

Article

GLBWOA: A Global–Local Balanced Whale Optimization Algorithm for UAV Path Planning

Qiwu Wu ^{1,†}, Weicong Tan ^{2,*,†} , Renjun Zhan ¹, Lingzhi Jiang ², Li Zhu ^{1,3} and Husheng Wu ¹ 

¹ School of Equipment Management and Support, Engineering University of PAP, Xi'an 710086, China; cli@hunnu.edu.cn (Q.W.); zhanrenjun@aliyun.com (R.Z.); zhuli_cnrs@foxmail.com (L.Z.); wuhusheng0421@163.com (H.W.)

² School of Information Engineering, Engineering University of PAP, Xi'an 710086, China; jianglingzhi1123@ustb.edu.cn

³ Key Laboratory of Counter-Terrorism Command & Information Engineering (Engineering University of PAP), Ministry of Education, Xi'an 710086, China

* Correspondence: tanweicong_pap@163.com

† These authors contributed equally to this work.

Abstract: To tackle the challenges of path planning for unmanned aerial vehicle (UAV) in complex environments, a global–local balanced whale optimization algorithm (GLBWOA) has been developed. Initially, to prevent the population from prematurely converging, a bubble net attack enhancement strategy is incorporated, and mutation operations are introduced at different stages of the algorithm to mitigate early convergence. Additionally, a failure parameter test mutation mechanism is integrated, along with a predefined termination rule to avoid excessive computation. The algorithm's convergence is accelerated through mutation operations, further optimizing performance. Moreover, a random gradient-assisted optimization approach is applied, where the negative gradient direction is identified during each iteration, and an appropriate step size is selected to enhance the algorithm's exploration capability toward finding the optimal solution. The performance of GLBWOA is benchmarked against several other algorithms, including SCA, BWO, BOA, and WOA, using the IEEE CEC2017 test functions. The results indicate that the GLBWOA outperforms other algorithms. Path-planning simulations are also conducted across four benchmark scenarios of varying complexity, revealing that the proposed algorithm achieves the lowest average total cost for flight path planning and exhibits high convergence accuracy, thus validating its reliability and superiority.

Keywords: whale optimization algorithms; complex environment; path planning; multiple strategies; digital elevation model



Citation: Wu, Q.; Tan, W.; Zhan, R.; Jiang, L.; Zhu, L.; Wu, H. GLBWOA: A Global–Local Balanced Whale Optimization Algorithm for UAV Path Planning. *Electronics* **2024**, *13*, 4598.

<https://doi.org/10.3390/electronics13234598>

electronics13234598

Academic Editor: Subir Halder

Received: 14 October 2024

Revised: 12 November 2024

Accepted: 18 November 2024

Published: 21 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Unmanned aerial vehicles and other intelligent agents have become the focal point of research in recent years, with their applications expanding across various domains, including military operations [1], agriculture [2], environmental monitoring [3], and particularly in emergency rescue [4] and logistics distribution [5]. The establishment of a safe flight path is crucial for the successful execution of missions. However, the task of planning a safe flight path for UAVs within complex environments presents considerable challenges. Such environments may encompass rugged mountainous terrain, urban structures, and geographical obstacles such as waterfalls, in addition to being influenced by dynamic factors like weather fluctuations and traffic conditions. Consequently, the development of efficient, safe, and feasible flight paths for UAVs has emerged as a pressing issue that necessitates urgent attention under these intricate conditions.

Currently, the predominant approach to unmanned aerial vehicle (UAV) path planning involves transforming the environmental model [6] into a mathematical framework. This

allows for the development of a safe, feasible, and stable flight path from the initial point to the destination, utilizing algorithms within specified constraints [7,8].

Intelligent algorithms utilized for autonomous UAV path planning can be categorized into three primary groups: traditional optimization algorithms, intelligent optimization algorithms, and machine learning algorithms. Traditional optimization algorithms have been extensively employed in various path-planning applications. Among these, the A* algorithm [9,10], recognized as a classical heuristic search algorithm, is particularly favored due to its straightforward implementation. However, its efficacy diminishes significantly when applied to large-scale and high-dimensional spaces, thereby constraining its capacity to address trajectory planning challenges that involve multiple constraints. The rapidly exploring random tree (RRT) algorithm [11,12] represents a notable path-planning technique based on spatial sampling, which does not necessitate the discretization of the flight environment, resulting in a more rapid search process. Nonetheless, this method often struggles to yield optimal trajectories. The artificial potential field method [13,14] is appreciated for its rapid computational speed and effective real-time performance in path planning; however, in expansive and high-dimensional spatial environments, it may encounter issues such as local oscillations and local minima, which can render the generated paths impractical. Collectively, these observations underscore the limitations of traditional optimization algorithms in effectively addressing multi-constraint path-planning problems, as they often fail to achieve a satisfactory balance between accuracy and time efficiency.

In recent years, numerous scholars have investigated and developed intelligent optimization algorithms inspired by biological behaviors observed in nature, as well as principles derived from mathematical functions, to address UAV path-planning challenges. Notable algorithms in this domain include the social spider algorithm (SSA) [15], grey wolf optimization (GWO) [16], beluga whale optimization (BWO) algorithm [17,18], sine cosine algorithm (SCA) [19], and butterfly optimization algorithm (BOA) [20–22], among others. These algorithms are capable of rapidly identifying optimal paths by employing a variety of search strategies and executing multiple iterations. However, it is important to note that many intelligent optimization algorithms are probabilistic stochastic search methods characterized by a significant degree of randomness when applied to specific engineering problems. Consequently, the interplay between global optimization and local optimization tends to be weak. Therefore, a primary challenge faced by many intelligent algorithms is to prevent convergence to local optima and to enhance the global convergence rate as much as possible.

Introduced by Mirjalili S et al. in 2016, the whale optimization algorithm [23] is a metaheuristic method that emulates the feeding behaviors of humpback whales. By emulating three predation tactics of humpback whales—encircling prey, bubble net feeding, and searching for food—the algorithm seeks the optimal solution. These strategies are characterized by robust search capabilities and computational stability. Jiang R [24] et al. proposed the whale army optimization algorithm, which introduces the armed forces procedure and adjusts the establishment of key parameters and foundation principles of the original whale algorithm, which is comparable to the traditional whale optimization algorithm as well as other high-performance group intelligent algorithms. The algorithm has faster convergence speed under lower computational complexity. Huang Y [25] et al. introduced a whale optimization algorithm designed for the path-planning challenges of autonomous underwater vehicles. This approach integrates segmented learning with an adaptive operator selection strategy, employing a dynamic partitioning method and a weighted mean scheme to create virtual individuals. These virtual entities are then incorporated into the whale optimization framework to form an evolutionary pool. As a result, the algorithm's optimization performance is enhanced. Simulation results demonstrate that the proposed method exhibits greater robustness and search capability compared to other comparative algorithms. Guo W [26] et al. proposed an improved whale optimization algorithm based on wavelet mutation strategy and social learning, designed a new linear incremental probability to improve the global exploitation ability, introduced an

adaptive neighborhood learning strategy to promote the exchange of information between individuals, and integrated the Morlet wavelet mutation mechanism to avoid the algorithm falling into local optimality. Wang C [27] and colleagues introduced an adaptive adjustment mechanism based on the whale optimization algorithm. This mechanism dynamically modifies the search process during iterations by incorporating controllable variables and utilizing a differential mutation evolutionary strategy. These enhancements effectively balance the algorithm's global and local optimality. Ultimately, the algorithm was applied to the path-planning problem, demonstrating superior performance compared to the original whale algorithm and six other advanced intelligent optimization methods. The proposed algorithm, along with the six high-performance optimization techniques, exhibited improved convergence speed, greater accuracy, and enhanced stability. The study presented in [28] introduced a novel whale optimization algorithm (NWOA) aimed at addressing the robot path-planning challenge within highly complex dynamic environments. This approach employs an adaptive strategy to hasten the algorithm's convergence. Additionally, it incorporates virtual obstacles to improve the algorithm's capacity to evade local optima and introduces a potential field factor to boost the robot's performance in obstacle avoidance. Simulation comparisons demonstrate the advantages of the proposed algorithm. The authors of [29] proposed an enhanced whale optimization algorithm to realize the path planning of UAV weather detection missions in complex environments and introduced real-time boundary processing, quasi-opposite-based learning, and an enhanced search mechanism into the standard whale optimization algorithm, which improves the convergence speed of the algorithm and the ability of global optimization, and the simulations show that the algorithm gives a higher-quality path plan than the other improved algorithms.

Furthermore, with the rapid advancements in key machine learning techniques, an increasing number of researchers have begun exploring their application to path-planning challenges [30–32]. As a result, the quest for more efficient and robust path-planning algorithms has become a critical area of research that warrants comprehensive investigation.

The aforementioned paper proposes various enhancement strategies for the whale optimization algorithm, resulting in varying degrees of improved optimization performance. However, it continues to encounter challenges pertaining to the accuracy of optimal value searches and the balance between global and local exploration capabilities. To address these issues, we introduce a global–local balanced whale optimization algorithm (GLBWOA) that incorporates three principal enhancement strategies:

- We propose an enhancement strategy for the bubble net attack, enabling individuals to possess the requisite capability to escape their current position at various stages. This approach aims to increase the algorithm's likelihood of transcending local optima.
- We present the failure parameter test mutation mechanism, which involves predefined trigger conditions. When the algorithm encounters a local optimum or exhibits slow progress, the mutation operation is activated to enhance its global search capability and its ability to escape from local optima.
- Inspired by the directional variations of the gradient vector, we propose a stochastic gradient-assisted optimization method that integrates the gradient into the conventional whale optimization algorithm to enhance its optimization performance. Furthermore, an energy reduction scheme is introduced to improve the algorithm's local exploration capabilities.

Ultimately, the algorithm introduced in this study is utilized for path planning in UAVs across various complex environments, demonstrating its effectiveness.

The remainder of this paper is organized as follows: Section 2 outlines the steps involved in formulating the objective function; Section 3 details the design of the improved algorithm; and Section 4 presents the simulation of the algorithms using the test function, along with the implementation of path-planning problems and their comparative results. Finally, we conclude the paper with a summary of our findings.

2. Problem Description

This research investigates the challenge of path planning for unmanned aerial vehicles through the formulation of a comprehensive cost function that assigns specific weights to different constraints. This methodology considers a wide array of constraints and seeks to determine the optimal trajectory by minimizing the associated cost function [33]. The constraints considered are more comprehensive than the cost function designed in the literature [34], and the comprehensive cost function in this paper can be flexible by adding constraints and adjusting weights at any time according to the different task requirements.

2.1. Restrictive Condition

UAVs prioritize range length when performing missions, and shorter ranges can greatly reduce fuel consumption to improve endurance. A UAV is usually controlled by a ground control station to fly sequentially along waypoints planned in a search map to form a flight path X_i . Each waypoint is a node of the path searched on a known map, with coordinates set to $M_{ij} = (x_{ij}, y_{ij}, z_{ij})$. The range length cost function F_1 can be obtained by accumulating the Euclidean distance between every two nodes:

$$F_1(Path_i) = \sum_{j=1}^{n-1} \left\| \vec{M_{ij}M_{i,j+1}} \right\|, \quad (1)$$

In addition to the duration of the flight, it is imperative to account for potential threats posed by obstacles encountered during the flight. The designated flight path must be designed to enable the UAV to navigate around these obstacles, thereby ensuring safe operational conditions throughout the flight. Let α represent the set of all potential threats, with the threat posed by obstacles being modeled as a cylinder characterized by a projection center coordinate O_α and a radius R_α . The UAV has a diameter denoted as D , while the length of the safety buffer is represented by S . Furthermore, the distance between the path, denoted as $\left\| \vec{M_{ij}M_{i,j+1}} \right\|$, is directly established by any two flight points of the UAV. The distance between the path and the center coordinate is d_α , as shown in Figure 1. Considering the above conditions, the threat cost function F_2 constitutes:

$$\begin{cases} F_2(Path_i) = \sum_{j=1}^{n-1} \sum_{\alpha=1}^{\alpha} K_\alpha(\vec{M_{ij}M_{i,j+1}}), \\ K_\alpha(\vec{M_{ij}M_{i,j+1}}) = \begin{cases} 0, & \text{if } d_\alpha > S + D + R_\alpha \\ (S + D + R_\alpha) - d_\alpha, & \text{if } D + R_\alpha < d_\alpha \leq S + D + R_\alpha \\ \infty, & \text{if } d_\alpha \leq D + R_\alpha. \end{cases} \end{cases} \quad (2)$$

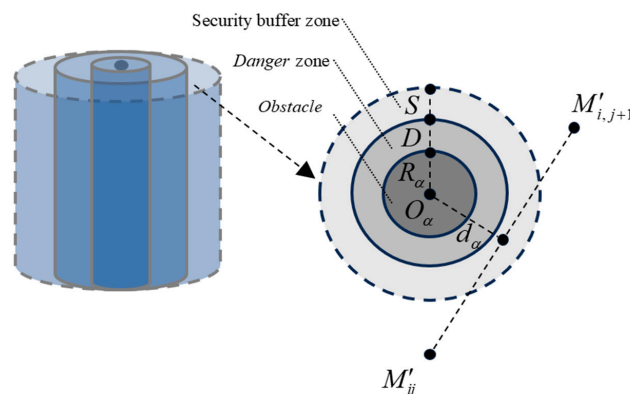


Figure 1. Cylindrical obstacle threat.

In certain scenarios, such as aerial photography and experimental missions, an unmanned aerial vehicle (UAV) is required to operate within designated airspace. Flying at excessive altitudes can compromise the resolution of aerial imagery, while insufficient altitudes pose unnecessary risks to ground personnel, flora, and fauna and may obstruct the field of view during filming. Therefore, the operational flight altitude is typically constrained within two defined limits, as illustrated in Figure 2. This approach enables the efficient allocation and utilization of airspace resources, reducing potential conflicts and interference. During flight operations, the actual altitude is determined by both the terrain elevation and the prescribed altitude limits. The following guidelines outline how to calculate the altitude cost at a given point within the specified range:

$$H_{ij} = \begin{cases} \left| h_{ij} - \frac{(h_{\max} + h_{\min})}{2} \right|, & \text{if } h_{\min} \leq h_{ij} \leq h_{\max} \\ \infty, & \text{otherwise,} \end{cases} \quad (3)$$

where the rule restricts the UAV to flying at the average of the two extremes and specifies that the cost of flying increases accordingly as the distance traveled away from the average altitude is increased, thus constituting the altitude cost function F_3 :

$$F_3(\text{Path}_i) = \sum_{j=1}^n H_{ij}. \quad (4)$$

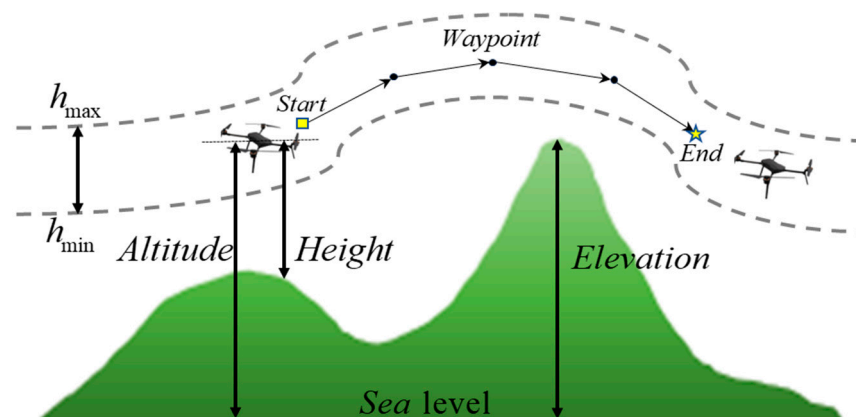


Figure 2. Altitude cost explanation.

Smoothing cost is an important consideration in UAV path planning and is a cost introduced to ensure that the flight path of the UAV is as smooth as possible to avoid sharp steering or altitude changes that lead to increased difficulty in controlling the airframe and accelerated fuel consumption. As shown in Figure 3, the smoothing cost includes the fuselage steering cost and the climb cost, where ψ_{ij} stands for the steering angle and $\theta_{i,j+1}$ stands for the climb angle. According to the UAV performance constraints, the maximum steering angle and maximum climb angle cannot exceed $\psi_{\max}$ and $\theta_{\max}$. Taking z as a unit vector in the direction of the z coordinate axis, the climb angle θ_{ij} can be expressed as:

$$\theta_{ij} = \arctan \left(\frac{z_{i,j+1} - z_{ij}}{\left\| \vec{M'_{ij}M'_{i,j+1}} \right\|} \right), \quad (5)$$

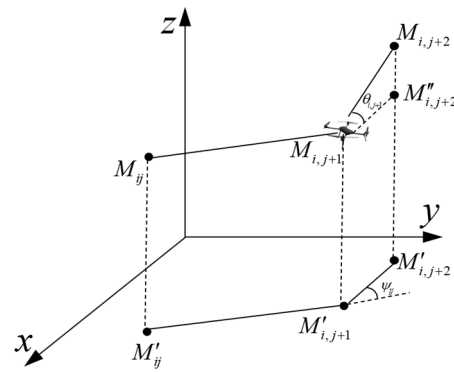


Figure 3. Calculation of turning and climbing angles.

The steering angle ψ_{ij} is expressed as

$$\psi_{ij} = \arctan \left(\frac{\left\| \vec{M'_{ij}M'_{i,j+1}} \times \vec{M'_{i,j+1}M'_{i,j+2}} \right\|}{\vec{M'_{ij}M'_{i,j+1}} \cdot \vec{M'_{i,j+1}M'_{i,j+2}}} \right), \quad (6)$$

The smoothing cost can be expressed as

$$F_4(Path_i) = \sigma_1 \sum_{j=1}^{n-2} \psi_{ij} + \sigma_2 \sum_{j=1}^{n-1} |\theta_{ij} - \theta_{i,j-1}|, \quad (7)$$

where σ_1 and σ_2 denote the cost coefficients for the steering angle and climb angle, respectively.

2.2. Integrated Cost Function

To ensure the UAV reaches its target safely and efficiently, the planned path must guide the UAV through collision-free flight, taking into account factors such as total path length cost, threat cost, altitude cost, and smoothing cost. The total cost function is formulated as follows:

$$F_{\text{cost}}(Path_i) = \sum_{\alpha=1}^4 w_{\alpha} F_{\alpha}(Path_i), \quad (8)$$

where F_1 to F_4 denote the costs of trajectory length, obstacle threat, navigational altitude, and path smoothing, respectively, and w_{α} is a weighting factor corresponding to the different costs.

2.3. Environmental Model

The path planning scenarios are derived from two distinct terrains located on Christmas Island, Australia [35], as depicted in Figure 4a,b. These scenarios employ authentic digital elevation model (DEM) maps obtained from LiDAR sensors. Each scenario is classified into simple and complex categories based on the number of obstacles present. The simple scenario is characterized by the presence of three cylindrical obstacles, whereas the complex scenario is designed to incorporate at least twice the number of structural obstacles found in the simple scenario. This differentiation aims to assess the performance of the path-planning algorithm under varying environmental conditions, leading to the establishment of four benchmark scenarios.

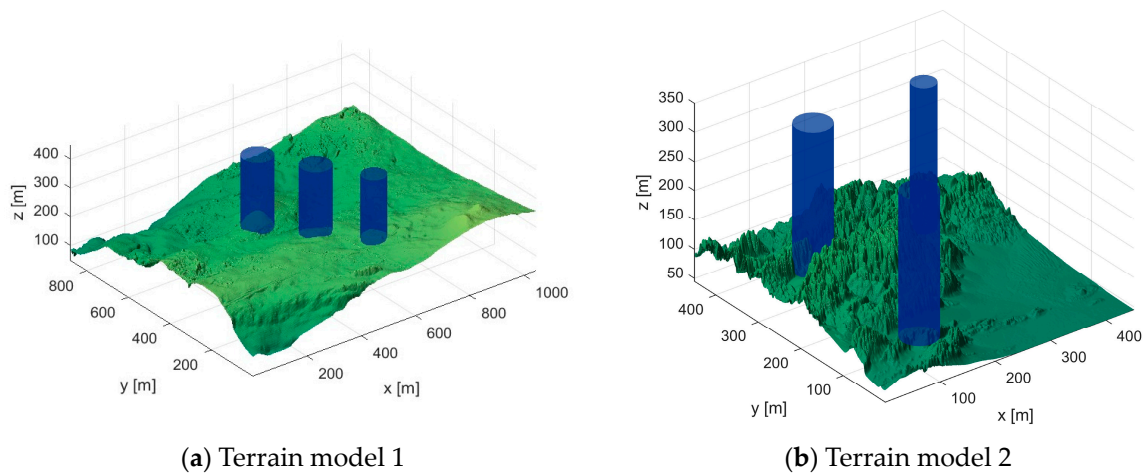


Figure 4. Terrain environment model for UAV path planning. (Blue cylinders are artificially added obstacles).

3. Algorithm Design

Among various swarm intelligence algorithms, the whale optimization algorithm (WOA) is known for its simplicity and minimal parameter requirements. However, it exhibits certain limitations when addressing the complex optimization challenges of UAV path planning. Therefore, this paper focuses on the analysis and enhancement of the WOA. In this subsection, after reviewing the standard WOA, we propose an improved global–local equilibrium whale optimization algorithm by integrating multiple strategies. These enhancements enable the algorithm to more efficiently identify optimal paths in UAV path-planning tasks, particularly in complex terrain environments.

3.1. Standard Whale Optimization Algorithm

The whale optimization algorithm mimics the unique hunting behavior of humpback whales. According to the feeding characteristics of whales, the whale’s feeding behavior is divided into three phases: constricted encircling feeding, bubble net attack, and stochastic search, and the specific analyses are described as follows:

3.1.1. Constriction-Enclosed Predation Phase

In nature, whales can find the location of prey and encircle them for predation, and the algorithm assumes that the optimal individual of the current population is the prey, and all other whales in the population encircle the location of the optimal whale to update their position, which is updated by Equations (9) and (10):

$$D = |C \cdot X^*(t) - X(t)|, \quad (9)$$

$$X(t+1) = X^*(t) - A \cdot D, \quad (10)$$

where t is the number of iterations; D denotes the bracketing step; A and C represent the coefficient vectors; $X^*(t)$ is the optimal position vector of the population; $X(t)$ is the position vector of the current population; A and C are updated by Equations (11) and (12):

$$A = 2a \cdot r_1 - a, \quad (11)$$

$$C = 2 \cdot r_2, \quad (12)$$

r_1 and r_2 are random numbers in the range $[0,1]$; the value of a decreases linearly from 2 to 0, denoted by:

$$a = 2 - 2 \cdot \frac{t}{T_{Max}}, \quad (13)$$

T_{Max} is the maximum number of iterations.

3.1.2. Bubble Network Attack Phase

Humpback whales swim towards their prey in a spiral trajectory when hunting, and in the whale optimization algorithm, individual whales update their position by using Equations (14) and (15):

$$D' = |X^*(t) - X(t)|, \quad (14)$$

$$X(t+1) = D' \cdot e^{bl} \cos(2\pi l) + X^*(t), \quad (15)$$

where b is a constant to change the shape of the spiral, usually set to 1; l is a random number in between $[-1, 1]$.

The development phase of the whale optimization algorithm consists of two phases: shrinking encirclement and bubble net attack. When $|A| < 1$, as the whale swims along the spiral trajectory around the prey in the shrinking encirclement, the whale has a 50% possibility of choosing to encircle the prey and a 50% possibility of choosing the bubble net attack, which is obtained by Equation (16):

$$X(t+1) = \begin{cases} X^*(t) - A \cdot D, & p < 0.5 \\ D' \cdot e^{bl} \cos(2\pi l) + X^*(t), & p \geq 0.5 \end{cases} \quad (16)$$

where p is a random number between $[0, 1]$.

3.1.3. Random Search Phase

Whales randomly search and feed based on their position when $|A| \geq 1$. In the WOA, whales update their position by Equations (17) and (18):

$$D = |C \cdot X^{\text{rand}}(t) - X(t)|, \quad (17)$$

$$X(t+1) = X^{\text{rand}}(t) - A \cdot D, \quad (18)$$

where $X^{\text{rand}}(t)$ is a randomly selected whale position vector.

3.2. Global–Local Balanced Whale Optimization Algorithm

WOA conducts global exploration through a stochastic search strategy after initializing the population. However, this approach may result in insufficient global search capability. As the algorithm progresses, local search is performed during the encircling predation and spiral updating phases, but the linearly decreasing convergence factor can cause an imbalance between global and local search. Consequently, the algorithm tends to fall into local optima in its later stages. To overcome these shortcomings, this subsection introduces a global–local equalization whale optimization algorithm. A bubble net attack enhancement strategy is implemented to prevent the algorithm from getting trapped in local optima, and a failure parameter test variation mechanism is incorporated to strengthen its global search capability. In the algorithm's later stages, a stochastic gradient-assisted optimization strategy is employed to further enhance its search performance. The three proposed improvement strategies are elaborated on in the following sections.

3.2.1. Bubble Net Attack Enhancement Strategy

In the standard whale optimization algorithm, the same spiral attack and contraction encircling strategy is used for all individuals during the exploitation phase to update positions. However, this approach does not account for the fact that the ability of an individual to escape its current position varies at different stages of the optimization process. Relying solely on these two modes of position updating can hinder the algorithm's ability to escape local optima, particularly when it begins to converge prematurely. To prevent the population from entering an aggregation state, which can cause WOA to

become trapped in a local optimum, a variation method is introduced to enhance the bubble net attack. A new variation operation is applied to update the position of the optimal whale individual when the random number is less than or equal to 0.6, while the original position update method remains unchanged. The variation formula is presented in Equation (19).

$$X_{i,j}(t+1) = X_{i,j}(t) + r_5 \times (X_{be,j}(t) - X_{i,j}(t)), \quad (19)$$

where $X_{be,j}(t)$ is the optimal solution of the j^{th} dimension of the current iteration, r_5 is a random number.

3.2.2. Failed Parameter Test Variation Mechanism

In optimization algorithms, termination criteria dictate when to stop the iterative process. Selecting appropriate termination criteria helps prevent over-computation while ensuring that the algorithm finds a satisfactory solution within a reasonable time frame. Based on the analysis above, to enhance the algorithm's performance, a failure parameter test variation mechanism is proposed. This mechanism triggers mutation operations by setting the following two termination conditions for the algorithm's execution:

1. A better value is not found for several consecutive iterations. For example, when the algorithm runs for 5 consecutive iterations without finding a better value, the mutation operation is triggered.
2. The change in the objective function value is less than a preset threshold. For example, the preset threshold parameter is set to 10^{-6} , and when the difference between the objective function value of the current iteration and the objective function value of the previous iteration is less than 10^{-6} , the variation operation is triggered.

The specific formula for the mutation operation is as follows:

Calculate the mutation step based on the current number of iterations t and the maximum number of iterations T :

$$step\_size = \left(\frac{ub - lb}{t/T} \right) \cdot randn(1, dim), \quad (20)$$

Calculate the variation vector:

$$m = step\_size \cdot ((rand(1, dim)) \cdot 2 - 1) \cdot \left(\frac{1}{randn(1, dim)} \right)^\beta, \quad (21)$$

where $\beta = 1.5$.

Apply the variation vector to the current whale position and perform a boundary check:

$$X_{i,j}(t+1) = X_{i,j}(t) + m \quad (22)$$

By introducing the above mutation operations, the aim is to increase the population diversity and avoid falling into local optimal solutions, thus improving the algorithm's global search capability and convergence speed.

3.2.3. Stochastic Gradient-Assisted Optimization Method

The standard whale optimization algorithm runs with incomplete local exploration. The fusion of gradient into a swarm intelligence algorithm to form a hybrid algorithm aims at combining the advantages of the two to improve the algorithm's performance in optimization. The change in the independent variable along the direction of the gradient vector can maximize the change in the function value. In the algorithm, the objective function is the optimization goal. Given the above analysis, a stochastic gradient-assisted optimization [36] method is proposed, in which after determining the negative gradient direction in each iteration, a suitable step size $step^{t+1}$ is chosen so that the objective function

value can be reduced to the maximum extent to enhance the exploration ability of the algorithm for the optimal solution. The specific operation is as follows:

$$X_i^{t+1} = X_i^t - step^{t+1} \cdot \nabla f(X_i^t), \quad (23)$$

where $-\nabla f(x_i^t)$ denotes the negative gradient direction, and $step$ denotes the optimal step size along that direction. X_i^{t+1} is the i individual in the $t + 1$ iteration. The gradient direction is defined as shown in Equation (24):

$$\nabla f = \frac{\partial f}{\partial X_1^t} \vec{e}_1 + \frac{\partial f}{\partial X_2^t} \vec{e}_2 + \cdots + \frac{\partial f}{\partial X_{Dim}^t} \vec{e}_{Dim}, \quad (24)$$

where Dim denotes the dimension of the optimization problem to be solved. $\vec{e}_i (i = 1, 2, \dots, Dim)$ is a set of unit orthogonal vectors. In order to enhance the ability of the swarm intelligence algorithm to solve the optimization problem, so that the hybrid algorithm is still able to solve the problem independently of the mathematical properties of the problem, the gradient is approximated using the forward difference formulae viewed as a definitional derivation. The formula is shown in Equation (25).

$$\frac{\partial f}{\partial X_i^t} \approx \frac{f(X_i^t + \varepsilon) - f(X_i^t)}{\varepsilon}, \quad (25)$$

where ε is a small positive number set to 10^{-6} .

The step size along the negative gradient direction is obtained by the line search method, as shown in Equation (26).

$$step^{t+1} = \begin{cases} step^t \zeta, & \text{if } f(X_i^t - step^t \zeta \nabla f(X_i^t)) \leq f\left(X_i^t - \frac{step^t}{\zeta} \cdot \nabla f(X_i^t)\right) \\ \frac{step^t}{\zeta}, & \text{else} \end{cases} \quad (26)$$

where ζ is a scale factor and ζ is set to 1.8.

3.3. Algorithmic Process

In this paper, a global local balanced whale optimization algorithm is proposed. Firstly, the original population initialization as well as the random search strategy are used, followed by a bubble attack enhancement strategy designed to improve the ability to jump out of the local optimum by introducing a mutation method while the original helix updating method remains unchanged in the bubble net predation phase. After that, a failure parameter test mutation mechanism is introduced, conditions are set to trigger the termination of the algorithm to avoid over-computation, and the global search ability is improved and the algorithm convergence speed is accelerated by increasing the mutation operation. Finally, the stochastic gradient is used to assist in the optimization search, so that the algorithm searches the global optimal value as much as possible, and balances the global and local exploration ability of the algorithm. The pseudo-code of the algorithm is shown in Algorithm 1. The specific flow of the algorithm is shown in Figure 5.

3.4. Computational Complexity

Computational complexity encompasses both time complexity and space complexity. Time complexity constitutes a fundamental component of algorithm analysis, serving as a metric for both the performance and computational efficiency of algorithms. In this context, let N represent the population size, n denote the dimension of the search space, and Max_iter signify the maximum number of iterations. The time complexity of the conventional whale optimization algorithm can be articulated as $T_{woa} = O(N \times n \times Max\_iter) = O(n)$. The generalized bubble whale optimization algorithm (GLBWOA) enhances the standard WOA by integrating three distinct strategies, one of which is the bubble net attack enhancement

strategy that does not introduce additional nested loops. As a result, the time complexity for this strategy remains $T_1 = O(N \times n \times \text{Max\_iter}) = O(n)$. Let t_1 denote the cycle running time for formulas (20), (21), and (22), while t_2 represents the time required for fitness evaluations. Consequently, the time complexity associated with the failure parameter test mutation mechanism can be expressed as $T_2 = O(N \times n \times (t_1 + t_2)) = O(n)$. Furthermore, let t_3 indicate the time required for executing boundary constraint operations within the stochastic gradient-assisted optimization method. Thus, the time complexity for this method is $T_3 = O(N \times n \times t_3) = O(n)$. In conclusion, the overall time complexity of GLBWOA can be formulated as $T_{\text{GLBWOA}} = T_1 + \text{Max\_iter}(T_2 + T_3) = O(n)$. When juxtaposed with the standard WOA, it is apparent that the time complexity remains invariant, thereby preserving the execution efficiency of the algorithm. The subsequent section will assess the performance of GLBWOA through a series of function set tests and path-planning simulations and further analyze the optimization efficiency of the algorithm by documenting its actual running time in path-planning simulations.

Algorithm 1: GLBWOA

```

1.   Set parameters, population size, number of iterations, etc.
2.   Initialize the population.
3.   Calculate the fitness value of the initial population.
4.   Find the current optimal individual and the optimal value.
5.   Main loop
6.   For 1:N
7.       Use Equations (11)–(13) to update the whale algorithm parameters
8.       If  $p < 0.5$ 
9.           If  $|A| \geq 1$ 
10.              Equation (18) is used to update the random search phase stage.
11.           else
12.              Equation (10) is used to update the closed predator stage.
13.           end
14.       else
15.           If  $\text{rand} > 0.6$ 
16.              Equation (15) is used to update the bubble net attack.
17.           else
18.              Equation (19) update enhanced bubble network attack strategy.
19.           end
20.       Calculate the fitness value of each individual, and select the optimal individual and the
           optimal value at this time.
21.       Record the number of optimal value update failures and calculate the difference between
           the current best fitness and the global best fitness.
22.           If Value of difference  $> 10^{-6}$  or the number of optimal value update failures  $> 5$ 
23.               Using the Formula (20)–(21) to update the mutation parameters.
24.               Equation (22) is used to update the failed mutation formula.
25.           end
26.       Set the gradient step factor.
27.       Using Equation (24) to update the negative gradient direction
28.       Equation (26) is used to update the gradient step size.
29.       Equation (23), update the gradient-assisted optimization formula.
           Using the greedy algorithm, the optimal individual is compared. If the fitness value of the
30.       optimal individual after auxiliary optimization is better, the individual is replaced by the
           original individual.
31.       end
32.       Output the optimal solution, the optimal value.
  
```

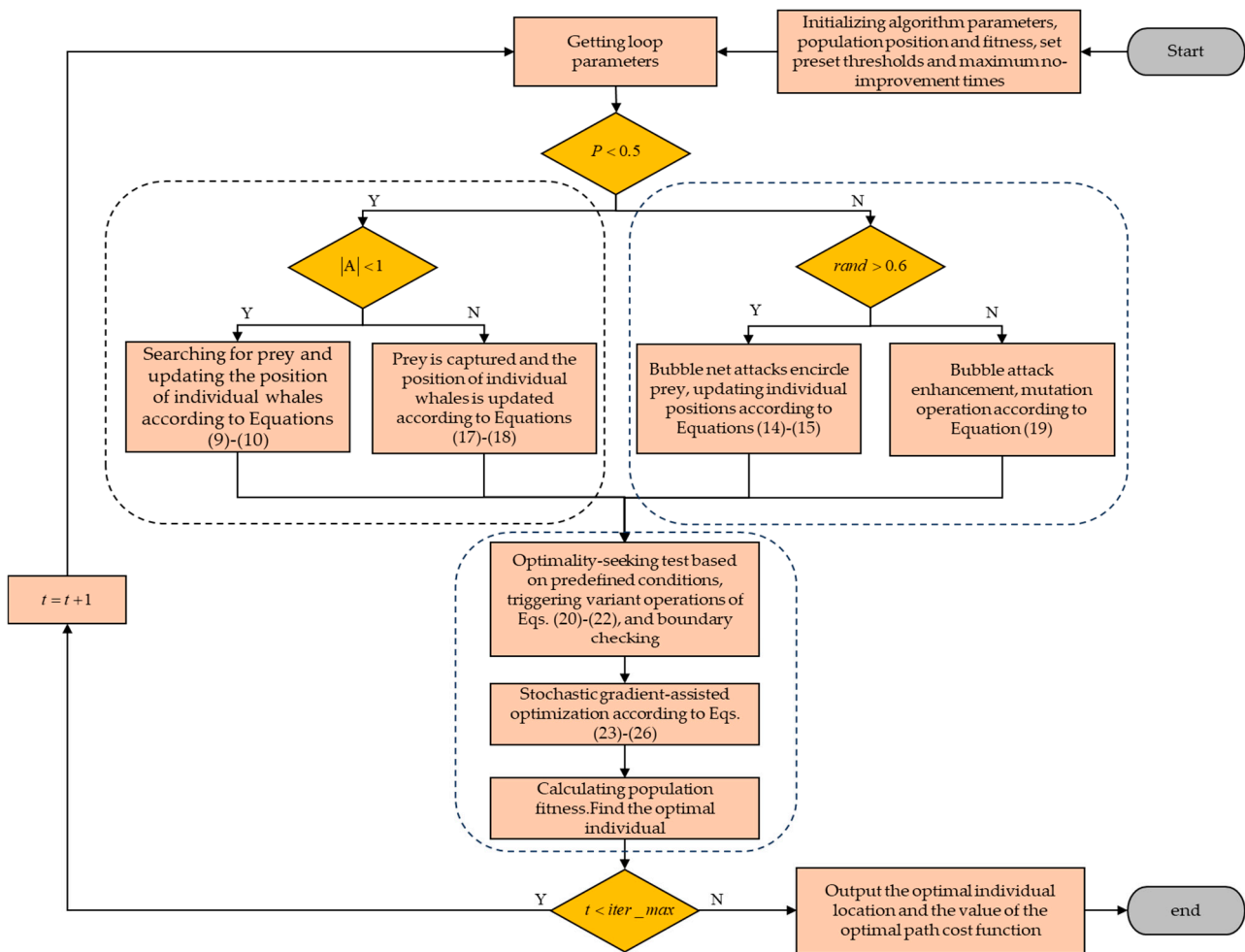


Figure 5. Global-local balanced whale optimization algorithm process.

Space complexity is a measure of the amount of storage space temporarily occupied by an algorithm during operation and is an important indicator of an algorithm's merit. The space complexity of standard WOA is $S_{WOA} = O(N \times n) + O(Max\_iter) = O(n)$. In GLBWOA, the space complexity of the position matrix is $S_1 = O(N \times n)$, the space complexity of the fitness storage is $S_2 = O(N)$, the space complexity of the leader information is $S_3 = O(n) + O(1)$, the space complexity of the convergence curve storage is $S_4 = O(Max\_iter)$, and the space complexity of the other temporary variables is $S_5 = O(1)$, so the overall space complexity of GLBWOA is $S_{GLBWOA} = O(N \times n) + O(N) + O(Max\_iter) = O(n)$, which shows that the space complexity of GLBWOA is consistent with that of the WOA in the large-scale problems. Space complexity is the same.

4. Simulation and Discussion

In this subsection, the test function is used to assess the optimization capabilities of the proposed algorithm, and the optimal path is identified within the actual elevation map. The superiority of the GLBWOA is demonstrated through a comparative analysis with other leading high-performance metaheuristic algorithms via numerical simulations. All simulations conducted in this study were performed on a Windows 11 platform equipped with a 13th Generation Intel Core™ i9-13900HX processor operating at 2.20 GHz. Additionally, MATLAB 2024a software was employed for the simulations.

4.1. Optimization Performance Test

The IEEE CEC2017 function is utilized to evaluate the optimization performance of the GLBWOA. In addition to comparing it with the basic WOA, three advanced intelligent

optimization algorithms—BWO, SCA, and BOA—are also employed for comparative analysis. To ensure a fair evaluation, all algorithms are configured to operate in dimensions of 30, 50, and 100. Simulations are conducted across these varying dimensions, with a maximum evaluation time set to 10,000 times the dimensionality. The population size is fixed at 30, and the maximum number of iterations is limited to 500. In this study, each algorithm is executed independently for 30 trials, and metrics such as the optimal value, mean, and standard deviation are recorded to assess the optimization performance of the algorithms.

The CEC2017 function test set contains four types of functions, where F1 to F3 are single-peak functions, F4 to F10 are basic multi-peak functions, F11 to F20 are hybrid functions, and F21 to F30 are composite functions. The performance of the algorithms tested on the different function types will be analyzed in turn below.

From the data presented in Table 1, it is evident that, for the single-peak function, the GLBWOA can reach the theoretical optimum more efficiently than the other algorithms within a limited number of iterations. This suggests that GLBWOA possesses a stronger convergence capability. However, it is also observed that the stability of the optimization search conducted by GLBWOA slightly diminishes under high-dimensional conditions.

Table 1. Algorithm test results. Significant values are in bold.

Func	Alg	d = 30			d = 50			d = 100		
		Best	Ave	Std	Best	Ave	Std	Best	Ave	Std
F1	Ours	4.9E + 04	1.4E + 05	6.6E + 04	1.0E + 03	9.3E + 03	8.1E + 03	5.7E + 03	2.7E + 04	2.2E + 04
	WOA	2.1E + 09	5.3E + 09	2.1E + 09	1.3E + 10	2.0E + 10	4.6E + 09	9.0E + 10	1.1E + 11	1.0E + 10
	BWO	4.0E + 10	5.3E + 10	5.8E + 09	9.4E + 10	1.1E + 11	4.7E + 09	2.5E + 11	2.6E + 11	6.9E + 09
	SCA	1.2E + 10	2.0E + 10	5.1E + 09	5.0E + 10	6.6E + 10	7.9E + 09	1.8E + 11	2.1E + 11	1.5E + 10
	BOA	3.7E + 10	5.3E + 10	6.7E + 09	8.6E + 10	1.1E + 11	9.6E + 09	2.3E + 11	2.6E + 11	1.2E + 10
F2	Ours	2.0E + 02	5.2E + 03	2.3E + 04	4.2E + 02	1.3E + 11	7.2E + 11	7.2E + 32	6.8E + 47	3.7E + 48
	WOA	2.2E + 26	5.7E + 34	1.4E + 35	7.0E + 62	4.8E + 76	2.0E + 77	9.4E + 146	1.9E + 175	6.6E + 04
	BWO	2.0E + 41	4.8E + 46	1.9E + 47	6.1E + 69	1.2E + 80	4.6E + 80	3.2E + 162	5.2E + 172	6.6E + 04
	SCA	1.3E + 33	2.6E + 38	8.6E + 38	1.9E + 64	8.5E + 70	2.8E + 71	1.7E + 150	5.8E + 167	6.6E + 04
	BOA	3.0E + 41	1.6E + 53	3.5E + 53	3.4E + 80	3.2E + 93	9.9E + 93	7.5E + 185	1.7E + 197	6.6E + 04
F3	Ours	5.8E + 03	3.4E + 04	2.3E + 04	5.4E + 04	1.1E + 05	3.1E + 04	2.5E + 05	4.3E + 05	1.0E + 05
	WOA	1.4E + 05	2.7E + 05	7.8E + 04	1.8E + 05	3.1E + 05	1.2E + 05	7.6E + 05	9.2E + 05	1.1E + 05
	BWO	7.0E + 04	8.2E + 04	4.6E + 03	1.6E + 05	2.4E + 05	3.5E + 04	3.4E + 05	3.8E + 05	4.0E + 04
	SCA	6.5E + 04	8.9E + 04	1.7E + 04	1.5E + 05	2.2E + 05	3.2E + 04	4.0E + 05	6.1E + 05	9.5E + 04
	BOA	6.6E + 04	8.0E + 04	6.2E + 03	1.7E + 05	3.7E + 05	1.6E + 05	3.2E + 05	5.7E + 05	3.5E + 05

Table 2 presents the test data for the algorithms applied to the multi-peak function with local extreme points. The optimal and average values achieved through GLBWOA optimization are superior across different dimensions, demonstrating the algorithm's robust capability to escape local optima. Notably, in function F4, GLBWOA significantly outperforms other algorithms across various performance indicators, underscoring its excellent stability.

Table 2. Algorithm test results. Significant values are in bold.

Func	Alg	d = 30			d = 50			d = 100		
		Best	Ave	Std	Best	Ave	Std	Best	Ave	Std
F4	Ours	4.9E + 02	5.2E + 02	2.0E + 01	5.0E + 02	6.7E + 02	6.7E + 01	9.7E + 02	1.1E + 03	8.9E + 01
	WOA	8.2E + 02	1.3E + 03	4.0E + 02	2.6E + 03	4.5E + 03	1.2E + 03	1.4E + 04	2.1E + 04	3.6E + 03
	BWO	9.4E + 03	1.3E + 04	1.3E + 03	2.7E + 04	3.4E + 04	2.9E + 03	6.7E + 04	9.8E + 04	8.6E + 03
	SCA	1.8E + 03	3.3E + 03	1.2E + 03	9.1E + 03	1.3E + 04	2.4E + 03	3.9E + 04	5.3E + 04	8.4E + 03
	BOA	1.5E + 04	2.1E + 04	3.9E + 03	3.1E + 04	4.0E + 04	4.0E + 03	9.1E + 04	1.1E + 05	1.0E + 04

Table 2. Cont.

Func	Alg	d = 30			d = 50			d = 100		
		Best	Ave	Std	Best	Ave	Std	Best	Ave	Std
F5	Ours	6.4E + 02	7.0E + 02	3.7E + 01	8.0E + 02	8.6E + 02	4.0E + 01	1.2E + 03	1.3E + 03	6.8E + 01
	WOA	7.8E + 02	8.7E + 02	4.9E + 01	1.0E + 03	1.1E + 03	9.0E + 01	1.8E + 03	1.9E + 03	1.5E + 02
	BWO	9.0E + 02	9.3E + 02	1.7E + 01	1.2E + 03	1.2E + 03	1.6E + 01	2.0E + 03	2.1E + 03	2.3E + 01
	SCA	7.8E + 02	8.2E + 02	2.1E + 01	1.1E + 03	1.1E + 03	3.9E + 01	1.9E + 03	2.1E + 03	1.1E + 02
	BOA	8.9E + 02	9.3E + 02	1.9E + 01	1.2E + 03	1.2E + 03	2.3E + 01	2.0E + 03	2.1E + 03	3.7E + 01
F6	Ours	6.3E + 02	6.4E + 02	6.4E + 00	6.4E + 02	6.5E + 02	5.4E + 00	6.5E + 02	6.6E + 02	3.7E + 00
	WOA	6.6E + 02	6.8E + 02	1.3E + 01	6.9E + 02	7.0E + 02	7.9E + 00	6.9E + 02	7.1E + 02	1.2E + 01
	BWO	6.8E + 02	6.9E + 02	5.6E + 00	6.9E + 02	7.0E + 02	3.8E + 00	7.1E + 02	7.1E + 02	2.2E + 00
	SCA	6.5E + 02	6.7E + 02	1.0E + 01	6.7E + 02	6.9E + 02	6.9E + 00	7.0E + 02	7.1E + 02	4.8E + 00
	BOA	6.8E + 02	6.9E + 02	5.2E + 00	6.9E + 02	7.0E + 02	5.7E + 00	7.1E + 02	7.1E + 02	2.7E + 00
F7	Ours	9.8E + 02	1.2E + 03	8.2E + 01	1.4E + 03	1.6E + 03	8.4E + 01	2.9E + 03	3.2E + 03	2.6E + 02
	WOA	1.1E + 03	1.3E + 03	7.6E + 01	1.7E + 03	1.9E + 03	8.3E + 01	3.5E + 03	3.8E + 03	1.5E + 02
	BWO	1.3E + 03	1.4E + 03	2.9E + 01	1.8E + 03	2.0E + 03	5.7E + 01	3.8E + 03	3.9E + 03	6.3E + 01
	SCA	1.1E + 03	1.3E + 03	6.0E + 01	1.7E + 03	1.9E + 03	9.6E + 01	3.8E + 03	4.2E + 03	2.5E + 02
	BOA	1.3E + 03	1.4E + 03	3.0E + 01	1.9E + 03	2.0E + 03	5.4E + 01	3.8E + 03	4.0E + 03	6.3E + 01
F8	Ours	9.0E + 02	9.5E + 02	2.9E + 01	1.1E + 03	1.2E + 03	4.5E + 01	1.6E + 03	1.7E + 03	7.9E + 01
	WOA	9.9E + 02	1.1E + 03	5.0E + 01	1.3E + 03	1.4E + 03	8.6E + 01	2.3E + 03	2.4E + 03	8.9E + 01
	BWO	1.1E + 03	1.1E + 03	1.9E + 01	1.5E + 03	1.5E + 03	2.1E + 01	2.5E + 03	2.6E + 03	3.1E + 01
	SCA	1.1E + 03	1.1E + 03	2.2E + 01	1.4E + 03	1.5E + 03	3.8E + 01	2.2E + 03	2.4E + 03	7.6E + 01
	BOA	1.1E + 03	1.1E + 03	2.1E + 01	1.4E + 03	1.5E + 03	3.3E + 01	2.5E + 03	2.6E + 03	4.0E + 01
F9	Ours	3.4E + 03	5.5E + 03	1.2E + 03	1.1E + 04	1.4E + 04	1.6E + 03	2.1E + 04	2.5E + 04	2.0E + 03
	WOA	5.5E + 03	1.2E + 04	4.5E + 03	2.4E + 04	4.2E + 04	1.2E + 04	5.8E + 04	7.8E + 04	1.7E + 04
	BWO	8.1E + 03	1.1E + 04	1.1E + 03	3.3E + 04	3.9E + 04	2.3E + 03	7.5E + 04	8.1E + 04	3.4E + 03
	SCA	6.3E + 03	8.7E + 03	1.7E + 03	2.4E + 04	3.3E + 04	5.5E + 03	7.7E + 04	9.5E + 04	1.1E + 04
	BOA	8.9E + 03	1.1E + 04	1.3E + 03	3.5E + 04	4.0E + 04	2.7E + 03	7.6E + 04	8.4E + 04	4.4E + 03
F10	Ours	3.7E + 03	5.4E + 03	7.9E + 02	5.8E + 03	8.2E + 03	9.0E + 02	1.4E + 04	1.7E + 04	1.5E + 03
	WOA	6.6E + 03	7.6E + 03	5.1E + 02	1.1E + 04	1.3E + 04	1.1E + 03	2.6E + 04	3.0E + 04	1.5E + 03
	BWO	8.6E + 03	8.9E + 03	2.3E + 02	1.4E + 04	1.5E + 04	4.8E + 02	3.1E + 04	3.2E + 04	5.4E + 02
	SCA	7.3E + 03	8.9E + 03	4.4E + 02	1.4E + 04	1.5E + 04	5.3E + 02	3.2E + 04	3.3E + 04	4.4E + 02
	BOA	8.7E + 03	9.3E + 03	2.9E + 02	1.5E + 04	1.6E + 04	4.4E + 02	3.1E + 04	3.3E + 04	7.4E + 02

Table 3 presents the test data for the algorithm applied to hybrid functions. Each function is either rotated or shifted and comprises three or more CEC2017 benchmark functions, with each subfunction assigned a specific weight. The difficulty of solving these functions has significantly increased compared to the previous ones. For the GLBWOA, its optimization accuracy remains unaffected by rotation or displacement. However, the individual indices for F20 are slightly lower than those of the other algorithms. In contrast, the optimal and average values for the other nine functions are considerably higher than those of the comparison algorithms. This demonstrates GLBWOA's strong global exploration capability and its superior ability to identify optimal solutions.

Table 3. Algorithm test results. Significant values are in bold.

Func	Alg	d = 30			d = 50			d = 100		
		Best	Ave	Std	Best	Ave	Std	Best	Ave	Std
F11	Ours	1.1E + 03	1.2E + 03	5.0E + 01	1.3E + 03	1.4E + 03	8.1E + 01	9.0E + 03	1.9E + 04	1.3E + 04
	WOA	3.7E + 03	1.0E + 04	4.3E + 03	4.1E + 03	8.5E + 03	2.2E + 03	1.5E + 05	2.8E + 05	1.2E + 05
	BWO	5.3E + 03	8.0E + 03	1.4E + 03	2.0E + 04	2.3E + 04	1.4E + 03	2.3E + 05	3.6E + 05	8.3E + 04
	SCA	2.4E + 03	4.0E + 03	1.0E + 03	7.0E + 03	1.2E + 04	2.5E + 03	1.1E + 05	1.8E + 05	3.7E + 04
	BOA	3.5E + 03	8.3E + 03	2.6E + 03	1.8E + 04	2.4E + 04	2.8E + 03	1.7E + 05	4.1E + 05	2.4E + 05

Table 3. Cont.

Func	Alg	d = 30			d = 50			d = 100		
		Best	Ave	Std	Best	Ave	Std	Best	Ave	Std
F12	Ours	1.6E + 06	5.1E + 06	2.6E + 06	2.9E + 06	1.1E + 07	6.8E + 06	7.5E + 06	3.4E + 07	1.6E + 07
	WOA	1.6E + 08	4.7E + 08	2.3E + 08	2.1E + 09	4.7E + 09	1.9E + 09	1.7E + 10	2.8E + 10	5.8E + 09
	BWO	8.4E + 09	1.2E + 10	1.5E + 09	4.4E + 10	6.5E + 10	8.9E + 09	1.5E + 11	1.9E + 11	1.2E + 10
	SCA	1.5E + 09	2.9E + 09	9.3E + 08	1.3E + 10	2.2E + 10	4.7E + 09	8.0E + 10	1.0E + 11	1.2E + 10
	BOA	4.8E + 09	1.4E + 10	4.2E + 09	5.0E + 10	8.3E + 10	1.7E + 10	1.4E + 11	2.0E + 11	2.3E + 10
F13	Ours	1.8E + 04	6.2E + 04	5.0E + 04	1.3E + 04	4.3E + 04	3.3E + 04	1.1E + 04	2.5E + 04	8.1E + 03
	WOA	1.6E + 06	1.9E + 07	4.5E + 07	1.2E + 08	6.0E + 08	3.7E + 08	1.2E + 09	2.7E + 09	1.1E + 09
	BWO	9.7E + 08	6.4E + 09	2.4E + 09	1.1E + 10	3.8E + 10	9.6E + 09	2.9E + 10	4.3E + 10	4.7E + 09
	SCA	1.7E + 08	1.1E + 09	5.9E + 08	3.4E + 09	6.4E + 09	1.7E + 09	1.3E + 10	1.8E + 10	3.6E + 09
	BOA	3.2E + 09	1.4E + 10	6.6E + 09	3.0E + 10	4.8E + 10	1.2E + 10	3.6E + 10	4.8E + 10	5.4E + 09
F14	Ours	2.1E + 04	3.4E + 05	2.9E + 05	1.1E + 05	5.5E + 05	2.3E + 05	7.0E + 05	2.0E + 06	8.3E + 05
	WOA	5.0E + 04	1.9E + 06	2.0E + 06	8.2E + 05	6.0E + 06	3.9E + 06	8.0E + 06	1.9E + 07	7.7E + 06
	BWO	1.2E + 06	3.9E + 06	2.1E + 06	8.6E + 06	5.6E + 07	2.5E + 07	4.6E + 07	7.7E + 07	1.9E + 07
	SCA	1.3E + 05	1.1E + 06	9.8E + 05	2.1E + 06	8.2E + 06	4.3E + 06	2.0E + 07	6.2E + 07	3.1E + 07
	BOA	8.3E + 04	3.6E + 06	3.3E + 06	2.1E + 07	1.3E + 08	9.6E + 07	4.2E + 07	1.4E + 08	7.0E + 07
F15	Ours	3.8E + 03	1.6E + 04	1.0E + 04	7.9E + 03	2.2E + 04	9.5E + 03	8.7E + 03	1.9E + 04	6.2E + 03
	WOA	9.7E + 04	1.4E + 07	3.6E + 07	1.9E + 06	5.2E + 07	4.9E + 07	1.1E + 08	4.7E + 08	3.0E + 08
	BWO	6.4E + 07	3.5E + 08	2.4E + 08	3.0E + 09	6.0E + 09	1.4E + 09	1.9E + 10	2.4E + 10	2.3E + 09
	SCA	5.0E + 06	5.3E + 07	5.4E + 07	2.8E + 08	1.1E + 09	4.5E + 08	2.9E + 09	6.6E + 09	1.6E + 09
	BOA	4.0E + 07	7.1E + 08	6.9E + 08	2.8E + 09	7.9E + 09	3.2E + 09	1.5E + 10	2.4E + 10	4.8E + 09
F16	Ours	2.6E + 03	3.1E + 03	3.9E + 02	2.9E + 03	4.1E + 03	5.2E + 02	5.4E + 03	7.0E + 03	8.3E + 02
	WOA	2.9E + 03	4.3E + 03	4.9E + 02	4.8E + 03	6.6E + 03	1.0E + 03	1.2E + 04	1.7E + 04	3.2E + 03
	BWO	4.4E + 03	5.6E + 03	3.2E + 02	7.1E + 03	8.8E + 03	8.6E + 02	2.0E + 04	2.3E + 04	1.1E + 03
	SCA	3.3E + 03	4.2E + 03	1.8E + 03	5.6E + 03	6.3E + 03	3.6E + 02	1.3E + 04	1.5E + 04	1.1E + 03
	BOA	4.7E + 03	7.8E + 03	9.3E + 02	8.0E + 03	1.1E + 04	1.6E + 03	2.3E + 04	2.6E + 04	1.9E + 03
F17	Ours	2.0E + 03	2.4E + 03	2.3E + 02	2.8E + 03	3.7E + 03	3.6E + 02	4.9E + 03	5.9E + 03	7.3E + 02
	WOA	2.1E + 03	2.8E + 03	3.5E + 02	3.9E + 03	4.9E + 03	7.9E + 02	9.6E + 03	3.7E + 04	2.9E + 04
	BWO	3.6E + 03	4.3E + 03	4.7E + 02	5.0E + 03	7.6E + 03	1.6E + 03	3.9E + 05	5.6E + 06	3.4E + 06
	SCA	2.4E + 03	2.8E + 03	1.9E + 02	4.1E + 03	5.1E + 03	4.8E + 02	1.5E + 04	6.5E + 04	5.4E + 04
	BOA	3.8E + 03	1.1E + 04	9.8E + 03	5.3E + 03	1.7E + 04	1.3E + 04	1.2E + 06	1.8E + 07	1.7E + 07
F18	Ours	1.3E + 05	1.2E + 06	1.3E + 06	2.0E + 05	3.5E + 06	2.4E + 06	1.8E + 06	3.1E + 06	1.0E + 06
	WOA	1.2E + 06	1.1E + 07	8.1E + 06	2.6E + 06	5.0E + 07	4.3E + 07	8.9E + 06	2.1E + 07	1.0E + 07
	BWO	5.0E + 06	3.6E + 07	1.6E + 07	6.5E + 07	1.5E + 08	3.9E + 07	7.6E + 07	2.1E + 08	6.2E + 07
	SCA	1.7E + 06	1.9E + 07	1.4E + 07	2.5E + 07	7.3E + 07	3.8E + 07	3.5E + 07	1.2E + 08	5.5E + 07
	BOA	4.9E + 06	9.4E + 07	1.1E + 08	3.8E + 07	2.1E + 08	1.2E + 08	7.3E + 07	2.5E + 08	1.2E + 08
F19	Ours	2.8E + 03	1.6E + 04	1.8E + 04	7.5E + 03	6.1E + 04	5.7E + 04	1.6E + 04	2.5E + 05	2.2E + 05
	WOA	1.2E + 06	2.0E + 07	1.6E + 07	3.0E + 06	2.2E + 07	2.2E + 07	2.6E + 08	5.3E + 08	2.5E + 08
	BWO	1.4E + 08	5.2E + 08	2.4E + 08	1.2E + 09	3.3E + 09	9.4E + 08	1.2E + 10	2.3E + 10	2.7E + 09
	SCA	2.9E + 07	1.2E + 08	7.5E + 07	2.2E + 08	7.3E + 08	3.2E + 08	2.4E + 09	5.4E + 09	1.4E + 09
	BOA	8.8E + 07	8.9E + 08	6.8E + 08	5.8E + 08	4.2E + 09	1.8E + 09	1.6E + 10	2.5E + 10	4.1E + 09
F20	Ours	2.4E + 03	2.8E + 03	2.1E + 02	3.1E + 03	3.6E + 03	2.7E + 02	4.7E + 03	5.8E + 03	5.6E + 02
	WOA	2.5E + 03	3.0E + 03	2.3E + 02	2.9E + 03	4.0E + 03	4.3E + 02	5.7E + 03	7.2E + 03	6.4E + 02
	BWO	2.8E + 03	3.1E + 03	1.2E + 02	3.5E + 03	4.1E + 03	1.8E + 02	7.3E + 03	7.8E + 03	2.3E + 02
	SCA	2.6E + 03	3.0E + 03	1.7E + 02	3.9E + 03	4.3E + 03	2.3E + 02	7.1E + 03	8.0E + 03	3.4E + 02
	BOA	2.8E + 03	3.1E + 03	1.3E + 02	3.6E + 03	4.3E + 03	2.4E + 02	7.5E + 03	8.1E + 03	2.8E + 02

F21–F30 are composite functions that consist of at least three hybrid functions or CEC2017 benchmark functions that have been rotated and shifted. Each subfunction includes additional bias values and a weight, which further complicates the optimization challenges faced by the algorithms. Table 4 presents the test data for each algorithm within the composite function. The simulation results indicate that, across various dimensions, the GLBWOA demonstrates superior performance in identifying optimal solutions. It excels in both optimal

value and average value metrics, particularly in functions F25, F28, and F30, where GLBWOA's performance indices significantly surpass those of other algorithms. This suggests that GLBWOA enhances local search capabilities compared to the WOA, leading to improved solution accuracy and a robust ability to avoid local optima, thereby facilitating the discovery of optimal solutions. Even when other algorithms exhibit low search accuracy or fail to converge, GLBWOA consistently maintains high solution accuracy.

To verify whether there is a significant difference between GLBWOA and the other algorithms, the experimental data were statistically analyzed using the Wilcoxon rank sum test [37]. For each test function, the Wilcoxon rank sum test was performed by combining the results of 30 independent optimization searches for GLBWOA with the results of 30 independent optimization searches for WOA, BWO, SCA, and BOA, respectively, at a significance level of 5%, with all algorithmic dimensions set to 30, 50, and 100 cases simultaneously, the population size set to 30, and the number of iterations set to 500 generations. The p -value of the test result is less than 0.05, indicating that there is a significant difference between the compared algorithms, and vice versa there is no significant difference. The symbols '<' '>', and '=' indicate that the performance of GLBWOA is better, worse, and equivalent to the comparison algorithms, respectively. The results in Table 5 show that almost all p -values are less than 5%, indicating that GLBWOA's ability to find the best is better than the remaining four comparison algorithms.

Table 4. Algorithm test results. Significant values are in bold.

Func	Alg	d = 30			d = 50			d = 100		
		Best	Ave	Std	Best	Ave	Std	Best	Ave	Std
F21	Ours	2.4E + 03	2.5E + 03	4.8E + 01	2.6E + 03	2.7E + 03	6.7E + 01	3.2E + 03	3.5E + 03	1.8E + 02
	WOA	2.5E + 03	2.6E + 03	6.5E + 01	2.8E + 03	3.1E + 03	1.1E + 02	4.1E + 03	4.4E + 03	2.0E + 02
	BWO	2.6E + 03	2.7E + 03	3.5E + 01	3.1E + 03	3.2E + 03	5.8E + 01	4.5E + 03	4.8E + 03	9.3E + 01
	SCA	2.6E + 03	2.6E + 03	2.5E + 01	2.9E + 03	3.0E + 03	5.0E + 01	4.1E + 03	4.2E + 03	8.8E + 01
	BOA	2.5E + 03	2.7E + 03	5.9E + 01	3.1E + 03	3.3E + 03	8.4E + 01	4.5E + 03	4.9E + 03	1.7E + 02
F22	Ours	2.3E + 03	4.4E + 03	2.5E + 03	8.5E + 03	1.0E + 04	9.5E + 02	1.8E + 04	2.0E + 04	1.3E + 03
	WOA	3.1E + 03	8.1E + 03	1.8E + 03	1.3E + 04	1.5E + 04	1.1E + 03	2.9E + 04	3.2E + 04	1.5E + 03
	BWO	7.4E + 03	8.7E + 03	6.5E + 02	1.5E + 04	1.7E + 04	6.9E + 02	3.4E + 04	3.5E + 04	5.0E + 02
	SCA	4.2E + 03	9.7E + 03	1.9E + 03	1.6E + 04	1.7E + 04	4.5E + 02	3.4E + 04	3.5E + 04	6.9E + 02
	BOA	5.0E + 03	7.2E + 03	1.5E + 03	1.3E + 04	1.7E + 04	9.0E + 02	3.4E + 04	3.6E + 04	5.9E + 02
F23	Ours	2.8E + 03	2.9E + 03	6.8E + 01	3.1E + 03	3.3E + 03	1.4E + 02	3.8E + 03	4.1E + 03	1.7E + 02
	WOA	2.9E + 03	3.2E + 03	1.2E + 02	3.4E + 03	3.8E + 03	1.9E + 02	4.9E + 03	5.3E + 03	2.6E + 02
	BWO	3.2E + 03	3.3E + 03	5.8E + 01	4.0E + 03	4.1E + 03	9.4E + 01	5.9E + 03	6.1E + 03	1.3E + 02
	SCA	3.0E + 03	3.1E + 03	3.9E + 01	3.5E + 03	3.7E + 03	8.1E + 01	5.1E + 03	5.3E + 03	1.2E + 02
	BOA	3.2E + 03	3.6E + 03	1.7E + 02	4.1E + 03	4.7E + 03	2.8E + 02	6.2E + 03	6.7E + 03	3.2E + 02
F24	Ours	3.0E + 03	3.1E + 03	8.3E + 01	3.3E + 03	3.5E + 03	1.4E + 02	4.5E + 03	5.1E + 03	2.5E + 02
	WOA	3.1E + 03	3.3E + 03	8.8E + 01	3.6E + 03	3.9E + 03	1.5E + 02	6.1E + 03	6.8E + 03	4.7E + 02
	BWO	3.4E + 03	3.6E + 03	1.0E + 02	4.1E + 03	4.5E + 03	1.5E + 02	8.7E + 03	9.3E + 03	4.3E + 02
	SCA	3.2E + 03	3.3E + 03	3.7E + 01	3.7E + 03	3.9E + 03	7.1E + 01	6.8E + 03	7.3E + 03	2.9E + 02
	BOA	3.5E + 03	4.1E + 03	2.9E + 02	4.9E + 03	5.5E + 03	3.3E + 02	9.7E + 03	1.3E + 04	1.3E + 03
F25	Ours	2.9E + 03	2.9E + 03	2.1E + 01	3.1E + 03	3.2E + 03	3.1E + 01	3.6E + 03	3.8E + 03	8.7E + 01
	WOA	3.1E + 03	3.2E + 03	7.7E + 01	4.3E + 03	5.4E + 03	6.9E + 02	9.5E + 03	1.1E + 04	1.0E + 03
	BWO	4.0E + 03	4.5E + 03	2.0E + 02	1.2E + 04	1.4E + 04	9.2E + 02	2.6E + 04	2.8E + 04	1.0E + 03
	SCA	3.3E + 03	3.6E + 03	2.2E + 02	6.7E + 03	9.0E + 03	1.3E + 03	1.8E + 04	2.2E + 04	2.0E + 03
	BOA	4.8E + 03	5.8E + 03	5.3E + 02	1.4E + 04	1.6E + 04	8.6E + 02	2.8E + 04	3.0E + 04	1.6E + 03
F26	Ours	2.8E + 03	5.6E + 03	1.8E + 03	4.0E + 03	1.0E + 04	2.2E + 03	2.1E + 04	2.5E + 04	2.1E + 03
	WOA	6.8E + 03	8.6E + 03	1.0E + 03	1.2E + 04	1.5E + 04	1.8E + 03	2.9E + 04	3.8E + 04	3.7E + 03
	BWO	9.2E + 03	1.1E + 04	5.0E + 02	1.6E + 04	1.7E + 04	4.3E + 02	4.8E + 04	5.1E + 04	1.3E + 03
	SCA	6.8E + 03	7.9E + 03	5.4E + 02	1.2E + 04	1.4E + 04	8.1E + 02	3.6E + 04	4.2E + 04	3.0E + 03
	BOA	1.0E + 04	1.2E + 04	8.3E + 02	1.7E + 04	1.8E + 04	7.0E + 02	5.3E + 04	5.8E + 04	2.2E + 03

Table 4. Cont.

Func	Alg	d = 30			d = 50			d = 100		
		Best	Ave	Std	Best	Ave	Std	Best	Ave	Std
F27	Ours	3.2E + 03	3.3E + 03	2.9E + 01	3.6E + 03	3.9E + 03	2.6E + 02	3.8E + 03	4.3E + 03	3.2E + 02
	WOA	3.3E + 03	3.5E + 03	1.4E + 02	4.0E + 03	4.9E + 03	6.6E + 02	5.0E + 03	6.5E + 03	1.0E + 03
	BWO	3.6E + 03	4.0E + 03	1.7E + 02	4.7E + 03	6.3E + 03	4.5E + 02	1.0E + 04	1.2E + 04	7.2E + 02
	SCA	3.4E + 03	3.6E + 03	6.9E + 01	4.6E + 03	5.0E + 03	2.4E + 02	7.6E + 03	8.7E + 03	6.7E + 02
	BOA	3.8E + 03	4.4E + 03	3.0E + 02	5.5E + 03	7.1E + 03	8.2E + 02	1.3E + 04	1.5E + 04	1.3E + 03
F28	Ours	3.3E + 03	3.3E + 03	2.3E + 01	3.4E + 03	3.5E + 03	6.1E + 01	3.8E + 03	4.0E + 03	1.5E + 02
	WOA	3.7E + 03	4.0E + 03	2.9E + 02	5.3E + 03	6.2E + 03	5.2E + 02	1.1E + 04	1.5E + 04	1.4E + 03
	BWO	5.7E + 03	6.5E + 03	3.8E + 02	1.2E + 04	1.2E + 04	4.9E + 02	2.6E + 04	2.8E + 04	8.4E + 02
	SCA	3.9E + 03	4.5E + 03	4.7E + 02	7.3E + 03	8.9E + 03	1.0E + 03	2.4E + 04	2.7E + 04	2.6E + 03
	BOA	6.9E + 03	8.1E + 03	5.1E + 02	1.3E + 04	1.5E + 04	8.1E + 02	3.4E + 04	3.7E + 04	1.8E + 03
F29	Ours	3.6E + 03	4.2E + 03	3.0E + 02	4.4E + 03	5.2E + 03	5.2E + 03	7.3E + 03	8.9E + 03	7.5E + 02
	WOA	4.3E + 03	5.4E + 03	5.8E + 02	7.2E + 03	9.4E + 03	9.4E + 03	1.4E + 04	2.0E + 04	2.7E + 03
	BWO	6.0E + 03	7.1E + 03	6.4E + 02	1.2E + 04	3.0E + 04	3.0E + 04	1.1E + 05	4.3E + 05	2.0E + 05
	SCA	4.6E + 03	5.2E + 03	2.4E + 02	6.6E + 03	9.2E + 03	9.2E + 03	1.9E + 04	3.2E + 04	8.0E + 03
	BOA	6.5E + 03	1.2E + 04	4.7E + 03	1.3E + 04	2.8E + 05	2.8E + 05	1.6E + 05	1.1E + 06	7.5E + 05
F30	Ours	6.7E + 04	5.3E + 05	2.7E + 05	2.9E + 06	6.5E + 06	1.7E + 06	1.4E + 06	4.0E + 06	1.6E + 06
	WOA	5.7E + 06	6.6E + 07	4.9E + 07	1.3E + 08	3.1E + 08	1.1E + 08	8.6E + 08	2.8E + 09	1.3E + 09
	BWO	2.8E + 08	1.1E + 09	4.7E + 08	3.4E + 09	5.2E + 09	9.0E + 08	3.2E + 10	4.1E + 10	2.8E + 09
	SCA	5.2E + 07	1.8E + 08	7.3E + 07	5.3E + 08	1.2E + 09	3.5E + 08	8.7E + 09	1.3E + 10	2.7E + 09
	BOA	2.9E + 08	1.4E + 09	1.1E + 09	2.2E + 09	7.8E + 09	2.9E + 09	2.5E + 10	4.0E + 10	6.5E + 09

The primary objective of the improved algorithm is to find the best solution as quickly as possible. As depicted in Figure 6, which showcases the average convergence curve derived from the function tests, the GLBWOA demonstrates superior global search capabilities relative to four other high-performance optimization algorithms. It exhibits an improved capacity to evade local optima and achieves swift convergence across all 30 optimization functions. These findings indicate that GLBWOA is not only proficient in rapidly identifying superior solutions but also excels in addressing complex, multi-constraint problems with increased efficiency.

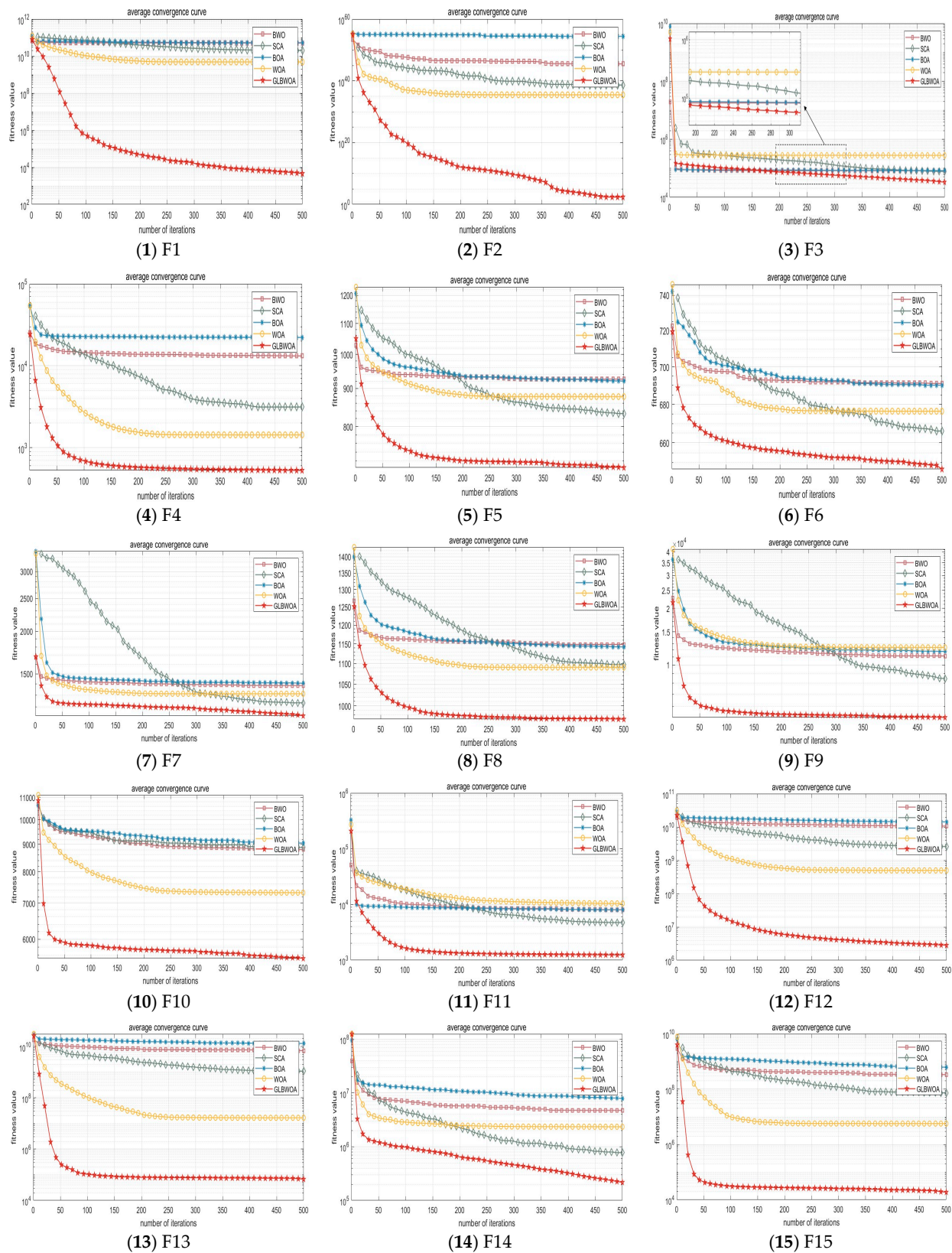


Figure 6. Cont.

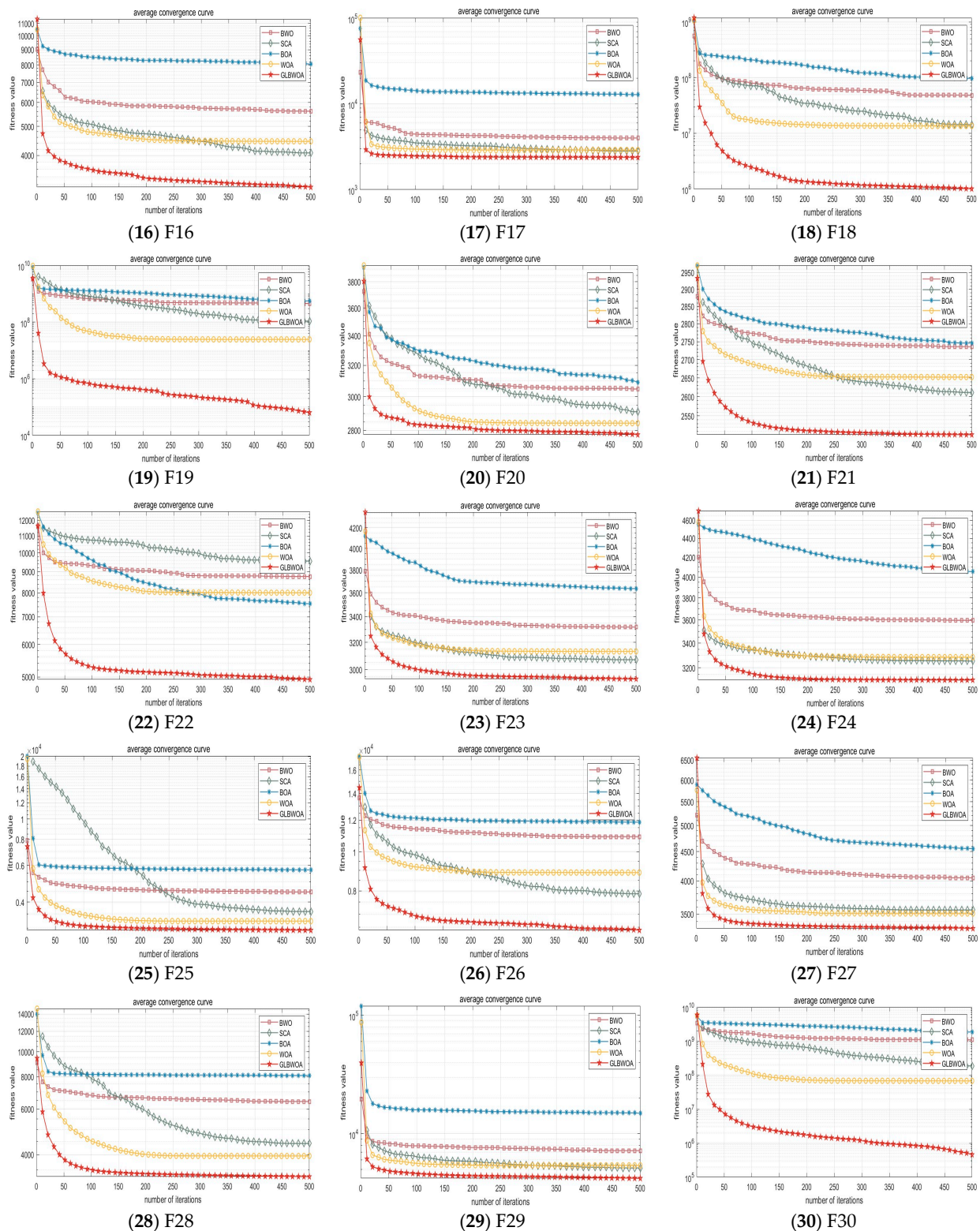


Figure 6. Average convergence curves of the algorithms.

In the analysis of convergence curves for thirty functions, the global best water wave optimization algorithm demonstrates superior convergence accuracy, particularly in the context of multi-modal functions. Specifically, in the case of function 3, while the initial fitness of GLBWOA is marginally higher than that of competing algorithms, it is observed that, as the number of iterations increases, the other four algorithms tend to converge prematurely,

resulting in entrapment within local optima. In contrast, GLBWOA maintains its capacity for continuous exploration of optimal solutions, thereby achieving enhanced optimization outcomes in the later stages of iteration, attributable to its distinct optimization strategies.

The findings indicate that the GLBWOA demonstrates significant efficiency in identifying optimal solutions across both low- and high-dimensional spaces. In comparison to other high-performance algorithms, GLBWOA exhibits superior capabilities in escaping local optima and possesses enhanced local search abilities.

4.2. Path-Planning Simulation

The simulations were conducted in two environments with different terrain structures on Christmas Island, Australia, where Map 1 has an extent of $1045 \times 879 \times Z$ m and Map 2 has an extent of $450 \times 450 \times Z$ m. The scenarios were classified as simple and complex according to the number and position of threatening cylinders, while the waypoints are selected as $n = 10$. To make a fair comparison, all the algorithmic parameters were unified, the population size is set to 30, and the maximum number of iterations is set to 200. Due to the stochastic nature of the metaheuristic algorithms, each algorithm is repeated independently 30 times to better illustrate the performance of the algorithms and ensure the reliability of the results, the total cost $F_{cost}(Path_i)$ is used as the main performance index of path planning, and according to the realistic flight requirements, the weighting coefficients of each cost function are set to $w_1 = 10$, $w_2 = 100$, $w_3 = 10$, $w_4 = 50$. The constraint parameters of the simulations are shown in Table 6.

Table 6. Constraint parameter setting.

Parameter	Numerical Value
UAV size D/m	10
Dangerous distance S/m	$2 \times D$
Maximum height h_{max}/m	200
Minimum height h_{min}/m	100
Maximum steering angle $\psi_{max}/(^{\circ})$	45
Maximum angle of climb $\theta_{max}/(^{\circ})$	45

The obstacle parameters for scenario 1 are shown in Table 7. Assuming that the starting point of the path is (200, 100, 150) and the end point is set to (800, 800, 150), the optimal path between the two points is planned. As shown in Figure 7, the best effect of path planning in scenario 1 is demonstrated. From the top view in Figure 7a and the side view in Figure 7b, it can be seen that all the algorithms can generate feasible paths that satisfy the requirements of five constraints, namely, path length, obstacle threat, height limitation, climb angle, and steering angle, and in order to observe the paths avoiding obstacles in a more intuitive manner, we chose to hide the terrain structure for observation. As shown in Figure 7c, it can be seen that the generated path can effectively avoid collision, and the path is smoother, in line with the real flight needs.

Table 7. Scenario 1 Parameter Setting.

Number	Location Coordinates $/m$	Threat Radius R/m
1	(400, 500, 200)	50
2	(600, 200, 200)	40
3	(500, 350, 200)	50

According to the distribution of terrain peaks, terrain 1 complex environment obstacles are set as in Table 8, with the same start and end points as above. Figure 8 shows the path-planning effect in the complex scene of terrain 1 and, with the increase in obstacles, the complexity of path solving rises, and the difficulty of algorithms to solve the optimal path increases. It can be seen that all the algorithms generate paths that can guide the UAV to

fly safely without collision, but different algorithms generate paths with large differences, among which BWO generates paths with more sharp turns, and the difficulty of actual flight is higher. BOA has the highest average total cost of planning and does not find the optimal path while, in comparison, the paths generated by SCA, WOA, and GLBWOA are more suitable for the actual flight of UAV, and GLBWOA has the fastest convergence speed among the three. The optimal path can be found in about 10 generations of iterations, and the average total cost is the lowest as in the case of WOA, reflecting the superiority of the algorithms proposed in this paper.

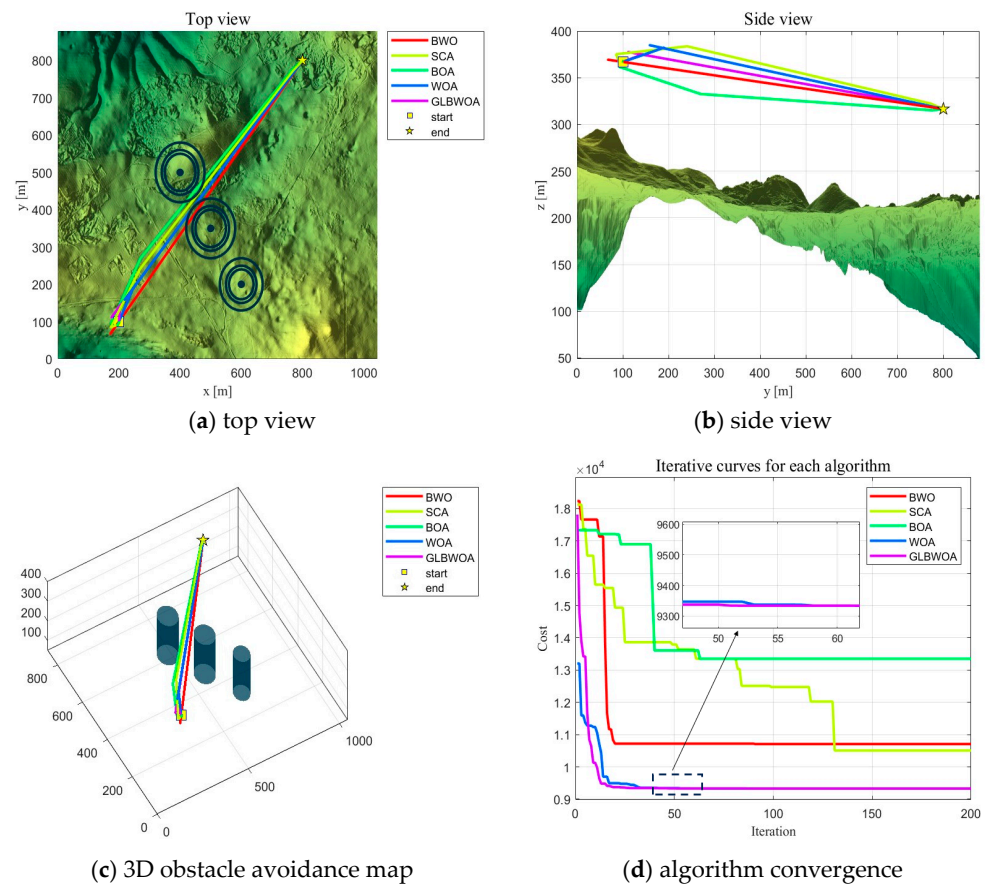


Figure 7. Path Planning in Scenario 1 ($n = 10$).

Table 8. Scenario 2 Parameter Setting.

Number	Location Coordinates /m	Threat Radius R/m
1	(400, 600, 200)	50
2	(300, 350, 100)	40
3	(500, 430, 150)	35
4	(200, 180, 200)	40
5	(200, 550, 200)	40
6	(650, 750, 150)	50
7	(820, 400, 175)	30
8	(730, 560, 200)	50
9	(450, 210, 200)	30

The path-planning simulation is continued under terrain 2 to further test the ability of the algorithm proposed in this paper to plan paths, where the path starting point is set to (10, 10, 200) and the end point is set to (400, 400, 150). The simple scenario obstacles are set as in Table 9, and the solved paths are shown in Figure 9, and it can be observed in Figure 9a,b that the paths generated by the GLBWOA are smoother in terms of path changes and without abrupt changes, which is better than the remaining four algorithms.

From Figure 9c, it can be seen that BOA and BWO fall into the local optimum at the early stage of the algorithm, which ultimately leads to premature convergence and planning paths with higher total cost, while GLBWOA has better optimality searching performance by quickly escaping after falling into the local extremes and continuously searching for the globally optimal paths on an ongoing basis. From Figure 9d, it is evident that the algorithm introduced in this study can swiftly escape local optima encountered during the initial stages. In contrast, other algorithms tend to remain trapped in local optima for extended durations throughout their iterations. This suggests that incorporating the bubble network attack enhancement strategy significantly improves the ability to escape local optima. A closer examination of the average convergence curve during the early iterations of the GLBWOA reveals that the algorithm activates the variance mechanism shortly after becoming stuck in a local optimum, enabling it to quickly identify the global optimum. This indicates that implementing the failure parameter testing mechanism markedly enhances the algorithm's global search capabilities.

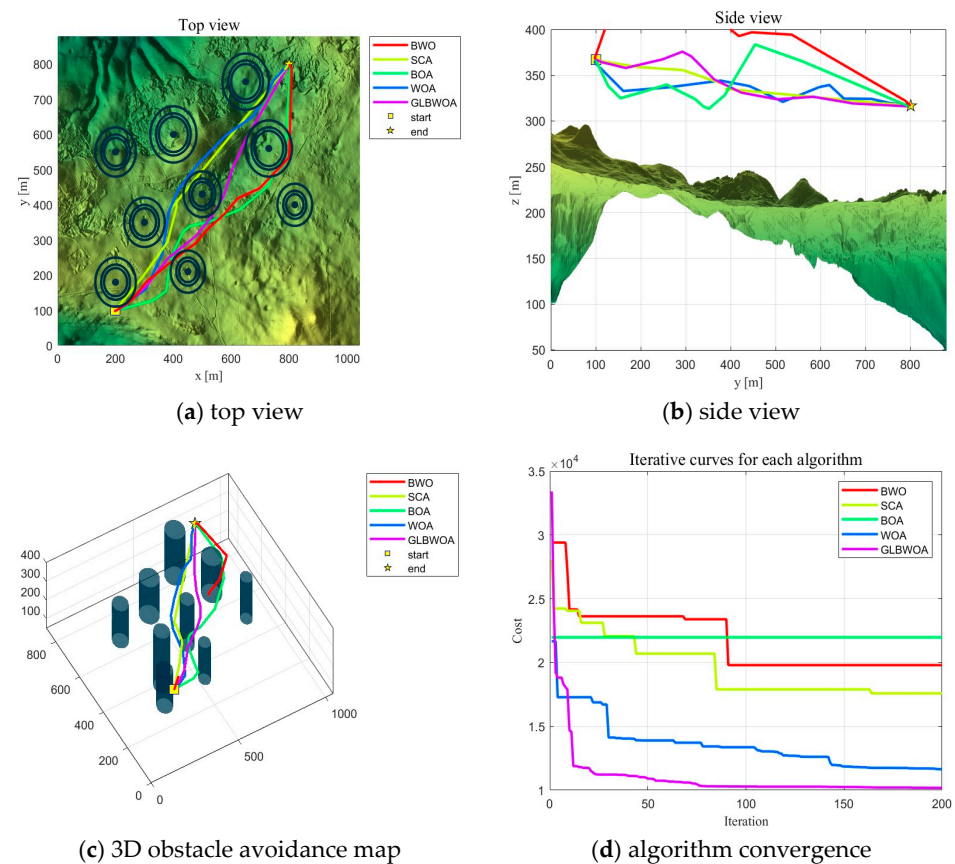


Figure 8. Path Planning in Scenario 2 ($n = 10$).

Table 9. Scenario 3 Parameter Setting.

Number	Location Coordinates /m	Threat Radius R/m
1	(100, 300, 100)	30
2	(300, 300, 100)	20
3	(100, 50, 100)	30

In the terrain 2 complex obstacle scenario to solve the safe flight path, the start and end points are the same as above, the obstacle parameters are set as in Table 10, and the results are shown in Figure 10, which shows that the related algorithms all give their respective path-planning results and the planned paths avoid the peaks and threat sources in the environment. It can be observed that although the scheme given by GLBWOA sacrifices the path length cost to plan the paths, the planned paths effectively reduce the altitude changes and steering

adjustments during the flight of the UAV and at the same time effectively avoid the obstacles, thus reducing the total cost, and the total cost is the lowest amongst the five algorithms. From Figure 10d, it can be seen that the BWO and WOA gradually begin to converge after around 30 generations of iterations, but GLBWOA is still searching for the globally optimal path, demonstrating that the algorithms have strong local exploration ability and global exploration performance, indicating that the introduction of the gradient kinetic energy strategy in the algorithms can escape from the local minima. The design of the update method effectively prevents the algorithm from missing the globally optimal solution.

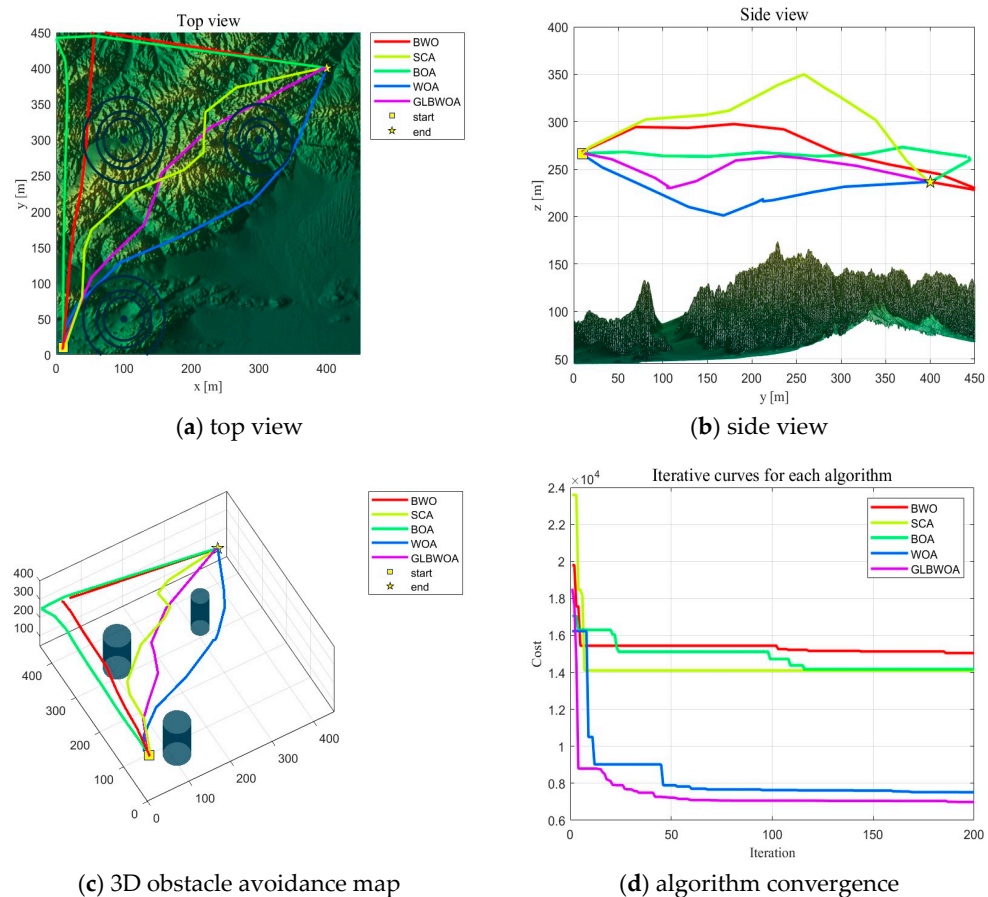


Figure 9. Path Planning in Scenario 3 ($n = 10$).

Table 10. Scenario 4 Parameter Setting.

Number	Location Coordinates /m	Threat Radius R/m
1	(300, 300, 100)	30
2	(200, 100, 100)	20
3	(100, 200, 100)	30
4	(300, 100, 100)	20
5	(200, 50, 100)	20
6	(150, 350, 100)	30

The flight data for path planning for the four benchmark scenarios are presented in Table 11. GLBWOA, along with the other algorithms, successfully generated collision-free UAV flight paths, and GLBWOA excelled in the optimal flight cost and average flight cost metrics. The only exception is in the terrain 1 simple obstacle scenario, where the total flight cost of GLBWOA is the same as the standard WOA. However, in the other three benchmark scenarios, the total flight cost of GLBWOA is reduced by 19%, 10.6%, and 24.9%, respectively, compared to the standard WOA. In addition, the improvements in convergence speed, accuracy, and stability highlight the effectiveness of the proposed

enhancement strategy. Meanwhile, Table 11 shows the average running time of each algorithm to determine the optimal path in 30 independent runs. It is clear that the GLBWOA has a slightly longer running time compared to the other algorithms but it is within an acceptable range. Although the extension of time is still within acceptable limits, it poses a limitation in application settings where the optimal path solution is required to be found in the shortest possible time. Therefore, our future research efforts will aim to reduce the running time of the algorithm while maintaining its performance level.

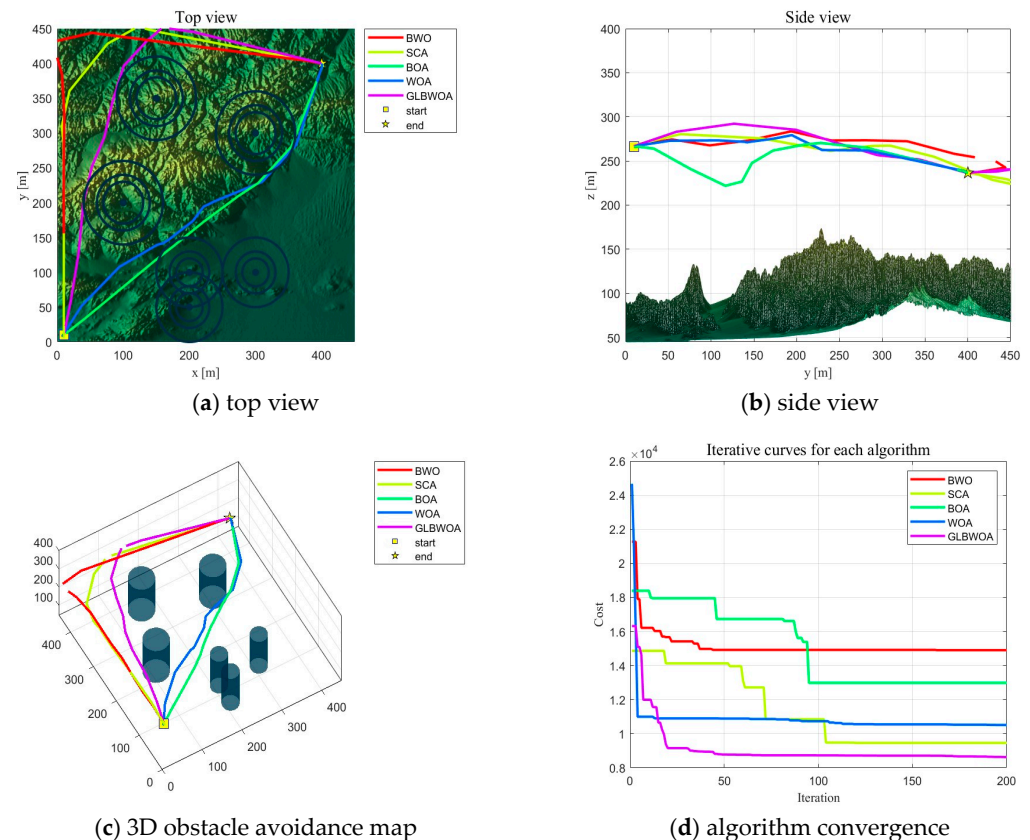


Figure 10. Path Planning in Scenario 4 ($n = 10$).

Table 11. Results of path-planning simulations. Significant values are in bold.

Alg	Scenario 1			Scenario 2			Scenario 3			Scenario 4		
	Best	Mean	Ave-Time	Best	Mean	Ave-Time	Best	Mean	Ave-Time	Best	Mean	Ave-Time
Ours	9330.37	9350.6	0.6597 s	10,185.1	11,381.5	1.3252 s	6989.56	7211.2	0.8263 s	8637.54	8900.2	1.3124 s
WOA	9330.37	9354.8	0.3390 s	12,644.5	12,774.6	0.7153 s	7817.26	8102.6	0.3634 s	11,514.3	11,902.6	0.4532 s
SCA	10,508.4	10,646.3	0.3217 s	17,588.3	18,363.2	0.7194 s	14,100.4	14,908.3	0.3544 s	9463.38	9779.6	0.5226 s
BWO	10,708.4	11,206.2	0.3696 s	19,799.7	19,828.3	0.7852 s	15,042.2	15,900.1	0.4290 s	14,921.2	15,112.3	0.6468 s
BOA	13,350.7	13,820.5	0.6195 s	21,975.4	22,131.8	1.1082 s	14,170.8	15,192.4	0.7109 s	12,984.4	13,155.7	0.8866 s

5. Conclusions

This study introduces a novel global–local balanced whale optimization algorithm designed to address the path-planning challenges faced by unmanned aerial vehicles in complex environments. A three-dimensional spatial model is constructed using digital elevation models, and a total cost function is formulated by integrating mission requirements with relevant constraints. To enhance the traditional whale optimization algorithm, a bubble net attack enhancement scheme is proposed to improve the algorithm’s ability to escape local optima. This enhancement is achieved through a mutation operation governed by predefined conditions, recognizing that an individual’s capacity to overcome

its current position varies at different stages of the optimization process. Additionally, a failure parameter testing mutation mechanism is incorporated to accelerate the algorithm's convergence rate. In the later phases of optimization, a stochastic gradient-assisted search strategy is employed to reinforce the algorithm's global search capabilities. The optimization performance of GLBWOA is evaluated using the CEC2017 function test set, where it demonstrates superior performance compared to four other high-performance algorithms across all metrics, indicating that the proposed strategies effectively balance the global and local search capabilities of the algorithm. In four benchmark scenarios, GLBWOA consistently achieves a lower average total flight cost and exhibits more accurate and faster convergence under identical algorithmic parameters. Future research will further assess the algorithm's performance by considering the impact of dynamic obstacles, weather variations, and other complex conditions on the UAV path-planning problem. Additionally, efforts will be made to implement the algorithm in real UAVs to conduct path planning in real-world environments, thereby validating its practical effectiveness.

Author Contributions: Conceptualization, Q.W. and W.T.; methodology, Q.W. and W.T.; software, W.T.; validation, Q.W. and W.T.; formal analysis, W.T.; investigation, W.T.; resources, L.Z.; data curation, H.W.; writing—original draft preparation, W.T.; writing—review and editing, Q.W. and W.T.; visualization, W.T.; supervision, W.T.; project administration, L.J.; funding acquisition, Q.W. and R.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Defense Science and Technology Innovation project, grant number ZZKY20223103; the Basic Frontier Innovation Project at the Engineering University of PAP, grant number WJY202429; the Basic Frontier Innovation Project at the Engineering University of PAP, grant number WJY202408; Key project of National Social Science Foundation, grant number 2023-SKJJ-A-116.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Elmeseiry, N.; Alshaer, N.; Ismail, T. A detailed survey and future directions of unmanned aerial vehicles (uavs) with potential applications. *Aerospace* **2021**, *8*, 363. [\[CrossRef\]](#)
2. Tsouros, D.C.; Bibi, S.; Sarigiannidis, P.G. A review on UAV-based applications for precision agriculture. *Information* **2019**, *10*, 349. [\[CrossRef\]](#)
3. Asadzadeh, S.; de Oliveira, W.J.; de Souza Filho, C.R. UAV-based remote sensing for the petroleum industry and environmental monitoring: State-of-the-art and perspectives. *J. Pet. Sci. Eng.* **2022**, *208*, 109633. [\[CrossRef\]](#)
4. Tang, P.; Li, J.; Sun, H. A Review of Electric UAV Visual Detection and Navigation Technologies for Emergency Rescue Missions. *Sustainability* **2024**, *16*, 2105. [\[CrossRef\]](#)
5. Du, P.; He, X.; Cao, H.; Garg, S.; Kaddoum, G.; Hassan, M.M. AI-based energy-efficient path planning of multiple logistics UAVs in intelligent transportation systems. *Comput. Commun.* **2023**, *207*, 46–55. [\[CrossRef\]](#)
6. Aggarwal, S.; Kumar, N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Comput. Commun.* **2020**, *149*, 270–299. [\[CrossRef\]](#)
7. Zhao, Y.; Zheng, Z.; Liu, Y. Survey on computational-intelligence-based UAV path planning. *Knowl.-Based Syst.* **2018**, *158*, 54–64. [\[CrossRef\]](#)
8. Tisdale, J.; Kim, Z.; Hedrick, J.K. Autonomous UAV path planning and estimation. *IEEE Robot. Autom. Mag.* **2009**, *16*, 35–42. [\[CrossRef\]](#)
9. Zhang, Z.; Jiang, J.; Wu, J.; Zhu, X. Efficient and optimal penetration path planning for stealth unmanned aerial vehicle using minimal radar cross-section tactics and modified A-Star algorithm. *ISA Trans.* **2023**, *134*, 42–57. [\[CrossRef\]](#)
10. Zhen, R.; Gu, Q.; Shi, Z.; Suo, Y. An improved A-star ship path-planning algorithm considering current, water depth, and traffic separation rules. *J. Mar. Sci. Eng.* **2023**, *11*, 1439. [\[CrossRef\]](#)
11. Hao, K.; Yang, Y.; Li, Z.; Liu, Y.; Zhao, X. CERRT: A Mobile Robot Path Planning Algorithm Based on RRT in Complex Environments. *Appl. Sci.* **2023**, *13*, 9666. [\[CrossRef\]](#)
12. Xin, P.; Wang, X.; Liu, X.; Wang, Y.; Zhai, Z.; Ma, X. Improved bidirectional RRT* algorithm for robot path planning. *Sensors* **2023**, *23*, 1041. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Zhao, W.; Li, L.; Wang, Y.; Zhan, H.; Fu, Y.; Song, Y. Research on A Global Path-Planning Algorithm for Unmanned Aerial Vehicle Swarm in Three-Dimensional Space Based on Theta*–Artificial Potential Field Method. *Drones* **2024**, *8*, 125. [\[CrossRef\]](#)

14. Chen, Y.-B.; Luo, G.-C.; Mei, Y.-S.; Yu, J.-Q.; Su, X.-I. UAV path planning using artificial potential field method updated by optimal control theory. *Int. J. Syst. Sci.* **2016**, *47*, 1407–1420. [\[CrossRef\]](#)
15. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control. Eng.* **2020**, *8*, 22–34. [\[CrossRef\]](#)
16. Yu, X.; Jiang, N.; Wang, X.; Li, M. A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning. *Expert Syst. Appl.* **2023**, *215*, 119327. [\[CrossRef\]](#)
17. Zhong, C.; Li, G.; Meng, Z. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. *Knowl.-Based Syst.* **2022**, *251*, 109215. [\[CrossRef\]](#)
18. Houssein, E.H.; Sayed, A. Dynamic candidate solution boosted beluga whale optimization algorithm for biomedical classification. *Mathematics* **2023**, *11*, 707. [\[CrossRef\]](#)
19. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [\[CrossRef\]](#)
20. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2019**, *23*, 715–734. [\[CrossRef\]](#)
21. Arora, S.; Singh, S. An improved butterfly optimization algorithm with chaos. *J. Intell. Fuzzy Syst.* **2017**, *32*, 1079–1088. [\[CrossRef\]](#)
22. Mortazavi, A.; Moloodpoor, M. Enhanced butterfly optimization algorithm with a new fuzzy regulator strategy and virtual butterfly concept. *Knowl.-Based Syst.* **2021**, *228*, 107291. [\[CrossRef\]](#)
23. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [\[CrossRef\]](#)
24. Jiang, R.; Yang, M.; Wang, S.; Chao, T. An improved whale optimization algorithm with armed force program and strategic adjustment. *Appl. Math. Model.* **2020**, *81*, 603–623. [\[CrossRef\]](#)
25. Huang, Y.; Li, Y.; Zhang, Z.; Sun, Q. A novel path planning approach for AUV based on improved whale optimization algorithm using segment learning and adaptive operator selection. *Ocean. Eng.* **2023**, *280*, 114591. [\[CrossRef\]](#)
26. Guo, W.; Liu, T.; Dai, F.; Xu, P. An improved whale optimization algorithm for forecasting water resources demand. *Appl. Soft Comput.* **2020**, *86*, 105925. [\[CrossRef\]](#)
27. Wang, C.-H.; Chen, S.; Zhao, Q.; Suo, Y. An efficient end-to-end obstacle avoidance path planning algorithm for intelligent vehicles based on improved whale optimization algorithm. *Mathematics* **2023**, *11*, 1800. [\[CrossRef\]](#)
28. Dai, Y.; Yu, J.; Zhang, C.; Zhan, B.; Zheng, X. A novel whale optimization algorithm of path planning strategy for mobile robots. *Appl. Intell.* **2023**, *53*, 10843–10857. [\[CrossRef\]](#)
29. Yin, S.; Yang, J.; Ma, L.; Fu, M.; Xu, K. An enhanced whale algorithm for three-dimensional path planning for meteorological detection of the unmanned aerial vehicle in complex environments. *IEEE Access* **2024**, *12*, 60039–60057. [\[CrossRef\]](#)
30. He, L.; Aouf, N.; Song, B. Explainable Deep Reinforcement Learning for UAV autonomous path planning. *Aerosp. Sci. Technol.* **2021**, *118*, 107052. [\[CrossRef\]](#)
31. de Castro, G.G.; Pinto, M.F.; Biundini, I.Z.; Melo, A.G.; Marcato, A.L.; Haddad, D.B. Dynamic path planning based on neural networks for aerial inspection. *J. Control. Autom. Electr. Syst.* **2023**, *34*, 85–105. [\[CrossRef\]](#)
32. Cui, Z.; Wang, Y. UAV path planning based on multi-layer reinforcement learning technique. *IEEE Access* **2021**, *9*, 59486–59497. [\[CrossRef\]](#)
33. Phung, M.D.; Ha, Q.P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Appl. Soft Comput.* **2021**, *107*, 107376. [\[CrossRef\]](#)
34. Liu, S.; Jin, Z.; Lin, H.; Lu, H. An improve crested porcupine algorithm for UAV delivery path planning in challenging environments. *Sci. Rep.* **2024**, *14*, 20445. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Hou, J.; Van Dijk, A.I.; Renzullo, L.J. Merging Landsat and airborne LiDAR observations for continuous monitoring of floodplain water extent, depth and volume. *J. Hydrol.* **2022**, *609*, 127684. [\[CrossRef\]](#)
36. Cao, R.; Si, L.; Li, X.; Guang, Y.; Wang, C.; Tian, Y.; Pei, X.; Zhang, X. A conjugate gradient-assisted multi-objective evolutionary algorithm for fluence map optimization in radiotherapy treatment. *Complex Intell. Syst.* **2022**, *8*, 4051–4077. [\[CrossRef\]](#)
37. He, Y.; Wang, M. An improved chaos sparrow search algorithm for UAV path planning. *Sci. Rep.* **2024**, *14*, 366. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.