IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 September 2023, accepted 26 September 2023, date of publication 2 October 2023,
date of current version 11 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3321104

RESEARCH ARTICLE

Three-Dimensional Marine Ranching Cage Inspection Path Planning Integrating the Differential Evolution and Particle Swarm Optimization Algorithms

YIJUN HU[1], JING WANG[2], HAOJIN HE[1], YIQIANG ZHANG[3], SHUO CAI[1], ANLU XIE[1], AND ZIJUN ZHENG[1]

[1]School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China
[2]Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China (UESTC), Shenzhen 518110, China
[3]College of Computer Science and Technology, Jilin University, Changchun 130012, China

Corresponding author: Jing Wang (wj8538@163.com)

ABSTRACT This study addressed the autonomous planning of three-dimensional (3D) underwater inspection paths for autonomous underwater vehicles (AUVs) in marine ranching by integrating differential evolution and particle swarm optimization (PSO) algorithms. First, a modified PSO algorithm incorporating swap operators, mutation, and crossover strategies was employed to enable autonomous obstacle avoidance during the inspection of offshore net cages in fish farms situated within a 3D marine environment. This approach addresses the problem of planning full-traversal paths for multiple inspection points. Second, the performance of the proposed algorithm was assessed through comparative tests with other algorithms. The proposed algorithm demonstrated significant improvements in convergence speed, accuracy, and stability under complex scenarios involving multiple optima and intense oscillations. To validate the superiority and overall planning proficiency of the modified method, an experimental setup comprising of two distinct 3D marine cage environments with a series of checkpoints was utilized. The experimental results demonstrated the ability of the proposed algorithm to generate an optimal path while traversing all inspection points of fish farm offshore net cages. By ensuring the safety of AUVs and closely adhering to the surfaces of offshore net cages during the inspection process, the algorithm exhibits remarkable adaptability to specific application scenarios, effectively mitigating concerns related to local optima and premature convergence.

INDEX TERMS Differential evolution, marine ranching, particle swarm optimization, path planning, three-dimensional space.

I. INTRODUCTION

As terrestrial resources become increasingly scarce, the utilization and development of marine resources have become prominent research targets worldwide. Ocean ranching is an environmentally friendly and innovative technique that provides a new direction for the marine aquaculture industry.

Currently, multiple challenges exist in the construction of intelligent marine ranches. Notably, the maintenance of marine net cages and acquisition of information regarding fish population growth are significant hurdles. Traditional underwater inspection operations rely on manual inspections, which suffer from low efficiency, high costs, and concerns regarding driver safety. Therefore, researchers have proposed the use of remotely operated vehicles (ROVs) [1], [2] and omnidirectional surface vehicles (OSVs) [3] as alternatives to manual underwater inspection. Nevertheless, the utilization of ROVs commonly necessitates the deployment of a substantial OSV outfitted with sophisticated tether management systems and dynamic positioning capabilities, making

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

ROV operations arduous and financially burdensome [4]. Moreover, the operational scope of OSVs is limited to water surfaces, which further limits their potential applications [5]. Compared to manned vessels, floating platforms, and other exploratory underwater vehicles, autonomous underwater vehicles (AUVs) have the advantages of high maneuverability and advanced intelligence, making them more suitable for underwater inspection tasks on marine farms [6]. These tasks include cage damage inspection and autonomous monitoring of fish growth. In the studies described in [7] and [8], AUVs were employed to inspect marine farms, thereby enhancing the efficiency of routine examinations conducted in aquaculture cages and the organisms within them and highlighting the crucial role of AUV inspection in marine ranching.

Path planning aims to determine the optimal path between the start and end points by satisfying a series of constraints while avoiding potential threat obstacles. In the inspection path planning of marine ranches, while ensuring the safety of the AUV itself, the inspection path must be ensured to fit the surface of the cage to monitor the cage and the organisms inside. To enable AUVs to navigate, assess, and avoid threats autonomously over long distances and to improve their survivability and adaptability, efficient and intelligent optimization algorithms are required to solve the three-dimensional (3D) underwater path planning problem. The evaluation criterion for the algorithm involves determining the shortest path that visits all the predefined points in a traversal. In recent years, many intelligent algorithms have been widely applied to underwater path planning for AUVs. These methods include the ant colony optimization (ACO) algorithm [9], [10], [11], [12], tuna algorithm [13], whale algorithm [14], [15], [16], grey wolf optimization algorithm [17], [18], artificial jellyfish search algorithm [19], water wave optimization algorithm [20], genetic algorithm (GA) [21], [22], and other methods [23].

Particle swarm optimization (PSO) effectively balances global and local search capabilities and is an intelligent and efficient swarm intelligence algorithm that effectively balances global and local search capabilities. It has a simple algorithm structure, high search precision, fast convergence speed, and no strict requirements for differentiability, derivability, and continuity of the optimization function. Therefore, it demonstrates prominent adaptability in solving complex underwater 3D path planning problems. Thus far, many scholars have made improvements to the PSO algorithm to cater to the path-planning needs of AUVs in complex underwater environments. Wang et al. [24] merged the PSO algorithm with an adaptive step-size cuckoo search algorithm. The enhanced algorithm empowers AUVs with efficient obstacle avoidance and path-planning capabilities in 3D environments. Sui et al. [25] proposed a two-layer hybrid algorithm that combines the ACO, PSO, and A* algorithm for 3D path planning. Experimental results have shown that this algorithm can generate optimal collision-free paths with short lengths and high safety. Zhang and Shi. [26] proposed an algorithm based on a deep Q-network and quantum particle swarm optimization that exhibits superior performance compared with traditional methods in 3D path planning. Li and Yu [27] proposed an improved compressed factor particle swarm optimization algorithm that achieved better planning efficiency and path quality and shorter planning time in 3D path planning tasks. The aforementioned methods can plan safe paths to avoid obstacles; however, they are not suitable for AUV inspection path planning close to the cage in marine ranch application scenarios, and they still face challenges such as local optimum trapping and premature convergence when solving 3D problems in complex scenarios.

Considering the requirements of intelligent marine ranching for net cage inspection, this article proposes a modified fusion algorithm based on the PSO and differential evolution (DE) algorithms. The algorithm utilizes the crossover strategy of the DE algorithm to help escape from local optima while optimizing using the PSO algorithm based on the exchange operator and adaptive weights. The primary objective is to make the AUV close to the cage during the inspection of the predetermined point of the fish cage and achieve shortest-path planning and intelligent obstacle avoidance, thereby facilitating low-power autonomous underwater inspection. Through simulation experiments, we compared the proposed algorithm with two improved PSO algorithms and found that the DE-PSO algorithm exhibited better convergence stability, effectively avoiding local optima and premature convergence. In addition, the developed algorithm can plan a shorter path, which will help AUVs realize the inspection of cages in marine ranching application scenarios, reduce energy consumption, and improve endurance.

The remainder of this paper is organized as follows. Section II presents a detailed description of the specific application background and a problem statement for the experimental research. Section III provides the essential preparatory knowledge for problem solving, including the basic models of the PSO and DE algorithm, and mathematical foundations of swap operators and sequences. Section IV provides the design of the DE-PSO algorithm, and presents the benchmark function test results of the DE-PSO algorithm along with the underwater 3D map construction and obstacle avoidance strategy. Section V describes the simulation results based on real-world applications and problem-solving scenarios. Finally, Section VI summarizes the achievements of this study and outlines potential future research directions and plans based on the test results and simulation experiments.

II. PROBLEM DESCRIPTION

In the context of autonomous net cage inspection in marine ranching, this article presents a novel optimization approach for AUV path planning that seeks to determine the optimal path for a vehicle during underwater operations.

This study was focused primarily on the application scenario of underwater submersible path planning. As shown in Fig. 1, marine ranching net cages are spherical fish-farming enclosures of various sizes and depths. To avoid collisions with the fish cage surface, AUVs were monitored at a certain

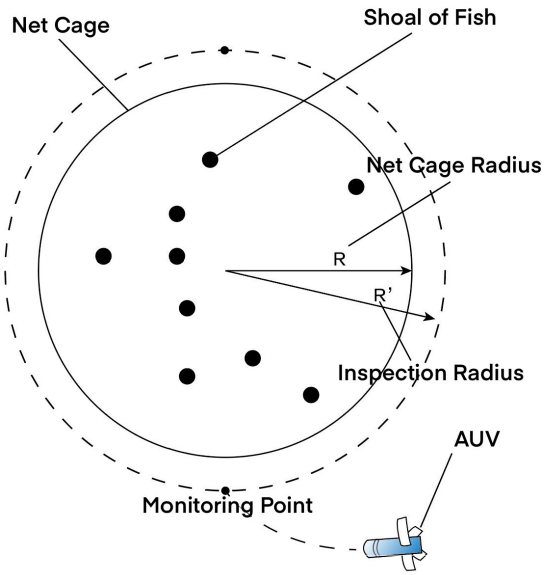


FIGURE 1. Illustration of fish cage monitoring point settings.

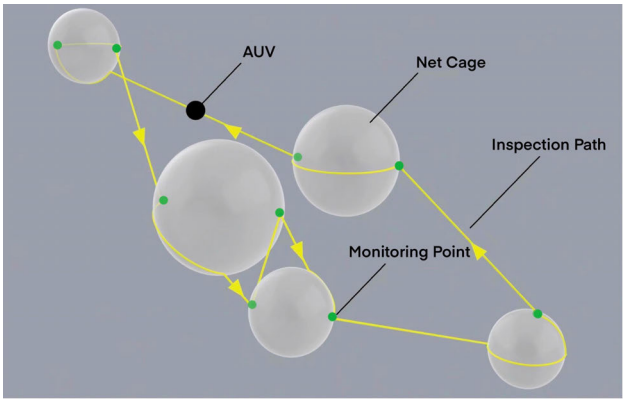


FIGURE 2. Schematic diagram of underwater AUV inspection route.

distance from the outer boundary of the cage sphere. For each fish farm cage, two monitoring points were established with the following specifications: the fish cage sphere had a radius of R and the monitoring sphere had a radius slightly greater than R. A monitoring point was randomly selected on the center-point plane of the monitoring sphere. The symmetric point of this selected point with respect to the center of the fish cage sphere was designated as the second monitoring point. As AUVs cannot pass through fish cages, fish cages can be treated as spherical obstacles when planning inspection paths. While traversing each monitoring point, the AUV avoided static obstacles around the fish cages to ensure operational safety. As shown in Fig. 2, the AUV was treated as a point mass. To obtain information from the farm fish cages efficiently and expeditiously, we employed a path-planning algorithm in this study to determine the shortest path that traverses all monitoring points. Simultaneously, an AUV must avoid static obstacles during inspection. After traversing all monitoring points, the vehicle returned to its starting point to collect and provide feedback on the fish cage information.

The following sections expand on the principles of the PSO and DE algorithms. Additionally, a 3D obstacle map constructed for path planning simulation experiments is presented. Based on the experimental results, the fitness of the DE-PSO algorithm in the aforementioned application scenario is discussed.

III. PRELIMINARY KNOWLEDGE

A. PSO ALGORITHM THEORETICAL FOUNDATION

1) FUNDAMENTAL PSO MODEL

The PSO algorithm is based on swarm intelligence that migrates individuals within a swarm to the optimal solution region according to their fitness in the environment.

In the fundamental PSO algorithm, because each particle consistently tracks the current optimal particle, particle x_i updates its velocity and position using the following formulas:

$$v_{id}(k+1) = v_{id}(k) + c_1 \cdot r_1 \left(pbest_i(k) - x_{id}(k)\right) + c_2 \cdot r_2 \left(gbest_g(k) - x_{id}(k)\right) \quad (1)$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1), \quad (2)$$

where $v_i(k)$ represents the velocity of particle i, $x_i(k)$ represents the position of particle i, $pbest_i(k)$ represents the current optimal position of an individual particle, and $gbest_i(k)$ represents the current optimal position of the entire swarm. r_1 and r_2 are random numbers between 0 and 1 that follow a uniform distribution. c_1 and c_2 are the acceleration coefficients (or learning factors), typically set between 0 and 2, representing the extent of trust of a particle in individual cognition and social knowledge, respectively, where $d = 1, 2, \cdots, n$ represents the dimension of the problem being solved.

2) LINEARLY DECREASING INERTIA WEIGHT (LDIW) PSO

The inertial weight w describes the influence of the previous-generation velocity of a particle on its current velocity. The global and local search capabilities of the algorithm can be controlled by adjusting the value of w. To make the algorithm more adaptable to path planning requirements, an attempt was made to change the inertia weight w dynamically, leading to the proposal of LDIW-PSO [28]. In this algorithm, the value of w is updated using the following equation:

$$w(k) = \frac{(T_{\max} - T)}{T_{\max}} \cdot (w_{ini} - w_{end}) + w_{end}, \quad (3)$$

where w_{ini} represents the initial value and w_{end} corresponds to the value of the maximum iteration (generation) for the inertia weight. $w_{ini} > w_{end}$ and T denote the current and maximum iteration counts, respectively.

3) LINEARLY CHAOTIC PARTICLE SWARM OPTIMIZATION (LCPSO)

The fundamental concept underlying the chaotic inertia weight entails the use of a chaotic mapping technique to determine the inertia weight coefficient [29]. The logistic

mapping formula is as follows:

$$u = \rho \cdot u \cdot (1 - u). \tag{4}$$

u was initialized to a random value of (0, 1). When $3.57 < \rho \leq 4$, the logistic map appears chaotic.

In light of the introduction of inertia weight in the PSO algorithm, Nkwanyana and Wang [30] devised the LCPSO algorithm, which utilizes both a linearly decreasing inertia weight and chaotic inertia weight, with the aim of further refining the convergence performance of the algorithm. The expressions of linear decreasing inertia weight w^l [31] and chaotic inertia weight w^c [29] are as follows:

$$w^l = w_{\max} - \frac{\Delta w}{t_{\max}} \cdot t \tag{5}$$

$$w^c = \Delta w \cdot \frac{\Delta t}{t_{max}} + w_{min} \cdot u \tag{6}$$

$$\Delta t = t_{\max} - t \tag{7}$$

$$\Delta w = w_{\max} - w_{\min}, \tag{8}$$

where w_{max} and w_{min} are the maximum and minimum values of the inertial weight, respectively, and t_{max} and t are the maximum and current numbers of iterations, respectively.

Let ζ represent the parameter that governs the proportion of weights. The fusion of two distinct weights into w^ζ is accomplished through the utilization of the following equation:

$$\begin{aligned} w^\zeta &= \zeta \cdot (w^l + w^c) \\ &= \zeta \cdot [(w_{\max} - \frac{\Delta w}{t_{\max}} \cdot t) + (\Delta w \cdot \frac{\Delta t}{t_{\max}} + w_{\min} \cdot u)] \\ &= \zeta \cdot [\frac{t_{\max} \cdot (2w_{\max} - w_{\min} \cdot u) - \Delta w \cdot \Delta t - t \cdot \Delta w}{t_{\max}}]. \end{aligned} \tag{9}$$

Then, the velocity update formula of the particle becomes as follows:

$$\begin{aligned} v_{id}(k+1) = w^\zeta \cdot v_{id}(k) \\ + c_1 \cdot r_1(pbest_i(k) - x_{id}(k)) \\ + c_2 \cdot r_2 \cdot (gbest_g(k) - x_{id}(k)). \end{aligned} \tag{10}$$

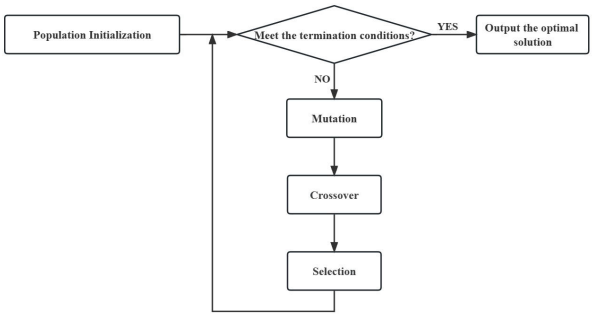
B. PSO ALGORITHM BASED ON SWAP OPERATORS AND SWAP SEQUENCE

The fundamental PSO and PSO algorithms that utilize swap operators and sequences to solve the multi-objective point-to-point shortest path problem exhibit significant differences in terms of particle encoding, particle update rules, and objective function computation methods.

The fundamental PSO algorithm is commonly applied to continuous optimization problems, whereas the PSO algorithm that utilizes swap operators and sequences is primarily used to solve discrete optimization problems.

The specific operations of the modified PSO algorithm incorporating swap operators [32] are as follows:

$$\alpha = p_{id}\,(t) - x_{id}\,(t) \tag{11}$$




```
Input :
particlesize = particle population size
i = an individual particle within the population
a, b = two different random particle individuals
c = randomly generated decimals between 0 and 1
CR = crossover rate
F = scaling factor
for i = 1 → particlesizes do
    while a! = i do
        a ← round(rand()*(particlesize-1)+1)
    end
    while b! = a and b! = i do
        b ← round(rand()*(particlesize-1)+1)
    end
    M_i ← f_i + F * (f_a + f_b)
    if c < CR then
        f_i ← M_i
    end
end
```

FIGURE 3. Flowchart and pseudocode of DE algorithm.

$$\beta = p_{gd}\,(t) - x_{id}\,(t), \tag{12}$$

where α represents the difference between the individual best positions $p_{id}\,(t)$ and $x_{id}\,(t)$, and β represents the difference between the global best positions $p_{gd}(t)$ and$x_{id}\,(t)$. Both α and β are basic swap sequences. Moreover, update the weight coefficient ω through the following formula:

$$\omega^* = \omega - \frac{\omega - 0.01}{\text{Particle size}}. \tag{13}$$

C. THEORETICAL FOUNDATIONS OF DE ALGORITHM

The DE algorithm is based on the GA and follows the fundamental idea of randomly selecting a target vector and two reference vectors from a swarm. The algorithm generates new offspring mutation vectors by utilizing the differences between the target and reference vectors and employing various differential strategies. New test vectors were obtained by crossing these mutation vectors with reference vectors with a certain probability. The fitness of the target and test vectors were compared, and the individual with the highest fitness was selected as part of the next-generation population for the iteration. This process was continued until the termination conditions were satisfied.

The flowchart and pseudocode of the DE algorithm are shown in Fig. 3.

1) POPULATION INITIALIZATION

Initially, NP individuals with D dimensions are randomly generated as the 0th generation population. Each vector is a feasible solution within the search range. The individual vectors can be generated as follows [33]:

$$x_{i,j}\,(k) = x_j^l + rand\,(0, 1) * \left(x_j^h - x_j^l\right), \tag{14}$$

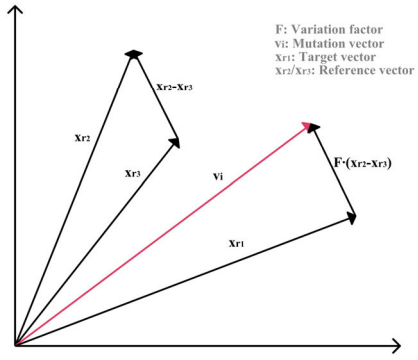


FIGURE 4. Illustration of mutation operation.

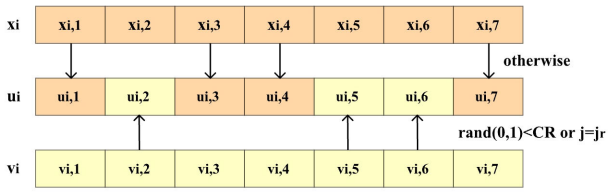


FIGURE 5. Schematic diagram of binomial crossover.

where i represents the individual vector i in the swarm, j represents the jth dimension component of the individual vector, x_j^l is the lower bound of the jth dimension component, and x_j^h is the upper bound of the jth dimension component.

2) MUTATION OPERATION

We selected the ith individual vector $x_i(k)$ in the swarm as the target vector. The mutation operation for the target vector was performed using the $DE/best/1/bin$.

$DE/best/1/bin$ is a widely used differential strategy. Taking the binomial crossover as an example, the mutation vector $v_i(k)$ of generation k can be represented as [34]

$$v_i(k) = x_{best} + F(x_{r1}(k) - x_{r2}(k)), \tag{15}$$

where F is the mutation factor. The process of generating vector $v_i(k)$ is illustrated in Fig. 4.

3) CROSSOVER OPERATION

To increase the swarm diversity, the DE algorithm introduces a discrete crossover operator. The mutation and target vectors are combined using a crossover operator to generate a new test vector, $u_i(k)$. Commonly used crossover operators include the binomial, exponential, and orthogonal operators [33].

Fig. 5 illustrates the binomial crossover operation.

4) SELECTION OPERATION

The fitness function f(x) is defined as follows:

$$f(x) = \frac{2}{\text{distance}(x)}, \tag{16}$$

where distance(x) represents the distance of the path represented by vector x, which is calculated as the Euclidean distance between the nodes. The fitness function takes the reciprocal of the distance and transforms the problem into a PSO that seeks the minimum.

DE conducts selection operations employing the subsequent equation:

$$x_i(k+1) = \begin{cases} u_i(k), & f(u_i(k)) \le f(x_i(k)) \\ x_i(k), & otherwise. \end{cases} \tag{17}$$

The selection operator in DE retains the best individual between the target vector and its corresponding test vector, ensuring that the fitness value of the offspring is always superior to that of the parent. This mechanism drives the swarm to evolve continuously towards the optimal solution, gradually converging towards the position of the global or satisfactory solution.

IV. ALGORITHM DESIGN

A. DE-PSO ALGORITHM

1) BASIC IDEA

As a swarm-intelligence search method, the PSO algorithm is highly sensitive to the quality of the initial swarm. When the initial swarm exhibits significant diversity, the PSO algorithm is prone to becoming stuck in local optima, converging prematurely, and failing to reach the global optimal solution.

To improve the solving efficiency, convergence rate, precision, and stability of the PSO algorithm, this paper proposes a novel approach called DE-PSO, which integrates the DE approach into the PSO algorithm. In this integration process, the scaling factor F and crossover probability CR are introduced along with the mutation, crossover, and selection operations. Section IV presents an experimental analysis to investigate the effects of the two introduced parameters F and CR on the performance of the algorithm. The DE-PSO algorithm generates a mutated particle using the position difference between any two distinct particles in the current swarm. It probabilistically selects whether the mutated particle should replace the original particle to participate in the next generation of evolution. By combining the strategies and characteristics of both algorithms, the DE-PSO algorithm effectively enhances swarm diversity, enabling it to escape local optima and avoid premature convergence while demonstrating superior stability and robustness when solving optimization problems.

2) DE-PSO ALGORITHM DESIGN

The DE-PSO algorithm follows a fundamental framework similar to that of the conventional PSO algorithms. However, the key difference lies in the iterative update process of the particles, in which the idea of a mutation operation from the DE approach is incorporated to improve the local search strategy. In addition, a mutation perturbation strategy is introduced to perturb each particle in the current swarm with a certain probability.

The specific implementation process of the integrated algorithm is as follows:

Step 1: Initialization of particle swarm and parameters.

In the search space, set swarm size M, randomly generate initial solutions $X = \{X_1, X_2, \cdots, X_M\}$, and set the initial velocity $v_i(k)$ and initial position $x_i(k)$ for each particle, where k represents the current iteration (generation). The specific parameter initialization includes the learning factors c_3, c_4, scaling factor $F \subset [0, 2]$, crossover rate $CR \subset (0, 1)$, and maximum iteration count $iter_{max}$.

Step 2: Evaluate individuals and determine the current optimal solution within the swarm.

Determine the current optimal position $pbest_i(k)$ for each particle. Calculate the fitness value f for each particle in the swarm using the fitness function, and determine the current optimal position within the swarm $gbest_g(k)$.

Step 3: Update particle velocity and position.

As each particle continuously tracks the current optimal particle, particle X_i updates its velocity and position using the following formulas:

$$\begin{aligned} v_i(k+1) = w \cdot v_i(k) \\ + c_3 \cdot r_1 \left(pbest_i(k) - x_i(k)\right) \\ + c_4 \cdot r_2 \left(gbest_g(k) - x_i(k)\right) \end{aligned} \tag{18}$$

$$x_i(k+1) = x_i(k) + v_i(k+1), \tag{19}$$

where r_1 and r_2 are random numbers in the range (0, 1).

Step 4: Perform mutation operation on current particles.

For a specific particle X_i within the current particle swarm with a positional vector x_i characterized by three coordinate elements xyz, six particles from the current particle swarm were randomly selected and denoted as X_{a1}, X_{b1}, X_{a2}, X_{b2}, X_{a3}, and X_{b3}. Assuming that the vector of the mutated particle is represented as X_{mi}, and its corresponding positional vector x_{mi} is defined by the three coordinate elements x_{mix}, x_{miy}, and x_{miz}, the formula governing the update of the position of the mutated individual is as follows:

$$\begin{cases} x_{mix} = x_{ix} + F \cdot (x_{a1x} - x_{b1x}) \\ x_{miy} = x_{iy} + F \cdot \left(x_{a2y} - x_{b2y}\right) \\ x_{miz} = x_{iz} + F \cdot (x_{a3z} - x_{b3z}). \end{cases} \tag{20}$$

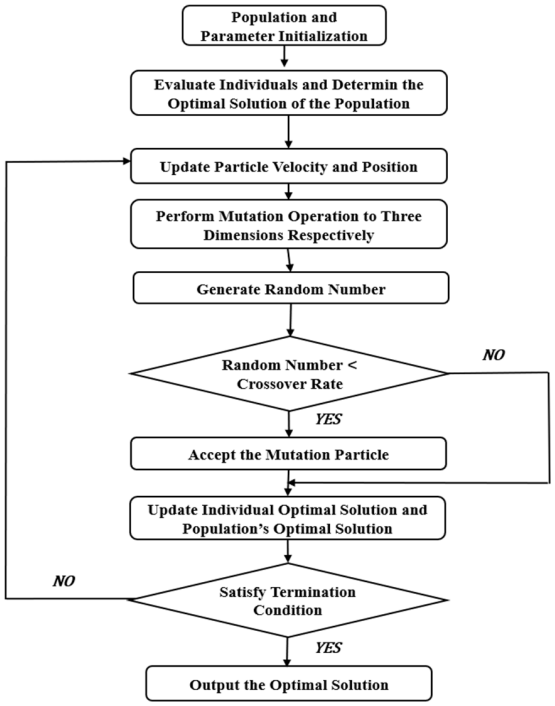
Step 5: Perform crossover and selection operations on current particles.

For element x_{ix} of positional vector x_i, a random number $C_{ix}(0 < C_{ix} < 1)$ was chosen. If the random number is less than crossover rate CR, the mutated positional element x_{mix} is selected and accepted. Similarly, for elements x_{iy} and x_{iz}, random numbers $C_{iy}(0 < C_{iy} < 1)$ and $C_{iz}(0 < C_{iz} < 1)$ are taken, and the aforementioned process is repeated. After completing the mutation of X_i, its fitness f_{mi} is calculated, and the subsequent step of optimal comparison is performed.

Step 6: Update individual and global optimal solutions.

Step 7: Increase the iteration count, and verify whether the termination condition is satisfied.

Set $k = k + 1$, and check whether the maximum iteration count M or the convergence requirement is reached. If the termination condition is not satisfied, return to Step 3. Otherwise, proceed to the next step.



Input :
$maxnum$ = maximum iteration count
$particlesize$ = particle population size
x_i = an individual particle within the population
x_{mi} = mutated individual
$x_{a1}, x_{a2}, x_{a3}, x_{b1}, x_{b2}, x_{b3}$ = six different random particle individuals
x_{ix}, x_{iy}, x_{iz} = the coordinate element of a specific particle
v_i = particle velocity
C_{ix}, C_{iy}, C_{iz} = randomly generated decimals between 0 and 1
$pbest_i$ = the particle with the minimum fitness
Output:
$gbest_g$ = global optimum

```
while k <= maxnum do
    for i = 1 → particlesizes do
        v_i(k+1)
          ←v_i(k)+c3*r1*(pbest_i(k)−X_i(k))+c4*r2*(gbest_g(k)−x_i(k))
        x_i(k+1) ←x_i(k) + v_i(k+1)
        x_mix ← x_ix + F * (x_a1x − x_b1x)
        x_miy ← x_iy + F * (x_a2y − x_b2y)
        x_miz ← x_iz + F * (x_a3z + x_b3z)
        if C_ix < CR then
            x_ix ← x_mix
        end
        if C_iy < CR then
            x_iy ← x_miy
        end
        if C_iz < CR then
            x_iz ← x_miz
        end
        if x_i < pbest_i then
            pbest_i ← x_i
        end
    end
    gbest_g = min(plbest)
end
```

FIGURE 6. Flowchart and pseudocode of DE algorithm.

Step 8: Output the optimal solution of the particle swarm.

The specific flow and pseudocode of the integrated algorithm are shown in Fig. 6.

B. PERFORMANCE TESTING AND ANALYSIS OF THE MODIFIED ALGORITHM

The performance of the proposed DE-PSO algorithm was evaluated by conducting performance tests using six representative benchmark test functions. The selected test functions are as follows:

(1) Spherical function (single-peak spherical functions)

$$\min f_1(X) = \sum_{i=1}^{n} x_i^2 \tag{21}$$

TABLE 1. Test function-related parameter settings.

Function	Theoretical optimal value	Search range
f_1	0	[–5, 5]
f_2	0	[–5, 5]
f_3	0	[–5, 5]
f_4	0	[–5, 5]
f_5	0	[–5, 5]
f_6	0	[–5, 5]

(2) Rosenbrock function (single-peak continuous function):

$$\min f_2(X) = \sum_{i=1}^{n-1}\left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right] \quad (22)$$

(3) Ackley function (multimodal, multilocal minimum function)

$$\min f_3(X) = 20 + e - 20\exp\left[-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right] - \exp\left[\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right] \quad (23)$$

(4) Rastrigin function (multimodal infinite extreme-value function)

$$\min f_4(X) = 10n + \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i)\right] \quad (24)$$

(5) Griewank function (multi-peak function)

$$\min f_5(X) = \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (25)$$

(6) Schaffer F6 function (highly oscillating multi-peak function)

$$\min f_6(X) = \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{1 + 0.001\left(x_1^2 + x_2^2\right)} + 0.5 \quad (26)$$

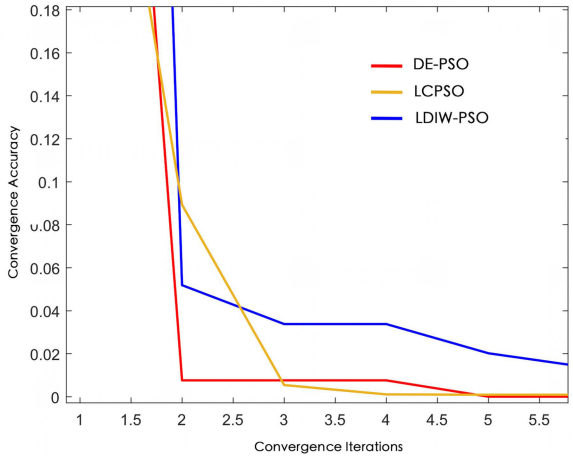
The theoretical optimal values and search ranges of the test functions are listed in Table 1.

The local convergence curves of the LDIW-PSO, LCPSO, and DE-PSO algorithms for the various test functions are shown in Fig. 7. To mitigate randomness in the stochastic search algorithms, all methods were executed 50 times, and the convergence rates and mean values of the algorithms were statistically analyzed. The test results are presented in Table 2, where the optimal data for each test result are in bold.

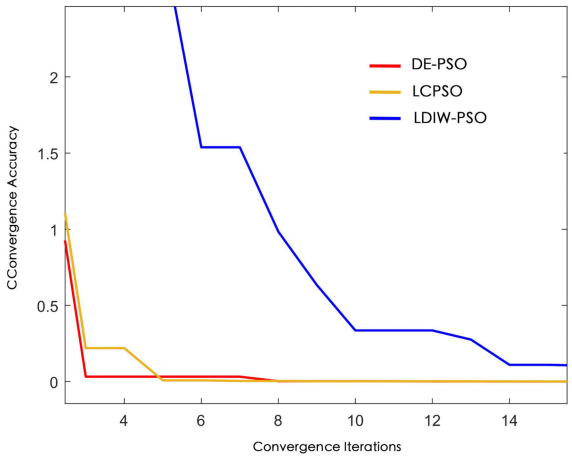
Fig. 7 shows that both the modified DE-PSO and LCPSO algorithms converged faster than the conventional LDIW-PSO algorithm across all tested functions. Moreover, except for f_3, the newly proposed DE-PSO algorithm outperformed the LCPSO algorithm in terms of convergence speed. Table 2 shows that all algorithms exhibited high convergence rates when handling single-peak or multimodal multiextremal functions. In addition, in terms of optimization accuracy, both the modified DE-PSO and LCPSO algorithms showed some improvements compared to the conventional algorithm. However, the DE-PSO algorithm proposed in this study surpasses the LDIW-PSO algorithm by to 1–2 orders of magnitude in functions such as f_1, f_2, f_3, and f_4, indicating its superior performance. Furthermore, the DE-PSO algorithm maintained a 100% convergence rate when solving the infinitely extreme value function, demonstrating exceptional performance. Moreover, when dealing with multimodal, multi-local minimum problems with insignificant changes, such as function f_3, the LCPSO algorithm exhibited an optimal convergence rate and accuracy. For handling the multi-peak function (f_5), the modified DE-PSO and LCPSO algorithms achieved comparable convergence accuracy, outperforming the conventional LDIW-PSO algorithm. However, when handling a highly oscillating multi-peak function (f_6), the LCPSO algorithm shows no significant advantage over the conventional LDIW-PSO algorithm in terms of convergence accuracy. Nevertheless, the DE-PSO algorithm proposed in this study exhibits superior performance in solving this problem. Regarding the convergence rate, although the DE-PSO algorithm demonstrates a slightly higher convergence than the LDIW-PSO and LCPSO algorithms, the improvement is not substantial.

The above analysis indicates that the DE-PSO algorithm exhibits faster convergence than the existing approaches. It also demonstrates higher convergence accuracy in low-dimensional problem solving. However, it does not exhibit a significant advantage in terms of the accuracy or convergence rate in high-dimensional problem solving. Overall, the DE-PSO algorithm, which incorporates mutation and crossover strategies, maintains swarm diversity when dealing with both single-peak and multimodal problems, enabling it to escape the local optima and achieve significant improvements in solution accuracy.

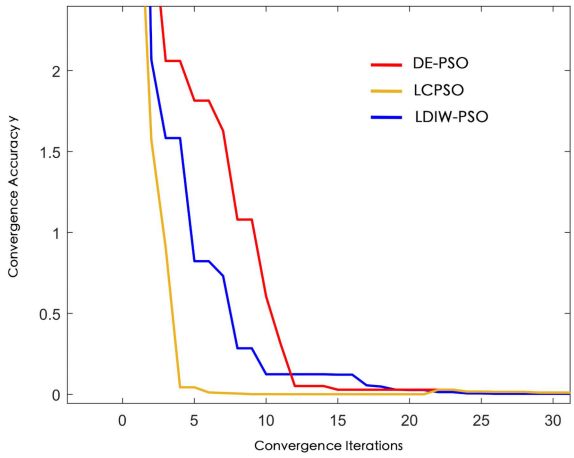
To investigate further the influence of the scaling factor and crossover probability on the performance of the proposed DE-PSO algorithm, we selected the Rosenbrock function (f_2), Rastrigin function (f_4), and Schaffer F6 function (f_6) to represent the single-peak, multimodal, and multi-peak functions, respectively, for the experimental analysis of the performance of the algorithm. First, we fixed the scaling factor F to 2 and sequentially set the crossover probability CR values to 0.2, 0.4, 0.6, and 0.8, for comparative simulation experiments. The DE-PSO algorithm test results for the different functions under the aforementioned settings are shown in Fig. 8. Next, by fixing the crossover probability at 0.6, we sequentially set the scaling factor F values to 0.2, 0.8, 1.2, and 1.8 for comparative simulation experiments. The convergence test results of the DE-PSO algorithm for different test functions under these parameter settings are shown in Fig. 9. Each test group was tested 50 times to ensure the robustness of the experiments, and the convergence rate and mean values of the algorithm results were statistically analyzed. Furthermore, based on the selected test functions,



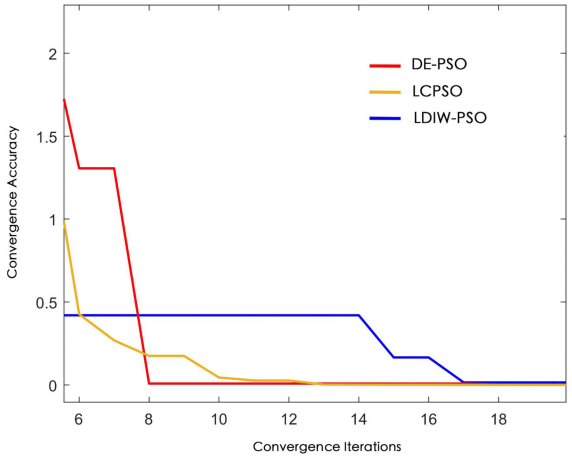
(a) Local convergence curve of the Spherical function



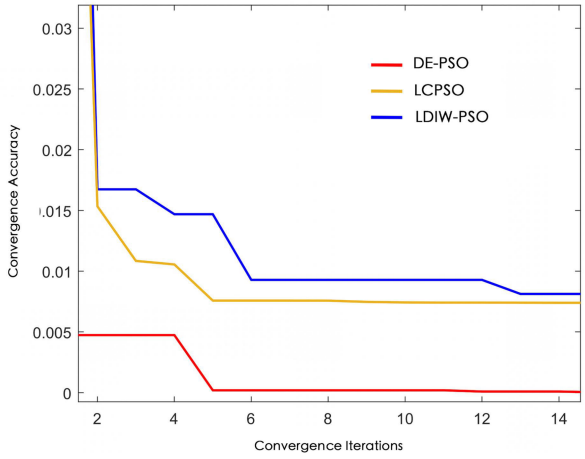
(b) Local convergence curve of the Rosenbrock function



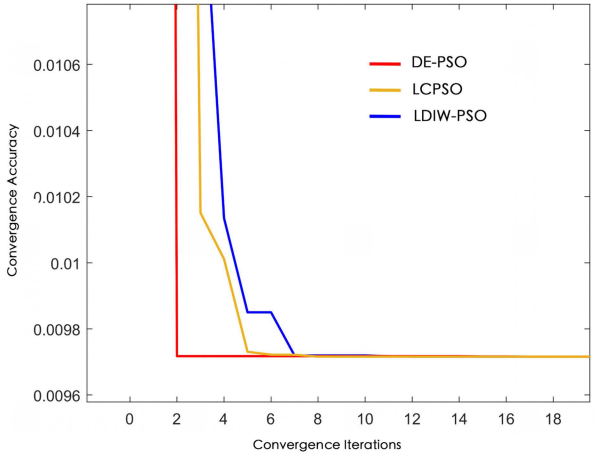
(c) Local convergence curve of the Ackley function



(d) Local convergence curve of the Rastrigin function



(e) Local convergence curve of the Griewank function



(f) Local convergence curve of the Schaffer F6 function

FIGURE 7. **Local convergence curves of PSO and DE-PSO on various test functions.**

the convergence speeds of the algorithm were compared for different values of CR and F, as shown in Figs. 10 and 11, respectively.

The results in Fig. 11 indicate that increasing the crossover probability CR or scaling factor F in the modified DE-PSO algorithm leads to improved convergence rates, and this

TABLE 2. Comparison of results for different algorithms on test functions.

Function	Algorithm	Convergence rate	Mean value
	LDIW-PSO	**50/50**	7.8651e–07
f_1	LCPSO	**50/50**	4.6575e–07
	DE-PSO	**50/50**	**7.0499e–09**
	LDIW-PSO	**50/50**	1.1512e–07
f_2	LCPSO	**50/50**	1.4388e–08
	DE-PSO	**50/50**	**3.7607e–09**
	LDIW-PSO	**50/50**	8.5541e–07
f_3	LCPSO	**50/50**	**2.8521e–08**
	DE-PSO	**50/50**	3.2401e–08
	LDIW-PSO	48/50	7.1538e–07
f_4	LCPSO	45/50	2.3734e–08
	DE-PSO	**50/50**	**2.0941e–08**
	LDIW-PSO	36/50	7.4131e–07
f_5	LCPSO	**40/50**	**1.3029e–08**
	DE-PSO	38/50	2.0517e–08
	LDIW-PSO	26/50	1.8907e–08
f_6	LCPSO	26/50	1.9607e–08
	DE-PSO	**27/50**	**1.2824e–09**

improvement is particularly significant when dealing with multi-peak problems. However, the convergence accuracy of the algorithm may exhibit certain fluctuations. Overall, it still shows a tendency to improve. However, after increasing the values of CR and F, the convergence speed decreased. Conversely, reducing the values of CR and F can lead to a faster convergence speed but may result in lower convergence rates and accuracies. Therefore, when applying the DE-PSO algorithm, the practical problem must be considered and the settings of parameters CR and F must be comprehensively evaluated. In particular, the parameter settings have a more significant impacts on the performance of the algorithm in high-dimensional multi-peak problems.

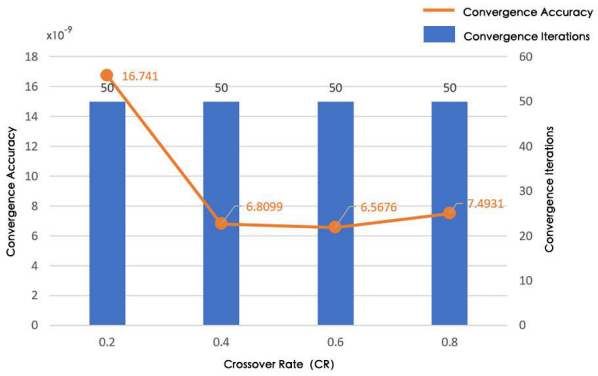
C. THREE-DIMENSIONAL ENVIRONMENTAL MODELING AND OBSTACLE AVOIDANCE STRATEGY FOR MARINE NET CAGES

1) THREE-DIMENSIONAL MARINE MAPPING

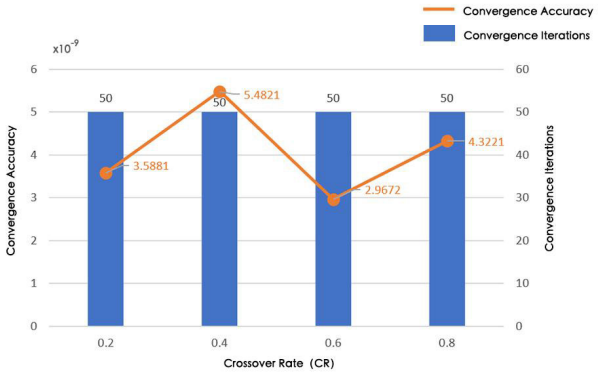
The modeling method for marine environments directly determines the difficulty of path planning problem solving. Therefore, the proper selection of environmental modeling methods directly affects the difficulty and efficiency of problem-solving. Path planning for AUVs requires particular consideration of the influence of obstacles.

In the 3D coordinate system $O-XYZ$, let $S=(s_x, s_y, s_z)$ represents the starting point of path planning for the AUV, $G=(g_x, g_y, g_z)$ denotes the target point, and $P_i=(p_x, p_y, p_z)$ represents intermediate nodes along the path.

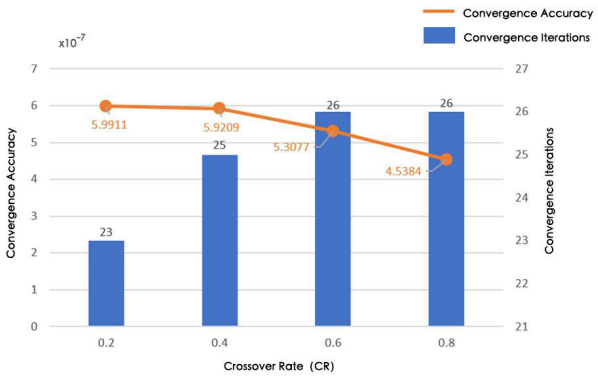
As shown in Fig. 12, path planning was conducted from $A(S)$ to $G(P)$, with lengths of $|AD|$ in the OX direction, $|AB|$ in the OY direction, and $|AE|$ in the OZ direction, constructing a cubic region $ABCD-EFGH$ as the path planning space. The obstacles O_i were represented using



(a) Rosenbrock function test results



(b) Rastrigin function test results

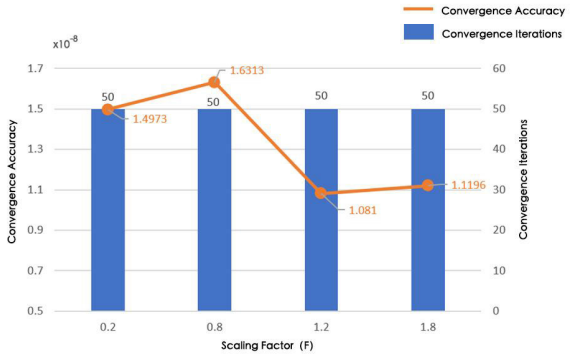


(c) Schaffer F6 function test results

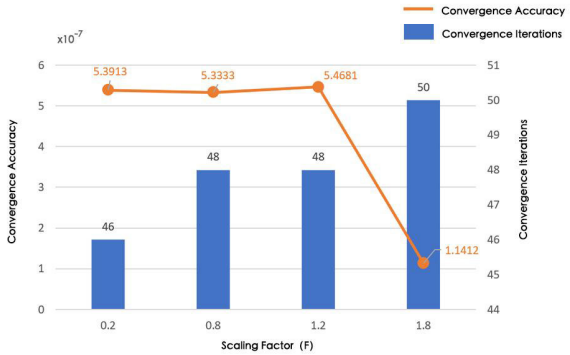
FIGURE 8. DE-PSO algorithm test results with different crossover probabilities.

their corresponding minimum bounding spheres, described as $\{(o_i, r_i)|i=1,2,\ldots,k\}$, where o_i represents the center of the circumscribed sphere of obstacle O_i, and r_i represents the radius of the sphere [35].

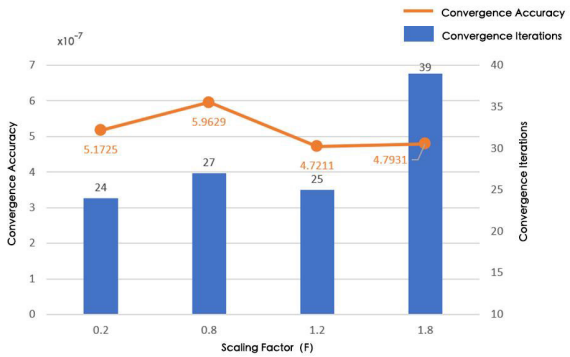
As shown in Fig. 13, planes were constructed by passing through the center o_i of each obstacle O_i perpendicular to the x-axis. Additionally, parallel planes were generated at a fixed step size λ in both the positive and negative directions along the x-axis. This process divided the path-planning space into evenly spaced partitions. The selection of waypoints λ was determined based on the size of the circumscribed sphere radius r_i for each obstacle. This division of the path-planning



(a) Rosenbrock function test results



(b) Rastrigin function test results



(c) Schaffer F6 function test results

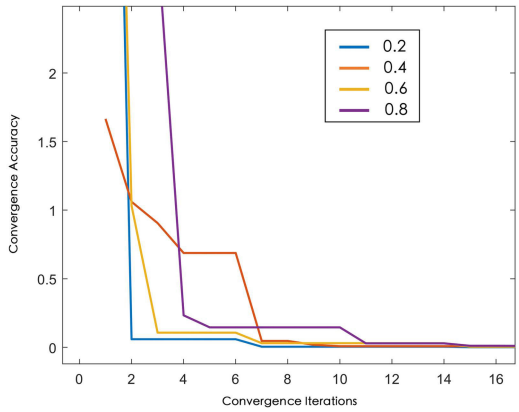
FIGURE 9. DE-PSO algorithm test results with different scaling factors.

space resulted in a two-dimensional plane comprising non-feasible, feasible, and obstacle regions.

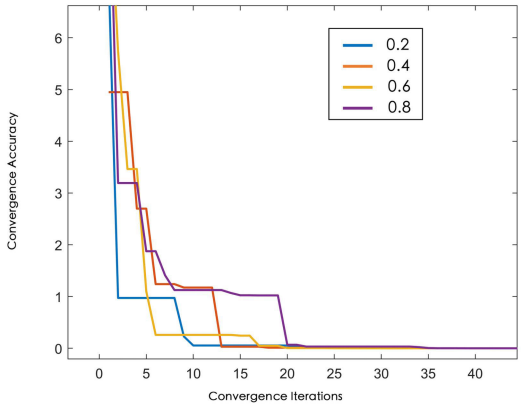
With this discretization, abstraction and modeling of the 3D marine map were completed. The goal of path planning is to select feasible points P_i and P_{i+1} from the non-obstacle regions within the feasible area while ensuring that no intersections occur between the lines connecting P_i and P_{i+1} and the circumscribed sphere of any obstacle to achieve the shortest possible planned path.

2) THREE-DIMENSIONAL OBSTACLE AVOIDANCE ALGORITHM PRINCIPLE

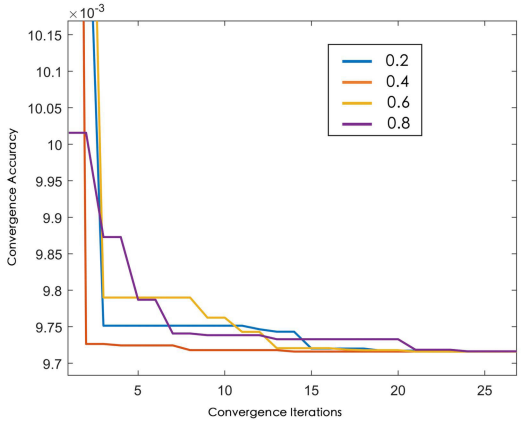
Let P_i represent the current position and P_{i+1} represent the position at the next time step. In the PSO algorithm, position P_{i+1} at the next time step is determined by the weighted



(a) Convergence speed results for the Rosenbrock function



(b) Convergence speed results for the Rastrigin function

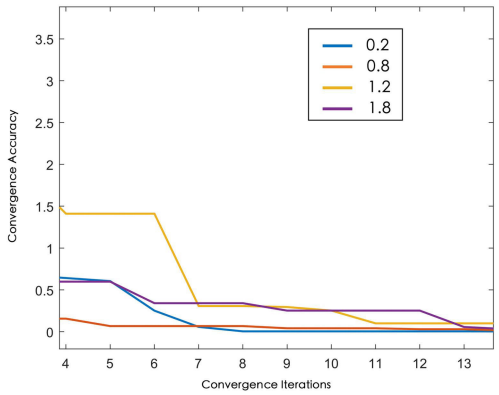


(c) Convergence speed results for the Schaffer F6 function

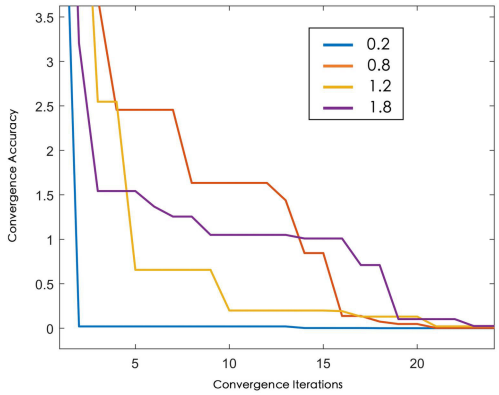
FIGURE 10. Convergence speed comparison of DE-PSO algorithm with different crossover probabilities.

sum of the individual optimal position P_i and the global optimal position P_b. Therefore, the PSO algorithm can be applied to an abstracted marine map model to determine the shortest-path solution.

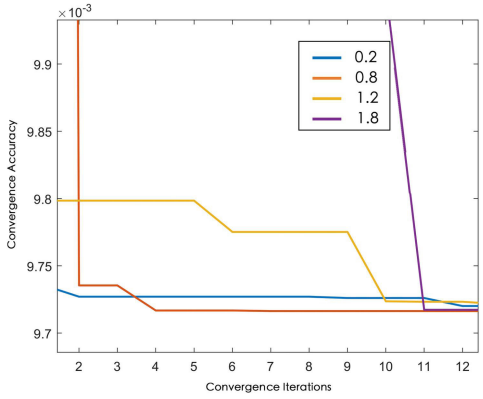
In the context of the path-planning problem, the objective was to select the shortest obstacle-free path as the optimal path. Therefore, to solve this problem, the performance indicator of the distance metric was used to represent the fitness function value.



(a) Convergence speed results for the Rosenbrock function



(b) Convergence speed results for the Rastrigin function



(c) Convergence speed results for the Schaffer F6 function

FIGURE 11. Convergence speed comparison of DE-PSO algorithm with different scaling factors.

Let each particle represent a feasible path L_i, and divide the path planning space into m blocks, resulting in a total of $m+1$ nodes. The fitness function for the path length is defined as follows:

$$D = \sum D_{P_i P_{i+1}}, \tag{27}$$

where $D_{p_i p_{i+1}}$ is the Euclidean distance between points P_i and P_{i+1} in 3D space:

$$D_{p_i p_{i+1}} = \sqrt{(p_x - p_{x+1})^2 + (p_y - p_{y+1})^2 + (p_z - p_{z+1})^2}. \tag{28}$$

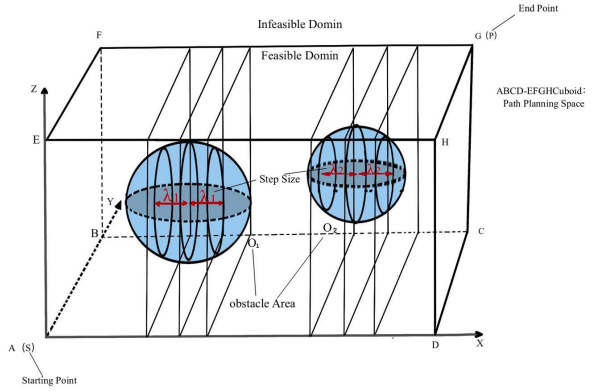


FIGURE 12. Three-dimensional marine map modeling schematic.

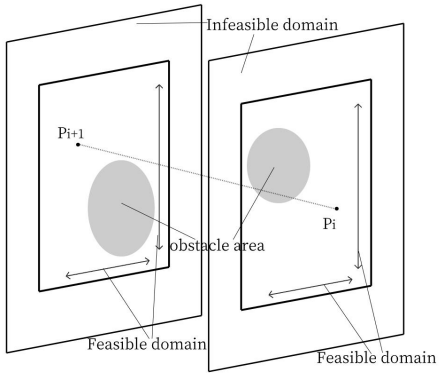


FIGURE 13. Illustration of waypoint selection using segmented planes.

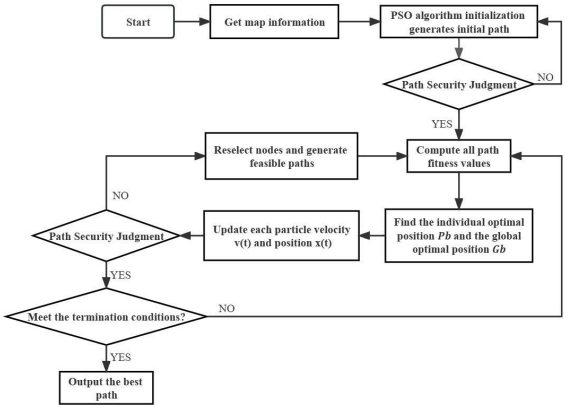


FIGURE 14. Flowchart of 3D obstacle avoidance algorithm.

A smaller value of fitness function D indicates a better path. The optimal path is determined through continuous iterations.

The minimum circumscribed sphere of an obstacle can be represented by the equation

$$(x - o_{ix})^2 + (y - o_{iy})^2 + (z - o_{iz})^2 = r_i^2. \tag{29}$$

As shown in Fig. 13, the selection of P_i should be selected in the non-obstacle region of the feasible area. For points inside the obstacle region and outside the feasible area, the

TABLE 3. Parameter settings for the modified PSO algorithm incorporating swap operators.

Parameter	Numerical setting
Population size (M)	100
Iteration count (iter)	2000
Initial inertia weight (w_0)	0.9
Swap operator learning factor c_1	0.7
Swap operator learning factor c_2	0.7

TABLE 4. Positions and sizes of spherical net cages.

Spherical cage code	Center coordinates	Radius
A_1	[300, 400, 460]	50
A_2	[100, 150, 150]	45
A_3	[600, 522, 510]	50
A_4	[350, 200, 440]	80
A_5	[150, 250, 300]	70
B_1	[300, 400, 460]	75
B_2	[100, 150, 150]	50
B_3	[600, 522, 510]	51
B_4	[350, 200, 440]	80
B_5	[150, 250, 300]	61

following boundary absorption process was applied:

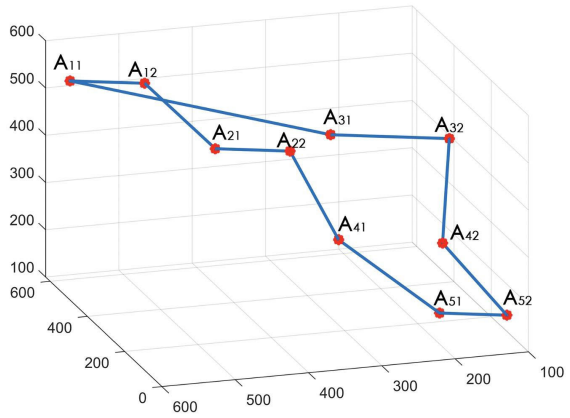
$$P_i' = \begin{cases} P_i + r_i - d\left(P_i, o_i\right), & P_i \text{with the obstacle area} \\ P_i - \delta\left(P_i, x, y, z\right), & P_i \text{within feasible area,} \end{cases} \tag{30}$$

where $d\left(P_i, o_i\right)$ represents the distance from point P_i to the center of sphere o_i and $\delta\left(P_i, x, y, z\right)$ represents the distance from point P_i to the boundary of the feasible area. This requirement ensured that point P_i inside the obstacle region was moved to the boundary of the obstacle region and point P_i outside the feasible area was moved to the boundary of the feasible area. However, the selected path points P_i and P_{i+1} along their connecting lines may intersect or be tangential to the circumscribed spheres of the obstacles. In real-world scenarios, the planned path of the AUV should not include points of contact with obstacles. Therefore, further analysis of the safety of the planned path is required.

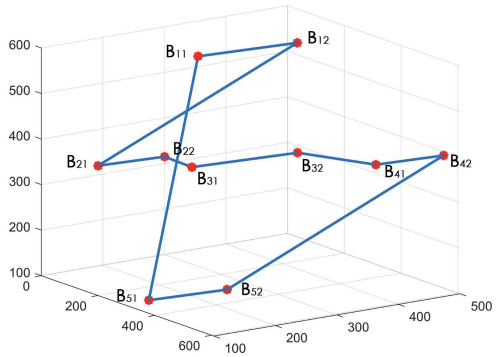
The spatial straight-line equation P_i for the line connecting P_{i+1} and L is defined as follows:

$$L: \begin{cases} x = x_{i+1} + k_x t \\ y = y_{i+1} + k_y t \\ z = z_{i+1} + k_z t, \end{cases} \tag{31}$$

where $[x_{i+1} y_{i+1} z_{i+1}]^T$ represents the position of the next search point and $[k_x k_y k_z]^T = [x_{i+1} - x_i y_{i+1} - y_i z_{i+1} - z_i]^T$ represents the direction vector. By combining the equations for the minimum circumscribed sphere of an obstacle,



(a) Simulation results of group A



(b) Simulation results of group B

FIGURE 15. Shortest path planning for concealed cage inspection points with obstacles.

we obtain the following equation:

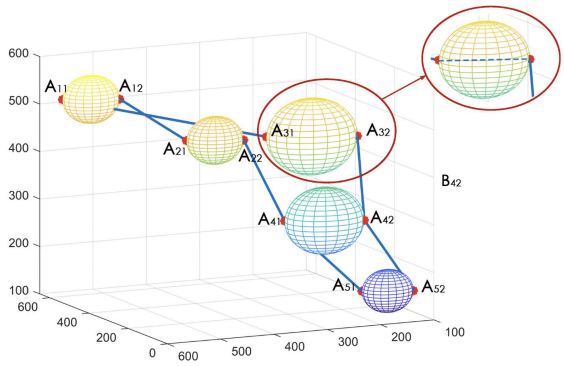
$$\begin{cases} \lambda_1 = k_x^2 + k_y^2 + k_z^2 \\ \lambda_2 = 2x_i k_x - 2k_x o_{ix} + 2y_i k_y - 2k_y o_{iy} + 2z_i k_z - 2k_z o_{iz} \\ \lambda_3 = (x_i - o_{ix})^2 + \left(y_i - o_{iy}\right)^2 + \left(y_i - o_{iy}\right). \end{cases} \tag{32}$$

The above equations transform the problem of determining the feasibility of path $P_i P_{i+1}$ to check whether $\Delta = \lambda_2^2 - 4\lambda_1\lambda_3$ [35]:

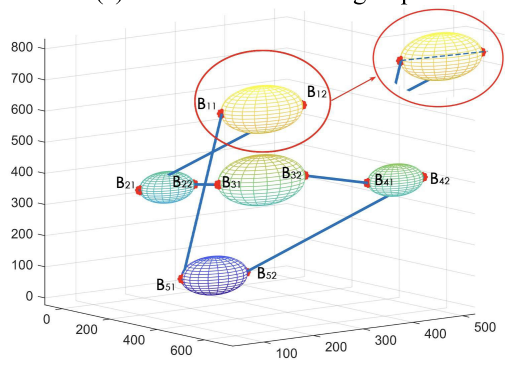
$$\begin{cases} \Delta < 0, & P_i P_{i+1} \text{ is feasible} \\ \Delta \geq 0, & P_i P_{i+1} \text{ is infeasible.} \end{cases} \tag{33}$$

If point $P_i P_{i+1}$ is infeasible, it must be reselected until there is no intersection or tangency with the circumscribed sphere of the obstacles in the P_{i+1} direction.

The PSO algorithm eventually generates an optimal collision-free path from point S to G based on the principle of selecting intermediate points along the path and ensuring path safety. The process of implementing the optimal 3D obstacle avoidance algorithm in 3D space using the PSO algorithm is illustrated in Fig. 14.

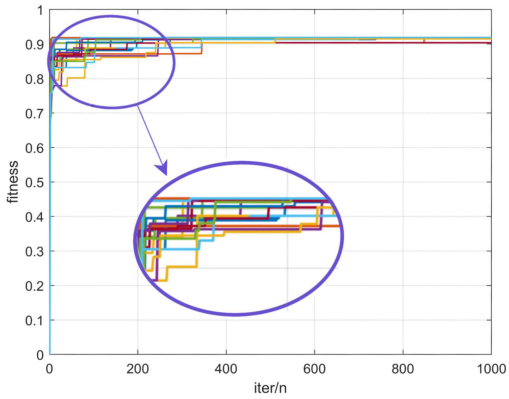


(a) Simulation results of group A

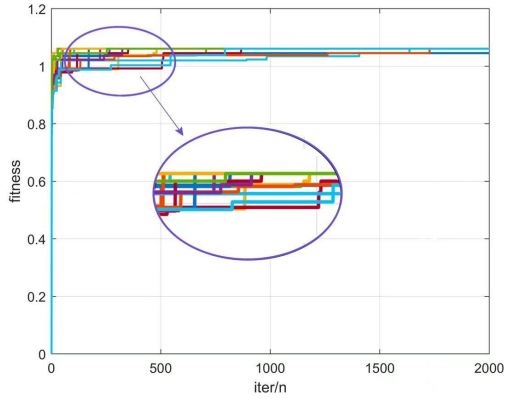


(b) Simulation results of group B

FIGURE 16. Shortest path planning for visible cage inspection points with obstacles.



(a) Convergence curves for group A



(b) Convergence curves for group B

FIGURE 17. Convergence curves for shortest path planning.

V. SIMULATION EXPERIMENT

This section demonstrates the effectiveness of the DE-PSO algorithm by presenting simulation experiments conducted for marine offshore net cage inspections. All simulation experiments were performed on a laptop computer running a x64 processor and equipped with a Windows 11 operating system within the MATLAB R2023a environment.

A. NUMERICAL SIMULATION AND SIMULATION OF SHORTEST PATH PLANNING FOR 3D MARINE NET CAGE INSPECTION POINTS

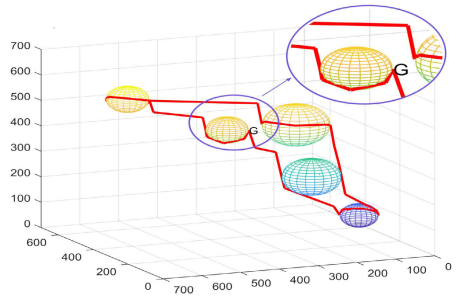
To accomplish multi-point inspection of marine offshore net cages, the first step is to find the shortest paths that traverse all predetermined inspection points. In this study, we employed a modified PSO algorithm that incorporates swap operators to conduct simulation experiments to solve this problem. Based on this formula, the algorithm required the parameter settings listed in Table 3.

In the simulation experiments, we constructed five spherical marine offshore net cages in 3D space with varying radii. The coordinates of the sphere centers and radius sizes were divided into two experimental groups, labeled Groups A and B, to verify the adaptability and feasibility of the algorithm for different environmental models. The specific parameters and location settings are presented in Table 4. In the abstract environmental model space, we set two predetermined inspection points for each net cage center along the diameter direction on the plane. Thus, the entire inspection path included 10 inspection points. Each inspection point was represented by A_{mn} or B_{mn}, where A and B denote the experimental groups; $m = 1, \cdots, 10$ represent the cage codes; and $n = 1$ and 2 represent the inspection point numbers.

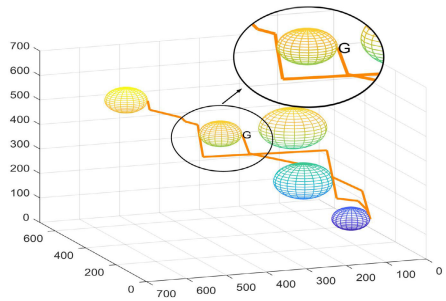
Figs. 15 and 16 show the results of applying the modified PSO algorithm incorporating swap operators to solve the shortest inspection path planning problem for Groups A and B, respectively, without considering AUV obstacle avoidance. The simulation results indicate that this

TABLE 5. Parameter settings for the DE-PSO algorithm.

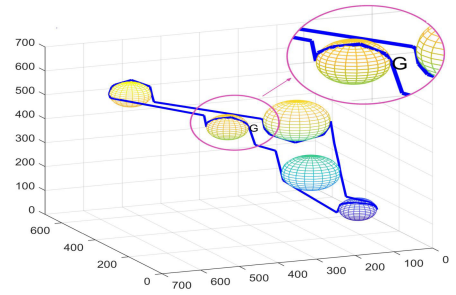
Parameter	Numerical setting
Population size (M)	60
Iteration count (iter)	400
PSO learning factor c_1	2.0
PSO learning factor c_2	1.5
Initial Inertia Weight (w_{init})	0.9
Final inertia weight (w_{end})	0.4
Number of path segments (path_num)	30
DE scaling factor (F)	0.8
DE crossover probability (CR)	0.6



(a) Simulation results for the PSO algorithm

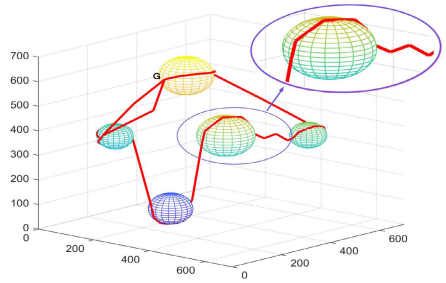


(b) Simulation results for the LCPSO algorithm

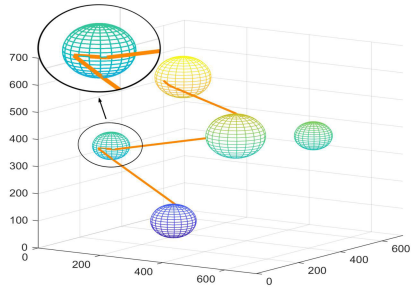


(c) Simulation results for the DE-PSO algorithm

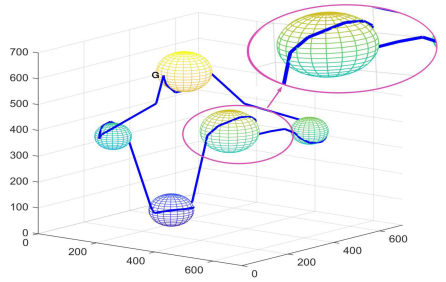
FIGURE 18. Autonomous obstacle avoidance for Group A in the shortest path planning.



(a) Simulation results for the PSO algorithm



(b) Simulation results for the LCPSO algorithm



(c) Simulation results for the DE-PSO algorithm

FIGURE 19. Autonomous obstacle avoidance for Group B in the shortest path planning.

algorithm could adaptively plan the shortest inspection paths. However, during the inspection route planning process, some inspection routes may pass through the marine offshore net cage to achieve the shortest path, which compromises the safety of the AUVs and hinders their ability to complete their inspection tasks autonomously. Fig. 17 illustrates the convergence status of the algorithm for Groups A and B after 20 repetitions. The results demonstrate that the modified PSO algorithm incorporating swap operators exhibits rapid convergence and is less prone to becoming trapped in local optima, thus satisfying the specific requirements of the application scenario.

B. SIMULATION OF AUTONOMOUS OBSTACLE AVOIDANCE SHORTEST PATH PLANNING FOR MARINE CAGE INSPECTION POINTS USING DE-PSO ALGORITHM

Real-time obstacle avoidance must be considered in autonomous AUV inspection tasks. As shown by the simulation experiment discussed in Section V-A, certain predetermined routes pass through a marine offshore net cage when only the shortest inspection path is considered, resulting in a lack of autonomous obstacle avoidance capability. Therefore, in addition to shortest path planning, we employed the DE-PSO algorithm, which is a modified PSO algorithm that integrates DE algorithms, to implement autonomous obstacle avoidance in AUVs. Moreover, a performance comparison was performed with conventional LDIW-PSO and LCPSO algorithms to demonstrate the superiority of the proposed algorithm in terms of premature convergence and other aspects. The parameter settings for the algorithm are listed in Table 5. The position and size data of the spherical cages for Groups A and B, used in the simulation experiments described in Section V-A, were retained. The parameter settings are listed in Table 4.

Figs. 18 and 19 present the experimental results of Groups A and B, respectively, where the conventional LDIW-PSO, LCPSO, and modified DE-PSO algorithms were employed to achieve autonomous obstacle avoidance and shortest path planning for AUVs. Point G represents the

TABLE 6. Simulation result data records.

Group	Algorithm selection	Simulation data	
A	LDIW-PSO	Mean path growth length 500.654 Standard deviation of path growth length 57.0913	Path growth length variance 3259.42 Mean path growth rate 23.72%
A	LCPSO	Unable to find a successful inspection route	
A	DE-PSO	Mean path growth length **488.928** Standard deviation of path growth length **36.9377**	Path growth length variance **1364.39** Mean path growth rate **22.75%**
B	LDIW-PSO	Mean path growth length **456.093** Standard deviation of path growth length 86.7143	Path growth length variance 7519.38 Mean path growth rate 17.81%
B	LCPSO	Unable to find a successful inspection route	
B	DE-PSO	Mean path growth length 461.729 Standard deviation of path growth length **62.5425**	Path growth length variance **3911.56** Mean path growth rate **17.59%**

departure and return points of the AUV. To avoid experimental randomness and validate the performance of the improved algorithm, Table 6 presents the mean path growth length, path growth length variance, standard deviation of path growth length, and mean path growth rate for Groups A and B, with each group independently replicated 10 times using three different algorithms considering obstacle avoidance. The optimal values of each test result are shown in bold.

A comparison of Figs. 18 and 19 reveals that, when dealing with autonomous obstacle avoidance at the same location, the DE-PSO algorithm outperformed the conventional LDIW-PSO and LCPSO algorithms. The former achieves effective obstacle avoidance while exhibiting higher convergence accuracy, smoother paths, and better planning results. Analysis of the simulation data in Table 6 reveals that the convergence stability of the DE-PSO algorithm is superior to that of the conventional LDIW-PSO algorithm. Both experimental groups showed reductions of more than 27% in the standard deviation, effectively avoiding issues such as the excessive length of autonomously planned obstacle avoidance routes owing to local optima and premature convergence, which could result in increased energy consumption for AUVs. Furthermore, the LCPSO algorithm failed to find a complete inspection route. The primary limitation of this algorithm is its inability to identify a local path when confronted with the task of discovering a route that requires reaching the opposite side of a spherical offshore net cage. When addressing intricate path-planning problems involving multiple inspection points, the paths generated by this algorithm possess noticeably heightened roughness. Consequently, it is unsuitable for marine offshore net cage inspection.

VI. CONCLUSION

This article proposes a novel PSO algorithm that integrates the DE approach to address autonomous path planning for AUVs in marine offshore net cage inspections. For the multipoint shortest-path search problem, the algorithm optimizes and modifies the velocity and position update formulas by introducing the concepts of swap operators and sequences into the PSO algorithm. Additionally, through mutation, crossover, and selection operations, the modified algorithm maintains a high swarm diversity when facing multi-obstacle avoidance problems, enabling it to avoid local optima and premature convergence when solving multimodal problems.

Numerical comparisons of the modified DE-PSO algorithm were performed by conducting various simulation experiments involving different 3D marine offshore net cage modeling scenarios. In addition, quantitative testing analysis of the convergence and convergence accuracy of the algorithm was conducted using six representative benchmark test functions. The results demonstrate that the PSO algorithm that integrates the DE approach exhibits superior solution accuracy in handling single-peak and multimodal problems. Furthermore, a comparative analysis of the simulation experiment results revealed that the DE-PSO algorithm demonstrates outstanding performance in addressing the path planning problem for 3D marine offshore net cage inspection.

Future work will focus on investigating the effects of complex natural marine environments on path planning for AUVs, as well as the influences of factors such as ocean current variations and marine organism movements on the safety and economic efficiency of the planned paths.

REFERENCES

[1] Q. Tao, K. Huang, C. Qin, B. Guo, R. Lam, and F. Zhang, "Omnidirectional surface vehicle for fish cage inspection," in *Proc. OCEANS MTS*, Charleston, SC, USA, Oct. 2018, pp. 1–6, doi: 10.1109/OCEANS.2018.8604674.

[2] S. J. Ohrem, E. Kelasidi, and N. Bloecher, "Analysis of a novel autonomous underwater robot for biofouling prevention and inspection in fish farms," in *Proc. 28th Medit. Conf. Control Autom. (MED)*, Sep. 2020, pp. 1002–1008, doi: 10.1109/MED48518.2020.9183157.

[3] T. X. Lin, Q. Tao, and F. Zhang, "Planning for fish net inspection with an autonomous OSV," in *Proc. Int. Conf. Syst. Sci. Eng.*, Kagawa, Japan, Aug. 2020, pp. 1–5, doi: 10.1109/ICSSE50014.2020.9219318.

[4] L. Christensen, J. Hilljegerdes, M. Zipper, A. Kolesnikov, B. Hülsen, C. E. S. Koch, M. Hildebrandt, and L. C. Danter, "The hydrobatic dual-arm intervention AUV cuttlefish," in *Proc. OCEANS*, Oct. 2022, pp. 1–8.

[5] S. Mayberry, J. Wang, Q. Tao, F. Zhang, A. Song, X. Hong, S. Dong, C. Webb, D. Dugaev, and Z. Peng, "First step towards μNet: Open-access aquatic testbeds and robotic ecosystem," in *Proc. 15th Int. Conf. Underwater Netw. Syst.*, Nov. 2021, pp. 5–6.

[6] D. Li, P. Wang, and L. Du, "Path planning technologies for autonomous underwater vehicles—A review," *IEEE Access*, vol. 7, pp. 9745–9768, 2019, doi: 10.1109/ACCESS.2018.2888617.

[7] I. Stenius, J. Folkesson, S. Bhat, C. I. Sprague, L. Ling, Ö. Özkahraman, N. Bore, Z. Cong, J. Severholt, C. Ljung, A. Arnwald, I. Torroba, F. Gröndahl, and J.-B. Thomas, "A system for autonomous seaweed farm inspection with an underwater robot," *Sensors*, vol. 22, no. 13, p. 5064, Jul. 2022.

[8] 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.

[9] H. Zhou, Z. Jiang, Y. Xue, W. Li, F. Cai, and Y. Li, "Research on path planning in 3D complex environments based on improved ant colony algorithm," *Symmetry*, vol. 14, no. 9, p. 1917, Sep. 2022.

[10] Z. Li, K. Chen, and J. Wang, "An adaptive environmental sampling path planning technique for autonomous underwater vehicles based on ant colony optimization algorithm," in *Proc. IEEE 4th Int. Conf. Civil Aviation Saf. Inf. Technol. (ICCASIT)*, Oct. 2022, pp. 727–730.

[11] R. Meng, A. Sun, Z. Wu, X. Du, and Y. Meng, "3D smooth path planning of AUV based on improved ant colony optimization considering heading switching pressure," *Sci. Rep.*, vol. 13, no. 1, p. 12348, Jul. 2023, doi: 10.1038/s41598-023-39346-5.

[12] M.-Y. Gao, B.-B. Hu, B. Liu, N. Qiu, and H.-T. Zhang, "Constrained path-planning control of unmanned surface vessels via ant-colony optimization," in *Proc. 40th Chin. Control Conf. (CCC)*, Shanghai, China, Jul. 2021, pp. 4079–4084.

[13] Z. Yan, J. Yan, Y. Wu, S. Cai, and H. Wang, "A novel reinforcement learning based tuna swarm optimization algorithm for autonomous underwater vehicle path planning," *Math. Comput. Simul.*, vol. 209, pp. 55–86, Jul. 2023, doi: 10.1016/j.matcom.2023.02.003.

[14] Z. Yan, J. Zhang, and J. Tang, "Whale optimization algorithm based on lateral inhibition for image matching and vision-guided AUV docking," *J. Intell. Fuzzy Syst.*, vol. 40, no. 3, pp. 4027–4038, Mar. 2021.

[15] Z. Yan, J. Zhang, J. Zeng, and J. Tang, "Three-dimensional path planning for autonomous underwater vehicles based on a whale optimization algorithm," *Ocean Eng.*, vol. 250, Apr. 2022, Art. no. 111070.

[16] Y. Huang, Y. Li, Z. Zhang, and Q. Sun, "A novel path planning approach for AUV based on improved whale optimization algorithm using segment learning and adaptive operator selection," *Ocean Eng.*, vol. 280, Jul. 2023, Art. no. 114591.

[17] S. P. Sahoo, B. Das, B. B. Pati, F. P. G. Marquez, and I. S. Ramirez, "Hybrid path planning using a bionic-inspired optimization algorithm for autonomous underwater vehicles," *J. Mar. Sci. Eng.*, vol. 11, no. 4, p. 761, Mar. 2023.

[18] Z. Yan and Y. Li, "Data collection optimization of ocean observation network based on AUV path planning and communication," *Ocean Eng.*, vol. 282, Aug. 2023, Art. no. 114912.

[19] 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.

[20] Z. Yan, J. Zhang, J. Zeng, and J. Tang, "Water wave optimization algorithm for autonomous underwater vehicle path planning problem," *J. Intell. Fuzzy Syst.*, vol. 40, no. 5, pp. 9127–9141, Apr. 2021.

[21] N. H. Tran, A. D. Nguyen, and T. N. Nguyen, "A genetic algorithm application in planning path using B-spline model for autonomous underwater vehicle (AUV)," *Appl. Mech. Mater.*, vol. 902, pp. 54–64, Sep. 2020.

[22] B. Li, R. Zhao, G. Xu, G. Wang, Z. Su, and Z. Chen, "Three-dimensional path planning for an under-actuated autonomous underwater vehicle," in *Proc. ISOPE Int. Ocean Polar Eng. Conf.*, 2019, pp. 1–2.

[23] H. S. Lim, P. King, C. K. H. Chin, S. Chai, and N. Bose, "Real-time implementation of an online path replanner for an AUV operating in a dynamic and unexplored environment," *Appl. Ocean Res.*, vol. 118, Jan. 2022, Art. no. 103006.

[24] L. Wang, J. Li, J. Qi, and J. He, "AUV underwater 3D path planning based on particle swarm optimization-adaptive step-size cuckoo search algorithm," in *Proc. 11th Int. Conf. Netw., Commun. Comput.*, Dec. 2022, pp. 215–225.

[25] F. Sui, X. Tang, Z. Dong, X. Gan, P. Luo, and J. Sun, "ACO+PSO+A: A bi-layer hybrid algorithm for multi-task path planning of an AUV," *Comput. Ind. Eng.*, vol. 175, Jan. 2023, Art. no. 108905.

[26] H. Zhang and X. Shi, "An improved quantum-behaved particle swarm optimization algorithm combined with reinforcement learning for AUV path planning," *J. Robot.*, vol. 2023, pp. 1–11, Apr. 2023.

[27] X. Li and S. Yu, "Three-dimensional path planning for AUVs in ocean currents environment based on an improved compression factor particle swarm optimization algorithm," *Ocean Eng.*, vol. 280, Jul. 2023, Art. no. 114610.

[28] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII* (Lecture Notes in Computer Science), vol. 1447, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds. Berlin, Germany: Springer, 1998, doi: 10.1007/BFb0040810.

[29] Y. Feng, G.-F. Teng, A.-X. Wang, and Y.-M. Yao, "Chaotic inertia weight in particle swarm optimization," in *Proc. 2nd Int. Conf. Innov. Comput., Inf. Control (ICICIC)*, Sep. 2007, p. 475.

[30] T. B. Nkwanyana and Z. Wang, "Improved particle swarm optimization base on the combination of linear decreasing and chaotic inertia weights," in *Proc. 12th Int. Conf. Comput. Intell. Commun. Netw. (CICN)*, Sep. 2020, pp. 460–465.

[31] J. Xin, G. Chen, and Y. Hai, "A particle swarm optimizer with multi-stage linearly-decreasing inertia weight," in *Proc. Int. Joint Conf. Comput. Sci. Optim.*, vol. 1, Apr. 2009, pp. 505–508.

[32] L. Huang, K. Wang, C. Zhou, W. Pang, and L. Dong, "Particle swarm optimization for traveling salesman problems," *J. Jilin Uni.*, vol. 4, pp. 477–480, Jan. 2003, doi: 10.13413/j.cnki.jdxblxb.2003.04.013.

[33] R. Storn and K. Price, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," Int. Comput. Sci. Inst., Berkeley, CA, USA, Tech. Rep., TR-95-012, 1995.

[34] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997, doi: 10.1023/A:1008202821328.

[35] H. Pan, "Research on AUV path planning based on modified particle swarm optimization algorithm," *Today's Mod. Coastal Soc., Tech. Sociol. Aspects Coastal Res.*, no. 111, pp. 279–282, Fall 2020.

YIJUN HU is currently pursuing the bachelor's degree with the South China University of Technology.

He is the co-third author of the paper "Real-Time Ocean Current Compensation for AUV Trajectory Tracking Control Using a Meta-Learning and Self-Adaptation Hybrid Approach" presented by Sensors 2023. His current research interests include swarm intelligence optimization algorithms and path planning for autonomous underwater vehicles (AUVs).

JING WANG received the B.E. degree in underwater acoustics and electronics engineering from the Harbin Shipbuilding Engineering Institute (now Harbin Engineering University), Harbin China, in 1992, the M.E. degree in signal and information processing from the Harbin Institute of Technology, Harbin, in 1995, and the Ph.D. degree in underwater acoustics engineering from Northwestern Polytechnical University, Xi'an, China, in 2004.

She is currently a Visiting Researcher with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China. She has been with Yunnan University for almost 30 years. Her research interests include fast algorithms for transportation equations, underwater acoustic signal processing, and embedded systems.

HAOJIN HE is currently pursuing the bachelor's degree with the South China University of Technology. His research interest includes path-following algorithms for underwater detection.

YIQIANG ZHANG is currently pursuing the master's degree with the School of Computer Science and Technology, Jilin University. His research interests include artificial intelligence and mapping, planning, and control of underwater vehicles.

SHUO CAI is currently pursuing the bachelor's degree with the South China University of Technology. His current research interests include underwater autonomous navigation systems for autonomous underwater vehicles (AUVs) and reinforcement learning algorithms.

ANLU XIE is currently pursuing the bachelor's degree with the South China University of Technology. His current research interest includes innovative exterior design for soft robots.

ZIJUN ZHENG is currently pursuing the bachelor's degree with the South China University of Technology. His current research interest includes path-following control in autonomous underwater vehicles (AUVs).