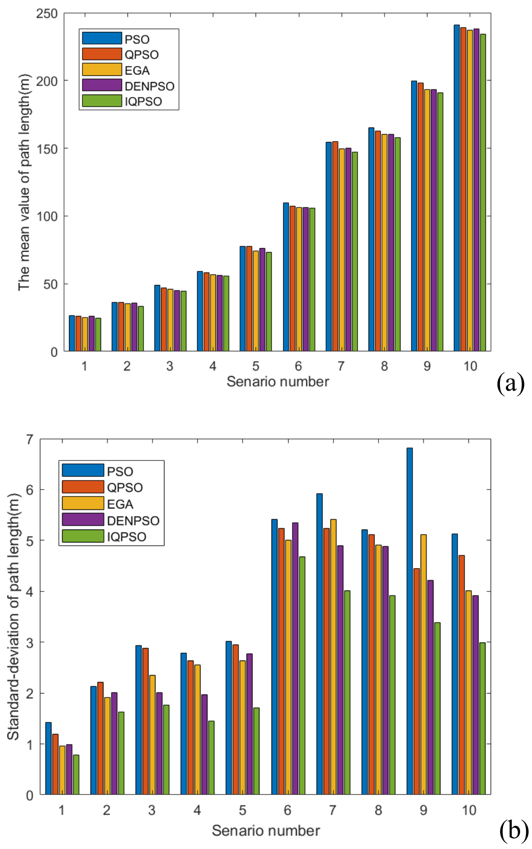

FIGURE 10. Comparison results: (a) mean value and (b) standard deviation of the path length in the 10 scenarios.

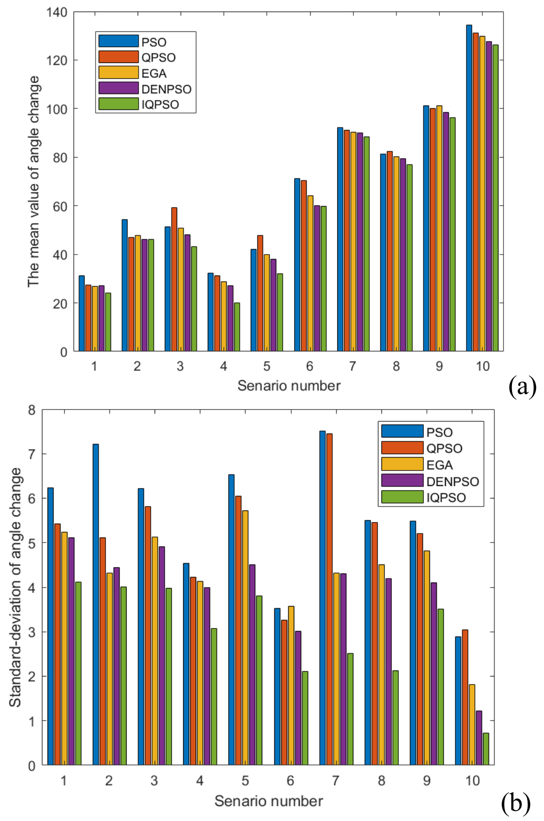

FIGURE 11. Comparison results: (a) mean value and (b) standard deviation of the angle change in the 10 scenarios.

1) CHANGING THE NUMBER OF PARTICLES

As shown in Fig. 13(a), as the number of particles increases, the fitness values of the PSO, traditional QPSO, DENPSO and IQPSO algorithms and the EGA increase gradually. The fitness values of these algorithms are all greater than 0.9. As the number of particles increases, the fitness values of the QPSO, IQPSO, and DENPSO algorithms and the EGA all

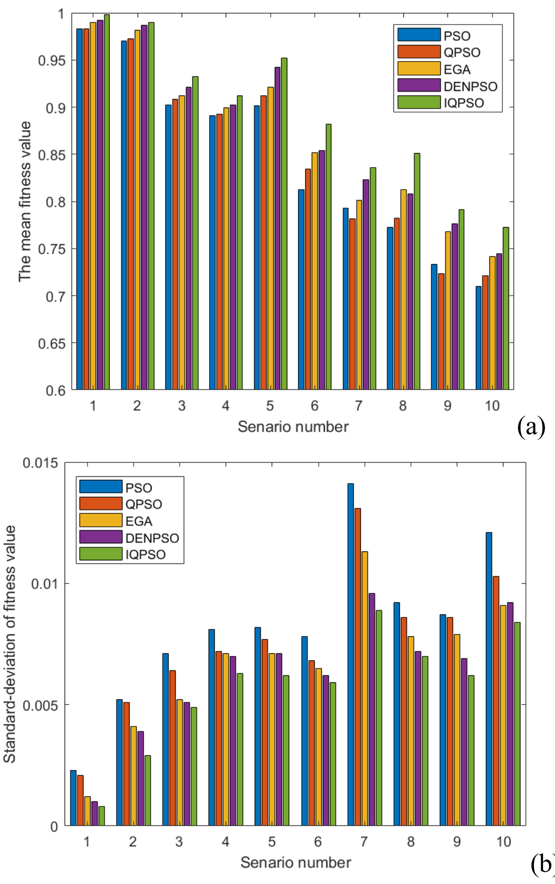

FIGURE 12. Comparison results: (a) mean value and (b) standard deviation of the fitness value in the 10 scenarios.

are smaller than that of the QPSO. In Fig. 13(b), the convergence locations of the PSO, traditional QPSO, DENPSO and IQPSO algorithms and the EGA decrease when the number of particles is in the range of [100,150) and increase when the number of particles is in the range of [150,200]. At the beginning, as the number of particles increases, the optimization ability of the algorithm becomes stronger, and the algorithm converges more easily. However, as the number of particles increases later, the complexity of the algorithm increases, which affects the convergence of the algorithm. In general, the convergence location of the IQPSO algorithm is better than that of the other algorithms at every stage.

2) CHANGING THE PARTICLE DIMENSION

In Fig. 14(a), as the particle dimension increases, the fitness value of the IQPSO algorithm increases gradually. Compared with the PSO, QPSO, EGA and DENPSO, the fitness value of the IQPSO algorithm is improved by approximately 0.0211, 0.111, 0.01 and 0.0094, respectively, on average. In Fig. 14(b), the convergence location of all the algorithms decreases as the particle dimension changes from 3 to 4 and increases when the particle dimension changes from 4 to 7. Appropriately increasing the particle dimension is conducive to algorithm convergence. However, as the particle dimension increases, the complexity of the algorithm increases, which is not conducive to algorithm convergence. However, the IQPSO algorithm has a better convergence location than the other algorithms at every stage. These data demonstrate that, as the particle dimension changes, the performance of the IQPSO algorithm is better.

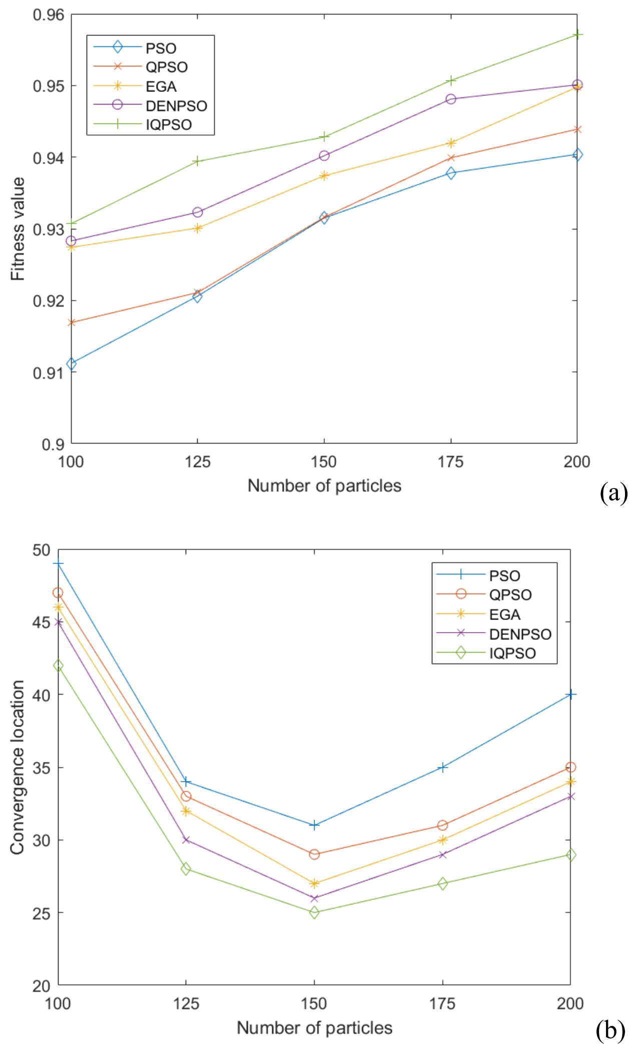

FIGURE 13. Effect of the number of particles on (a) the fitness value and (b) the convergence location of the algorithms.

Fig. 15 shows the change curve of the fitness value as the number of iterations of the algorithms increases. This result is a randomly selected set of experiments in which the number and dimension of particles are 150 and 3, respectively. As seen from Fig. 15, the convergence locations of the PSO, traditional QPSO, DENPSO and IQPSO algorithms and the EGA are 29, 51, 28, 15 and 26, respectively, and the fitness values after convergence of the three algorithms are 0.9475 (PSO), 0.9434 (QPSO), 0.9616 (EGA), 0.9632 (DENPSO) and 0.9701 (IQPSO). Thus, compared with the other algorithms, the IQPSO algorithm has a higher fitness value and convergence speed. When the number of iterations is 10, the IQPSO algorithm becomes trapped in a local optimum.

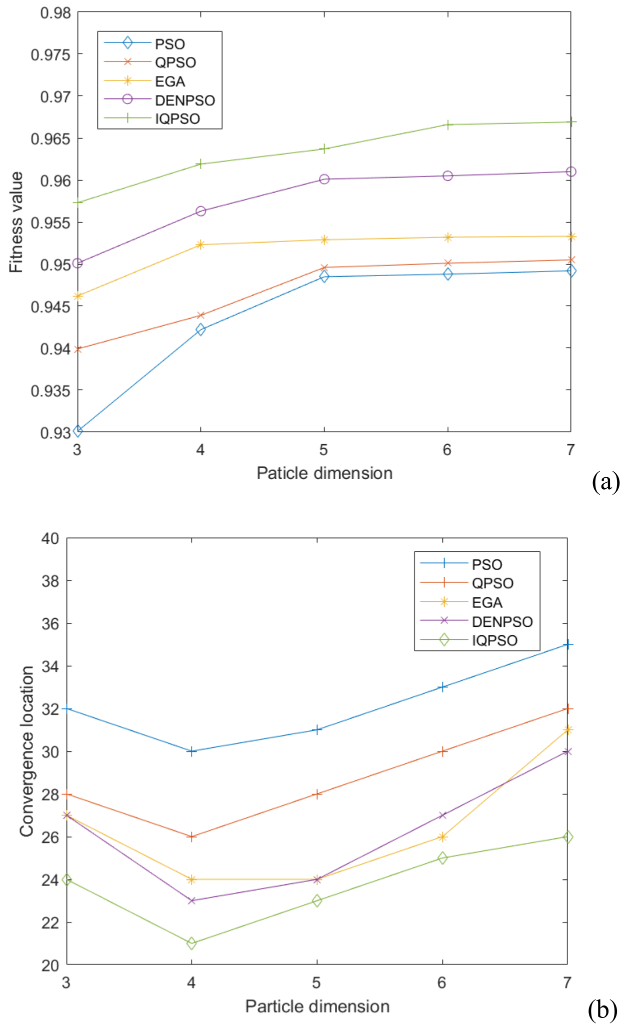

FIGURE 14. Effect of the particle dimension on (a) the fitness value and (b) the convergence location of the algorithms.

However, it can quickly jump out of the local optimum and reach a higher fitness value. The results show that compared with the other four algorithms, the IQPSO algorithm has a faster convergence speed and better solution results.

V. CONCLUSION

To address the problem of AUV path planning, this paper analyses the motion model of an AUV under water and proposes a multi-objective path planning optimization method that considers the path safety, path length and angle change on the premise of spherical modelling of obstacles. The method proposed in this paper does not require mesh modelling of the 3D environment, which saves time, reduces the calculation cost and enables greater flexibility. Aiming at the slow convergence speed of the traditional optimization algorithm, this paper strongly improves the convergence and optimization ability of the QPSO algorithm. The simulation results show that under different experimental environments, the IQPSO algorithm has a stronger path planning ability and a higher stability than the PSO, DENPSO and QPSO algorithms and the EGA and can converge to the optimal solution faster. Since the AUV usually plans the path in advance before executing the task, there is no need to execute the algorithm when tracking the trajectory. At the same time, the algorithm is not suitable for avoiding moving obstacles. Therefore, dynamic real-time path planning for AUVs will be a research direction in the future.

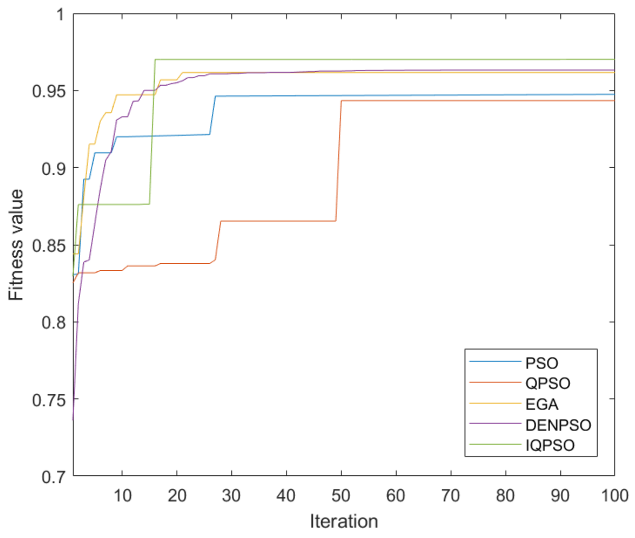

FIGURE 15. Fitness value change diagram.

REFERENCES

[1] M. Kwiesielewicz, W. Piotrowski, R. Smierzchalski, and R. Sutton, "AUV path planning for navigational constrains by evolutionary computation," *IFAC Proc. Volumes*, vol. 33, no. 21, pp. 313–316, Aug. 2000.

[2] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technol.*, vol. 15, no. 4, pp. 582–606, Aug. 2019.

[3] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Comput. Commun.*, vol. 149, pp. 270–299, Jan. 2020.

[4] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[5] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symp. (ICRA)*, San Francisco, CA, USA, Apr. 2000, pp. 995–1001.

[6] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, USA, May 1994, pp. 3310–3317.

[7] S. Shao, Y. Peng, C. He, and Y. Du, "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization," *ISA Trans.*, vol. 97, pp. 415–430, Feb. 2020.

[8] C. Xiong, D. Chen, D. Lu, Z. Zeng, and L. Lian, "Path planning of multiple autonomous marine vehicles for adaptive sampling using Voronoi-based ant colony optimization," *Robot. Auto. Syst.*, vol. 115, pp. 90–103, May 2019.

[9] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man Cybern., B (Cybern.)*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[10] J. Chen, F. Ye, and T. Jiang, "Path planning under obstacle-avoidance constraints based on ant colony optimization algorithm," in *Proc. IEEE 17th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2017, pp. 1434–1438.

[11] F. H. Ajeil, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, "Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments," *Sensors*, vol. 20, no. 7, p. 1880, Mar. 2020.

[12] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Louis, MO, USA, Mar. 1985, pp. 500–505.

[13] Z. Zhou, J. Wang, Z. Zhu, D. Yang, and J. Wu, "Tangent navigated robot path planning strategy using particle swarm optimized artificial potential field," *Optik*, vol. 158, pp. 639–651, Apr. 2018.

[14] X. Liu, X. Du, X. Zhang, Q. Zhu, and M. Guizani, "Evolution-algorithm-based unmanned aerial vehicles path planning in complex environment," *Comput. Electr. Eng.*, vol. 80, Dec. 2019, Art. no. 106493.

[15] W. M. Hirsch and G. B. Dantzig, "The fixed charge problem," *Nav. Res. Logistics Quart.*, vol. 15, no. 3, pp. 413–424, Sep. 1968.

[16] M. Elhoseny, A. Tharwat, and A. E. Hassanien, "Bezier curve based path planning in a dynamic field using modified genetic algorithm," *J. Comput. Sci.*, vol. 25, pp. 339–350, Mar. 2018.

[17] S. MahmoudZadeh, A. M. Yazdani, K. Sammut, and D. M. W. Powers, "Online path planning for AUV rendezvous in dynamic cluttered undersea environment using evolutionary algorithms," *Appl. Soft Comput.*, vol. 70, pp. 929–945, Sep. 2018.

[18] S. Chowdhury, M. Marufuzzaman, H. Tunc, L. Bian, and W. A. Bullington, "Modified Ant Colony Optimization algorithm to solve a dynamic traveling salesman problem: A case study with drones for wildlife surveillance," *J. Comput. Des. Eng.*, vol. 6, no. 3, pp. 368–386, Jul. 2019.

[19] D. Ramtake, S. Kumar, and V. K. Patle, "Route optimisation byant colony optimisationtechnique," *Procedia Comput. Sci.*, vol. 92, pp. 48–55, 2016.

[20] M. N. Zafar and J. C. Mohanta, "Methodology for path planning and optimization of mobile robots: A review," *Procedia Comput. Sci.*, vol. 133, pp. 141–152, 2018.

[21] H. Duan, Y. Yu, X. Zhang, and S. Shao, "Three-dimension path planning for UCAV using hybrid meta-heuristic ACO-DE algorithm," *Simul. Model. Pract. Theory*, vol. 18, no. 8, pp. 1104–1115, Sep. 2010.

[22] M. Nazarahari, E. Khanmirza, and S. Doostie, "Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm," *Expert Syst. Appl.*, vol. 115, pp. 106–120, Jan. 2019.

[23] D. R. Parhi and S. Kundu, "Navigational control of underwater mobile robot using dynamic differential evolution approach," *Proc. Inst. Mech. Eng., M, J. Eng. Maritime Environ.*, vol. 231, no. 1, pp. 284–301, Feb. 2017.

[24] J. Wu, C. Song, C. Fan, A. Hawbani, L. Zhao, and X. Sun, "DENPSO: A distance evolution nonlinear PSO algorithm for energy-efficient path planning in 3D UASNs," *IEEE Access*, vol. 7, pp. 105514–105530, 2019.

[25] J. Han, "An efficient approach to 3D path planning," *Inf. Sci.*, vol. 478, pp. 318–330, Apr. 2019.

[26] C. Lucas, D. Hernández-Sosa, D. Greiner, A. Zamuda, and R. Caldeira, "An approach to multi-objective path planning optimization for underwater gliders," *Sensors*, vol. 19, no. 24, p. 5506, Dec. 2019.

[27] C. Lin, H. Wang, J. Yuan, D. Yu, and C. Li, "An improved recurrent neural network for unmanned underwater vehicle online obstacle avoidance," *Ocean Eng.*, vol. 189, Oct. 2019, Art. no. 106327.

[28] A. G. D. S. Silva Junior, D. H. D. Santos, A. P. F. D. Negreiros, J. M. V. B. D. S. Silva, and L. M. G. Gonçalves, "High-level path planning for an autonomous sailboat robot using Q-Learning," *Sensors*, vol. 20, no. 6, p. 1550, Mar. 2020.

[29] X. Cao, H. Sun, and G. E. Jan, "Multi-AUV cooperative target search and tracking in unknown underwater environment," *Ocean Eng.*, vol. 150, pp. 1–11, Feb. 2018.

[30] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, Oct. 1995, pp. 39–43.

[31] J. Sun, W. Xu, and B. Feng, "A global search strategy of quantum-behaved particle swarm optimization," in *Proc. IEEE Conf. Cybern. Intell. Syst.*, Singapore, Dec. 2004, pp. 111–116.

[32] T. K. Truong, L. J. Wang, I. S. Reed, and W. S. Hsieh, "Image data compression using cubic convolution spline interpolation," *IEEE Trans. Image Process.*, vol. 9, no. 11, pp. 1988–1995, Nov. 2000.

[33] H. Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 6, pp. 508–517, Dec. 1978.

LEI WANG received the Ph.D. degree in control theory and engineering from the Dalian University of Technology, China, in 2012. He is currently an Associate Professor with Henan Polytechnic University. His research interests include wireless ad hoc networks, embedded systems, networked control systems, and the Internet of Things.

LILI LIU received the B.S. degree from Yuncheng University, in 2018. She is currently pursuing the master's degree with Henan Polytechnic University. Her current research interests include underwater location optimization and path planning optimization.

JUNYAN QI received the M.S. degree in computer science from the Dalian University of Technology, China, in 2007. Since 2000, she has been with Henan Polytechnic University, China, where she is currently an Associate Professor. Her research interests include wireless data mining, distributed computing, and future Internet.

WEIPING PENG received the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications, China, in 2011. He is currently an Associate Professor with Henan Polytechnic University. His research interests include information security, the security and application of the Internet of Things, and mobile edge computing.