

# Three-Dimensional Complete Coverage Path Planning for UAVs Using an Improved Multi-Objective Dung Beetle Optimization Algorithm

Ziye Shi<sup>1</sup>, Wenxing Wu<sup>2</sup>, Liqin Tian<sup>1\*</sup>

<sup>1</sup>Graduate School, University of Emergency Management, Langfang, China

<sup>2</sup>School of Computer Science, Qinghai Normal University, Xining, China

Email: \*tianliqin@ncist.edu.cn

**How to cite this paper:** Shi, Z.Y., Wu, W.X. and Tian, L.Q. (2026) Three-Dimensional Complete Coverage Path Planning for UAVs Using an Improved Multi-Objective Dung Beetle Optimization Algorithm. *Journal of Computer and Communications*, 14, 163-188.

<https://doi.org/10.4236/jcc.2026.147009>

**Received:** June 24, 2026

**Accepted:** July 25, 2026

**Published:** July 28, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

To address coverage blind spots and path redundancy in UAV complete coverage path planning in complex three-dimensional environments, this paper constructs a multi-objective optimization model. The model targets coverage completeness, the number of nodes, and path length. We also address insufficient population diversity, an imbalance between exploration and exploitation, and a tendency to fall into local optima during optimization. To do so, we propose an improved multi-strategy multi-objective dung beetle optimization algorithm. This algorithm is termed the Improved Multi-Objective Dung Beetle Optimizer (IMODBO). It enhances population diversity using Latin hypercube sampling for initialization. It introduces dynamic parameters to balance exploration and exploitation. It also designs a sine-cosine perturbation strategy to improve local search capability. Results on the CEC2020 benchmark suite verify that the proposed algorithm delivers better convergence and solution distribution performance for most test functions, with only a few individual metrics of special Pareto front problems slightly inferior to those of partial comparison algorithms. Simulation on three-dimensional terrains shows that, with the same number of nodes, our algorithm increases the coverage rate from 74.6% to 87.2%. It also reduces the flight path length, thereby effectively mitigating the coverage blind spot problem. This work provides technical support for complete 3D coverage operations by UAVs.

## Keywords

UAV, Dung Beetle Optimization Algorithm, Latin Hypercube Sampling, Perturbation Strategy, Coverage Path

## 1. Introduction

With the diversified of low-altitude application scenarios, unmanned aerial vehicles (UAVs) are increasingly widely used in complex three-dimensional environments, such as personnel search, material delivery, and path planning in emergency rescue operations. Particularly in unstructured and complex three-dimensional environments, achieving efficient coverage of the target area is a core prerequisite for ensuring search success rates, delivery accuracy, and mission timeliness [1]. Existing path planning schemes often suffer from coverage blind spots or path redundancy due to insufficient handling of the altitude dimension and spatial continuity. Therefore, developing an efficient complete coverage path planning algorithm suitable for complex three-dimensional environments is of great significance for supporting the national low-altitude economic development strategy and improving emergency rescue efficiency.

Compared with two-dimensional path planning problems in emergency scenarios, three-dimensional coverage path planning must simultaneously consider altitude changes, terrain undulations, and multiple conflicting optimization objectives, which increases the difficulty of solving the problem [2] [3]. To address this problem, existing research has made certain progress. Sharma and Voruganti [4] noted the complexity introduced by elevation changes and proposed the TANSIGA-II method, which achieves significant energy savings in real terrain by simultaneously optimizing travel distance, cumulative elevation change, and the number of turns. Wu *et al.* [5] proposed a coverage method based on visibility graphs for complex concave areas, which effectively reduces occluded blind spots, but this method still simplifies the three-dimensional space into multiple two-dimensional slices for processing, neglecting the continuity of the altitude dimension.

To address the problem of complete coverage path planning for UAVs in complex environments, existing research mainly falls into two major directions: improvements to classical path planning algorithms and optimization using heuristic intelligent algorithms [6]. In the direction of classical algorithm improvements, algorithms such as A\* [7], Dijkstra [8], and RRT [9] are widely used due to their strong solution stability and reliable paths. Regarding, RRT algorithm improvements, Khuat *et al.* [10] proposed the MultiRRT algorithm, which extends RRT to multi-UAV cooperative scenarios. By introducing node reduction and Bezier interpolation, this algorithm achieves cooperative path planning for multiple target points while satisfying dynamic and safety constraints. Furthermore, Sharma *et al.* [11] pointed out in their review study that although classical algorithms perform robustly in structured environments, they still have significant limitations when facing unknown dynamic threats, real-time responsiveness requirements, and energy constraints, which drive continuous exploration toward heuristic intelligent algorithms.

In recent years, heuristic intelligent algorithms have emerged as a research hotspot for three-dimensional complete coverage path planning for emergency

rescue UAVs, thanks their strong global search capabilities and adaptability to multi-objective optimization. Among them, the Sparrow Search Algorithm (SSA) [12], Whale Optimization Algorithm (WOA) [13] [14], and Ant Colony Optimization (ACO) [15] have been widely applied. To address the limitations of two-dimensional environments, researchers have begun exploring the application of SSA in three-dimensional spaces. For example, He and Wang [16] proposed an improved SSA algorithm by integrating good point set initialization and piecewise dynamic step size strategies, effectively enhancing the path planning performance of UAVs in complex three-dimensional environments. Yang *et al.* [17] extended the improved SSA to the field of three-dimensional UAV trajectory planning, integrating terrain elevation and threat zone modeling to achieve a Pareto optimal solution set under multiple constraints. In WOA applications, researchers have proposed various improvement strategies to enhance its performance in three-dimensional path planning. For instance, Wang *et al.* [18] proposed an improved whale optimization algorithm (CSRD-WOA) by integrating cuckoo search and random difference strategies, which effectively improves global search capability and convergence accuracy in complex three-dimensional terrain environments, although its application scenario is limited to static, a priori known terrain. In ACO research, Shi *et al.* [19] proposed a hybrid strategy integrating simulated annealing with ACO for complete coverage tasks of ground robots, thereby improving path completeness.

In summary, existing research still has deficiencies in three-dimensional complete coverage path planning. Traditional algorithms focus on optimizing single-path performance and are poorly suited to coverage traversal tasks. Although heuristic algorithms possess global search capabilities, their improvements primarily focus on their own performance neglecting coverage completeness constraints, leading to paths with omissions or redundancies. To address these issues, this paper constructs a multi-objective collaborative optimization model that comprehensively considers coverage completeness, path nodes, and path length, and proposes an improved multi-objective dung beetle optimization algorithm for complex three-dimensional environments. This algorithm enhances global exploration and local exploitation to efficiently solve the multi-objective optimization model, thereby generating high-quality complete coverage flight paths.

## 2. Model Formulation

This paper considers a single UAV performing a complete coverage task in a known three-dimensional environment. To develop an efficiently optimizable mathematical model for the proposed algorithm, we make the following assumptions. First, the environment can be accurately modeled using a three-dimensional grid map [20]. Second, the UAV can hover at fixed points and adjust its flight altitude continuously to adapt to the terrain. Third, its onboard sensors have a fixed field of view and detection range, so coverage is determined by the UAV's position and attitude. Fourth, the mission requires generating a flight trajectory

that fully covers all target areas by traversing a minimal number of path nodes and keeping the path short, while satisfying the UAV's kinematic constraints. Building on these assumptions, we proceed to model the problem in detail.

Based on the above assumptions, this section systematically constructs a multi-objective optimization model for three-dimensional complete coverage path planning from three aspects: problem formulation, objective functions, and constraints.

## 2.1. Problem Modeling

The three-dimensional full-coverage path planning problem for unmanned aerial vehicles (UAVs) essentially consists of finding an optimal flight path  $P$  from the starting point  $P_{start}$  to the end point  $P_{end}$  in a pre-constructed three-dimensional grid map. The path  $P = \{p_1, p_2, \dots, p_n\}$  is a sequence composed of a series of consecutive nodes, where  $p_1$  is equivalent to  $P_{start}$ ,  $p_n$  is equivalent to  $P_{end}$ , and each node  $p_i$  corresponds to a coordinate  $(x_i, y_i, z_i)$  in three-dimensional space. The core decision variable of the problem is the node sequence  $P^*$ .

To convert disordered scattered individuals into an ordered sequence of three-dimensional nodes, a deterministic TSP decoding strategy without randomness is adopted. Taking the coordinate origin as the reference, the node with the shortest Euclidean distance is selected as the unified global starting point to eliminate result fluctuations caused by random starting positions. Taking the current node as the benchmark, the unvisited node with the minimum Euclidean distance is chosen as the next waypoint in each iteration to generate a traversal index sequence. With the first point of the sequence as the start point and the last point as the end point, an ordered three-dimensional node path is constructed.

## 2.2. Objective Functions

**Table 1.** Definitions of key variables and regional concepts.

| Nomenclature        | Physical Meanings of Variables                                           | Simulation Values                                                                                      |
|---------------------|--------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| $W, H$              | Length and width of the square planar target area to be inspected by UAV | $W = H = 10$                                                                                           |
| $grid\_step$        | Discrete sampling step of terrain grid                                   | 0.5                                                                                                    |
| $Grid\_cover\_unit$ | Final effective coverage marker of grid                                  | 1: Distance requirement satisfied with no occlusion;<br>0: Inspection blind area                       |
| $R$                 | 3D effective sensing radius of a single UAV point                        | 2                                                                                                      |
| $V_{total}$         | Total terrain area of the entire target area                             | Total surface area of grid cells in target zone                                                        |
| $V_{cover}$         | Total terrain area covered by UAV under path P                           | Total area of all grids with $Grid\_cover\_unit = 1$ covered by UAVs under the current inspection path |

To address the three-dimensional full-coverage path planning problem for UAVs, this paper constructs a multi-objective comprehensive function with coverage rate, path nodes and path length as the core optimization objectives. The optimization process is designed to achieve efficient and complete coverage path planning for UAVs in the target area within three-dimensional space. **Table 1** presents the definitions of key variables and regional concepts.

### 2.2.1. Coverage Completeness

Coverage completeness serves as a core indicator for measuring the completion rate of UAV emergency rescue missions, which is directly related to the omnidirectional and non-missing inspection of disaster-affected areas and the success rate of search and rescue. In this model, the monitoring area is discretized into a series of tiny curved grid cells. A grid cell is regarded as covered only if both of the following conditions are satisfied:

- 1) Geometric reachability: The three-dimensional Euclidean distance between the center point of the grid cell and at least one UAV path node does not exceed the UAV sensing radius  $R$ .
- 2) Unobstructed line of sight: There are no topographic obstacles between the grid cell and the corresponding UAV node. Then the coverage rate  $f_1$  can be expressed as Equation (1). As the coverage metric,  $f_1$  ranges from 0 to 1. A larger value indicates more complete regional coverage and better algorithm performance.

$$f_1 = \frac{V_{cover}}{V_{total}}. \quad (1)$$

### 2.2.2. Path Nodes

The number of nodes directly affects the complexity of UAV flight control, coverage effectiveness, and the execution efficiency of rescue missions. The node distribution is strongly coupled with the coverage capability of the UAV. The effective radius of a node is set as  $r = 1$ , and the 2D projected coverage area of a single node is  $S = \pi r^2 \approx 3.14$ . Given that path  $P$  consists of  $n$  nodes, the objective function  $f_2$  for minimizing path nodes can be expressed as Equation (2).

$$f_2 = n. \quad (2)$$

However, in complex three-dimensional unstructured environments, the existence of terrain undulations and obstacles may result in 3D blind spots due to line-of-sight obstruction even if a monitoring point lies within the theoretical sensing radius of the sensor, leading to failure of actual coverage. To systematically identify such blind spots, a blind area detection mechanism is introduced. Let the node coordinate be  $p_i = (x_i, y_i, z_i)$  and the monitoring point coordinate be  $q_j = (x_j, y_j, z_j)$ . The terrain surface is described by the function  $z = f(x, y)$ . The criteria for blind area judgment are as follows:

- 1) Generation of monitoring points via terrain grid sampling: The planar length and width of the simulation area satisfy  $W = H = 10$ . The discrete grid step size

is set to 0.5 for uniform grid division of the entire domain; Traverse the central coordinates  $(x_j, y_j)$  of all grids, substitute them into the terrain function  $z = f(x, y)$  to calculate the surface elevation, and construct the 3D monitoring point  $q_j = (x_j, y_j, z_j)$ .

2) Line-of-sight obstruction judgment: The target point is considered invisible if the line segment connecting the node and the monitoring point intersects the terrain surface. The parametric equation of the line segment is expressed as Equation (3).

$$\frac{x - x_i}{x_j - x_i} = \frac{y - y_i}{y_j - y_i} = \frac{z - z_i}{z_j - z_i}. \quad (3)$$

It is solved simultaneously with the terrain Equation (4), If the system of equations has a solution and the solution lies between the two points, an obstruction exists and the point is identified as a blind spot.

$$z = f(x, y). \quad (4)$$

3) Midpoint Height Verification: To further verify the obstruction, the midpoint  $M$  of the connecting line is calculated as  $M = \frac{x_i + x_j}{2}, \frac{y_i + y_j}{2}, \frac{z_i + z_j}{2}$ , and whether this midpoint is higher than the terrain surface is determined, as expressed in Equation (5).

$$\rho(p_j, q_j) = \begin{cases} 1 & \text{if } f\left(\frac{x_i + x_j}{2}, \frac{y_i + y_j}{2}\right) \geq \frac{z_i + z_j}{2} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

If  $\rho = 0$ , the midpoint is below the terrain, indicating terrain obstruction, and the target point belongs to a blind area.

Through the above blind area detection mechanism, coverage blind spots caused by terrain obstruction can be identified and avoided in path planning, thereby improving the integrity and reliability of three-dimensional full coverage.

### 2.2.3. Path Length

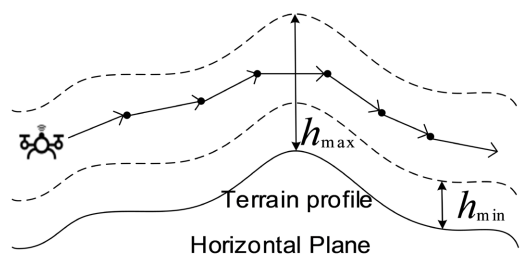
In emergency scenarios, the UAV path length is directly related to the timeliness of rescue operations and resource efficiency. A shorter flight path can effectively conserve the UAV's endurance and accelerate the response speed of search and rescue. Therefore, path length is an important indicators for evaluating path planning. The path length cost is expressed as Equation (6).

$$f_3 = \sum_{i=1}^{n-1} \|p_{i+1} - p_i\|. \quad (6)$$

where  $\|\cdot\|$  denotes the Euclidean distance.

### 2.3. Flight Height Constraint

During rescue missions, the UAV must maintain its flight height within a reasonable range and avoid significant fluctuations to ensure flight safety and stability. The schematic diagram of UAV flight height is shown in **Figure 1**.



**Figure 1.** Schematic diagram of UAV flight height.

Therefore, the flight altitude of the UAV is usually restricted and must be maintained between the minimum altitude  $h_{\min}$  and the maximum altitude  $h_{\max}$ . The formula for calculating flight altitude is expressed as Equation (7).

$$H_i = \begin{cases} \left| h_i - \frac{h_{\max} - h_{\min}}{2} \right| & h_{\min} \leq h_i \leq h_{\max} \\ \infty & \text{others} \end{cases} \quad (7)$$

To generate a high-quality full-coverage path, this paper constructs a comprehensive optimization objective that includes coverage rate, path nodes, and path length. The comprehensive objective function is formulated as a multi-objective minimization problem, as shown in Equation (8).

$$\min \{-f_1, f_2, f_3\} \text{ s.t. } H_i. \quad (8)$$

### 3. Algorithm Design

When solving the multi-objective 3D full-coverage path planning problem for UAVs, issues such as coverage blind spots, redundant paths, and poor terrain adaptability arise. The Dung Beetle Optimizer (DBO) has been widely applied in path planning problems due to its strong global search capability and good adaptability to multi-objective optimization [21]-[23]. In this section, the DBO is improved to address its limitations, including uneven population initialization, fixed parameters that cannot adapt to iterative requirements, and insufficient local search ability that easily leads to local optima. Furthermore, the improved algorithm is extended to a multi-objective version to solve the 3D full-coverage path planning problem for UAVs.

#### 3.1. Dung Beetle Optimizer Algorithm

The Dung Beetle Optimizer (DBO) is a global optimization algorithm inspired by the social behavior of dung beetles. In this algorithm, dung beetles are divided into four populations: roller beetles, breeding beetles, small beetles, and thief beetles. Each population updates its position according to specific behavioral rules, and finally outputs the current optimal solution and its fitness value. The updating mechanisms of each population are as follows:

##### 3.1.1. Roller Beetles

In the natural environment, dung beetles can roll dung balls in a straight line by

means of celestial light sources. When there is no obstacle along the path, the position update of roller beetles can be expressed as Equations (9) and (10).

$$x_i(t+1) = x_i(t) + \alpha \cdot k \cdot x_i(t-1) + b \cdot \Delta x. \quad (9)$$

$$\Delta x = |x_i(t) - X^W|. \quad (10)$$

where  $t$  denotes the current iteration number;  $x_i(t)$  represents the position of the  $i$ -th dung beetle at the  $t$ -th iteration;  $k \in (0, 0.2]$  stands for the deflection coefficient;  $b \in (0, 1)$ ;  $\alpha \in \{-1, 1\}$ ;  $\Delta x$  indicates the variation degree of light intensity; and  $X^W$  is the worst position in the current population.

If an obstacle is encountered during the ball-rolling process, the dung beetle adjusts its forward direction through a dancing behavior and updates its position according to Equation (11).

$$x_i(t+1) = x_i(t) + \tan(\theta) |x_i(t) - x_i(t-1)|. \quad (11)$$

where  $\theta \in [0, \pi]$ , The position is not updated when  $\theta = 0$ ,  $\pi/2$ , or  $\pi$ .

### 3.1.2. Breeding Beetles

During the iteration of the Dung Beetle Optimizer, breeding beetles are responsible for guiding local search to achieve fine exploitation. To ensure the safety of offspring, female dung beetles select safe oviposition areas for their progeny. Equations (12) and (13) define the boundaries.

$$Lb^* = \max(X^* \cdot (1-R), Lb). \quad (12)$$

$$Ub^* = \min(X^* \cdot (1+R), Ub). \quad (13)$$

where  $X^*$  denotes the current local optimal position;  $Lb^*$  and  $Ub^*$  correspond to the lower and upper bounds of the oviposition area, respectively;  $R = (1-t)/T_{\max}$ , in which  $T_{\max}$  is the preset maximum number of iterations of the algorithm;  $Lb$  and  $Ub$  represent the lower and upper bounds of the optimization problem, respectively.

After the oviposition area is determined, each female breeding beetle produces only one egg per iteration, and the position update of its brood ball follows Equation (14).

$$B_i(t+1) = X^* + b_1 \cdot (B_i(t) - Lb^*) + b_2 \cdot (B_i(t) - Ub^*). \quad (14)$$

where  $b_1$  and  $b_2$  are independent random vectors with a dimension of  $1 \times D$ , and  $D$  denotes the dimension of the problem.

### 3.1.3. Small Beetles

After maturation, small beetles leave the nest to forage. The boundaries of their optimal foraging area are defined as:

$$Lb^b = \max(X^b \cdot (1-R), Lb). \quad (15)$$

$$Ub^b = \min(X^b \cdot (1+R), Ub). \quad (16)$$

where  $X^b$  denotes the global optimal position.

The position update formula for small beetles is given in Equation (17).

$$x_i(t+1) = x_i(t) + C_1 \cdot (x_i(t) - Lb^b) + C_2 \cdot (x_i(t) - Ub^b). \quad (17)$$

where  $C_1$  is a random number following a normal distribution;  $C_2$  is a random vector in the range of (0, 1);  $Lb^b$  and  $Ub^b$  represent the lower and upper bounds of the foraging area, respectively.

#### 3.1.4. Thief Beetles

Some dung beetles obtain food by stealing dung balls from other individuals. Assuming  $X^b$  is the current optimal food source, thief beetles tend to move around it. Their position update formula is presented in Equation (18).

$$x_i(t+1) = X^b + S \cdot g \cdot (|x_i(t) - X^*| + |x_i(t) - X^b|). \quad (18)$$

where  $S$  is a fixed constant, and  $g$  is a  $1 \times D$ -dimensional random vector following a normal distribution.

### 3.2. Improved Dung Beetle Optimizer Algorithm

#### 3.2.1. Improved Initialization Strategy

The standard Dung Beetle Optimizer uses random initialization to generate the initial population, which may result in an uneven distribution of individuals in the search space and degrade the algorithm's global exploration ability in the early stage. In this paper, the Latin Hypercube Sampling (LHS) strategy is introduced for population initialization to ensure that sample points are uniformly distributed across each dimension, thereby improving the global exploration efficiency of the algorithm at the early stage. The initialization formula is given in Equation (19).

$$P_{ij} = Lb_j + LHS_{ij} \cdot (Ub_j - Lb_j). \quad (19)$$

where  $LHS_{ij} \in [0, 1]$  denotes uniformly distributed sampling points;  $Lb_j$  is the lower bound of the  $j$ -th dimension;  $Ub_j$  is the upper bound of the  $j$ -th dimension; and  $P_{ij}$  represents the value of the  $i$ -th individual in the  $j$ -th dimension of the initial population.

#### 3.2.2. Dynamic Parameter Adaptation

In the standard Dung Beetle Optimizer, the position update of roller beetles is the critical component of global exploration. In the original position update formula for roller beetles, the step size coefficient  $b$  and deflection coefficient  $k$  are fixed values, which cannot adapt to the search requirements at different iteration stages. In this paper, a dynamic adaptive mechanism is introduced to adjust parameters adaptively during iterations, so as to balance global exploration and local exploitation capabilities. The improved expressions are presented in Equations (20) and (21).

$$b(t) = 0.4 \cdot \exp\left(-\left(\frac{5t}{T_{\max}}\right)^2\right). \quad (20)$$

$$k(t) = 0.5 + 0.1 \cdot \frac{t}{T_{\max}}. \quad (21)$$

where the step size coefficient uses a Gaussian decay strategy. It maintains a large value in the early iteration stage to enhance global exploration, and gradually decays in the later stage to strengthen local exploitation. The deflection coefficient adopts a linear growth strategy to continuously improve the convergence accuracy of the algorithm.

### 3.2.3. Sine-Cosine Hybrid Disturbance Strategy

In the standard Dung Beetle Optimizer, the position update of small beetles only relies on normally distributed random numbers and boundary constraints, lacking a local disturbance mechanism. Thus, the algorithm tends to fall into local optima in late iterations stage and fails to conduct fine search around the optimal solution. In this paper, a sine-cosine hybrid disturbance strategy is introduced. Because of its periodic fluctuation characteristics, this strategy enhances local exploitation and escape from local optima. The strategy adaptively selects disturbance modes according to different algorithm iteration stages, and its formula is given in Equation (22).

$$\delta(t) = \begin{cases} a(t) \cdot \sin(r_2) \cdot r_3 & t < \frac{T_{\max}}{3} \\ a(t) \cdot \frac{\sin(r_2) + \cos r_2}{2} \cdot r_3 & \frac{T_{\max}}{3} \leq t \leq \frac{2T_{\max}}{3} \\ a(t) \cdot \cos(r_2) \cdot r_3 & \text{others} \end{cases} \quad (22)$$

where  $a(t) = 2 - t/T_{\max}$  denotes the decay factor, and  $r_2, r_3$  are random numbers. In the early iteration stage, this strategy mainly adopts sine disturbance to enhance global exploration; in the middle stage, hybrid disturbance is used to balance exploration and exploitation; in the late stage, cosine disturbance is adopted to strengthen local fine search. The position update formula of small beetles after disturbance is given as follows:

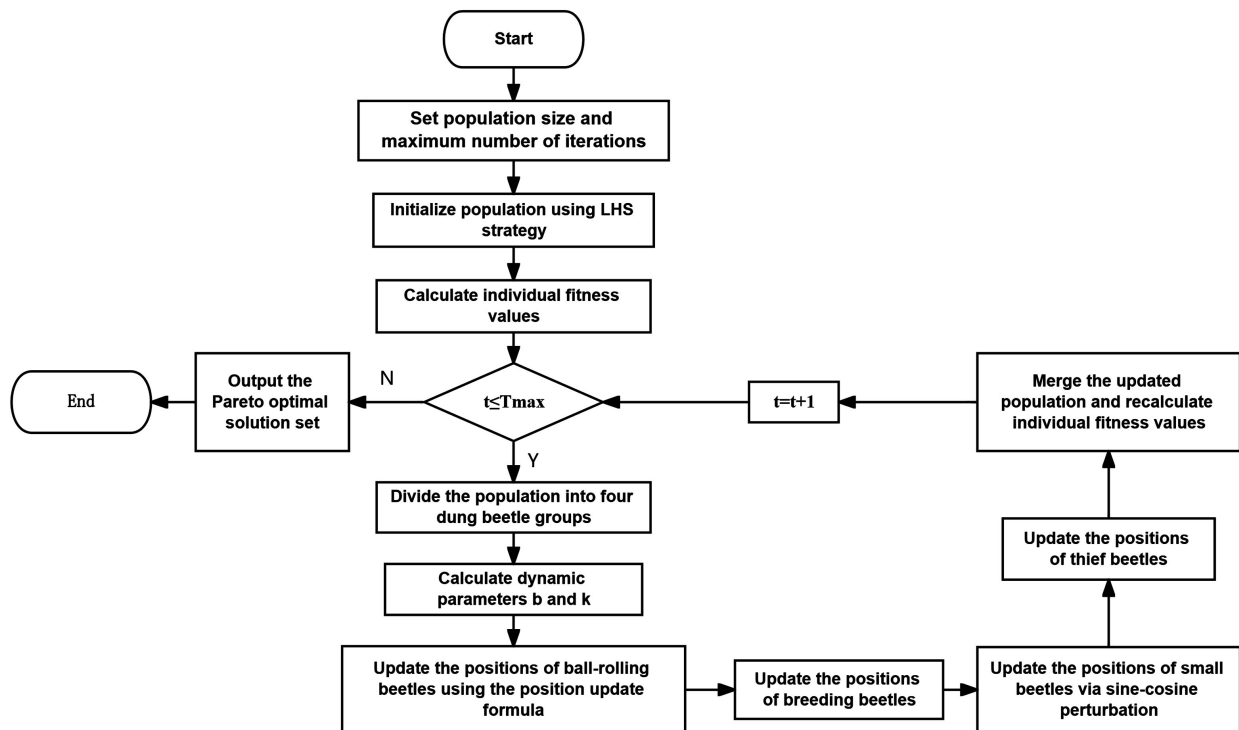
$$x_i(t+1) = x_i(t) + C_1 \cdot (x_i(t) - Lb^b) + C_2 \cdot (x_i(t) - Ub^b) + 0.1 \cdot \delta(t) \cdot (Ub^b - Lb^b). \quad (23)$$

Aiming at the shortcomings of the standard Dung Beetle Optimizer applied in 3D path planning, improvements are made from three aspects: population initialization, parameter design and disturbance strategy. The LHS strategy improves population diversity and avoids aggregation of initial coverage nodes; dynamic adaptive parameters balance global exploration and local exploitation; the sine-cosine disturbance enhances local search ability and the capability to escape from local coverage blind spots. The collaborative effect of the three improvements significantly promotes the convergence speed, accuracy and robustness of the algorithm in complex 3D space, and provides effective support for generating high-quality full-coverage paths.

### 3.3. Algorithm Flow

At the initial stage of the algorithm, the population size and the maximum number

of iterations are set, and the population is initialized using the LHS strategy. Afterwards, the fitness value of each individual is calculated. It is judged whether the current iteration number reaches the preset maximum iterations. If not, the population is divided into four subpopulations: roller beetles, breeding beetles, small beetles and thief beetles. The dynamic parameters  $b$  and  $k$  are calculated. Then the positions of roller beetles, breeding beetles, small beetles (updated with sine-cosine disturbance) and thief beetles are updated in sequence according to their respective position update formulas. After all subpopulation updates are completed, individuals are merged to form a new generation population. The individual fitness values are recalculated, the iteration number is increased by 1, and the above process is repeated. If the current iteration reaches the maximum number of iterations, the loop terminates, and the final Pareto-optimal solution set is output. The flow chart of the IMODBO algorithm is shown in **Figure 2**.



**Figure 2.** Flow chart of the IMODBO algorithm.

## 4. Experiments

### 4.1. Experimental Setup

To test the comprehensive performance of the IMODBO algorithm, the CEC2020 test suite is adopted in this study. This test suite contains 24 standard test functions that effectively simulate the challenges of nonlinearity, multimodality and complex spatial structures in practical optimization problems. Multi-environment evaluation can systematically verify IMODBO's core capabilities for maintaining population diversity, avoiding premature convergence, and balancing

global exploration and local exploitation. It is especially suitable for verifying IMODBO's robustness in solving multimodal and multi-constraint problems in complex 3D path planning.

To comprehensively evaluate the proposed IMODBO algorithm, comparative experiments are conducted against six multi-objective optimization algorithms, including three classical algorithms: Non-dominated Sorting Genetic Algorithm II (NSGA-II) [24], Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [25], Multi-objective Whale Optimization Algorithm (MOWOA) [26], and three state-of-the-art algorithms in recent years: Multi-objective Sparrow Search Algorithm (MOSSA) [27], Multi-objective Grey Wolf Optimizer (MOGWO) [28], and Multi-objective Dung Beetle Optimizer (MODBO) [29]. For controlled-variable comparisons, all comparative experiments in this study are conducted within a unified hardware and software environment. The detailed experimental platform configuration is as follows: Windows 11 (64-bit) operating system, Intel Core i5-13490F processor, and algorithms implemented in MATLAB 2022b. Meanwhile, systematic parameter tuning and verification are performed on the key parameters of all compared algorithms to minimize deviations caused by parameter configurations. The final parameter settings are listed in Table 2. Under the unified experimental environment and optimized parameter settings, the performance results of all compared algorithms are comparable, which avoids potential interference from differences in systems and configurations.

**Table 2.** Parameter settings of seven compared algorithms.

| Algorithm | Parameter Settings                                                               |
|-----------|----------------------------------------------------------------------------------|
| NSGAII    | $P_c = 0.8$ , $P_r = 0.1$                                                        |
| MOEA/D    | $archive_{size} = 100$ , $T = 0.1$ , $n_r = 0.01$ , $\delta = 0.9$ , $\eta = 30$ |
| MOWOA     | No parameter setting                                                             |
| MOSSA     | $R_2 = 0.8$ , $SD = 0.1$ , $PD = 0.2$                                            |
| MOGWO     | $nGrid = 10$ , $\alpha = 0.1$ , $\beta = 10$ , $\gamma = 2$                      |
| MODBO     | $K = 0.2$ , $b = 0.2$ , and $S = 0.25$                                           |
| IMODBO    | $K = 0.2$ , $b = 0.2$ , and $S = 0.25$                                           |

## 4.2. Experimental Metrics

To verify the optimization performance of the proposed algorithm and conduct a fair comparison with the selected compared algorithms, two metrics, namely Inverted Generational Distance (IGD) and Purity-Spread Performance (PSP), are adopted in this paper to quantitatively evaluate the quality of Pareto front solution sets generated by each algorithm from multiple dimensions.

### 4.2.1. IGD

The IGD metric measures the convergence and uniformity of the approximate Pareto front obtained by the algorithm relative to the true Pareto front. A smaller IGD value indicates that the solution set obtained by the algorithm is closer to the

true Pareto front and is distributed more uniformly along the front. Its calculation formula is given in Equation (24):

$$\text{IGD}(P^*, A) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|}. \quad (24)$$

where  $P^*$  is the uniformly sampled point set on the true Pareto front,  $A$  is the approximate solution set obtained by the algorithm, and  $d(v, A)$  denotes the Euclidean distance from point  $v$  to the nearest point in solution set  $A$ .

#### 4.2.2. PSP

PSP is a comprehensive metric for evaluating the solution set quality of multi-objective optimization algorithms, accounting for both purity and spread. PSP is defined as the ratio of Coverage Rate (CR) to decision-space inverted generational distance (IGD<sub>x</sub>). Its calculation formula is given in Equation (25):

$$\text{PSP}(A) = \frac{\text{CR}(A)}{\text{IGD}_x(A)}. \quad (25)$$

where CR measures the proportion of true Pareto optimal solutions obtained by the algorithm, reflecting the purity of the solution set. A larger CR value indicates more true optimal solutions acquired by the algorithm and more complete path coverage. IGD<sub>x</sub> measures the proximity between the algorithm's solution set and the true Pareto optimal solution set in the decision space. A smaller IGD<sub>x</sub> value means that the paths generated by the algorithm are geometrically closer to the theoretically optimal paths in physical space, corresponding to a better PSP performance.

### 4.3. Experimental Results

In this experiment, to ensure the fairness and statistical reliability of the results, each algorithm is independently run 10 times on each test problem with 500 iterations. **Table 3** and **Table 4** respectively report the mean and variance of IGD and PSP metrics for IMODBO and six compared algorithms on the CEC2020 test suite.

**Table 3.** IGD values obtained by IMODBO and other compared algorithms.

| function |      | NSGAI    | MOEA/D   | MOWOA    | MOSSA    | MOGWO    | MODBO    | IMODBO   |
|----------|------|----------|----------|----------|----------|----------|----------|----------|
| MMF1     | Mean | 2.82E-03 | 4.19E-03 | 1.62E-02 | 2.24E-03 | 9.74E-03 | 2.49E-03 | 2.30E-03 |
|          | SD   | 7.08E-05 | 1.30E-04 | 7.02E-04 | 6.64E-05 | 3.81E-03 | 6.96E-05 | 1.03E-04 |
| MMF2     | Mean | 7.45E-03 | 3.77E-03 | 1.05E-01 | 5.70E-03 | 4.14E-02 | 1.06E-02 | 8.69E-03 |
|          | SD   | 5.24E-03 | 3.37E-05 | 4.12E-02 | 7.29E-04 | 2.89E-02 | 1.75E-03 | 7.08E-04 |
| MMF4     | Mean | 2.65E-03 | 3.81E-03 | 9.73E-03 | 2.39E-03 | 9.89E-03 | 2.29E-03 | 2.19E-03 |
|          | SD   | 1.56E-04 | 1.75E-05 | 1.40E-03 | 1.33E-04 | 2.27E-03 | 1.60E-04 | 1.47E-04 |
| MMF5     | Mean | 3.94E-03 | 4.04E-03 | 1.45E-02 | 2.22E-03 | 1.87E-03 | 2.43E-03 | 2.28E-03 |
|          | SD   | 4.10E-03 | 7.99E-05 | 2.05E-03 | 6.95E-05 | 9.25E-05 | 8.71E-05 | 4.09E-05 |

## Continued

|           |      |          |          |          |          |          |          |          |
|-----------|------|----------|----------|----------|----------|----------|----------|----------|
| MMF7      | Mean | 2.61E-03 | 3.88E-03 | 1.08E-02 | 2.32E-03 | 3.72E-03 | 2.27E-03 | 2.25E-03 |
|           | SD   | 8.26E-05 | 6.83E-05 | 8.64E-04 | 6.29E-05 | 3.91E-04 | 6.72E-05 | 5.98E-05 |
| MMF8      | Mean | 2.58E-03 | 4.21E-03 | 2.45E-02 | 2.31E-03 | 2.78E-03 | 2.58E-03 | 2.50E-03 |
|           | SD   | 1.11E-04 | 3.63E-06 | 4.99E-03 | 8.23E-05 | 4.57E-03 | 8.58E-05 | 6.58E-05 |
| MMF10     | Mean | 2.09E-01 | 2.56E-01 | 2.62E-01 | 8.66E-02 | 3.65E-02 | 6.69E-02 | 5.07E-02 |
|           | SD   | 1.10E-01 | 1.13E-01 | 4.07E-02 | 2.26E-02 | 5.53E-03 | 9.16E-03 | 8.56E-03 |
| MMF11     | Mean | 2.13E-02 | 7.76E-02 | 7.57E-02 | 9.61E-03 | 1.06E-02 | 9.73E-03 | 9.44E-03 |
|           | SD   | 1.40E-03 | 1.16E-04 | 1.16E-02 | 5.08E-04 | 8.78E-04 | 3.98E-04 | 5.42E-04 |
| MMF12     | Mean | 9.98E-03 | 6.72E-03 | 5.93E-02 | 5.59E-03 | 2.85E-03 | 3.78E-03 | 3.05E-03 |
|           | SD   | 1.19E-02 | 5.76E-04 | 1.92E-02 | 1.04E-03 | 3.59E-04 | 6.66E-04 | 3.49E-04 |
| MMF13     | Mean | 2.76E-02 | 4.64E-01 | 1.24E-01 | 1.58E-02 | 1.66E-02 | 1.40E-02 | 1.20E-02 |
|           | SD   | 1.42E-03 | 9.92E-04 | 2.70E-02 | 2.64E-03 | 1.72E-03 | 8.55E-04 | 3.41E-04 |
| MMF14     | Mean | 1.34E-01 | 2.48E-01 | 1.58E-01 | 1.02E-01 | 1.03E-01 | 9.12E-02 | 9.19E-02 |
|           | SD   | 6.06E-03 | 1.64E-02 | 9.03E-03 | 4.23E-03 | 1.83E-02 | 2.71E-03 | 2.20E-03 |
| MMF15     | Mean | 1.99E-01 | 2.95E-01 | 1.95E-01 | 1.10E-01 | 9.47E-02 | 9.26E-02 | 9.20E-02 |
|           | SD   | 2.45E-02 | 1.99E-02 | 1.97E-02 | 4.70E-03 | 1.41E-02 | 2.70E-03 | 2.10E-03 |
| MMF1_e    | Mean | 2.38E-02 | 5.85E-02 | 8.61E-02 | 3.11E-03 | 9.14E-03 | 6.05E-03 | 5.77E-03 |
|           | SD   | 2.50E-02 | 2.16E-02 | 3.04E-02 | 2.79E-04 | 2.43E-03 | 4.63E-04 | 5.14E-04 |
| MMF14_a   | Mean | 1.44E-01 | 2.34E-01 | 1.45E-01 | 1.07E-01 | 1.36E-01 | 9.71E-02 | 9.62E-02 |
|           | SD   | 9.50E-03 | 1.74E-02 | 9.78E-03 | 6.24E-03 | 3.49E-02 | 5.94E-03 | 3.81E-03 |
| MMF15_a   | Mean | 2.29E-01 | 2.90E-01 | 1.88E-01 | 1.15E-01 | 1.17E-01 | 9.85E-02 | 9.57E-02 |
|           | SD   | 1.52E-02 | 1.88E-02 | 1.64E-02 | 7.31E-03 | 2.56E-02 | 2.73E-03 | 1.64E-03 |
| MMF10_l   | Mean | 2.82E-01 | 2.20E-01 | 2.22E-01 | 2.29E-01 | 2.22E-01 | 2.32E-01 | 2.13E-01 |
|           | SD   | 2.72E-02 | 1.34E-02 | 5.01E-02 | 1.13E-02 | 7.34E-03 | 9.03E-03 | 5.08E-03 |
| MMF11_l   | Mean | 9.50E-02 | 1.69E-01 | 1.28E-01 | 9.30E-02 | 9.59E-02 | 9.43E-02 | 9.30E-02 |
|           | SD   | 2.89E-04 | 1.08E-04 | 1.45E-02 | 4.04E-04 | 1.18E-03 | 1.22E-03 | 1.79E-04 |
| MMF12_l   | Mean | 8.59E-02 | 9.09E-02 | 1.25E-01 | 8.87E-02 | 8.56E-02 | 8.50E-02 | 8.48E-02 |
|           | SD   | 1.46E-04 | 5.24E-04 | 1.35E-02 | 3.06E-03 | 4.24E-04 | 9.28E-04 | 9.75E-04 |
| MMF13_l   | Mean | 1.49E-01 | 6.26E-01 | 1.97E-01 | 1.39E-01 | 1.53E-01 | 1.51E-01 | 1.47E-01 |
|           | SD   | 8.81E-04 | 7.41E-04 | 1.28E-02 | 1.17E-02 | 3.82E-03 | 3.93E-03 | 1.04E-03 |
| MMF15_l1  | Mean | 2.27E-01 | 2.66E-01 | 2.34E-01 | 1.87E-01 | 2.08E-01 | 1.87E-01 | 1.90E-01 |
|           | SD   | 8.01E-03 | 1.73E-02 | 1.14E-02 | 2.94E-03 | 1.74E-02 | 2.89E-03 | 4.55E-03 |
| MMF15_a_l | Mean | 2.32E-01 | 2.79E-01 | 2.28E-01 | 2.00E-01 | 1.97E-01 | 1.90E-01 | 1.89E-01 |
|           | SD   | 8.54E-03 | 2.02E-02 | 9.40E-03 | 5.25E-03 | 1.79E-02 | 3.20E-03 | 3.00E-03 |
| MMF16_l1  | Mean | 1.91E-01 | 2.81E-01 | 2.21E-01 | 1.65E-01 | 1.75E-01 | 1.62E-01 | 1.62E-01 |
|           | SD   | 6.99E-03 | 1.76E-02 | 8.99E-03 | 3.25E-03 | 1.89E-02 | 1.54E-03 | 3.03E-03 |
| MMF16_l2  | Mean | 2.66E-01 | 2.69E-01 | 2.61E-01 | 2.88E-01 | 2.61E-01 | 2.33E-01 | 2.32E-01 |
|           | SD   | 1.48E-02 | 1.40E-02 | 1.23E-02 | 5.64E-03 | 1.77E-02 | 4.98E-03 | 4.42E-03 |
| MMF16_l3  | Mean | 2.23E-01 | 2.83E-01 | 2.42E-01 | 1.97E-01 | 2.13E-01 | 1.97E-01 | 1.98E-01 |
|           | SD   | 9.82E-03 | 1.53E-02 | 8.63E-03 | 4.09E-03 | 1.68E-02 | 2.61E-03 | 4.55E-03 |

**Table 4.** PSP values obtained by IMODBO and other compared algorithms.

| Function |      | NSGAI    | MOEA/D   | MOWOA    | MOSSA    | MOGWO    | MODBO    | IMODBO   |
|----------|------|----------|----------|----------|----------|----------|----------|----------|
| MMF1     | Mean | 9.14E+00 | 6.81E+00 | 7.85E+00 | 1.15E+01 | 8.38E+00 | 1.89E+01 | 1.85E+01 |
|          | SD   | 1.39E+00 | 2.28E+00 | 1.09E+00 | 2.79E+00 | 1.36E+00 | 9.10E-01 | 7.66E-01 |
| MMF2     | Mean | 1.35E+01 | 9.44E+00 | 1.21E+01 | 2.40E+01 | 6.34E+00 | 2.06E+01 | 2.92E+01 |
|          | SD   | 7.64E+00 | 4.15E+00 | 4.38E+00 | 1.73E+01 | 3.87E+00 | 4.32E+00 | 3.87E+00 |
| MMF4     | Mean | 7.23E+00 | 3.66E+00 | 1.19E+01 | 1.53E+01 | 9.70E+00 | 2.04E+01 | 2.42E+01 |
|          | SD   | 1.70E+00 | 1.27E+00 | 1.45E+00 | 3.52E+00 | 1.96E+00 | 1.44E+00 | 1.81E+00 |
| MMF5     | Mean | 4.93E+00 | 3.24E+00 | 5.14E+00 | 8.55E+00 | 4.52E+00 | 1.01E+01 | 2.00E+01 |
|          | SD   | 6.75E-01 | 7.71E-01 | 5.12E-01 | 1.15E+00 | 9.80E-01 | 9.19E-01 | 4.61E-01 |
| MMF7     | Mean | 1.35E+01 | 6.99E+00 | 1.36E+01 | 2.84E+01 | 1.38E+01 | 2.87E+01 | 7.11E+01 |
|          | SD   | 2.83E+00 | 3.75E+00 | 2.16E+00 | 4.41E+00 | 2.25E+00 | 3.00E+00 | 1.65E+00 |
| MMF8     | Mean | 7.09E-01 | 5.99E-01 | 2.58E+00 | 1.01E+00 | 9.50E-01 | 4.58E+00 | 7.18E+00 |
|          | SD   | 3.25E-01 | 3.08E-01 | 4.91E-01 | 7.96E-01 | 5.47E-01 | 1.77E+00 | 1.72E+00 |
| MMF10    | Mean | 2.63E+01 | 6.63E+01 | 1.54E+01 | 3.52E+01 | 5.51E+01 | 4.42E+01 | 4.55E+01 |
|          | SD   | 6.70E+01 | 1.35E+02 | 9.66E+00 | 9.02E+00 | 1.11E+01 | 9.27E+00 | 8.49E+00 |
| MMF11    | Mean | 1.73E+02 | 2.60E+02 | 3.21E+01 | 2.45E+02 | 5.49E+01 | 1.62E+02 | 1.52E+02 |
|          | SD   | 1.72E+01 | 2.93E+00 | 1.03E+01 | 6.77E+00 | 7.42E+00 | 1.16E+01 | 1.08E+01 |
| MMF12    | Mean | 3.48E+02 | 3.89E+02 | 3.89E+01 | 4.17E+02 | 3.74E+01 | 2.83E+02 | 3.13E+02 |
|          | SD   | 1.58E+02 | 1.51E+01 | 3.15E+01 | 6.01E+01 | 1.24E+01 | 3.79E+01 | 6.92E+00 |
| MMF13    | Mean | 9.97E+00 | 1.27E+01 | 8.16E+00 | 2.36E+01 | 1.20E+01 | 2.64E+01 | 2.69E+01 |
|          | SD   | 2.44E+00 | 2.98E+00 | 1.36E+00 | 2.57E+00 | 4.01E+00 | 3.09E+00 | 1.49E+00 |
| MMF14    | Mean | 7.80E+00 | 7.96E+00 | 9.04E+00 | 9.40E+00 | 6.91E+00 | 1.38E+01 | 1.43E+01 |
|          | SD   | 7.97E-01 | 4.70E-01 | 5.82E-01 | 9.28E-01 | 1.70E+00 | 7.04E-01 | 4.66E-01 |
| MMF15    | Mean | 9.59E+00 | 8.29E+00 | 1.05E+01 | 1.72E+01 | 1.40E+01 | 2.01E+01 | 2.17E+01 |
|          | SD   | 1.21E+00 | 6.20E-01 | 1.32E+00 | 1.12E+00 | 1.13E+00 | 4.97E-01 | 6.31E-01 |
| MMF1_e   | Mean | 3.27E-01 | 2.18E+00 | 1.76E+00 | 9.18E-01 | 2.09E-01 | 2.89E+00 | 3.18E+00 |
|          | SD   | 4.24E-01 | 2.34E-01 | 7.35E-01 | 5.10E-01 | 7.85E-01 | 7.61E-01 | 1.51E-01 |
| MMF14_a  | Mean | 5.94E+00 | 7.51E+00 | 7.80E+00 | 8.07E+00 | 5.62E+00 | 9.63E+00 | 1.07E+01 |
|          | SD   | 6.80E-01 | 1.45E-01 | 4.50E-01 | 6.52E-01 | 1.95E+00 | 1.28E+00 | 7.82E-01 |
| MMF15_a  | Mean | 6.60E+00 | 7.77E+00 | 9.31E+00 | 1.42E+01 | 8.70E+00 | 1.62E+01 | 1.68E+01 |
|          | SD   | 6.63E-01 | 4.49E-01 | 9.73E-01 | 1.52E+00 | 1.68E+00 | 4.49E-01 | 3.97E-01 |
| MMF10_l  | Mean | 2.84E+00 | 3.89E-01 | 2.91E+00 | 1.02E+00 | 3.95E+00 | 3.08E+00 | 1.48E+01 |
|          | SD   | 3.33E+00 | 3.54E-01 | 4.70E+00 | 2.41E+00 | 6.03E-01 | 8.98E-01 | 1.07E+00 |
| MMF11_l  | Mean | 3.68E-01 | 1.30E-01 | 3.42E+00 | 6.01E-01 | 3.60E+00 | 3.28E+00 | 9.04E+00 |
|          | SD   | 1.01E-01 | 5.15E-02 | 3.94E+00 | 3.38E-02 | 3.52E-01 | 6.57E-01 | 7.09E-01 |

## Continued

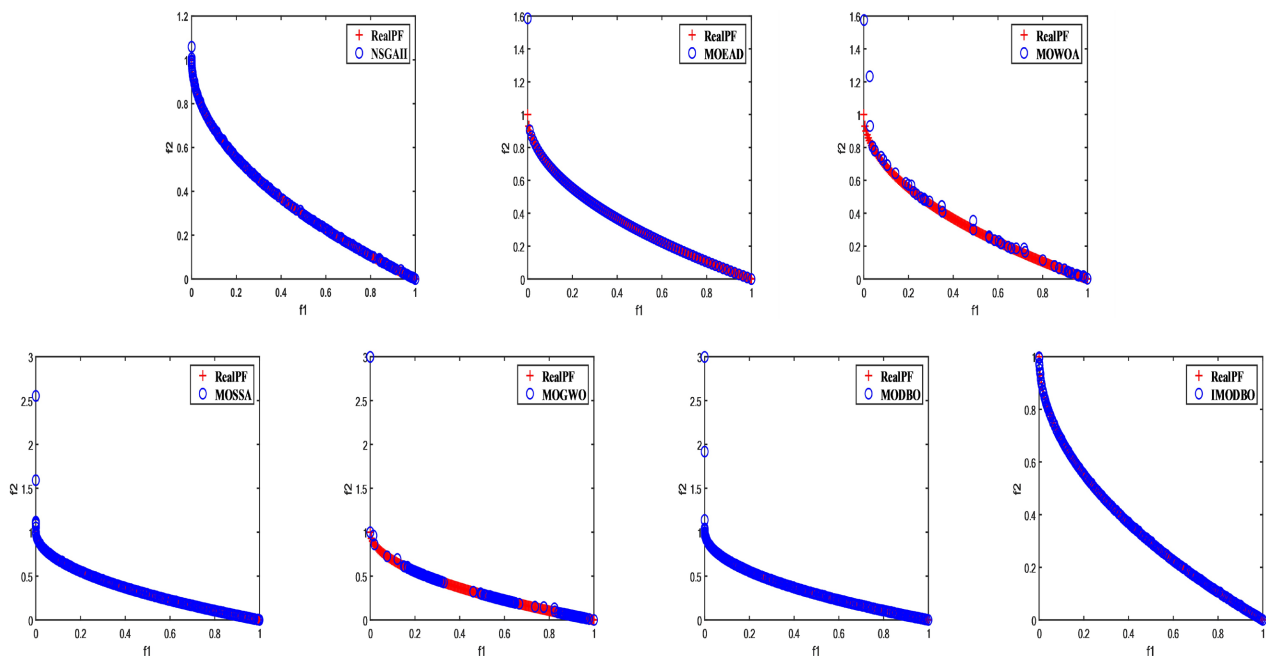
|           |      |          |          |          |          |          |          |          |
|-----------|------|----------|----------|----------|----------|----------|----------|----------|
| MMF12_l   | Mean | 2.47E-01 | 7.59E-02 | 1.33E+01 | 8.23E-01 | 3.22E+00 | 3.42E+00 | 3.63E+00 |
|           | SD   | 6.36E-02 | 8.37E-02 | 5.37E+00 | 1.63E-01 | 3.24E-01 | 4.99E-01 | 8.16E-01 |
| MMF13_l   | Mean | 1.25E+00 | 1.48E+00 | 4.31E+00 | 1.76E+00 | 2.32E+00 | 2.41E+00 | 2.56E+00 |
|           | SD   | 2.74E-01 | 1.99E-01 | 7.32E-01 | 5.27E-02 | 2.26E-01 | 1.36E-01 | 1.31E-01 |
| MMF15_l1  | Mean | 3.07E+00 | 4.54E+00 | 6.45E+00 | 5.47E+00 | 4.16E+00 | 4.72E+00 | 8.10E+00 |
|           | SD   | 1.38E+00 | 1.15E+00 | 7.30E-01 | 5.02E-01 | 1.05E+00 | 6.96E-01 | 4.69E-01 |
| MMF15_a_l | Mean | 3.28E+00 | 4.88E+00 | 6.45E+00 | 5.04E+00 | 5.49E+00 | 4.59E+00 | 7.35E+00 |
|           | SD   | 4.11E-01 | 4.37E-01 | 7.59E-01 | 2.68E-01 | 1.58E+00 | 6.32E-01 | 1.73E-01 |
| MMF16_l1  | Mean | 4.19E+00 | 6.29E+00 | 6.79E+00 | 6.49E+00 | 5.41E+00 | 6.60E+00 | 7.95E+00 |
|           | SD   | 8.75E-01 | 2.30E-01 | 5.28E-01 | 3.02E-01 | 1.08E+00 | 4.59E-01 | 6.46E-01 |
| MMF16_l2  | Mean | 1.94E+00 | 7.90E+00 | 5.58E+00 | 4.43E+00 | 2.95E+00 | 3.52E+00 | 3.27E+00 |
|           | SD   | 1.09E+00 | 7.60E-01 | 8.92E-01 | 7.89E-01 | 6.94E-01 | 8.31E-01 | 6.19E-01 |
| MMF16_l3  | Mean | 3.18E+00 | 4.84E+00 | 5.81E+00 | 5.39E+00 | 4.07E+00 | 5.15E+00 | 7.78E+00 |
|           | SD   | 5.37E-01 | 2.83E-01 | 8.03E-01 | 3.14E-01 | 1.39E+00 | 3.52E-01 | 6.00E-01 |

**Table 3** shows that IMODBO exhibits better convergence performance and distribution characteristics across most test functions. Its average IGD values on MMF1, MMF8, and MMF16\_l3 are close to the optimal average IGD values. On MMF10, MMF12, MMF14, and MMF1\_e, the average IGD values of IMODBO are better than those of all compared algorithms except MOGWO. Due to the complex characteristics of these functions, such as strong multimodality, discontinuity, or structural transformation, MOGWO demonstrates stronger global multimodal search capability and greater adaptability to complex structures through its distributed leadership mechanism based on social hierarchy.

The results in **Table 4** show that IMODBO outperforms the compared algorithms on most test functions. However, on problems with linear Pareto sets such as MMF10, MMF11, and MMF12, the average PSP value of IMODBO is slightly lower than that of MOEA/D. This is because the decomposition strategy of MOEA/D systematically searches along predefined directions to efficiently obtain uniformly distributed solution sets, whereas IMODBO relies on group competition and random disturbances, lacking a directional guidance mechanism for linear structures. Its diversity strategy, in contrast, reduces search efficiency and the quality of the solution set distribution in such problems, leading to insufficient performance on the PSP metric. IMODBO ranks second only to MOWOA on MMF12\_l and MMF13\_l, mainly because these two types of functions have highly asymmetric, skewed Pareto fronts that place extremely high demands on the algorithm's exploration capability in sparse regions. MOWOA can more effectively cover these sparse regions through its strong global search mechanism, which combines spiral updates and random walks. Although IMODBO introduces a sine-cosine disturbance to enhance diversity, its exploration efficiency in sparse

regions is insufficient in this scenario, leading to poor performance. In summary, the IGD and PSP values of IMODBO are superior to or comparable to those of other compared algorithms on most test functions. Although there is room for improvement in some optimization scenarios with specific structural characteristics, overall, it is suitable for most multi-objective optimization needs in complex 3D path planning and can provide reliable support for efficient full-coverage operations of UAVs.

The algorithm's performance can be further verified through visual analysis of convergence curves. **Figures 3-6**, respectively, show the PS (Population Spread) or PF (Pareto Front) of the IMODBO algorithm and the comparison algorithms for the MMF1, MMF8, MMF15, and MMF14\_a test functions. The PS graph reflects the dynamic distribution state of the population during the optimization process, while the PF graph reflects the convergence and distribution characteristics of the Pareto front obtained by the algorithm in the objective space. **Figure 3** shows that on the MMF1 test function, the PF convergence curve of IMODBO converges to the global optimal region faster than other algorithms and maintains good population diversity. **Figure 4** presents the PFs of each algorithm on the MMF15 test function, where the PF curves perform well in both convergence and coverage, especially being closer to the true front in the objective space and exhibiting a wider distribution. **Figure 5** shows the PS of each algorithm on MMF8; solutions generated by IMODBO exhibit a wider coverage range and a more complete structure, thereby effectively avoiding local optima. **Figure 6** presents the PS of the IMODBO algorithm on the MMF14\_a test function, and its solution set still effectively approaches the true front with more complete coverage, even in a complex multi-modal environment.



**Figure 3.** PF of compared algorithms on MMF1.

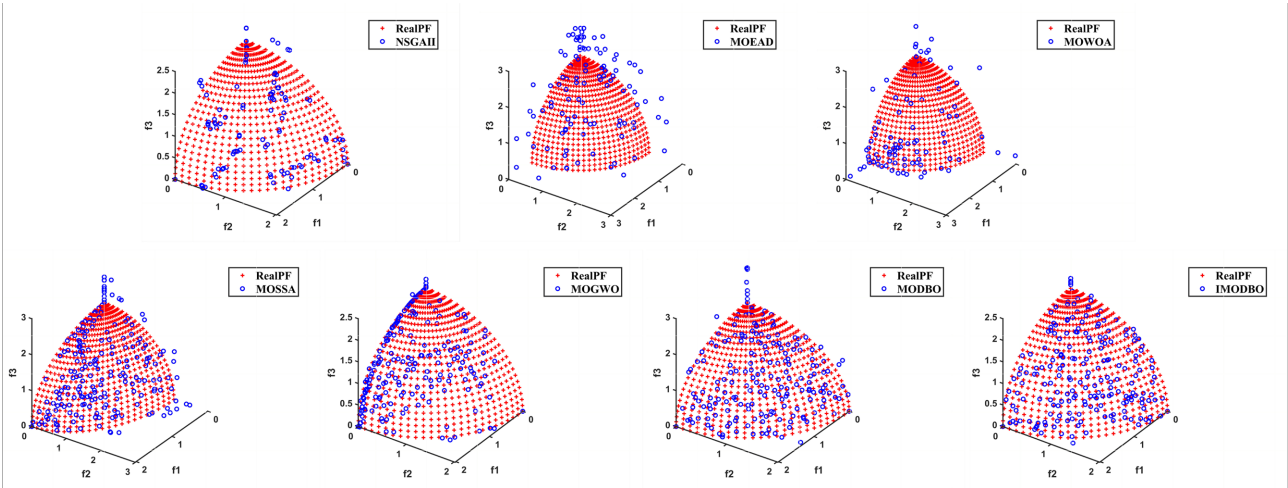


Figure 4. PF of compared algorithms on MMF15.

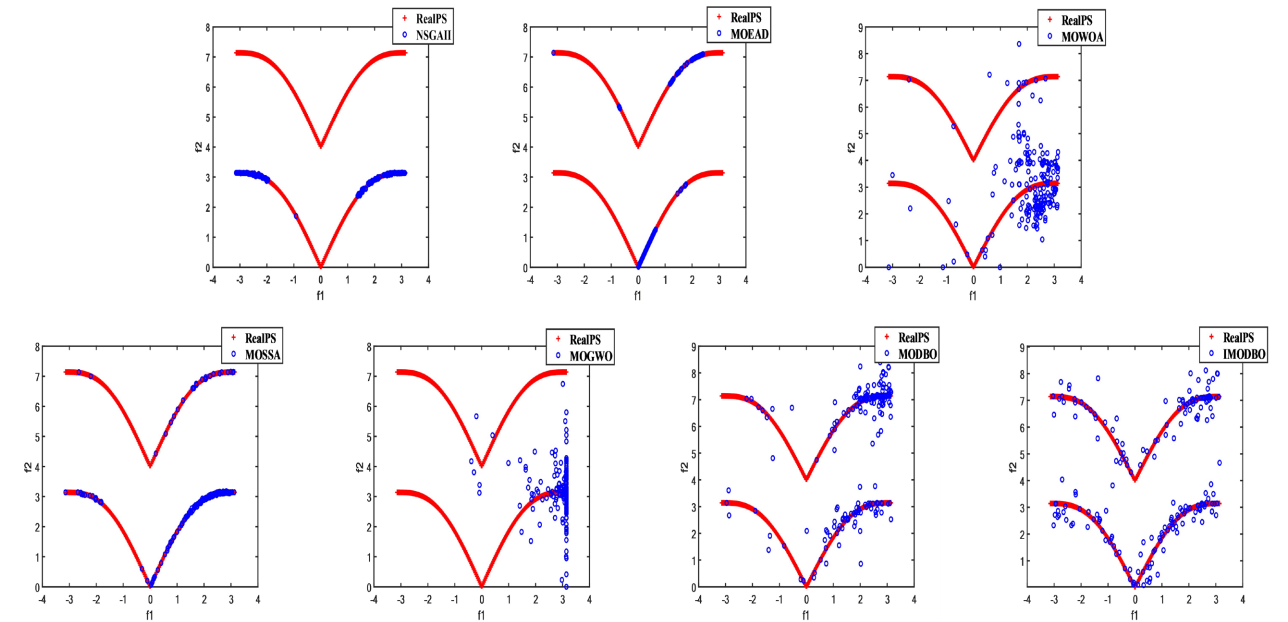


Figure 5. PS of compared algorithms on MMF8.

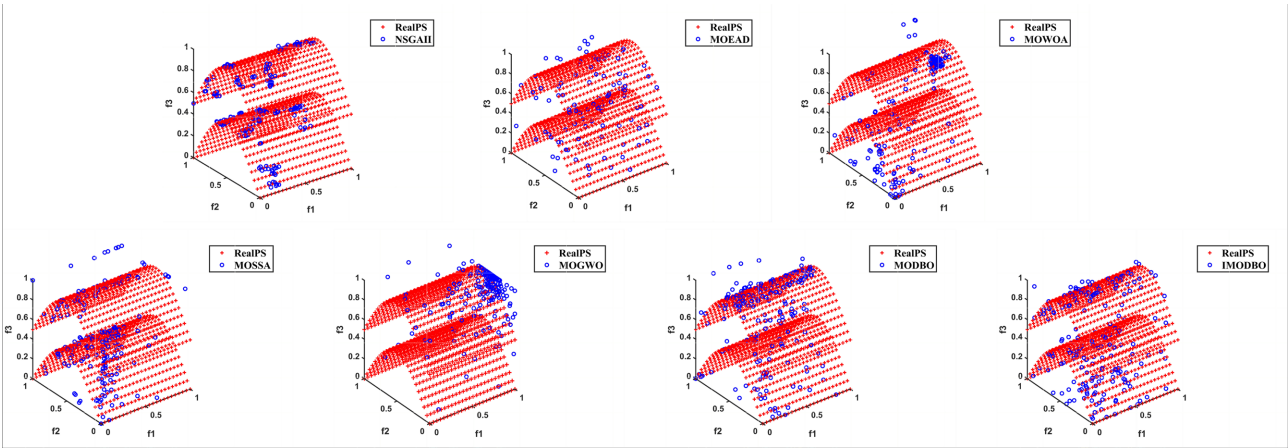


Figure 6. PS of compared algorithms on MMF14\_a.

## 5. Simulation Test of the Improved Algorithm

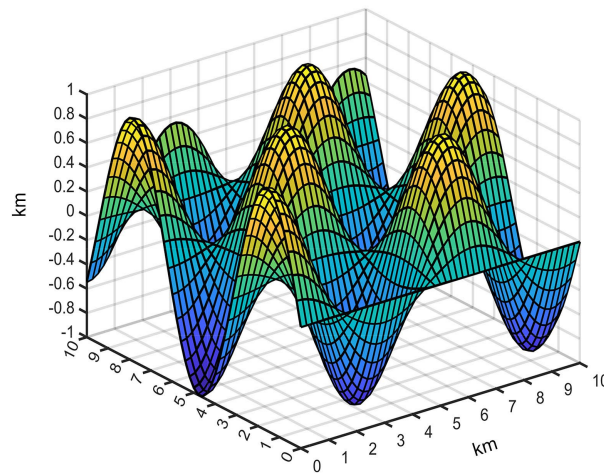
To verify the effectiveness of the proposed IMODBO algorithm in 3D coverage path planning, simulation comparisons between the traditional MODBO and IMODBO algorithm are conducted under conditions. The experiment quantitatively evaluates the quality of flight paths generated by the two algorithms from three core metrics: coverage rate, number of nodes, and path length.

### 5.1. Model Establishment

To verify the effectiveness of the proposed algorithm in complex 3D environments, a simulation scenario with unstructured undulating characteristics is constructed in this experiment, as shown in **Figure 7**. Its surface presents typical saddle-shaped features: the central area is relatively flat, while the terrain height changes significantly and nonlinearly when extending to the surrounding areas, forming multiple local peaks and valleys. This terrain model can effectively simulate complex, non-flat landforms such as hills and valleys in the real world, posing severe challenges for UAV path planning and coverage.

The monitoring area is set as a square plane with a side length of 10. The terrain height of this area is generated by a continuous and differentiable binary function, as shown in Equation (26). Under this terrain condition, the UAV needs to avoid terrain obstacles while achieving complete coverage of the target area, so as to effectively test the adaptability and robustness of the algorithm in complex environments.

$$z(x, y) = \sin(y) \cdot \cos(x). \quad (26)$$



**Figure 7.** 3D diagram of the terrain model.

### 5.2. Complete Parameter Settings for Simulation

#### 1) Mission Area and Grid Resolution

The planar inspection area is a square with  $x \in [0, 10]$  and  $y \in [0, 10]$  and the grid resolution is set to 0.5. The terrain elevation function is defined as  $z(x, y) = \sin(y) \cdot \cos(x)$ , which generates terrain with continuously varying ele-

vation across the whole domain.

#### 2) UAV Spatial Constraints

The flight altitude constraint is  $h_{\min} = -1$ ,  $h_{\max} = 1$ . The z-coordinate of all UAV waypoints is strictly restricted within  $[-1, 1]$ ; solutions exceeding this range are regarded as infeasible.

#### 3) Start and End Point Rules

Taking  $(0, 0, 0)$  as the reference point, the node closest to the origin is selected as the starting point, without manually fixed start and end coordinates.

#### 4) Algorithm Population and Termination Criterion

The population size  $N_p = 50$ , and the maximum iteration number  $T_{\max} = 500$ . The algorithm terminates once the iteration count reaches 500 and outputs the final Pareto solution set, with no early convergence stopping condition.

### 5.3. Algorithm Performance

To eliminate the randomness of single simulation results and suppress numerical fluctuations caused by random initialization, five independent repeated experiments were conducted for both MODBO and the proposed IMODBO algorithm under identical terrain and simulation parameter settings. The mean values and standard deviations of coverage rate, path length, and path node number were statistically calculated based on the five groups of experimental data to quantitatively evaluate the stability of algorithm performance. The statistical results are illustrated in **Table 5**.

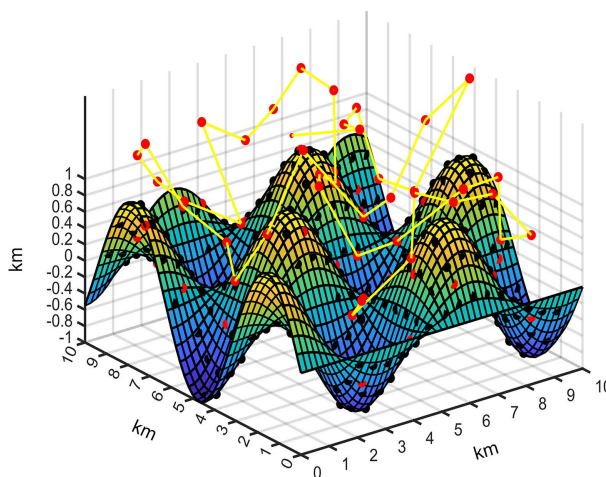
**Table 5.** Quantitative comparison between IMODBO and MODBO.

| algorithm | Mean coverage rate | Standard deviation of coverage rate | Mean path length | Standard deviation of path length | Mean number of path nodes | Standard deviation of the number of path nodes |
|-----------|--------------------|-------------------------------------|------------------|-----------------------------------|---------------------------|------------------------------------------------|
| MODBO     | 72.84              | 1.4876                              | 72.1694          | 2.3748                            | 41.8                      | 2.1679                                         |
| IMODBO    | 86.80              | 0.3670                              | 71.6046          | 1.1020                            | 41.2                      | 1.9235                                         |

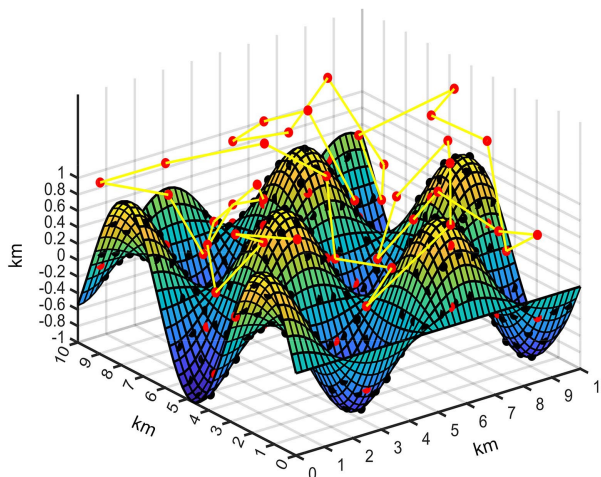
Statistical results show that the IMODBO achieves an average coverage rate of 86.80% in repeated experiments, which is 13.96 percentage points higher than that of the MODBO. Furthermore, the standard deviations of coverage rate and path length of the IMODBO are significantly lower than those of the original MODBO. This indicates that the proposed improved algorithm effectively suppresses the result fluctuations caused by random initialization and delivers more stable performance. Benefiting from multi-objective collaborative optimization, the IMODBO yields a shorter average flight path in multiple trials and effectively eliminates route redundancy. The average number of path nodes of the IMODBO is 41.2, compared with 41.8 for the MODBO. The improved algorithm simplifies redundant nodes during path planning and completes full-domain search and rescue tasks with fewer nodes. Moreover, the node standard deviation of the IMODBO is 1.9235, which is lower than the MODBO's value of 2.1679. This verifies that the node numbers solved by the improved algorithm present smaller fluctuations, and the scale of route nodes generated in each planning is more stable and consistent.

Multiple mutually non-dominated Pareto optimal solutions were obtained after the iterative optimization of the IMODBO algorithm. To satisfy the engineering requirements of UAV post-disaster emergency full-coverage search and rescue, a weighted evaluation criterion was adopted to determine the unique executable path. Firstly, min-max normalization was performed on three minimization objectives to eliminate dimensional differences. According to the mission characteristics prioritizing full-area search and rescue, the weights were set as ( $w_1 = 0.6$ ,  $w_2 = 0.2$ ,  $w_3 = 0.2$ ) to construct a comprehensive evaluation function. The comprehensive scores of all non-dominated solutions were calculated, and the solution with the minimum score was selected as the optimal 3D full-coverage path of the UAV for simulation and comparison. The optimal schemes obtained by screening are presented in **Table 6**.

**Figures 8-11** show the UAV path planning simulation results of the two algorithms under the corresponding model. Among them, **Figure 8** and **Figure 9** are side views, and **Figure 10** and **Figure 11** are top views.

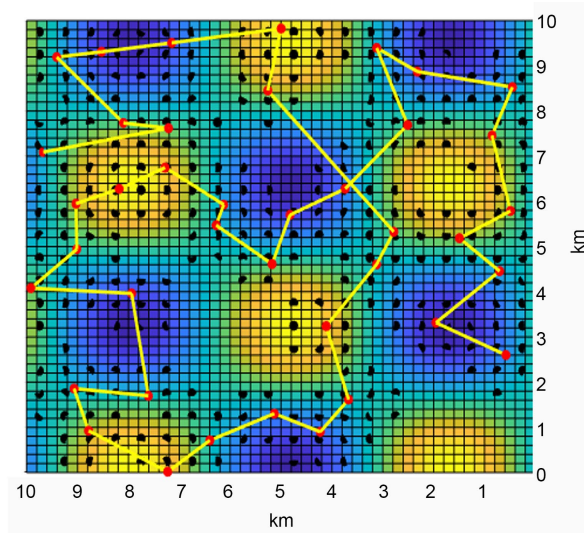


**Figure 8.** Side-view simulation result of MODBO in the model.

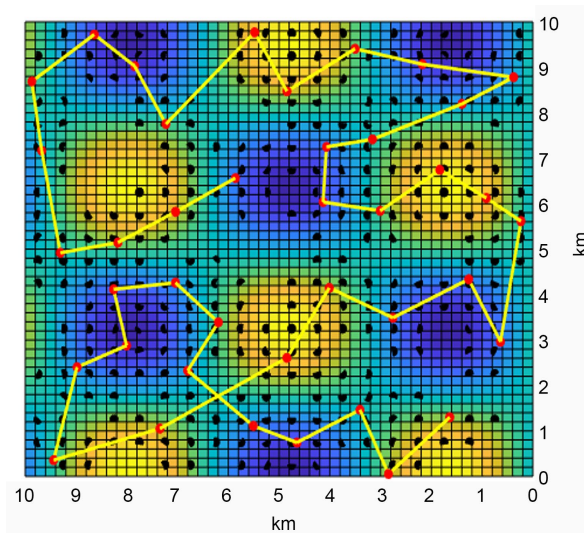


**Figure 9.** Side-view simulation result of IMODBO in the model.

From the side views in **Figure 8** and **Figure 9**, the path generated by the IMODBO algorithm fits the terrain undulation well. The UAV's flight altitude is adaptively adjusted to terrain changes, effectively ensuring coverage performance while avoiding the risk of collision with the ground. In contrast, the path planned by MODBO presents abrupt altitude changes in some areas, which may cause potential safety hazards in actual flight.



**Figure 10.** Top-view simulation result of MODBO in the model.



**Figure 11.** Top-view simulation result of IMODBO in the model.

It can be observed from the top views in **Figure 10** and **Figure 11** that the nodes of the path planned by IMODBO are distributed more uniformly and reasonably with appropriate spacing, which can construct a network structure with better coverage efficiency. In contrast, the node distribution of MODBO presents obvious aggregation. Nodes are excessively dense in some areas while relatively sparse in others, resulting in reduced overall coverage efficiency.

**Table 6.** Quantitative comparison of optimal schemes.

| algorithm | coverage rate | node | path length |
|-----------|---------------|------|-------------|
| MODBO     | 74.6%         | 40   | 71.1091     |
| IMODBO    | 87.2%         | 40   | 70.7435     |

To demonstrate the effectiveness of the proposed algorithm in simulation experiments, the IMODBO algorithm is compared with the original MODBO algorithm. **Table 6** shows that the IMODBO algorithm achieves significant advantages in 3D full-coverage path planning. Specifically, IMODBO significantly improves the coverage to 87.2%, which is 12.6% higher than MODBO's 74.6%. This indicates that its deployment strategy can more effectively cover complex 3D terrains. The total path length planned by IMODBO is 70.7435, which is shorter than MODBO's 71.1091. It improves coverage quality while reducing path length. With the same number of nodes, IMODBO significantly increases the coverage rate and shortens the flight path length.

## 6. Conclusions

To address coverage blind areas and path redundancy in UAV full-coverage path planning in complex 3D environments, this paper constructs a multi-objective collaborative optimization model that integrates coverage integrity, the number of path nodes, and path length, and proposes the IMODBO algorithm to efficiently solve the model. To address the shortcomings of the standard Dung Beetle Optimization (DBO) algorithm in handling such high-dimensional and complex problems, this paper systematically enhances the global exploration capability, local exploitation accuracy, and robustness of the algorithm in escaping local optima by introducing the Latin Hypercube Sampling (LHS) initialization strategy, dynamic adaptive parameter adjustment mechanism, and sine-cosine hybrid disturbance strategy.

Experimental results show that the proposed IMODBO algorithm exhibits superior comprehensive performance on the standard multi-objective test set. In the simulation verification for the 3D full-coverage path planning task, compared with the original MODBO algorithm, IMODBO significantly increases the coverage rate from 74.6% to 87.2% while maintaining the same number of nodes, and simultaneously shortens the total flight path length. This fully demonstrates that the method proposed in this paper can effectively overcome the coverage blind spot problem caused by insufficient handling of the height dimension in existing studies, and provides a practical technical solution for UAVs to perform efficient, complete autonomous operation tasks in unstructured 3D scenarios.

Future research work will focus on extending the algorithm to dynamic obstacle environments to enhance its real-time adaptability in real emergency scenarios, and on exploring the multi-UAV collaborative coverage mechanism to further improve the operation efficiency in large-scale areas.

## Funding

This work was supported by the Fundamental Research Funds for the Central Universities (Grant No.: ZY20250106); Hebei Major Science and Technology Support Program (262Y5403D); Hebei Provincial Innovation Capacity Enhancement Program (25350802D); Langfang Municipal Science and Technology Support Program (2025011099).

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Yoon, S., Kim, Y. and Kim, T.H. (2025) Adaptive Coverage Path Planning for Multiple Fixed-Wing UAVs with Learning-Based Initialization. *International Journal of Control, Automation and Systems*, **23**, 3359-3369. <https://doi.org/10.1007/s12555-025-0263-7>
- [2] Muñoz, J., López, B., Quevedo, F., Monje, C.A., Garrido, S. and Moreno, L.E. (2021) Multi UAV Coverage Path Planning in Urban Environments. *Sensors*, **21**, Article 7365. <https://doi.org/10.3390/s21217365>
- [3] Swain, S., Khilar, P.M. and Senapati, B.R. (2023) An Efficient Path Planning Algorithm for 2D Ground Area Coverage Using Multi-UAV. *Wireless Personal Communications*, **132**, 361-407. <https://doi.org/10.1007/s11277-023-10614-x>
- [4] Sharma, M. and Voruganti, H.K. (2025) Energy-Efficient Coverage Path Planning for Mobile Robots on Uneven Terrain via the TA-NSGA-II Approach. *Arabian Journal for Science and Engineering*. <https://doi.org/10.1007/s13369-025-10844-0>
- [5] Wu, W., Wang, Z., Lin, L., Chang, X. and Tian, L. (2025) An Efficient Coverage Path Planning Method for UAV in Complex Concave Regions. *Scientific Reports*, **15**, Article No. 37227. <https://doi.org/10.1038/s41598-025-20978-8>
- [6] Rahman, M., Sarkar, N.I. and Lutui, R. (2025) A Survey on Multi-UAV Path Planning: Classification, Algorithms, Open Research Problems, and Future Directions. *Drones*, **9**, Article 263. <https://doi.org/10.3390/drones9040263>
- [7] Huang, J., Chen, C., Shen, J., Liu, G. and Xu, F. (2025) A Self-Adaptive Neighborhood Search A-Star Algorithm for Mobile Robots Global Path Planning. *Computers and Electrical Engineering*, **123**, Article 110018. <https://doi.org/10.1016/j.compeleceng.2024.110018>
- [8] Alshammrei, S., Boubaker, S. and Kolsi, L. (2022) Improved Dijkstra Algorithm for Mobile Robot Path Planning and Obstacle Avoidance. *Computers, Materials & Continua*, **72**, 5939-5954. <https://doi.org/10.32604/cmc.2022.028165>
- [9] Lei, S., Li, T., Gao, X., Xue, P. and Song, G. (2025) Research on Improved RRT Path Planning Algorithm Based on Multi-Strategy Fusion. *Scientific Reports*, **15**, Article No. 13312. <https://doi.org/10.1038/s41598-025-92675-5>
- [10] Khuat, T.H., Bui, D., Nguyen, H.T.T., Trinh, M.L., Nguyen, M.T. and Phung, M.D. (2025) Multi-Goal Rapidly Exploring Random Tree with Safety and Dynamic Constraints for UAV Cooperative Path Planning. *IEEE Transactions on Vehicular Technology*, **74**, 13446-13457. <https://doi.org/10.1109/tvt.2025.3560658>
- [11] Sharma, G., Jain, S. and Sharma, R.S. (2025) Path Planning for Fully Autonomous UAVs-A Taxonomic Review and Future Perspectives. *IEEE Access*, **13**, 13356-13379. <https://doi.org/10.1109/access.2025.3529754>

- [12] Luo, Y., Lin, S., Wang, Y. and Liang, K. (2025) Path Planning for Mobile Robots in Complex Environments Based on Enhanced Sparrow Search Algorithm and Dynamic Window Approach. *Robotica*, **43**, 1929-1952. <https://doi.org/10.1017/s026357472500061x>
- [13] Xie, X., Yang, Y. and Zhou, H. (2025) Multi-Strategy Hybrid Whale Optimization Algorithm Improvement. *Applied Sciences*, **15**, Article 2224. <https://doi.org/10.3390/app15042224>
- [14] Wu, Q. (2025) An Enhanced Whale Optimization Algorithm with Inertia Weight and Dynamic Parameter Adaptation for Wireless Sensor Network Deployment. *Computing*, **107**, Article No. 167. <https://doi.org/10.1007/s00607-025-01509-9>
- [15] Yu, L., Lian, J., Li, Y., Du, Y. and Wang, D. (2026) An Enhanced Ant Colony Optimization for Path Planning of a Mobile Robot. *Optimal Control Applications and Methods*, **47**, 387-398. <https://doi.org/10.1002/oca.70064>
- [16] He, Y. and Wang, M. (2025) Dynamic Step Opposition-Based Learning Sparrow Search Algorithm for UAV Path Planning. *Cluster Computing*, **28**, Article No. 44. <https://doi.org/10.1007/s10586-024-04691-1>
- [17] Yang, Y., Sun, L., Fu, Y., Feng, W. and Xu, K. (2025) Three-Dimensional UAV Trajectory Planning Based on Improved Sparrow Search Algorithm. *Symmetry*, **17**, Article 2071. <https://doi.org/10.3390/sym17122071>
- [18] Wang, H., Hao, Z. and Zhang, Y. (2025) Research on Three-Dimensional Path Planning of Unmanned Aerial Vehicle Based on Improved Whale Optimization Algorithm. *PLOS ONE*, **20**, e0316836. <https://doi.org/10.1371/journal.pone.0316836>
- [19] Shi, K., Wu, W., Wu, Z., Jiang, B. and Karimi, H.R. (2024) Coverage Path Planning for Cleaning Robot Based on Improved Simulated Annealing Algorithm and Ant Colony Algorithm. *Signal, Image and Video Processing*, **18**, 3275-3284. <https://doi.org/10.1007/s11760-023-02989-y>
- [20] Shi, H., Li, X., Pan, W., Hwang, K. and Li, Z. (2017) A Novel Fuzzy Three-Dimensional Grid Navigation Method for Mobile Robots. *International Journal of Advanced Robotic Systems*, **14**.
- [21] Yu, S., Zhu, J., Shen, J., Yang, X. and Wang, P. (2023) A Three-Dimensional Full-Coverage Operation Path Planning Method for Plant Protection Unmanned Aerial Vehicles Based on Energy Consumption Modeling. *Electronics*, **12**, Article 4051. <https://doi.org/10.3390/electronics12194051>
- [22] Zheng, X., Liu, R. and Li, S. (2025) A Novel Improved Dung Beetle Optimization Algorithm for Collaborative 3D Path Planning of UAVs. *Biomimetics*, **10**, Article 420. <https://doi.org/10.3390/biomimetics10070420>
- [23] Zhang, L., Li, Y., Yu, Y. and Retscher, G. (2026) A UAV Path-Planning Method Based on Multi-Mechanism Improved Dung Beetle Optimizer Algorithm in Complex Constrained Environments. *Symmetry*, **18**, Article 383. <https://doi.org/10.3390/sym18020383>
- [24] Zhang, M., Chen, J., Ouyang, H., Long, S., Li, S., Zhang, C., et al. (2025) Solving the 3D Multi-Objective Dynamic AUV Path Planning Problem Based on the Improved Morphin-Altruistic NSGAI Algorithm. *Engineering Science and Technology, an International Journal*, **69**, Article 102137. <https://doi.org/10.1016/j.jestech.2025.102137>
- [25] Ho-Huu, V., Hartjes, S., Visser, H.G. and Curran, R. (2018) An Improved MOEA/D Algorithm for Bi-Objective Optimization Problems with Complex Pareto Fronts and Its Application to Structural Optimization. *Expert Systems with Applications*, **92**, 430-446. <https://doi.org/10.1016/j.eswa.2017.09.051>

- [26] Li, M., Yu, X., Fu, B. and Wang, X. (2025) A Modified Whale Optimization Algorithm with Multi-Strategy Mechanism for Global Optimization Problems. *Neural Computing and Applications*, **37**, 22339-22352. <https://doi.org/10.1007/s00521-023-08287-5>
- [27] Li, L., Xiong, J., Tseng, M., Yan, Z. and Lim, M.K. (2022) Using Multi-Objective Sparrow Search Algorithm to Establish Active Distribution Network Dynamic Reconfiguration Integrated Optimization. *Expert Systems with Applications*, **193**, Article 116445. <https://doi.org/10.1016/j.eswa.2021.116445>
- [28] Makhadmeh, S.N., Alomari, O.A., Mirjalili, S., Al-Betar, M.A. and Elnagar, A. (2022) Recent Advances in Multi-Objective Grey Wolf Optimizer, Its Versions and Applications. *Neural Computing and Applications*, **34**, 19723-19749. <https://doi.org/10.1007/s00521-022-07704-5>
- [29] Wang, Z.L. and Shao, P. (2023) A Multi-Strategy Dung Beetle Optimization Algorithm for Optimizing Constrained Engineering Problems. *IEEE Access*, **11**, 98805-98817. <https://doi.org/10.1109/access.2023.3313930>