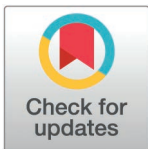


RESEARCH ARTICLE

Research on three-dimensional path planning of unmanned aerial vehicle based on improved Whale Optimization Algorithm

HaoCheng Wang[‡], ZeXian Hao[‡], Yu Zhang^{✉*}

Northeast Forestry University, Harbin, China

[‡] HW and ZH are jointly listed as the first authors.* zhangyu0902@163.com

Abstract

Addressing the insufficient optimization performance in drone 3D path planning and the issues of inadequate optimization precision and tendency to fall into local optima in the existing Whale Optimization Algorithm (WOA), this paper proposes a drone 3D path planning method based on an improved Whale Optimization Algorithm (CSR-D-WOA). Firstly, to enhance the search efficiency and fitness accuracy of the Whale Algorithm, the Cuckoo Search and Random Differential Strategy were introduced and compared with the traditional Particle Swarm Optimization algorithm, Whale Algorithm, and Cuckoo Search Algorithm. Experimental results demonstrate that the CSR-D-WOA algorithm improves global search capabilities and prevents premature convergence, significantly enhancing optimization precision and convergence speed. Secondly, applying the CSR-D-WOA algorithm to drone 3D path planning issues, the simulation results show that the CSR-D-WOA algorithm can effectively manage path planning in complex terrains, showcasing its application potential in drone path planning.

OPEN ACCESS

Citation: Wang H, Hao Z, Zhang Y (2025) Research on three-dimensional path planning of unmanned aerial vehicle based on improved Whale Optimization Algorithm. PLoS ONE 20(2): e0316836. <https://doi.org/10.1371/journal.pone.0316836>

Editor: Muhammad Suhail, Hanshan Normal University, CHINA

Received: July 31, 2024

Accepted: December 17, 2024

Published: February 24, 2025

Copyright: © 2025 Wang et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data availability statement: <https://doi.org/10.7910/DVN/UPYELW>

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

1. Introduction

The rapid development of drone technology has brought revolutionary changes to various application fields, especially in precision agriculture [1], urban planning, and emergency rescue, where effective path planning is a key factor in ensuring efficient and safe task execution by drones [2]. Path planning directly affects task execution efficiency and drone energy management [3].

In recent years, numerous scholars both domestically and internationally have explored drone path planning methods and developed various algorithms. These algorithms are generally divided into two categories: traditional algorithms, such as the artificial potential field method [4] and the A* algorithm [4], and intelligent algorithms, such as the Particle Swarm Optimization algorithm [5], Grey Wolf Optimization algorithm [6], and the Whale Optimization Algorithm. The Whale Optimization Algorithm (WOA), proposed by Mirjalili and others [7], has proven superior to other algorithms in function optimization due to its simple principles, minimal parameter settings, and robust optimization performance, finding successful application in numerous fields, including engineering optimization and function

maximization. However, the basic WOA has limitations in parameter control adjustment and maintaining population diversity, which restricts its balance between global search and local precision search, often leading it to fall prematurely into local optima [8]. Recent improvements to the Whale Algorithm include adding local search mechanisms and improving its convergence to adapt to more complex optimization problems [9]. These improvements demonstrate the feasibility and necessity of further optimizing the Whale Algorithm for applications such as path planning. Kun Wu and others [10] have used greedy optimization strategies to improve the Whale Algorithm, expanding its search range to enhance global optimization effects. However, this method is still limited by its convergence speed and optimization precision. Mehmet Enes Avcu [11] and others have applied the Whale Algorithm to drone path planning to avoid collisions in cluttered areas. Although effective, the optimization performance is still suboptimal and requires further improvement.

This paper proposes an improved Whale Algorithm that integrates Cuckoo Search [12] and Random Differential [13] strategies, ensuring the global search capability of the Whale Optimization Algorithm and its optimization precision. And the improved Whale Optimization Algorithm has been applied to drone 3D path planning for the first time.

2. Introduction to the Whale Optimization Algorithm

The Whale Optimization Algorithm (WOA) simulates the foraging behavior of whales, primarily involving two strategies: bubble-net feeding and random search.

2.1. Bubble-net encircling mechanism

The foraging behavior of humpback whales involves encircling prey and moving in a spiral to achieve local optimization. In the Whale Optimization Algorithm (WOA), each whale position in the search space represents a potential solution, where the individual closest to the optimal value of the objective function is considered the best. Other whales will update their positions by moving closer to this global best position, simulating the real-life behavior of whales encircling their prey can be described as shown in Equation (1):

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

In the formula: $D = |C \cdot X^*(t) - X(t)|$; t represents the current iteration count. X^* represents the global best position vector. X represents the current whale position vector.

A and C are parameter matrices, $A = 2a \cdot r_1 - a$, $C = 2 \cdot r_2$, r_1 and r_2 are random numbers between 0 and 1. $a = 2 - 2t / t_{max}$, t_{max} represents the maximum number of iterations. During the spiral update of positions, the distance between the current position and the best whale individual is first calculated, and then movement towards the best individual is performed in a spiral motion. The mathematical model for this phase can be expressed as shown in Equation (2):

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

In the equation: $D'_p = |X^*(t) - X(t)|$, l is a random number between -1 and 1, b is a constant which determines the shape of the helix.

When whales are within their enclosure, they approach their prey by feeding with bubble nets. In this case, when the amplitude of the spiral update is less than 1, the whale will perform a local optimal search while surrounding prey and performing spiral updates with a 50% probability. The mathematical model for this process is shown in Equation (3):

$$X(t+1) = \begin{cases} X^*(t) - A \cdot D & p < 0.5 \\ X^*(t) + D'_p \cdot e^{bl} \cdot \cos(2\pi l) & \geq 0.5 \end{cases} \quad (3)$$

2.2. Search and prey mechanism

In addition to the bubble-net method, whales also employ a random search strategy to locate prey. When the coefficient A exceeds the range of $[-1,1]$, the position of the whale is randomly updated based on the distance data D . This deviation from the original target enhances the algorithm's global search capability, aiding in the search for solutions across a broader search space. As shown in [Equations \(4\)](#) and [\(5\)](#):

$$D = |C \cdot X_{\text{rand}} - X(j)| \quad (4)$$

$$X(j+1) = X_{\text{rand}} - A \cdot D \quad (5)$$

X_{rand} Represents the position of a randomly selected individual whale from the current population.

3. Introduction to improvement strategy

3.1. Introduction to the random differential algorithm

The differential algorithm is an evolutionary optimization technique that mimics biological evolutionary mechanisms, specifically designed for optimizing continuous variables [\[14\]](#). It continuously generates new candidate solutions through mutation and crossover operations, and evaluates the fitness values of these new solutions, retaining the optimal solution until the best result is achieved. In the random differential strategy, random elements are incorporated into the original differential algorithm. The specific formula is shown as [Equation \(6\)](#).

$$Y(t+1) = m_1(Y_{\text{best}}(t) - Y(t)) - m_2(Y^*(t) - Y(t)) \quad (6)$$

In the formula, m_1 and m_2 are random number between 0 and 1, $Y^*(t)$ represents a randomly selected individual.

By applying the random differential strategy to update positions again after an individual has updated its location, and retaining the best-performing position, this method can significantly prevent the population from being confined to local optima. Thus, the global search capability and optimization performance of the algorithm are enhanced.

3.2. Introduction to the Cuckoo Search

The Cuckoo Search strategy is inspired by the breeding behavior of cuckoos, utilizing strategies generated through Levy flights and random walks [\[15\]](#). The execution of Levy flights is modeled by the following formula(7):

$$X(t+1) = X(t) + \alpha \oplus \text{Levy}(\eta) . \quad (7)$$

In the formula, $X(t)$ is the t -th generation of the solution, $X(t+1)$ is the $(t+1)$ th generation of the new solution. α is the step size coefficient, $\text{Levy}(\eta)$ is a random number drawn from the Levy distribution. The formula for $\text{Levy}(\eta)$ is shown in [Equation \(8\)](#).

$$\text{Levy}(\eta) \sim \frac{\phi \times u}{|v|^{\frac{1}{\eta}}} \tag{8}$$

In the formula, μ and v are random numbers uniformly distributed according to the normal distribution. η is the distribution parameter for Levy flights, typically set at

$$1.5. \phi = \left[\left(\Gamma(1 + \eta) \sin\left(\frac{\pi\eta}{2}\right) \right) / \left(\Gamma\left(\frac{1 + \eta}{2}\right) \eta \times 2^{\frac{\eta-1}{2}} \right) \right]^{1/\eta}, \Gamma \text{ is the Gamma function.}$$

Since the algorithm generates new solutions around the existing best solution, Formula (9) can be rewritten as:

$$X(t + 1) = X(t) + \alpha_0 \times \left[\frac{\phi \times u}{|v|^{\frac{1}{\eta}}} \right] \times (X(t) - X_{best}) \tag{9}$$

α_0 is a scaling factor set to 0.01 in Formula (10), and X_{best} is the current optimal solution. Random walk is shown in Equation (10):

$$X(t + 1) = \begin{cases} X(t) + r(X(m) - X(n)) & r > P_a \\ X(t) & otherwise \end{cases} \tag{10}$$

r is the probability of performing a Levy flight, and P_a is a constant between 0 and 1. m and n correspond to different individuals.

3.3. Integration of Cuckoo Search and random differential strategy

Although the Cuckoo Search algorithm effectively enhances global search and prevents premature convergence into local optima through Levy flights, it is less efficient in adjusting local solutions [16]. In contrast, the Random Differential Algorithm maintains population diversity and effectively finds local optima through differential, crossover, and selection operations [17]. Therefore, combining these two algorithms can enhance global search capabilities, increase search randomness, and improve the quality of solutions and the convergence speed of the algorithm. This paper modifies Formula (10) to Formula (11) and names the improved Whale Algorithm as: Cuckoo Search-Random Differential Strategy [18] based Whale Optimization Algorithm (CSR-D-WOA).

$$X(t + 1) = \begin{cases} X(t) + \alpha_0 \times \frac{\phi \times u}{|v|^{\frac{1}{\eta}}} \times (X(t) - X_{best}) & r > P_a \\ r_1(X_{best}(t) - X(t)) - r_2(X^*(t) - X(t)) & otherwise \end{cases} \tag{11}$$

3.4. Whale Optimization Algorithm with multi strategy fusion

In the CSR-D-WOA, parameter selection significantly impacts algorithm performance. Key parameters have been designed as shown in Table 1, optimized based on extensive testing and sensitivity analysis to improve search efficiency and solution quality.

Table 1. Parameter settings.

Parameter	NP	dim	t_{max}	b	η	α	Pa
Size	500	30	30	1	1.5	0.01	0.8

<https://doi.org/10.1371/journal.pone.0316836.t001>

4. Algorithm performance testing and analysis

4.1. Test function setup

To fully validate the performance of CSRD-WOA, three types of test functions were selected: unimodal functions, multimodal functions, and fixed multimodal functions. Detailed descriptions are provided in [Tables 2–4](#).

4.2. Comparative experiment

The CSRD-WOA algorithm is applied to the benchmark test functions described in Section [4.1](#), resulting in the statistical outcomes presented in [Table 5](#).

Table 2. Unimodal test functions.

Number	Expression	Function optimal solution
F_1	$f_1(x) = \sum_{i=1}^n (x_i - z_i)^2$	0
F_2	$f_2(x) = \max\{ x_i , 1 \leq i \leq 30\}$	0
F_3	$f_3(x) = \sum_{i=1}^{30} ix_i^4 + \text{random}[0,1)$	0

<https://doi.org/10.1371/journal.pone.0316836.t002>

Table 3. Multimodal test functions.

Number	Expression	Function optimal solution
F_1	$f_4(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	0
F_1	$f_5 = -20 \exp\left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i^2}\right) - \exp\left(\frac{1}{30} \sum_{i=1}^{30} \cos 2\pi x_i\right) + 20 + c$	0
F_1	$f_6(x) = \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0

<https://doi.org/10.1371/journal.pone.0316836.t003>

Table 4. Fixed multimodal test functions.

Number	Expression	Function optimal solution
F_1	$f_7(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	0.998003837794450
F_1	$f_8(x) = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	0.397887357729738
F_1	$f_9(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2.9999999999992

<https://doi.org/10.1371/journal.pone.0316836.t004>

Table 5. Benchmark test function result.

Function	Evaluation criteria	PSO	CS	WOA	CSRWOA
F1	Best	372.3384	21.7327	0	0
	Mean	1906.5221	1221.8361	821.3629	212.7033
	Std	2490.9898	1786.2417	3501.8532	1549.3213
F2	Best	6.1713	12.4933	0.0693	0
	Mean	14.2241	22.76	12.9049	0.0009
	Std	6.456	9.6169	21.0429	0.0093
F3	Best	0.3997	0.0602	0.0033	0.0006
	Mean	2.8156	0.1771	0.5106	0.0045
	Std	9.9699	0.2356	2.5482	0.0284
F4	Best	221.4351	195.8508	119.215	3.4013
	Mean	28.9113	36.5113	57.1616	22.8166
	Std	14.5254	34.0211	61.1915	92.8571
F5	Best	5.5849	19.9387	5.06E-14	8.88E-16
	Mean	8.8591	19.9392	1.3684	0.0518
	Std	2.3626	0.0014	3.1324	0.4341
F6	Best	2.4873	1.0751	0.0222	0
	Mean	24.6904	8.8593	6.2936	0.3364
	Std	27.332	12.0609	28.8159	6.6783
F7	Best	2.9821	0.998	0.998	0.998
	Mean	3.0101	0.998	1.0718	0.998
	Std	0.439	0	0.2605	0
F8	Best	0.3979	0.3979	0.3979	0.3979
	Mean	0.3993	0.3979	0.3981	0.3979
	Std	0.0163	0	0.0014	0
F9	Best	3	3	3	3
	Mean	3.0495	3	3.002	3.0001
	Std	0.6172	0	0.0212	0.0016

<https://doi.org/10.1371/journal.pone.0316836.t005>

Additionally, [Figs 1–9](#) are displays images of several benchmark test functions. Among them, F1 and F2 are unimodal test functions, F4 and F5 are multimodal test functions, and F7 and F8 are fixed multimodal test functions. These three categories of test functions are used to evaluate function convergence, optimization capability, and the ability to handle complex problems, respectively. Based on the content of [Figs 1–9](#) and [Table 5](#), it is evident that the improved algorithm surpasses the original WOA in terms of convergence, optimization capability, and handling complex problems, indicating that the improvement strategies employed in this study are effective. Although the improved algorithm slightly underperforms the original WOA and PSO algorithms in some test results, the effectiveness of the adopted improvement strategies remains evident.

4.3. Ablation experiment

To demonstrate that the improvement strategies have enhanced the original WOA, this section employs ablation experiments [\[19\]](#) to verify the effectiveness of these strategies. The larger the obtained value, the higher the degree of improvement. The calculation formulas for the ablation experiment are shown in [Equation \(12\)](#).

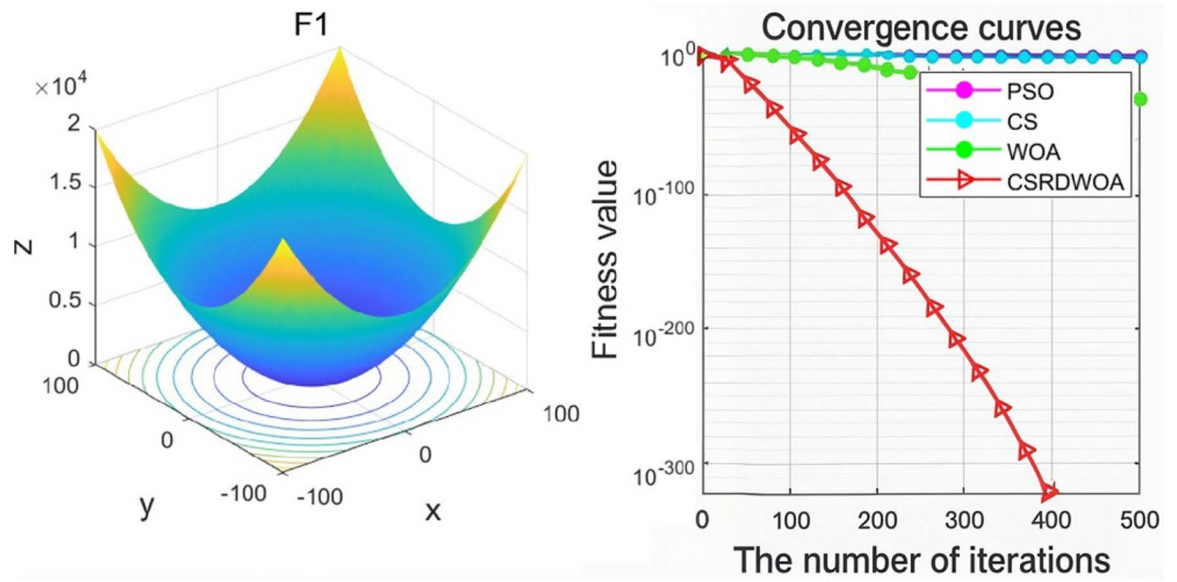


Fig 1. F_1 .

<https://doi.org/10.1371/journal.pone.0316836.g001>

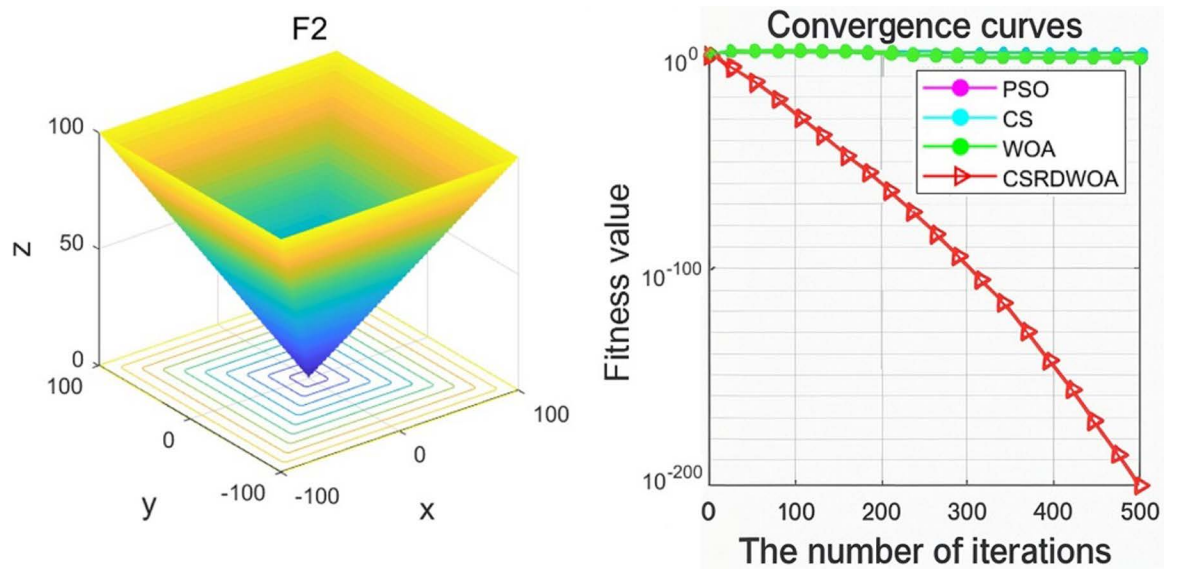


Fig 2. F_2 .

<https://doi.org/10.1371/journal.pone.0316836.g002>

$$\gamma_1 = \frac{f_{CS-WOA}}{f_{WOA}} \tag{12}$$

$$\gamma_2 = \frac{f_{CS-WOA}^{best}}{f_{WOA}^{best}} \tag{13}$$

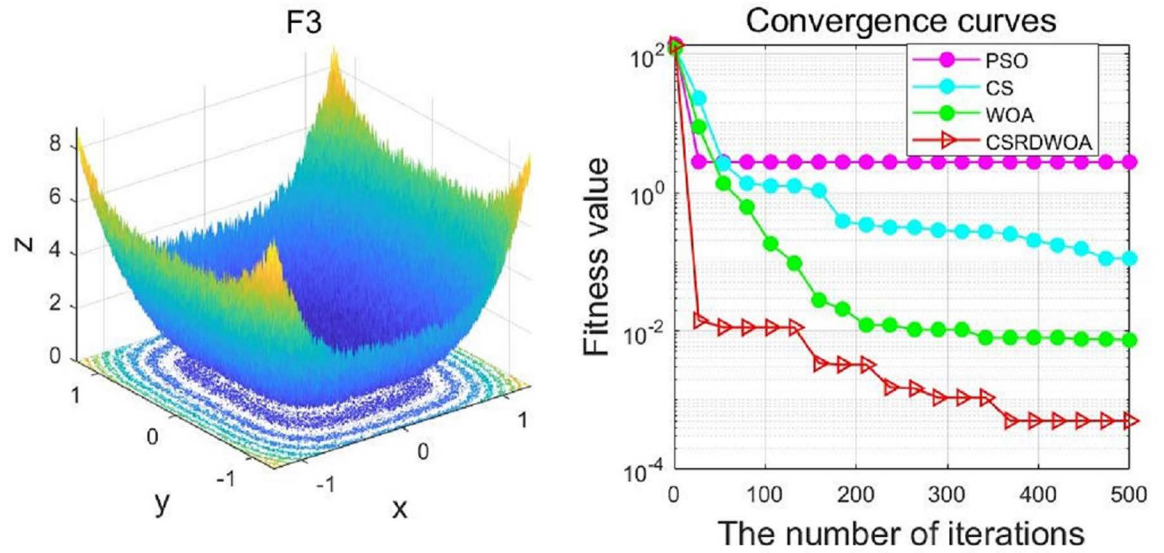


Fig 3. F_3 .

<https://doi.org/10.1371/journal.pone.0316836.g003>

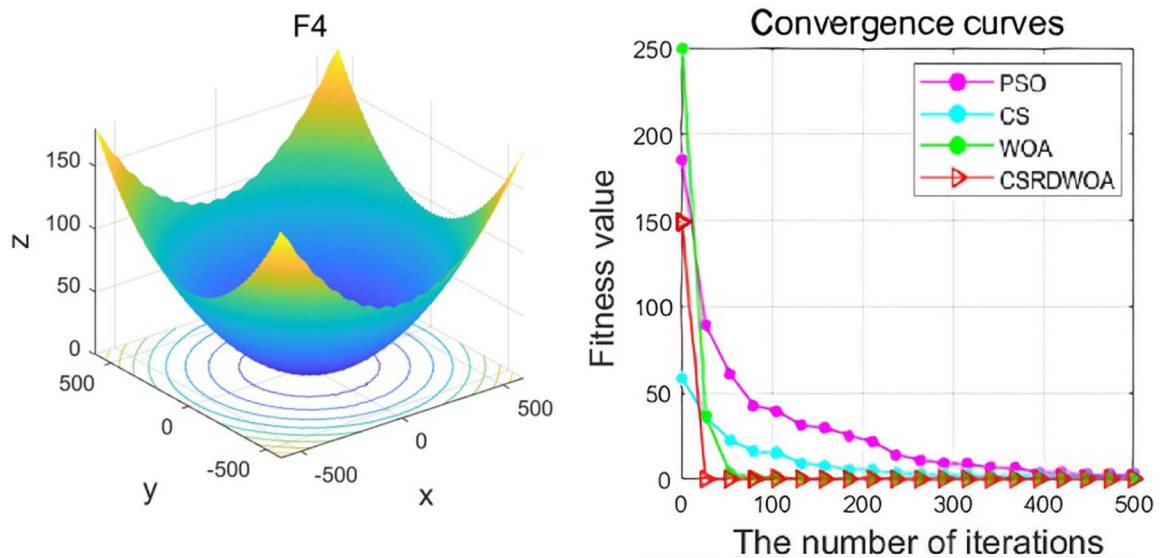


Fig 4. F_4 .

<https://doi.org/10.1371/journal.pone.0316836.g004>

The larger the obtained value, the better the improvement of the algorithm. In Formula (13), f_{CS-WOA}^{best} and f_{WOA}^{best} respectively represent the average best fitness values of the CSRDWAO algorithm and the WOA algorithm. f_{CS-WOA}^{best} and f_{WOA}^{best} are the best fitness values of the CSRDWAO and WOA algorithms on the test functions, respectively. Smaller values of γ_2 indicate poorer optimization performance of the control algorithm, highlighting that the missing strategy is crucial for enhancing the performance of the original algorithm. The results of the ablation experiment are shown in Table 6.

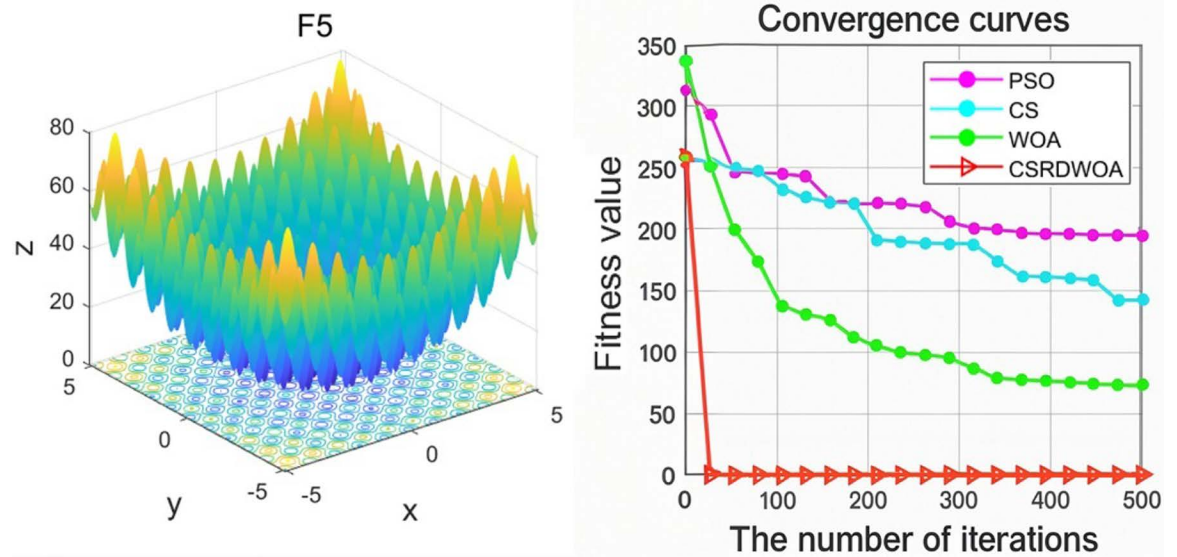


Fig 5. F_5 .
<https://doi.org/10.1371/journal.pone.0316836.g005>

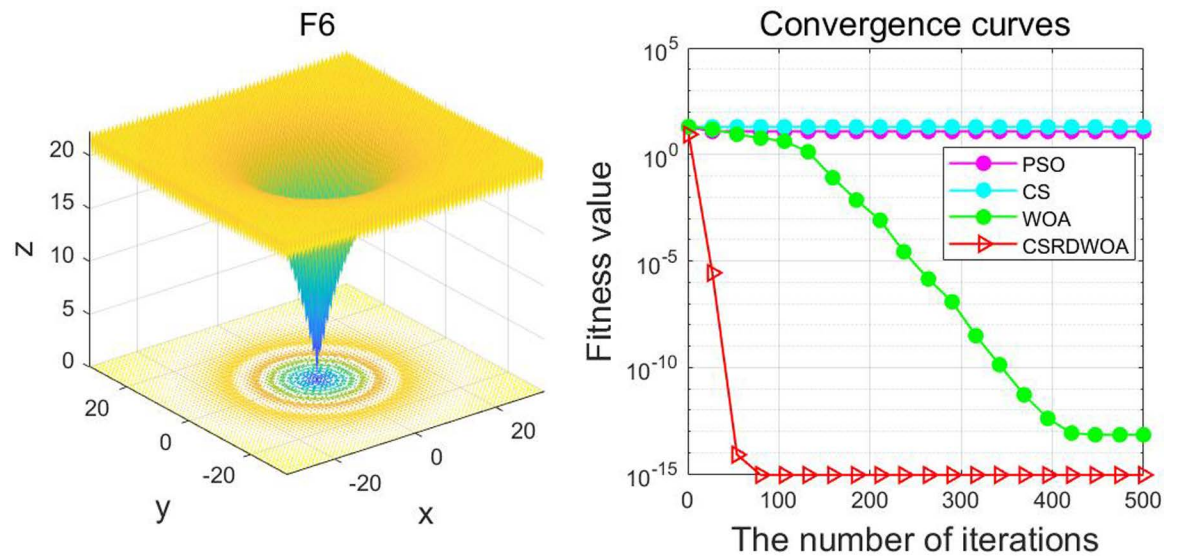


Fig 6. F_6 .
<https://doi.org/10.1371/journal.pone.0316836.g006>

The results of the ablation experiments reveal that the improvement strategies adopted for CSR-D-WOA enhance its performance compared to the original WOA. Although the improvements are not significant for some functions, the strategies can still be considered effective. Therefore, based on the results of the benchmark test functions and the ablation experiments, it can be concluded that the improvement strategies employed in this study have, to a certain extent, improved the performance of WOA.

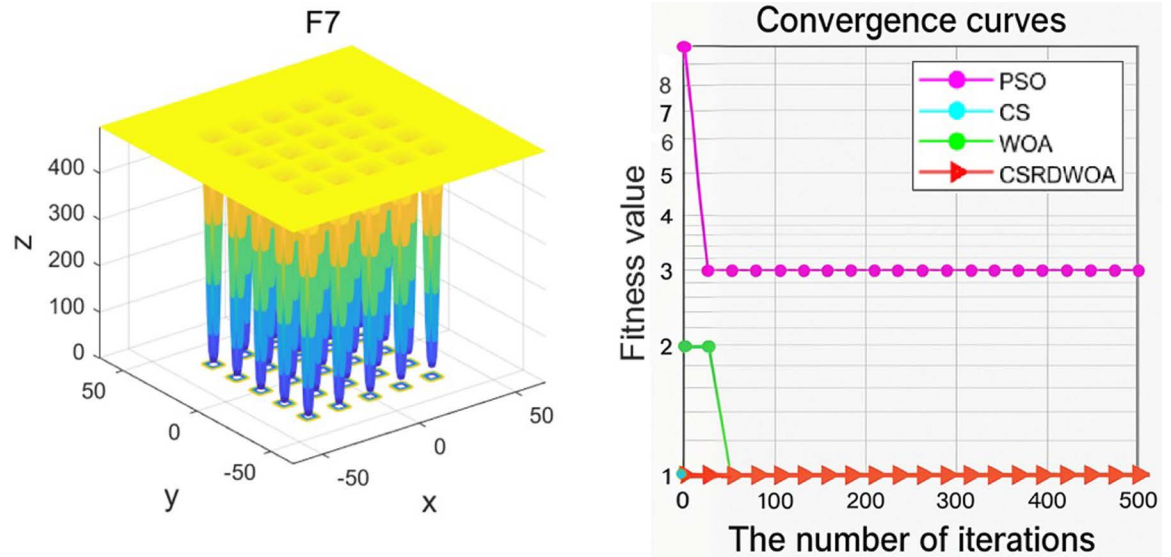


Fig 7. F_7 .

<https://doi.org/10.1371/journal.pone.0316836.g007>

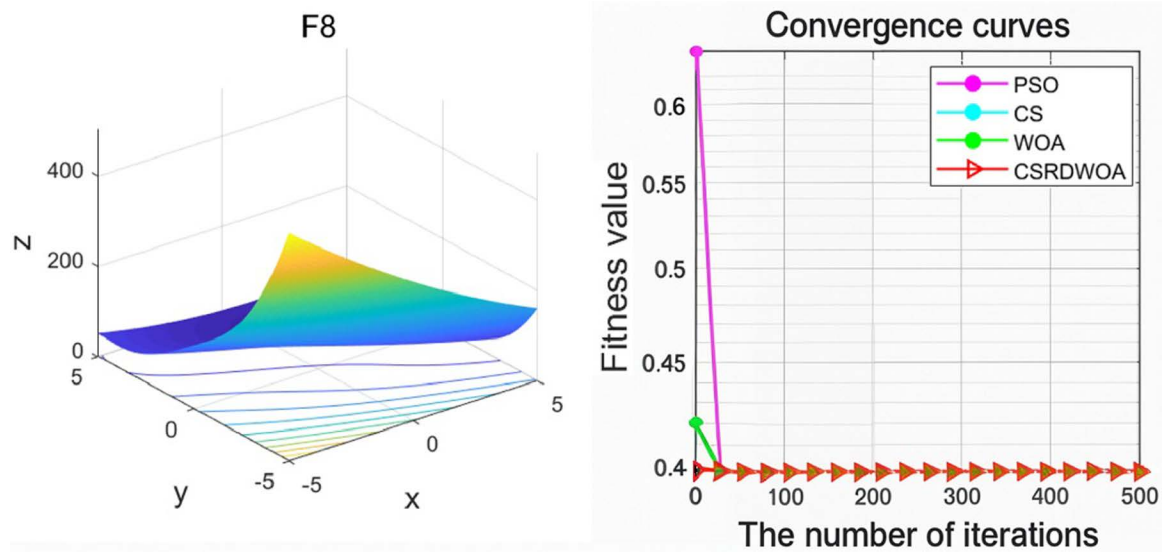


Fig 8. F_8 .

<https://doi.org/10.1371/journal.pone.0316836.g008>

5. Drone 3D path planning based on the improved Whale Optimization Algorithm

To verify the applicability of CSRD-WOA, the algorithm is applied to three-dimensional path planning for drones in complex environments [20]. The drone’s path planning needs to consider path cost, threat cost, flight altitude, and smoothness cost [21]. The total cost of drone flight is given by Equation (14).

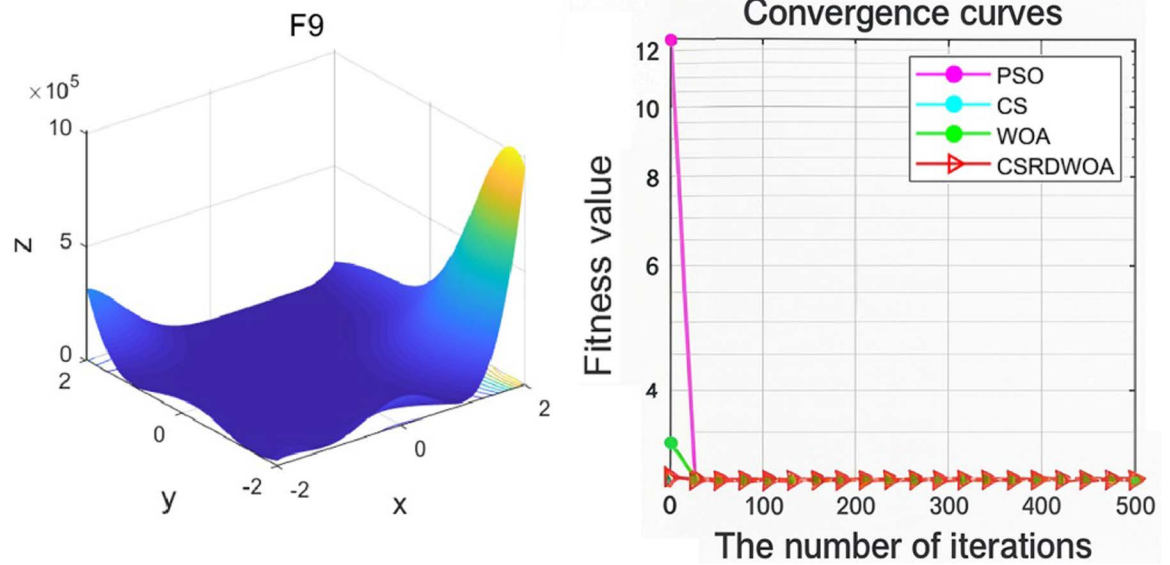


Fig 9. F_9 .

<https://doi.org/10.1371/journal.pone.0316836.g009>

Table 6. Ablation experiment results.

Test function	γ_2
F1	0
F2	7.9445e-196
F3	0.1666
F4	0
F5	0.0175
F6	0
F7	1
F8	1
F9	1

<https://doi.org/10.1371/journal.pone.0316836.t006>

$$F(X_i) = \sum_{k=1}^4 b_n F_n(X_i) \tag{14}$$

Among them, X_i are the decision variables, b_n are the function weight parameters, and F_n is the nth cost. The cost function is composed of three parts: the threat cost, the altitude cost, and the smoothness cost. Below, we will introduce these three cost functions.

5.1. Cost function

Threat environment cost. Drone path planning takes safety into account, simplifying potential threats as fixed-radius cylindrical areas to optimize the route and ensure flight safety. As shown in Fig 10.

Let M be the set of cylindrical collections representing all potential threat obstacles, with the center of the base circle denoted by C and radius R . The diameter of the drone is D , and the vertical distance between adjacent path nodes is dim . S represents the range of the

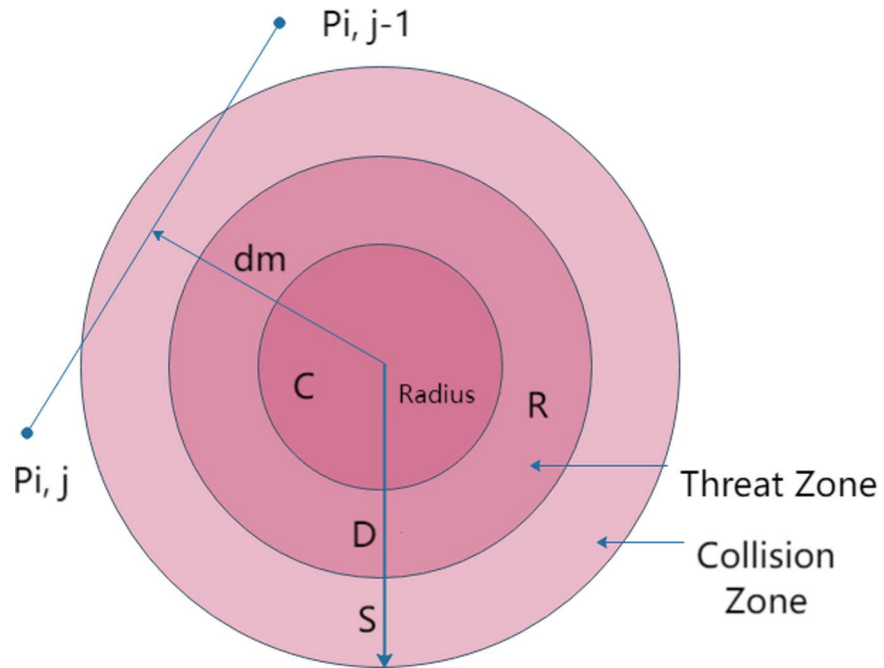


Fig 10. Threat area diagram.

<https://doi.org/10.1371/journal.pone.0316836.g010>

obstacle risk area. In order to quantify the safety of the drone’s path, the threat cost is calculated to evaluate the path risk, using Equation (15) to compute the threat cost:

$$\left\{ \begin{array}{l} F_1(J_i) = \sum_{j=1}^{n-1} \sum_{m=1}^M T_m(P_{ij}P_{ij+1}) \\ T_m(P_{i,j}P_{i,j+1}) = \begin{cases} 0, & d_m > S + D + R_m \\ (S + D + R_m) - d_m, & D + R_m < d_m \leq S + D + R_m \\ \infty, & d_m \leq D + R_m \end{cases} \end{array} \right. \quad (15)$$

When the drone is far from the danger zone, the threat level is zero; however, as it approaches the danger zone and the vertical distance (dm value) decreases, the threat correspondingly increases.

Flight altitude cost. When a drone performs measurement and search tasks, its flight altitude is restricted by a minimum height h_{min} and a maximum height h_{max} . A reasonable flight altitude ensures the safety of the drone’s flight and the precision of the collected images, optimizing task execution. The altitude cost is as shown in Equations (16) and (17).

$$H_{ij} \left\{ \begin{array}{l} \left| h_{ij} - \frac{(h_{max} + h_{min})}{2} \right|, h_{min} \leq h_{ij} \leq h_{max} \\ \infty, \text{ otherwise} \end{array} \right. \quad (16)$$

$$F_2(X_i) = \sum_{j=1}^n H_{ij} \tag{17}$$

The altitude cost aims to ensure that the drone remains within the set flight altitude, penalizing heights that exceed this range in order to reduce risks and improve efficiency.

Smoothness cost. The drone’s heading is controlled by horizontal turning angles and vertical pitch angles, which need to be set appropriately to comply with flight restrictions and ensure path safety and efficiency in response to sudden and environmental challenges. The horizontal turning angle is the angle between continuous path segments on the horizontal plane. This angle is calculated using Equation (18):

$$\phi_{ij} = \arctan \left(\frac{\| \overrightarrow{P'_{ij}P'_{i,j+1}} \times \overrightarrow{P'_{i,j+1}P'_{i,j+2}} \|}{\| \overrightarrow{P'_{ij}P'_{i,j+1}} \| \| \overrightarrow{P'_{i,j+1}P'_{i,j+2}} \|} \right). \tag{18}$$

The vertical pitch angle refers to the angle difference between two consecutive path segments in the vertical direction, as shown in Equation (19):

$$\psi_{ij} = \arctan \left(\frac{z_{i,j+1} - z_{ij}}{\| \overrightarrow{P'_{ij}P'_{i,j+1}} \|} \right) \tag{19}$$

The calculation of the smoothness cost is as shown in Equation (20):

$$F_3(X_i) = a_1 \sum_{j=1}^{n-2} \phi_{ij} + a_2 \sum_{j=1}^{n-1} |\psi_{ij} - \psi_{i,j-1}| \tag{20}$$

In the equation, a_1 and a_2 are the penalty coefficients for the horizontal turning angle and the vertical pitch angle, respectively.

Flight distance cost. The calculation of the flight distance cost involves measuring the displacement between adjacent data points along the drone’s flight path. By calculating the magnitude of the coordinate differences between these points, the length of each segment can be obtained. Summing these lengths yields the total flight distance cost. Assuming the three-dimensional coordinate data of the drone’s flight trajectory is $\{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_N, y_N, z_N)\}$, where N is the number of data points. The calculation formula for the flight path cost is shown in Equation (21).

$$F_4 = \sum_{i=1}^{N-1} \|diff_i\| . \tag{21}$$

Based on the aforementioned four types of flight costs, the total cost of the drone flight can be determined by Equation (22).

$$F_{total} = \omega_1 \times F_1 + \omega_2 \times F_2 + \omega_3 \times F_3 + \omega_4 \times F_4 \tag{22}$$

In the equation, $\omega_1, \omega_2, \omega_3, \omega_4$ respectively represent the weights for the threat environment cost, altitude cost, smoothness cost, and flight distance cost, with values set at: 10, 100, 10, 50. The flight cost parameters are as shown in Table 7. This weight coefficient value setting helps to provide a more intuitive observation of drone flight costs.

5.2. Simulation testing and analysis

The simulation environment is configured on a Windows 10 system, using MATLAB 2019b to create two types of terrain: mountainous terrain with multiple structures and relatively flat mountainous terrain. Various threats are added to both types of terrain to simulate environments of different complexities for drone path planning experiments.

Case study one. Terrain environment parameters: The terrain environment for Case Study One is shown in Fig 11. The deep blue cylinders represent the hazardous areas in the terrain. The test environment is set to an area of 400m by 400m, with a maximum flight altitude of 350m.

In the terrain scenario, the drone starts from the coordinates (10, 10, 200) m and flies to the endpoint at (400, 400, 150) m.

Table 7. Parameter settings.

Parameter	S	D	a_1	a_2	ω_1	ω_2	ω_3	ω_4
Size	20	10	1	1	10	100	10	50

<https://doi.org/10.1371/journal.pone.0316836.t007>

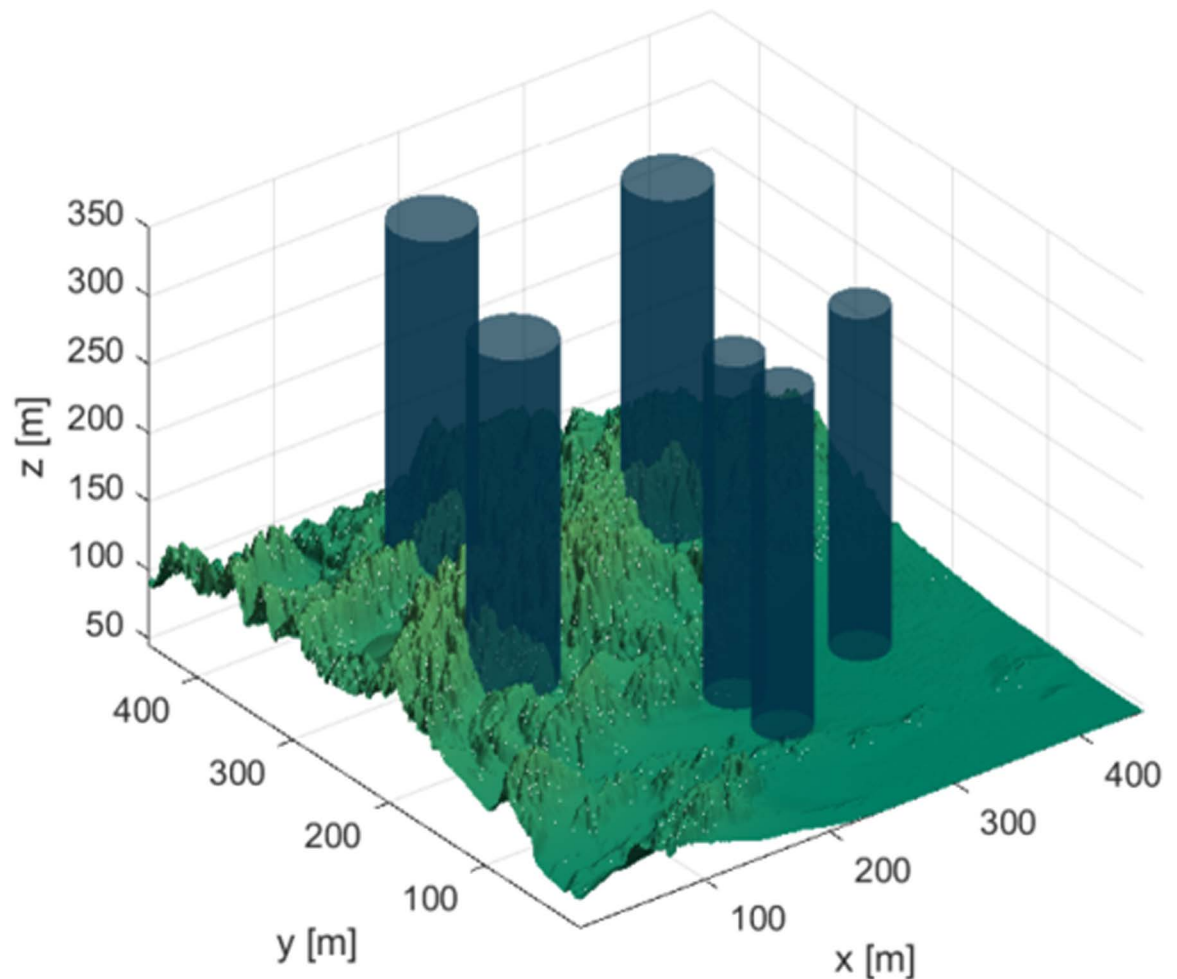


Fig 11. Terrain map for case study one.

<https://doi.org/10.1371/journal.pone.0316836.g011>

The coordinates of the hazardous areas in the terrain of Case Study One are shown in [Table 8](#).

Simulation experiment: Comparative experiments are conducted using the CSRD-WOA algorithm proposed in this paper with the PSO algorithm, WOA algorithm, and CS algorithm. The resulting optimal fitness iteration curve for the flight is shown in [Fig 12](#).

From the fitness curve diagram in [Fig 12](#), it can be seen that the CSRD-WOA algorithm found the optimal solution. Additionally, the various costs associated with the droneflight are presented in [Table 9](#).

As seen from [Table 9](#), the CSRD-WOA algorithm achieved the optimal fitness value, and all associated costs are lower than those of the other algorithms. This indicates that the drone path planning optimized by CSRD-WOA can completely avoid threat areas while reducing flight costs. [Figs 13–15](#) are shows the side view, top view, and 3D view of the flight path, respectively. These views illustrate the drone flight paths optimized by each algorithm, revealing that the path optimized by CSRD-WOA is safer and smoother.

Table 8. Coordinates and radius of hazardous areas in case study one.

Serial number	Coordinates (unit: m)	Threat radius (unit: m)
1	(300, 300, 100) m	50 m
2	(200, 100, 100) m	40 m
3	(100, 200, 100) m	50 m
4	(300, 100, 100) m	40 m
5	(200, 50, 100) m	40 m
6	(150, 350, 100) m	50 m

<https://doi.org/10.1371/journal.pone.0316836.t008>

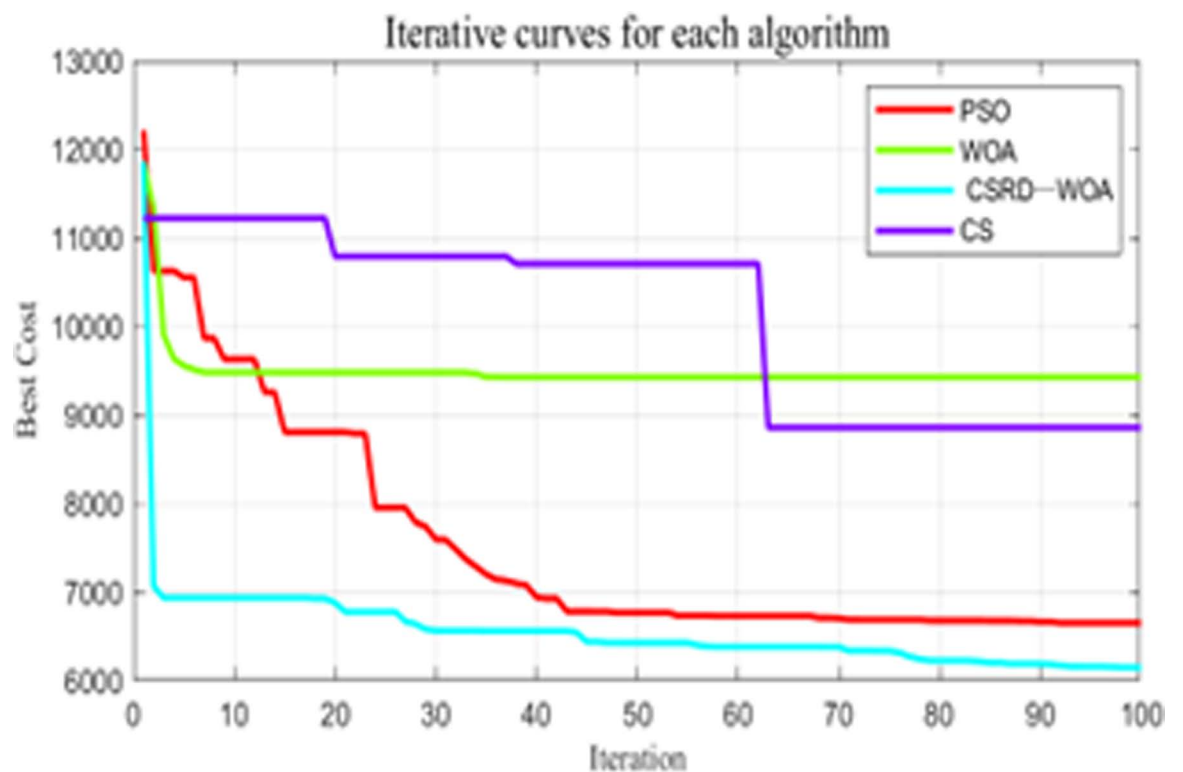


Fig 12. Flight optimization iterations.

<https://doi.org/10.1371/journal.pone.0316836.g012>

Table 9. Flight cost.

Algorithm name	Optimal fitness value	Flight distance cost	Threat cost	Altitude cost	Smoothness cost
PSO	6648.9351	671.7923	5.3762e+43	203.2753	278.7714
WOA	9422.5884	709.0645	6.9252	168.5617	0
CS	8853.7605	721.4416	5.3762e+43	454.9298	329.5859
CSR-D-WOA	6133.2575	600.1567	0	14.551	0

<https://doi.org/10.1371/journal.pone.0316836.t009>

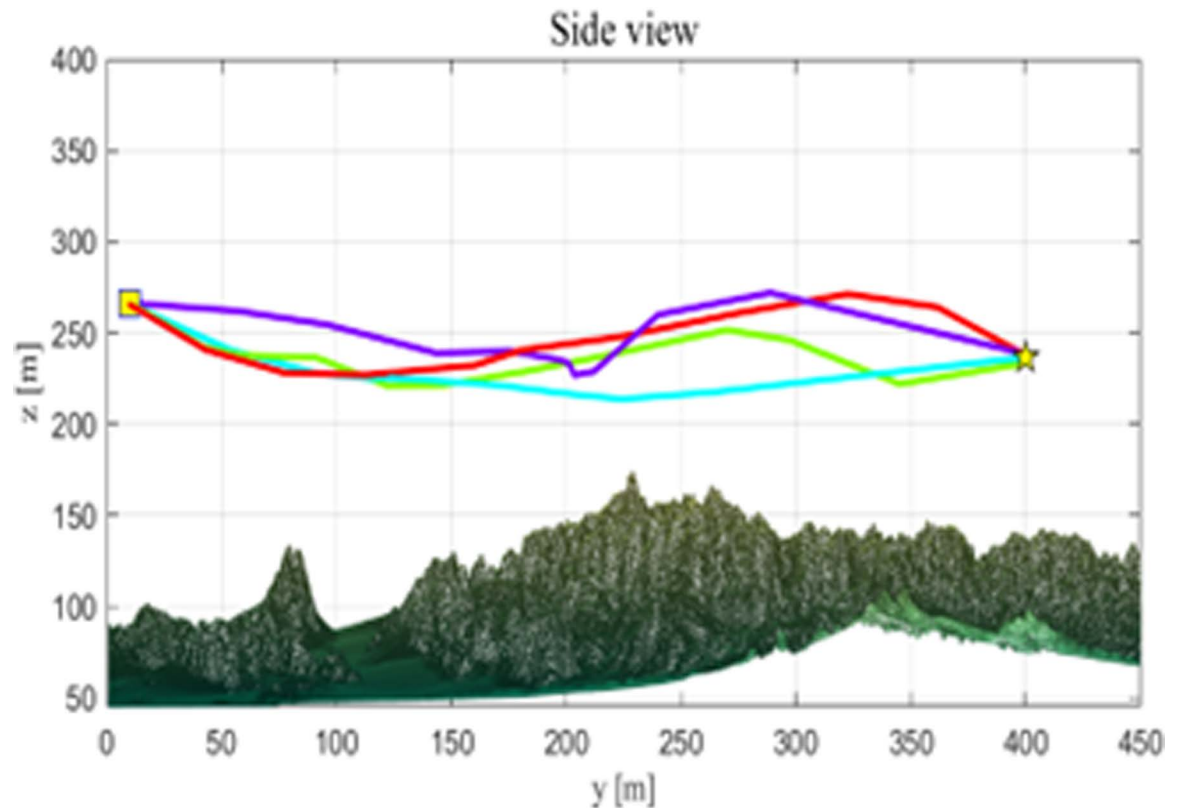


Fig 13. Side view.

<https://doi.org/10.1371/journal.pone.0316836.g013>

Case study two. To verify the accuracy and reliability of the experimental results, this study designed a series of new simulation experiments based on different ground conditions.

Terrain environment parameters: The terrain environment for Case Study Two is shown in Fig 16. The deep blue cylinders represent the hazardous areas in the terrain. The test environment is set to an area of 400m by 400m, with a maximum flight altitude of 350m. In the terrain scenario, the drone starts from the coordinates (10, 10,200) m and flies to the endpoint at (400, 400, 150) m.

The coordinates of the hazardous areas in the terrain of Case Study Two are shown in Table 10.

Simulation experiment: During the flight, the maximum turning angle of the drone is restricted to 45°. The simulation population size is 500, with a maximum of 100 iterations. The final optimal fitness for the flight, flight distance cost, threat cost, altitude cost, and smoothness cost are shown in Table 11, and the optimal fitness iteration curve for the flight is shown in Fig 17. Comparative experiments are conducted between the CSR-D-WOA algorithm and the PSO algorithm, WOA algorithm, and CS algorithm.

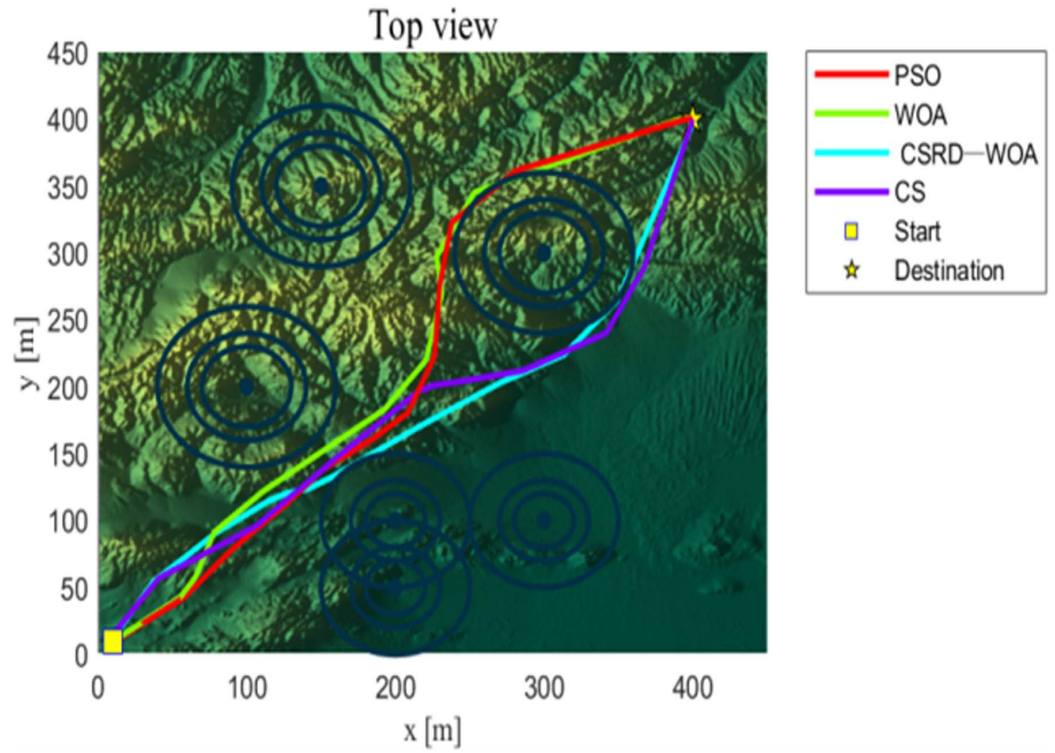


Fig 14. Top view.

<https://doi.org/10.1371/journal.pone.0316836.g014>

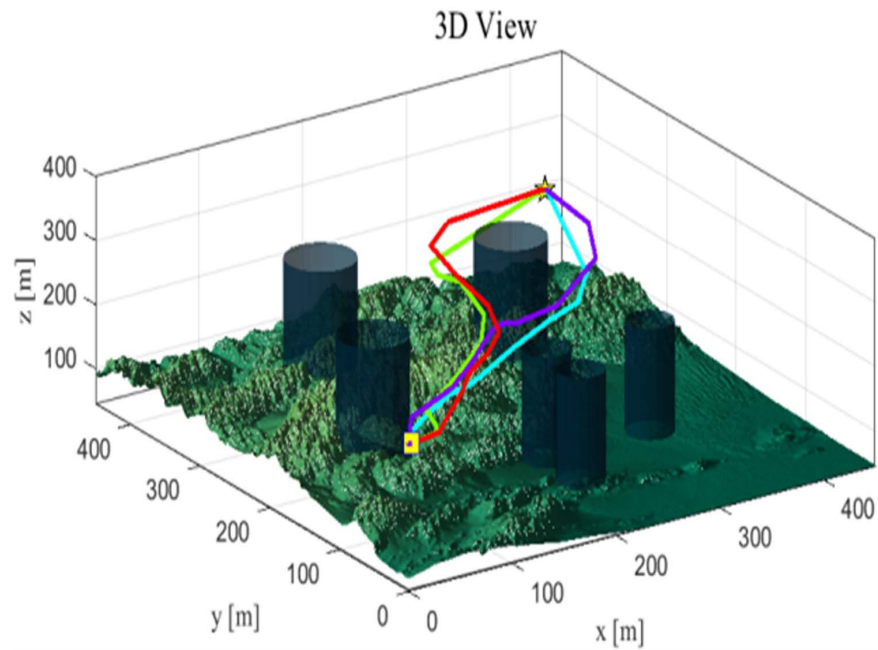


Fig 15. 3D view.

<https://doi.org/10.1371/journal.pone.0316836.g015>

Table 10. Coordinates and radius of the hazardous area in case two.

Number,	Coordinates (unit: m)	Threat radius (unit: m)
1	(85, 290, 80)	30 m
2	(120, 100, 70)	20 m
3	(220, 210, 90)	30 m
4	(230, 360, 80)	20 m
5	(300, 100, 70)	20 m
6	(350, 300, 70)	30 m

<https://doi.org/10.1371/journal.pone.0316836.t010>

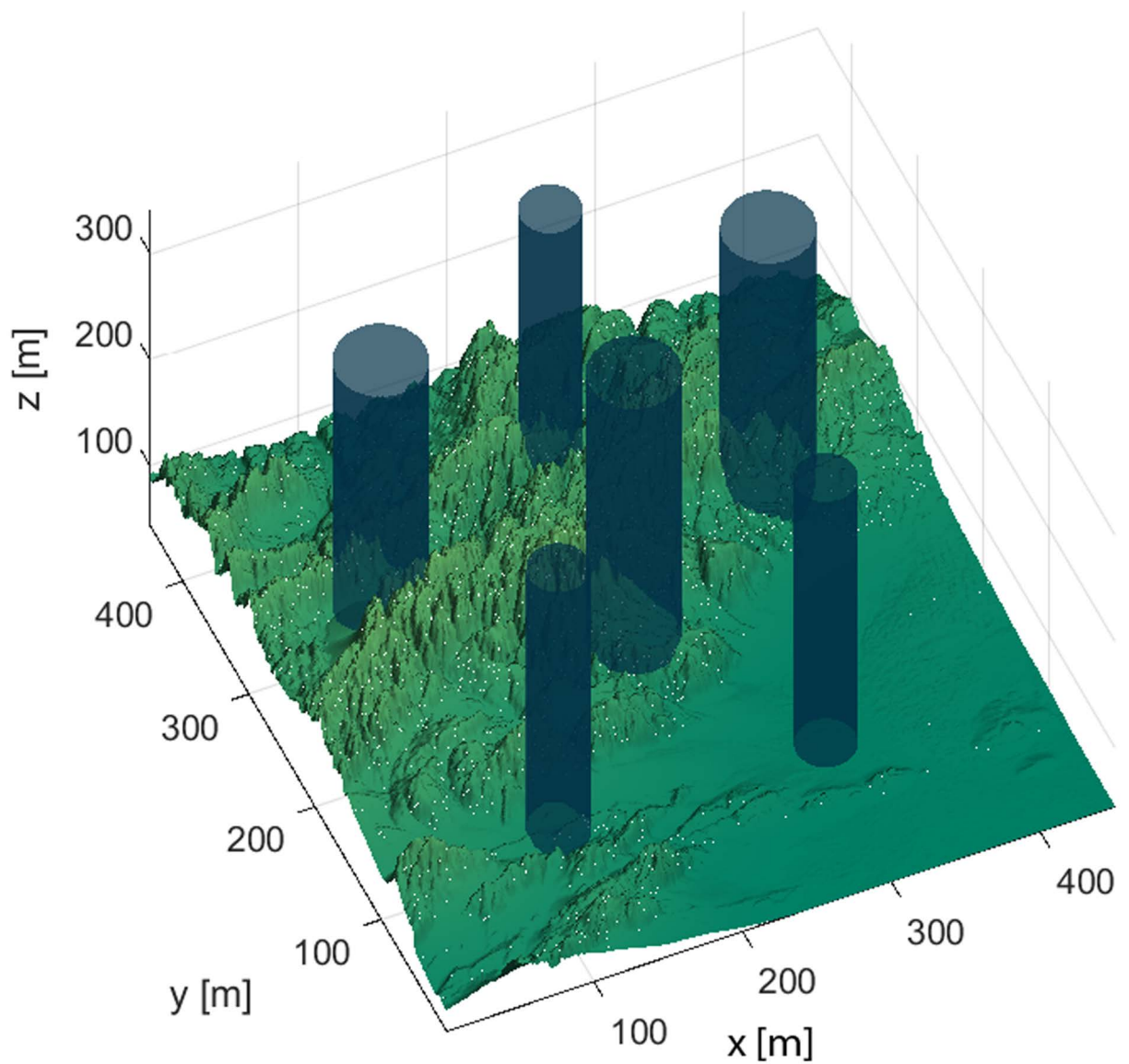


Fig 16. Terrain map for case study two.

<https://doi.org/10.1371/journal.pone.0316836.g016>

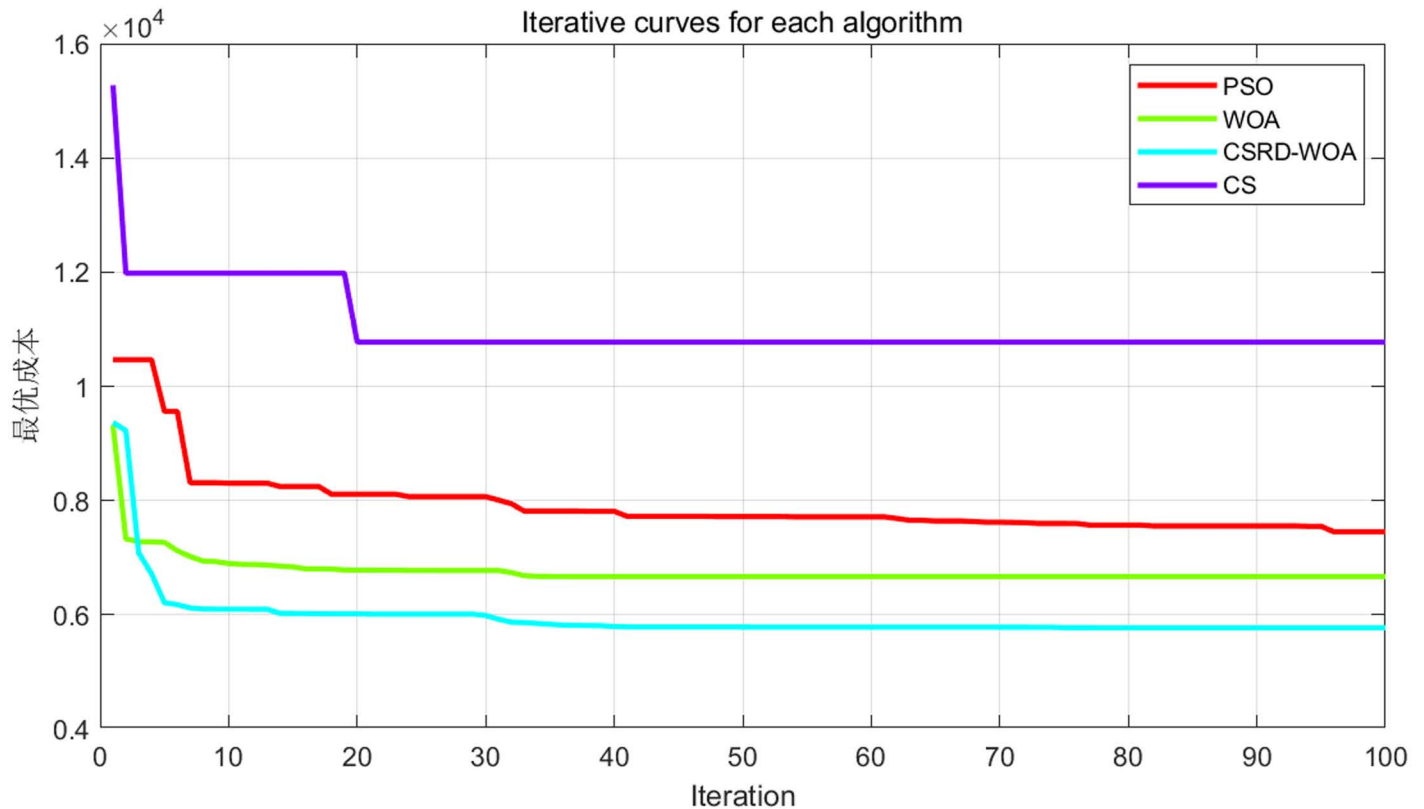


Fig 17. Flight optimization iterations.

<https://doi.org/10.1371/journal.pone.0316836.g017>

Table 11. The cost of the flight in case two.

Algorithm name	Optimal fitness value	Flight distance cost	Threat cost	Altitude cost	Smoothness cost
PSO	7452.86	707.21	71.37	239.85	214.24
WOA	6669.11	604.91	0	62.00	0
CS	10775.75	681.65	2.69E + 43	217.30	514.07
CSRD-WOA	5773.15	577.31	0	0.0035	0

<https://doi.org/10.1371/journal.pone.0316836.t011>

From the fitness curve diagram in Fig 17, it can be seen that the CSRD-WOA algorithm found the optimal solution, while the CS algorithm, WOA algorithm, and PSO algorithm all fell into local optima. This indicates that the precision of the CS, WOA, and PSO algorithms in finding the optimal solution is far inferior to that of CSRD-WOA. As seen from Table 11, the CSRD-WOA algorithm outperforms the other three algorithms in total cost, flight distance cost, threat cost, altitude cost, and smoothness cost. This suggests that the CSRD-WOA algorithm is more suitable for finding the optimal path for drones. Furthermore, the flight path of CS passes through the threat area, which is very disadvantageous for the safety of drone flight. Figs 18–20 are shows the side view, top view, and 3D view of the flight path, respectively.

From the 3D visual diagram, top view, and side view of the flight paths shown in Figs 18–20, it can be observed that the drone’s trajectory based on the CSRD-WOA algorithm is straighter, shorter, smoother, and successfully avoids all hazardous areas.

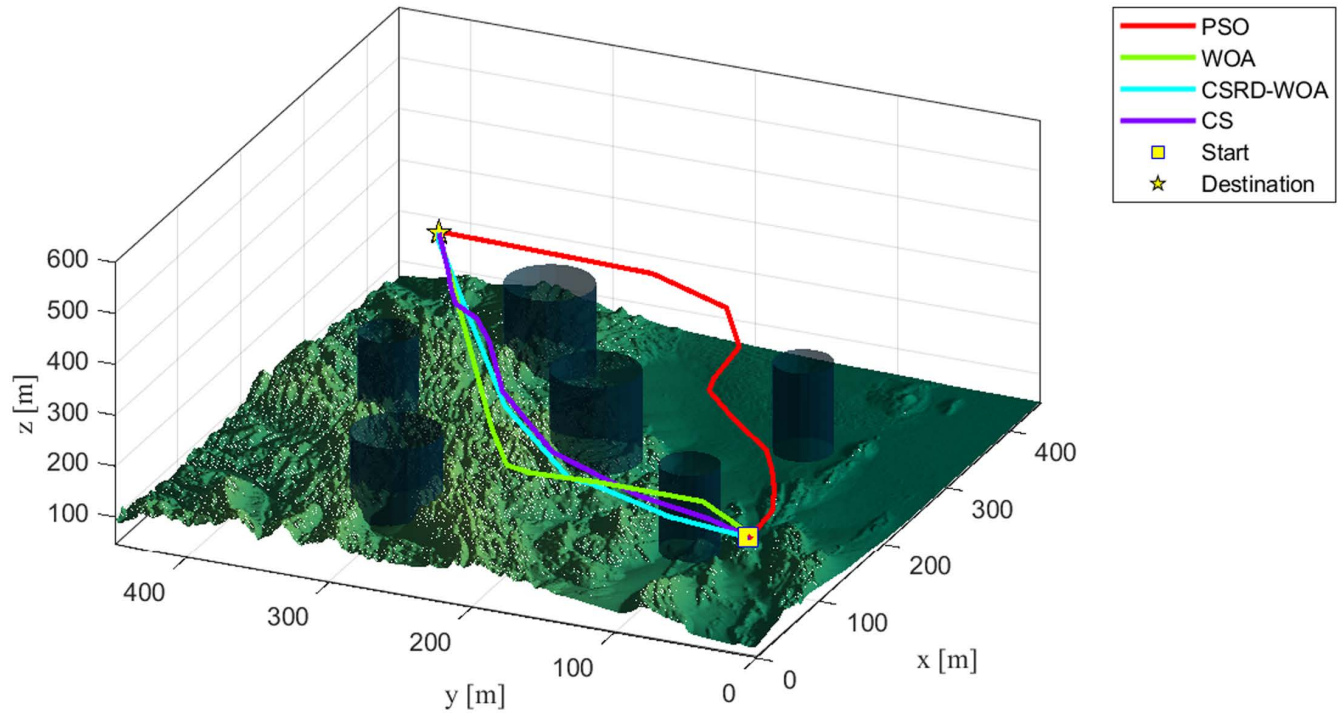


Fig 18. 3D view.

<https://doi.org/10.1371/journal.pone.0316836.g018>

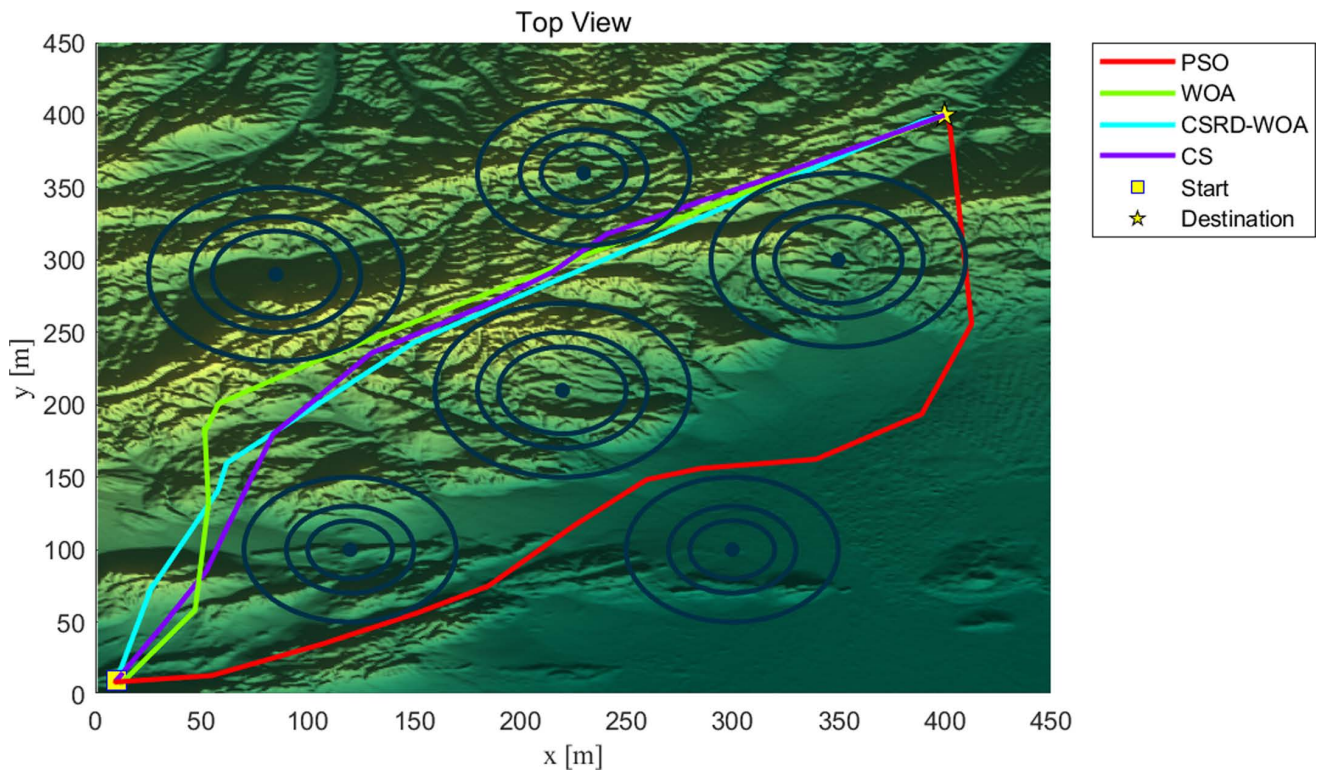


Fig 19. Top view.

<https://doi.org/10.1371/journal.pone.0316836.g019>

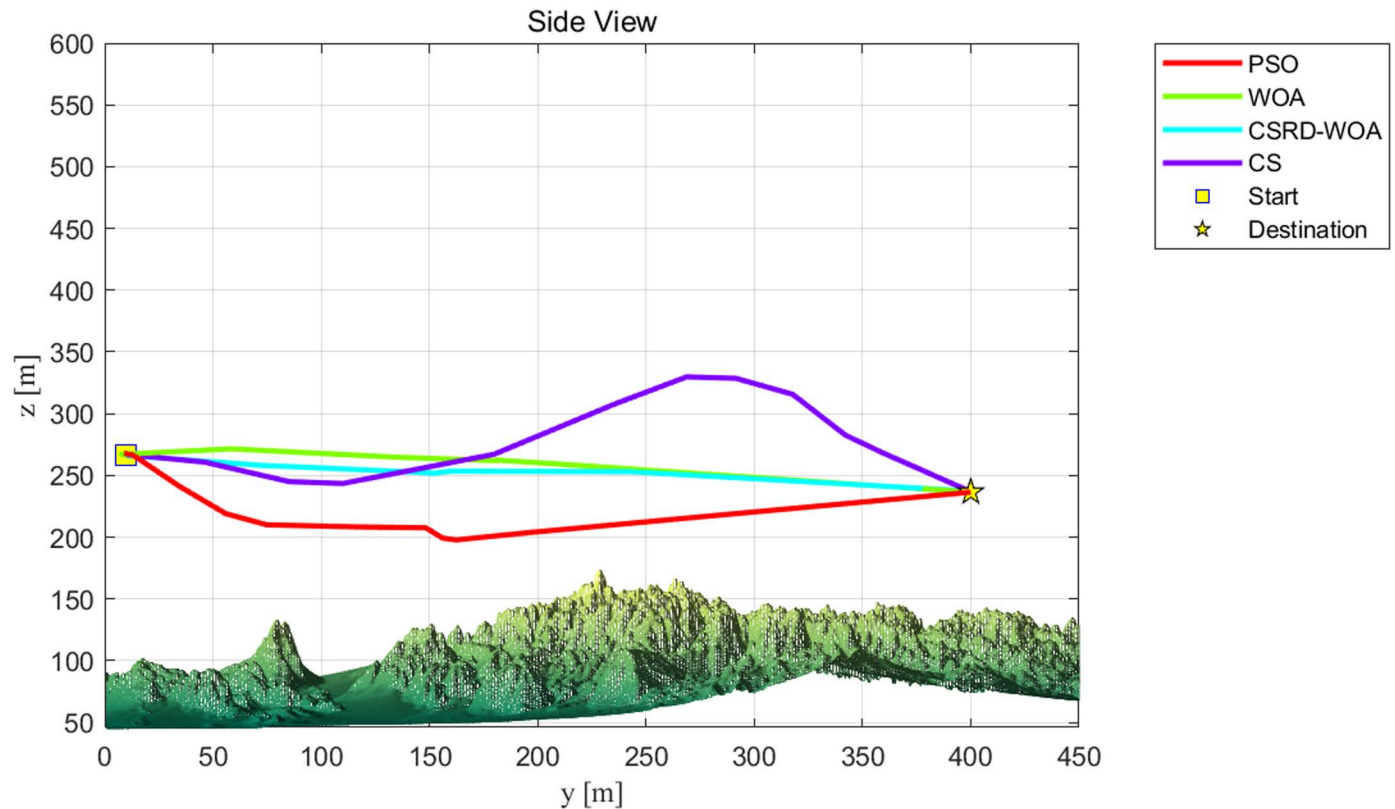


Fig 20. Side view.

<https://doi.org/10.1371/journal.pone.0316836.g020>

In summary, in both scenarios, CSRD-WOA demonstrates higher efficiency in avoiding various threat terrains, and it achieves the lowest total cost, flight distance cost, threat cost, altitude cost, and smoothness cost in path planning.

6. Conclusion

This paper proposes a three-dimensional path planning method for drones based on the CSRD-WOA algorithm. Firstly, after the completion of iterations in the whale algorithm, the diversity and randomness of the population are increased using the strategy of cuckoo-random differential, which compensates for the drawback of the population being prone to local optima in the early stage. This strategy allows the population to escape local optima and improves global search capability. Secondly, a model of drone terrain, threat sources, and drone constraints is constructed, and corresponding cost functions are defined. Finally, simulation results show that the CSRD-WOA algorithm performs excellently in drone path planning, enabling the drone to quickly traverse dangerous areas and avoid obstacles within the shortest distance, demonstrating significant practical value.

Author contributions

Writing – original draft: Zexian Hao.

Writing – review & editing: Haocheng Wang, Yu Zhang.

References

1. Luo W, Zhang GQ, Yuan QB, Zhao YX, Chen HC, Zhou JJ, et al. High-precision tracking and positioning for monitoring Holstein cattle. *PLoS One*. 2024;19(5):e0302277. <https://doi.org/10.1371/journal.pone.0302277> PMID: 38743665
2. Cheng HH, Yang S, Qix H. Online obstacle avoidance path planning method for quadrotor UAVs in urban environments. *Computer Sci*. 2019;46(4):241–6.
3. Evdokimenkov VN, Krasilshchikov MN, Lyapin NA. Guaranteeing UAV trajectory control when approaching a maneuvering air target. *J Comput Syst Sci Int*. 2018;57(5):789–800. <https://doi.org/10.1134/s1064230718050040>
4. Liu Y, Chen C, Wang Y, Zhang T, Gong Y. A fast formation obstacle avoidance algorithm for clustered UAVs based on artificial potential field. *Aerosp Sci Technol*. 2024;147:108974. <https://doi.org/10.1016/j.ast.2024.108974>
5. Martins OO, Adekunle AA, Olaniyan OM, Bolaji BO. An improved multi-objective a-star algorithm for path planning in a large workspace: design, implementation, and evaluation. *Scientific African*. 2022;15:e01068. <https://doi.org/10.1016/j.sciaf.2021.e01068>
6. Tan L, Zhang H, Shi J, Liu Y, Yuan T. A robust multiple Unmanned Aerial Vehicles 3D path planning strategy via improved particle swarm optimization. *Comput Electr Eng*. 2023;111:108947. <https://doi.org/10.1016/j.compeleceng.2023.108947>
7. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Adv Eng Softw*. 2014;69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
8. Mirjalili S, Lewis A. The whale optimization algorithm. *Adv Eng Softw*. 2016;95(95):51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
9. Che Z, Peng C, Yue C. Optimizing LSTM with multi-strategy improved WOA for robust prediction of high-speed machine tests data. *Chaos Solit Fract*. 2024;178:114394. <https://doi.org/10.1016/j.chaos.2023.114394>
10. Abdel-Basset M, Mohamed R, Chakraborty RK, Ryan MJ, Mirjalili S. An efficient binary slime mould algorithm integrated with a novel attacking-feeding strategy for feature selection. *Comput Ind Eng*. 2021;153:107078. <https://doi.org/10.1016/j.cie.2020.107078>
11. Wu K, Tan S. UAV route planning based on improved whale optimization algorithm. 2020;41(Sup.2):107–14.
12. Avcu ME, Gökçe H, Şahin İ. Chapter 32 - A WOA-based path planning approach for UAVs to avoid collisions in cluttered areas. In: Mirjalili S, editor. *Handbook of Whale Optimization Algorithm*. Academic Press; 2024. p. 449–61.
13. Yu X, Luo W. Reinforcement learning-based multi-strategy cuckoo search algorithm for 3D UAV path planning. *Expert Syst Appl*. 2023;223:119910. <https://doi.org/10.1016/j.eswa.2023.119910>
14. Zhang X, Liu Q, Qu Y. An adaptive differential evolution algorithm with population size reduction strategy for unconstrained optimization problem. *Appl Soft Comput*. 2023;138:110209. <https://doi.org/10.1016/j.asoc.2023.110209>
15. Lin M, Wang Z, Chen D, Zheng W. Particle swarm-differential evolution algorithm with multiple random mutation. *Appl Soft Comput*. 2022;120:108640. <https://doi.org/10.1016/j.asoc.2022.108640>
16. Zaman K, Zhaoyun S, Shah B, Hussain A, Hussain T, Khan US, et al. Efficient power management optimization based on whale optimization algorithm and enhanced differential evolution. *Alexandria Eng J*. 2023;79:652–70. <https://doi.org/10.1016/j.aej.2023.08.045>
17. Fadhil S, Zaher H, Ragaa N, Oun E. A modified differential evolution algorithm based on improving a new mutation strategy and self-adaptation crossover. *MethodsX*. 2023 Jun 30;11:102276. <https://doi.org/10.1016/j.mex.2023.102276> PMID: 38098772
18. Liang M, Wang L, Wang L, Zhong Y. A hypervolume-based cuckoo search algorithm with enhanced diversity and adaptive scaling factor. *Appl Soft Comput*. 2024;151:111073. <https://doi.org/10.1016/j.asoc.2023.111073>
19. Yu M-Y, Li T, Xu J, et al. Enhanced dwarf mongoose optimization algorithm with multi-strategy fusion. *J Beijing Univ Aeronaut Astronaut*.
20. Foreman VL, Favaró FM, Saleh JH, Johnson CW. Software in military aviation and drone mishaps: Analysis and recommendations for the investigation process. *Reliab Eng Syst Safe*. 2015;137:101–11. <https://doi.org/10.1016/j.ress.2015.01.006>
21. Phung MD, Ha QP. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Appl Soft Comput*. 2021;107:107376. <https://doi.org/10.1016/j.asoc.2021.107376>