

## RESEARCH ARTICLE

# Improved Dung Beetle Optimizer Algorithm With Multi-Strategy for Global Optimization and UAV 3D Path Planning

LIXIN LYU<sup>1,2</sup>, HONG JIANG<sup>1</sup>, AND FAN YANG<sup>1,2</sup><sup>1</sup>School of Information and Artificial Intelligence, Anhui Business College, Wuhu, Anhui 241002, China<sup>2</sup>College of Industrial Education, Technological University of the Philippines, Manila 1000, Philippines

Corresponding author: Lixin Lyu (Lixin\_Lyu@abc.edu.cn)

This work was supported by the Key Research Project of Natural Sciences in Colleges and Universities, Anhui Province of China, under Grant 2023AH052298.


**ABSTRACT** In high-dimensional scenarios, path planning is a challenging and computationally complex optimization task that requires finding optimal paths within complex domains. Metaheuristic (MH) algorithms offer a practical approach to addressing this issue. The Dung Beetle Optimizer (DBO), categorized as a MH algorithm, takes inspiration from the biological behaviors exhibited by dung beetles. However, DBO exhibits limitations, including inadequate global search capabilities and a tendency to converge on local optima. To address these challenges, this paper proposes a multi-strategy Improved Dung Beetle Optimization algorithm (IDBO) for UAV 3D path planning. Initially, cubic chaos mapping is applied for population initialization, enhancing diversity. Subsequently, a novel global exploration strategy replaces the DBO's original rolling phase, improving information exchange and minimizing parameter dependence. Third, an adaptive t-distribution is introduced to adjust dung beetle positions, balancing exploration and exploitation. Finally, an enhanced population update strategy is proposed, utilizing varied behavioral logic at different algorithm stages to improve solution quality and search efficiency. Additionally, performance comparisons with six advanced algorithms on the CEC2017 test suite, and the validation of IDBO's effectiveness via the Wilcoxon rank-sum and Friedman mean rank test. Meanwhile, in UAV 3D path planning experiment, IDBO achieves the best cost index, which is 1.34% higher than the best cost of original DBO, and is also significantly better than the most advanced algorithms such as WOA, GSA, HHO, COA, and the standard deviation is reduced by 99.93% compared with DBO algorithm, which proves the effectiveness and robustness of IDBO in UAV 3D path planning.

**INDEX TERMS** Dung beetle optimizer, metaheuristic, global optimization, UAV 3D path planning.

## I. INTRODUCTION

The use of Unmanned Aerial Vehicles (UAVs) in diverse applications is rapidly expanding due to their high autonomy, cost-effectiveness, real-time responsiveness, flexibility in deployment, and scalability. These vehicles are equipped with sophisticated sensors and advanced navigation technologies [1], enabling them to perform complex flight missions across varied environments. Over recent years, technological advancements and the increasing commercial

use of drones have significantly enhanced their practical significance [2]. UAVs are now applied in many scenarios, including agricultural monitoring, forest fire detection, rescue operations, and infrastructure inspection. However, the expanding range of application domains has made traditional two-dimensional trajectories insufficient for the complex requirements of today's tasks, highlighting the importance of three-dimensional trajectory planning [3]. This task is challenging as it involves navigating through diverse terrains and obstacles, considering airborne entities, and identifying an optimal or near-optimal flight path. The growing diversity of UAV types and models, along with a broader range

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

of flight missions, has significantly widened the scope of path selection solutions, presenting a complex optimization challenge [4]. As a result, solving the three-dimensional path planning problem for UAVs has become a classic NP-hard problem, leading to a focus on swarm intelligence optimization algorithms as a primary research approach to address these intricate issues [5].

Metaheuristic (MH) algorithms, inspired by natural biological behaviors, harness patterns found in nature to iteratively refine solutions, aiming for efficient outcomes within constrained timeframes [6], [7]. Their simplicity, versatility, and ease of use have made them applicable across a variety of domains, including but not limited to image segmentation [8], path planning [9], [10], agricultural monitoring [11], forest fire detection [12], rescue operations [13], rotor system [14], and cubic transmission [15]. This broad utility underscores their value in addressing diverse optimization challenges, particularly where traditional deterministic methods fall short.

When tackling 3D path planning for UAVs, an NP-Hard problem, the limitations of traditional algorithms become clear due to the task's complexity and dynamic conditions [16]. This difficulty is heightened by changing obstacles and various constraints in UAV path planning, making quick solution finding tough [17], [18]. MH algorithms are more effective here, known for their ability to closely approximate the best solutions efficiently [19]. Their adaptability and quick performance are key for large-scale 3D UAV tasks, backed by empirical studies [20], [21] [22]. J. Sanchez-Garcia et al. proposed a new distributed dynamic particle swarm optimization algorithm for UAV network (dPSO-U) to generate the trajectory of UAV network in the disaster scenario search task. Compared with the optimal trajectory planning algorithm covering the entire disaster scenario, the proposed algorithm can find victims faster and converge faster [23]. Gewen Huang et al. proposed a multi-UAV path planning model with the energy constraint (MUPPEC). The proposed method considers the energy consumption of the UAV in different states such as acceleration, cruising speed, deceleration and hovering to minimize the total monitoring time [24].

The No Free Lunch (NFL) Theorem reminds us there's no one perfect algorithm for every problem [25]. It highlights the importance of customizing algorithms for specific tasks. Thus, developing MH algorithms tailored for 3D UAV path planning is crucial for better efficiency [26], aligning with the NFL Theorem and stressing the need for specific solutions for complex problems.

The DBO algorithm, a meta-heuristic approach, was conceived by Professor Shen Bo's team in 2023. It draws its inspiration from the five distinct behaviors of dung beetles: rolling balls, dancing, breeding, foraging, and stealing [27]. DBO is characterized by its straightforward structure, minimal parameter requirements, and robust prospecting and exploration abilities. It finds extensive applications in

tackling optimization issues in areas like high-dimensional feature selection and data clustering [28]. Nevertheless, despite these attributes, DBO exhibits limitations such as slower convergence rates and a propensity to become entrapped in local optima. Furthermore, if UAVs path planning is reduced solely to factors like flight distance and time, it risks oversimplification, potentially failing to capture the optimal flight path in practical scenarios. To date, only a handful of researchers have experimented with applying DBO to three-dimensional UAVs trajectory planning. Like all optimization algorithms, achieving an optimal balance between exploration and exploitation is vital for determining the ideal flight path [29]. In essence, as a nascent algorithm, DBO necessitates additional investigation and refinement to more effectively address the intricate demands of three-dimensional path planning.

In the research of modern optimization algorithms, finding an effective strategy to balance exploration and exploitation is always one of the core problems. Kahraman et al. [30] proposed a selection method based on Fitness Distance Balance (FDB) to solve the problem of premature convergence in the selection process of metaheuristic algorithm. This approach optimizes the selection strategy of the population by considering the fitness of the candidate solution and its distance to the global optimal solution. In addition, in order to better deal with constrained optimization problems, Ozkaya et al. [31] introduced the Fitness-Distance-Constraint (FDC) model and the dynamic guidance mechanism. These methods have shown excellent performance in improving the performance of the algorithm in the constrained environment. Kahraman et al. [32] uses NSM score instead of fitness value to design the update mechanism. Although the computational complexity is increased, the results show that the NSM version has obvious advantages in finding the optimal solution. Although these advances have promoted the performance of the algorithm on static and standard test problems, there are still many challenges when dealing with dynamic and complex practical application problems, such as UAV path planning.

To improve the convergence speed and exploration capabilities of the DBO algorithm, this paper introduces an advanced version of DBO, incorporating a novel global search strategy and an adaptive t-distribution. Initially, Cubic chaos mapping is employed to augment the randomness and diversity during the population initialization of DBO. Subsequently, the newly devised global search strategy replaces the conventional DBO rolling phase, facilitating improved information exchange among DBO entities, diminishing parameter dependency, and bolstering global exploration effectiveness. Furthermore, the integration of an adaptive t-distribution perturbation approach during the dung beetle foraging phase ensures a balanced interplay between exploration and exploitation. Lastly, the implementation of a contemporary population update strategy, which infuses adaptive factors and elements of randomness, substantially upgrades the algorithm's overall performance.

Experimental findings utilizing 29 functions from the CEC2017 benchmark suite demonstrate that the IDBO algorithm markedly enhances the global optimization capability. This advancement effectively boosts both the convergence rate and the precision of the algorithm. The results demonstrate that the new algorithm proposed in this study outperforms three competing algorithms: GSA [33], GWO [34], WOA [6], HHO [35], DMO [36], COA [37] and the original DBO algorithm [27]. Through solving the UAVs three-dimensional path planning problem, it is verified that the IDBO algorithm also possesses high applicability for engineering problems.

In response to the limitations of DBO, this paper introduces the IDBO algorithm and applies it to address three-dimensional path planning for UAVs. The key contributions of this research are as follows:

- 1) To address the limitations of traditional DBO, we've enhanced the algorithm with several key improvements: the implementation of a Cubic chaotic map, a new global search strategy, an adaptive t-distributed perturbation strategy, and a novel adaptive and random population update method.
- 2) We rigorously tested the enhanced IDBO's exploration and exploitation abilities using the CEC 2017 benchmark for optimization. The tests clearly showed the algorithm's skill in navigating and utilizing the solution space effectively.
- 3) Additionally, we evaluated IDBO's practical value by applying it to 3D path planning for UAVs. This test highlighted the algorithm's capability to efficiently solve complex real-world engineering challenges with high accuracy.

The second part introduces the improvement work of DBO algorithm and MH algorithm in UAV trajectory planning. Section III outlines the structure of the original algorithm and introduces the proposed method. In Section IV, we carry out pertinent experimental tests and provide an in-depth analysis of the proposed algorithm. The fifth part is the application analysis of IDBO algorithm in UAV three-dimensional trajectory planning. The sixth part is the summary of the paper.

## II. RELATION WORKS

In recent years, with the widespread application of UAVs in various fields, the field of UAV three-dimensional path planning has aroused widespread research interest [38]. Effective path planning is essential for drones to fulfill their missions safely and efficiently. Faced with the NP-hard complexity inherent in three-dimensional UAV trajectory planning and the necessity for real-time combat capabilities, numerous researchers have delved into an array of optimization algorithms and strategies. Among these, swarm intelligence optimization algorithms have gained prominence in UAV trajectory planning [39], attributed to their high efficiency and swift response. Typically inspired by natural

behaviors and phenomena, these algorithms demonstrate significant potential in identifying optimal or near-optimal solutions for intricate path planning challenges. This section aims to explore recent developments in this area, emphasizing enhancements to the DBO and the utilization of various heuristic algorithms in addressing three-dimensional path planning issues in UAV operations.

Fang Zhu and their research team employed an optimal point set strategy that takes into account the convergence and dynamic balance of egg-laying and foraging insect behaviors. Additionally, they integrated a t-distribution mutation strategy, inspired by quantum computing principles, to enhance the DBO. This enhancement greatly improved the algorithm's ability to overcome local optima [28]. Shen and their research team utilized the Beta distribution to dynamically generate reflection solutions and incorporated the Levy distribution to manage particle crossings of boundaries. This algorithm incorporates two distinct crossover operators to enhance the update phase, expedite algorithm convergence, and strike a balance between exploration and exploitation capabilities [40]. Additionally, Wang Zilong and team proposed a multi-strategy DBO algorithm, known as GODBO, which leverages the current population's optimal value for directional migration and reverse learning, thereby boosting the algorithm's exploratory power [41]. Longhai Li and associates introduced a fitness-distance balance strategy and implemented a spiral foraging approach to refine the algorithm's search precision, expand its exploratory ability, and circumvent local optima. By integrating an optimal dimension Gaussian mutation strategy, they increased population diversity and hastened the algorithm's convergence speed [42]. Concurrently, Xu-ruo Wei and others merged the Simulated Annealing (SA) algorithm with the DBO algorithm to diminish the likelihood of converging to local extremes [43]. Furthermore, Zhang Zhanyou and their research collaborators introduced an optimal point-set strategy. They integrated the spiral search method with the whale optimization algorithm to achieve a balance between local and global search capabilities. Additionally, they leveraged Levi's flight strategy to enhance the algorithm's capacity to escape local optima [44]. Shuang Li et al. developed an improved DBO algorithm that integrates an adaptive t-distribution strategy [45], thereby augmenting the local search efficiency in the algorithm's later stages. Zhang Haiyang and colleagues implemented TENT chaotic mapping and reverse learning technology to intensify population randomness. They optimized the weight of the algorithm using the Levy flight strategy and further refined this strategy to maintain a high level of global development capability, ultimately aiming to expedite the optimization process [46]. A novel method for updating addresses was also introduced. Ran Zhang et al. incorporated sinusoidal chaos mapping into the DBO's initial population, which enhanced both the initial population's quality and the algorithm's stability. Subsequently, they employed sinusoidal chaotic mapping in

combination with DBO [47] to address the objective function of task allocation.

Recent advancements in UAV path planning have seen various innovative approaches. Shi Jiaqi and colleagues enhanced the grey wolf optimization algorithm with a spiral update position method, significantly reducing UAV flight times by 22.8%, improving convergence speed, and smoothing flight paths [48]. Zhu Hongyue's team boosted the PSO algorithm by integrating an improved nonlinear dynamic inertia weight, adaptive speed adjustment, and chaotic initialization, leading to faster convergence and smoother paths [49]. Hu Gang and associates enhanced the HBA algorithm using the Bernoulli shift map and other strategies, applying it successfully to UAV path planning [50]. Pan and colleagues proposed a golden eagle optimization algorithm with a dual-learning strategy, proving effective in power detection tasks for UAVs [51]. Jiang's research team introduced a sophisticated 3D path planning group optimization algorithm, featuring a segmented fitness strategy [52]. Additionally, Zhang and colleagues presented a novel *Drosophila* optimization algorithm incorporating phase angle coding and mutation adaptation [53]. Finally, Chen and team's flower pollination algorithm, based on neighborhood global learning, offered a novel solution to UAV path planning [54]. Yilmaz et al. [55] proposed an innovative adaptive evolutionary optimization algorithm to deal with the routing optimization problem of multiple drones and truck delivery systems. It is called Fitness-Distance Balance-based Evolutionary Algorithm (FDB-EA). The algorithm is especially designed to solve the high geometric complexity and multiple local solution trap problems encountered in the Traveling Salesman Problem and Unmanned Aerial Vehicle (TSP-D). These diverse methodologies underscore the dynamic progress in UAV path planning, focusing on efficiency, accuracy, and adaptability to complex environments.

While many researchers have done extensive work in employing metaheuristic algorithms to address UAVs path planning, some have improved evolutionary algorithms and achieved favorable outcomes. However, although DBO has been applied in various domains, research in the context of UAVs path planning remains limited. To achieve optimal path planning, a deeper investigation into the two core mechanisms of swarm intelligence algorithms, exploitation and exploration, is still necessary.

In our study, the improved IDBO algorithm by integrating multiple strategies can effectively strikes a balance between exploration and exploitation processes. Experimental results demonstrate that the improved algorithm outperforms recent algorithms, including GSA [33], GWO [34], WOA [6], HHO [35], DMO [36], COA [37] and the original DBO algorithm [27]. The enhanced algorithm yields superior results in the context of three-dimensional trajectory planning for UAVs.

### III. THE PROPOSED METHODOLOGY

This section briefly describes the behavior of the original DBO and the corresponding mathematical model. In addition, this section highlights the proposed IDBO algorithm, which contains Cubic chaotic map, global exploration strategy of Osprey optimization algorithm, and adaptive t-distributed perturbation strategy.

#### A. THE ORIGINAL DBO

The DBO algorithm draws inspiration from diverse dung beetle behaviors, encompassing ball rolling, dancing, foraging, stealing, and reproduction. In DBO, the population is categorized into four distinct types of search agents, each governed by unique update rules:

##### 1) BALL-ROLLING DUNG BEETLE

Dung beetles roll their feces into balls and use celestial cues such as the sun and moon to navigate efficiently. When encountering an obstacle, dung beetles will often climb onto the ball and dance to determine a new direction of movement. The location update formula is as follows:

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

$$\Delta x = |x_i(t) - X^w| \quad (2)$$

where, the current iteration number is represented by  $t$ , and  $x_i(t)$  denotes the position of the  $i$ th beetle at that iteration. The deflection coefficient  $k$ , a constant within  $(0, 0.2]$ , influences the beetle's directional changes, while  $b$ , another constant in the range  $(0, 1)$ , plays a part in position calculation. The natural coefficient  $\alpha$ , which is set to either 1 or -1, adds variability to the update process. The global worst position in the optimization context is represented by  $X^w$ , and  $\Delta x$  models changes in light intensity.

When a dung beetle encounters an obstacle during its ball-rolling journey, it executes a distinctive behavior referred to as a 'dance' to reorient itself and discover an alternative path. This dance behavior is emulated in the algorithm through the use of a tangent function, mimicking the beetle's process of determining a new direction for rolling when faced with an obstacle. Consequently, the update for the position of the rolling dung beetle is defined as follows:

$$x_i(t+1) = x_i(t) + \tan(\theta) |x_i(t) - x_i(t-1)| \quad (3)$$

where,  $\theta$  is the angle of deflection represented by radian and  $\theta \in [0, \pi]$ .

##### 2) BROOD BALL

Some of the dung balls collected by the beetles are used as food, and the other is pushed to a safe place to lay eggs, which are used as brood balls to breed the next generation. The boundaries of the brood balls are strictly restricted as follows:

$$Lb^* = \max(X^* \times (1 - R), Lb)$$

$$Ub^* = \min(X^* \times (1 - R), Ub) \quad (4)$$

where, the current local best position is represented by  $X^*$ , which is akin to the dung beetle's ideal spawning site. The lower bound ( $Lb$ ) and upper bound ( $Ub$ ) of the egg-laying area are defined to replicate the range within which dung beetles select their egg-laying locations. The factor  $R = 1 - t/T_{max}$  adjusts these bounds over time, accounting for the progression of the algorithm through its iterations, where  $t$  is the current iteration and  $T_{max}$  is the maximum number of iterations. This adjustment reflects the dung beetle's decreasing range of choices as conditions evolve. Additionally, the overall search space has its boundaries defined by  $Lb$  and  $Ub$ , representing the broadest area within which the dung beetles operate.

After pushing the brood ball to the defined spawning area, females lay eggs in it, and each female lays only one egg per iteration. The position of the brood ball is updated as follows:

$$B_i(t+1) = X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*) \quad (5)$$

In the model,  $B_i(t)$  represents the position of the  $i$ th sphere (analogous to the egg ball laid by the dung beetle) at the  $t$ th iteration. To introduce randomness and variability in the model, akin to the unpredictability in natural processes, two independent random vectors are used:  $b_1$  and  $b_2$ . These vectors have a size of  $1 \times D$ , where  $D$  denotes the dimensionality of the problem.

### 3) SMALL DUNG BEETLE

When the small beetles in the brood ball are mature, they will come out to search for food. Therefore, it is necessary to establish the optimal foraging area to guide them to search for food to achieve the purpose of space exploration. The optimal foraging area boundary is defined as follows:

$$\begin{aligned} Lb^b &= \max(X^b \times (1 - R), Lb) \\ Ub^b &= \min(X^b \times (1 - R), Ub) \end{aligned} \quad (6)$$

where,  $X^b$  signifies the global optimal position, with the lower and upper bounds of this optimal foraging area represented as  $Lb$  and  $Ub$ , respectively. As a result, the update for the location of the small dung beetle is defined as follows:

$$x_i(t+1) = x_i(t) + C_1 \times (x_i(t) - Lb^b) + C_2 \times (x_i(t) - Ub^b) \quad (7)$$

where,  $x_i(t)$  denotes the position of the  $i$ th dung beetle at the  $t$ th iteration of the algorithm. Additionally,  $C_1$  is a random number that follows a normal distribution.  $C_2$  is a random vector whose values lie within the range of (0, 1).

### 4) THIEF DUNG BEETLE LOCATION UPDATE

While not all dung beetles exert strong effort in pushing, some resort to stealing dung balls. Assuming that the global optimal position serves as the most suitable target for theft, the position update formula for the thief dung beetles is

expressed as follows:

$$x_i(t+1) = X^b + S \times g \times (|x_i(t) - X^*| + |x_i(t) - X^b|) \quad (8)$$

where,  $x_i(t)$  signifies the location of the  $i$ -th 'thief' (a metaphorical agent in the algorithm) at the  $t$ -th iteration. The variable  $g$  is a random vector that follows a normal distribution. This vector has a dimension size of  $1 \times D$ , where  $D$  represents the dimensionality of the problem.  $S$  is a constant value used in the algorithm.

## B. THE PROPOSED IDBO

Considering the aforementioned analysis, we improved the DBO algorithm from three perspectives:

- 1) The Cubic chaos mapping is used to initialize the positions of the dung beetles. Utilizing its nonlinear and dynamic characteristics, this method can generate a more diverse set of initial solutions, aiding the algorithm in searching through a broader solution space.
- 2) A new global search strategy is proposed, which reduces the dependence on algorithm parameters and enhances the global search capability.
- 3) During the foraging phase of dung beetles, the foraging behavior of smaller dung beetles is perturbed with a  $t$ -distribution, balancing exploration and exploitation, thereby improving the convergence speed.
- 4) An improved population update strategy is proposed, which adopted different population update methods in different stages of the algorithm to cope with the search requirements of different stages and explore a wider solution space.

### 1) IMPROVED POPULATION INITIALIZATION WITH CUBIC CHAOTIC MAP

A discrete, high-quality initial population can accumulate rich search experience for DBO, laying the foundation for intelligent searches in heuristic algorithms. Existing algorithms commonly utilize pseudorandom numbers to initialize candidate solutions. Such configuration can maximize the algorithm's global performance. However, the strong randomness means the algorithm can't maintain stable objective optimization accuracy. Furthermore, relying on pseudorandom number initialization can lead to insufficient population traversal, leading to a decline in population diversity. To enhance exploration capabilities and elevate the level of population diversity [56], we use chaotic maps to improve the population initialization. Chaotic map is a kind of mathematical map showing dynamic, complex and unpredictable behavior, and Cubic chaotic map is one of the commonly used forms [57], [58]. The Cubic chaotic map refers to a type of mathematical map characterized by cubic nonlinearity, which is often used in chaos theory and nonlinear dynamics. The map is defined by

the recurrence relation:

$$x_{n+1} = \mu \cdot x_n \cdot (1 - x_n^2) \quad (9)$$

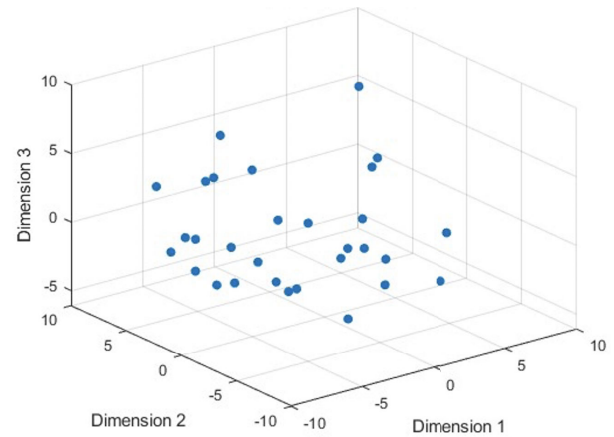
where  $x_n$  is a number between 0 and 1 that represents the system state at iteration  $n$ , and  $\mu$  sets a constant multiplier for the cubic map. In this case, it's set to 1.0, which means it does not change the scale of the result. This chaos is characterized by its sensitivity to initial conditions, leading to a behavior that appears random and unpredictable.

In the population initialization phase, a matrix named Cubic is initialized with random values between 0 and 1 for each member of the population ( $N$ ) across all dimensions ( $dim$ ). For each individual in the population, the Cubic chaos equation is iteratively applied across all dimensions. The chaos parameter  $\mu$ , set to a value of 1.0 in this case, governs the chaotic behavior of the map. The iterative process updates each dimension of an individual based on the previous dimension's value, thereby introducing a chaotic sequence in each dimension. The resulting matrix Cubic, now filled with values generated by the chaotic process, serves as the initial population for the DBO algorithm. This population exhibits a high degree of diversity due to the inherent unpredictability and randomness of the Cubic map. In the comparison of the initialization population distribution shown in Fig. 1, we can observe that after using Cubic chaotic map for population initialization, the population distribution obtained is significantly more dispersed. This result highlights the effectiveness of Cubic chaotic map in optimizing the population distribution and provides a broader exploration basis for the subsequent search process.

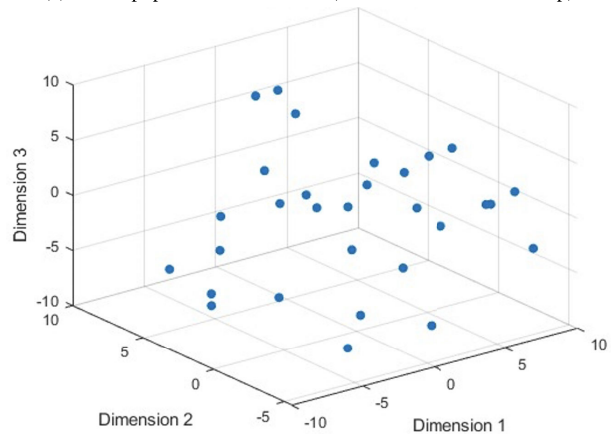
In summary, the utilization of Cubic chaos mapping during the initialization phase of the DBO guarantees that the initial population covers an extensive solution space. This mitigates the risk of premature convergence towards a local optimum. Furthermore, by diversifying the initial population, the DBO algorithm can explore a broader spectrum of the search space, thus enhancing the probability of discovering the global optimal solution.

## 2) NEW GLOBAL SEARCH STRATEGY

For low-dimensional test functions, the introduction of the aforementioned strategy somewhat balances the global exploration and local exploitation processes. However, when applied to large-scale complex optimization problems, the singular search pattern of DBO can easily lead the dung beetle population into local optima traps. Hence, enriching the search mode of random individuals is another crucial approach to improving the global search performance of heuristic algorithms. Jui and Ahmed [59] in AMVO-SCA rely on the average of multiple locations and mathematical guidance steps defined by trigonometric functions to provide a smoother and more globally aware search process, but increase the computational complexity. Mohd Tumari et al. [60] proposed an improved marine predators algorithm to determine new search points by calculating the average of the optimal solution and the current solution, promoting a



(a) Initial population distribution (without Cubic chaotic map)



(b) Initial population distribution (with Cubic chaotic map)

**FIGURE 1. Initial population distribution.**

balance between exploration (deviation from the current path) and exploitation (optimization around the best solution). As a result, local optima are avoided. But direct use of averaging does not inherently include radically changing trajectories close or far away, and while averaging can help mitigate rapid convergence in non-optimal regions, it may not be as effective in environments with changing landscape dynamics or extremely rugged environments.

In order to overcome the limitations of ball rolling in DBO that only depends on the worst solution and lack of communication with other dung beetles, as well as the problem of complex parameter settings [61], we propose a new global search strategy. In this paper, the new global exploration strategy is used to randomly detect the position of one of the fecal balls and roll it. The new global exploration strateg in the first stage is formulated as follows:

$$x_{i,j}^{p1} = (1 - r_{i,j}) \cdot x_{i,j} + r_{i,j} \cdot (SF_{i,j} - I_{i,j} \cdot x_{i,j}) \quad (10)$$

where,  $x_{i,j}^{p1}$  denotes the position vector of the  $i$ th dung beetle individual in the current iteration.  $x_{i,j}$  represents the position vector of the  $i$ th dung beetle individual in the previous iteration.  $r_{i,j}$  is random number in the range  $[0,1]$ .  $SF_{i,j}$  represents a better solution selected from the current population based on

the fitness value, if no solution is found that is better than the current individual,  $SF_{i,j}$  is set to the global optimum. If there exists at least one better solution, then with 50% probability the global optimal solution is chosen directly and with the other half probability one of these better solutions is randomly selected as  $SF_{i,j}$ .  $I_{i,j}$  is a randomly generated integer (1 or 2) that randomly decides whether to move closer (if  $I_{i,j}$  is 1) or farther (if  $I_{i,j}$  is 2) toward  $SF_{i,j}$  during the update process.

In the initial stage, a novel global exploration strategy supplants the original position update formula of the dung beetle algorithm during the rolling phase. This strategy introduces randomness and enhanced global search capabilities into the DBO, enabling the simulated dung beetles to update their positions not solely based on the poorest solution, but also to explore other potentially superior locations at random. Such enhancements significantly mitigate premature convergence and facilitate a more efficient exploration of the solution space, consequently elevating the likelihood of identifying the global optimum.

Furthermore, diminishing the quantity of parameters bolsters both the practicability and stability of the algorithm. In optimization contexts, an excess of parameters can escalate the complexity of calibration and potentially diminish the algorithm's adaptability to particular problems. Streamlining these parameters enhances the algorithm's flexibility and user-friendliness.

### 3) ADAPTIVE T-DISTRIBUTED PERTURBATION STRATEGY

Adaptive t-distribution is an improved probability distribution that is commonly used in optimization algorithms to enhance exploration and convergence performance [62], [63]. The foraging behavior of the dung beetle is modified by the t-distribution mutation perturbation, where the iteration number variation formula acts as a flexible parameter for the t-distribution. This approach equips the Dung Beetle Optimizer (DBO) with enhanced global development capabilities during the initial iterations and superior local exploration prowess in the latter stages. Such a strategy accelerates the algorithm's convergence rate. The detailed method for location updating is outlined below:

$$X_{new}^j = X_{best}^j + t(C_{iter}) \cdot X_{best}^j \quad (11)$$

$$C_{iter} = 1/\exp(-4 \times (t/M)^2) \quad (12)$$

In the equation:  $X_{new}^j$  denotes the position vector of the  $i$ th dung beetle individual in the current iteration.  $X_{best}^j$  represents the global best solution found in the current iteration phase.  $C_{iter}$  is the parameter of the adaptive t-distribution,  $t(C_{iter})$  is a random number generated from an adaptive t-distribution.  $M$  represents the maximum number of iterations.

The adaptive t-distribution modulates the randomness in the dung beetle's movement based on the current iteration count, a key factor in augmenting the exploration and convergence potential of the DBO. This adaptive technique harmonizes the algorithm's exploratory and developmental capacities, accelerates its convergence rate, and enhances

both its efficiency and effectiveness in addressing optimization challenges.

### 4) POPULATION UPDATE STRATEGY BASED ON ADAPTATION AND RANDOMNESS

To further balance the global exploration and local development capabilities of the algorithm and to prevent convergence on local optima, a novel population update strategy is employed. This strategy aims to elevate the algorithm's overall performance by incorporating adaptive factors and randomness. It enhances the algorithm's functionality at various stages, employing distinct behavioral logic to improve solution quality and search efficiency. The precise method for location updating is outlined below:

$$X_{new}^j = X^j + r_1 \cdot (X_{best}^j - X_{worse}^j)/(f_{max} - f_{min}) \quad (13)$$

$$X_{new}^j = X^j + r_1 \cdot (X_{best}^j - X_{worse}^j) + r_1 \cdot (X_{bestself}^j - X^j) \quad (14)$$

$$X_{new}^j = X^j + r_1 \cdot (X_{best}^j - X_{worse}^j) + r_1 \cdot (X_{bestself}^j - X^j) + r_2 \cdot \Delta X^j \quad (15)$$

Among them,  $X^j$  represents the position vector of the  $i$ th dung beetle individual in the current iteration.  $X_{best}^j$  is the global best solution found in the current iteration phase.  $X_{worse}^j$  represents the global worst solution found in the current iteration phase.  $r_1$  is a random number between 0 and 0.5,  $r_2$  is a random number between 0 and 0.2,  $X_{bestself}^j$  represents an individual's historical optimal position,  $\Delta X^j$  is the amount of change in the position of the individual in the last iteration.

In the first iteration of the algorithm, the position update of each individual follows the following logic: Each individual has a 50% chance to choose Eq. (13) or Eq. (14) to update the population position, and this random selection mechanism provides the algorithm with the ability to initially explore the solution space. In Eq. (13), the individual positions are updated according to the difference between the global best and worst positions. This method aims to guide individuals to move towards the optimal solution region, while taking into account the information of the worst solution to increase diversity. In Eq. (14), the individual position considers both the global optimal position and the individual historical optimal position. This approach combines global information and individual experience to facilitate meticulous search within known favorable regions.

From the second iteration, a new component is added to the update strategy of individual position. Each individual has a 50% chance to choose Eq. (13) or Eq. (15) to update the population position. In Eq. (15), the change of the individual position in the previous iteration is introduced. This element incorporates information based on the previous step's position change, thereby enhancing the algorithm's exploration capacity and preventing it from getting trapped in local optima.

This strategy combines the best global, worst-case, and historical positions to explore solutions broadly in the early

stage. In later stages, it delves deeper into promising areas by considering past location changes. The strategy adapts based on the algorithm's progress, responding flexibly to different search requirements. The introduction of random elements increases search process diversity, exploring a wider solution space.

The enhanced population update strategy, discussed in this section, effectively boosts the optimization algorithm's performance on complex problems. It achieves this by incorporating distinct behavioral rules for different algorithm stages. This approach not only improves the algorithm's ability to explore globally but also ensures its effective progression in local regions, thereby enhancing the chances of discovering high-quality solutions.

### 5) IDBO ALGORITHM DESCRIPTION

In the enhanced IDBO, Cubic chaos mapping is employed to attain a superior initial solution quality. Furthermore, a novel global search strategy replaces the original position update approach during the rolling phase. This change enhances information sharing among individuals, simplifies parameter selection, and yields better global search performance. Following the location update, to achieve a more balanced exploration-development trade-off, an adaptive t-distribution is introduced to update the positions of certain dung beetles. This adjustment leads to obtain a higher fitness evaluation. An enhanced population update strategy is proposed, which uses different behavioral logic in different algorithm stages to improve solution quality and search efficiency. The detailed process of IDBO is depicted in Fig.2, with the implementation steps outlined as follows:

- 1) Define the maximum number of iterations as  $M$ , the dimension as  $D$  and the population size as  $N$ . Utilize Uubic chaos mapping for population initialization and subsequent calculation of individual fitness values.
- 2) Update the location of the pushing dung beetle. If  $\lambda < \gamma$ , for barrier-free state, use Eq. (10) to update the position; otherwise, if it is in obstructed state, update according to Eq. (3), where  $\lambda \in [0, 1]$  and  $\gamma = 0.8$ .
- 3) Update the location of brood ball by Eq. (5) and use the upper and lower bounds in Eq. (4) to constrain the new position.
- 4) The t-distribution variation perturbation is performed on the foraging behavior of dung beetles by Eq. (11) and Eq. (12).
- 5) Update the location of little dung beetle by Eq. (7).
- 6) Update the location of the thief dung beetle by Eq. (8).
- 7) Update the global optimal position  $X^b$  and the worst position  $X^w$ .
- 8) Update the position of each dung beetle. If  $t = 1$ ,  $\lambda < \beta$ , for each dung beetle, use Eq. (14) to update the location; otherwise, update by Eq. (13); If  $t > 1$ ,  $\lambda < \beta$ , for each dung beetle, use Eq. (15) to update the location; otherwise, update by Eq. (13). where  $\lambda \in [0, 1]$  and  $\beta = 0.5$ .

- 9) Check whether the algorithm has reached the specified number of iterations. If it has, halt the execution and return the optimal position. If not, proceed to the next step (2) to continue searching for the optimal solution.

### C. COMPUTATIONAL COMPLEXITY OF IDBO

The computational complexity of the IDBO algorithm is primarily influenced by two key factors: solution initialization and the execution of core functions. These core functions encompass fitness function calculations and solution updates. The computational complexity is influenced by crucial variables, including the number of solutions ( $N$ ), the maximum iteration limit ( $T$ ), and the problem's dimension ( $D$ ). Specifically, the complexity of initializing solutions is represented as  $O(N)$ , indicating its direct relationship with the number of solutions. As  $N$  increases, the computational complexity of the initialization phase also rises accordingly. The overall time complexity for the core functions of the algorithm is  $O(T \times N \times D)$ , considering the number of iterations ( $T$ ), the count of solutions ( $N$ ), and the problem dimension ( $D$ ). IDBO modifies this with Eqs. (9), (10), and (11)-(15), including enhancements to population diversity using the Cubic chaos mapping, adoption of new global search strategy to reduce parameter dependence and enhance information exchange, the introduction of a population adaptive update based on the t-distribution variation perturbation and population update strategy based on adaptation and randomness. The Cubic chaos mapping strategy, which requires computation for each individual, exhibits a complexity of  $O(N)$ . The update process delineated in Eq.(10) hinges on the population size, search dimension, and the upper limit of iterations, leading to a time complexity of  $O(T \times N \times D)$ . Similarly, the updates as per Eqs. (13)-(15) are contingent upon the same factors — population size, search dimension, and maximum iterations — yielding an identical time complexity of  $O(T \times N \times D)$ . Furthermore, the computational complexity for both Eq. (11) and Eq. (12) is also  $O(T \times N \times D)$ . Consequently, the overall time complexity of IDBO is  $O(IDBO) = O(N) + O(T \times N \times D) + O(T \times N \times D) + O(T \times N \times D) = O(T \times N \times D)$ , consistent with the original algorithm.

## IV. ALGORITHM PERFORMANCE TESTING AND ANALYSIS

The simulation environment of this study runs on a Windows 11 64-bit operating system, with a CPU model of AMD Ryzen 74800H, a base frequency of 2.30GHz, and equipped with 16GB RAM. The algorithms were implemented on the Matlab 2023b platform.

### A. TEST FUNCTIONS AND PARAMETER SETTINGS

To assess the efficacy of the newly proposed IDBO algorithm, it was tested using the CEC2017 test function set ( $Dim = 30$ ). The CEC series comprises a diverse array of fundamental test functions, serving not only as benchmarks for comparing the performance of various optimization algorithms but also as tools to emulate the complexity of real-world problems.



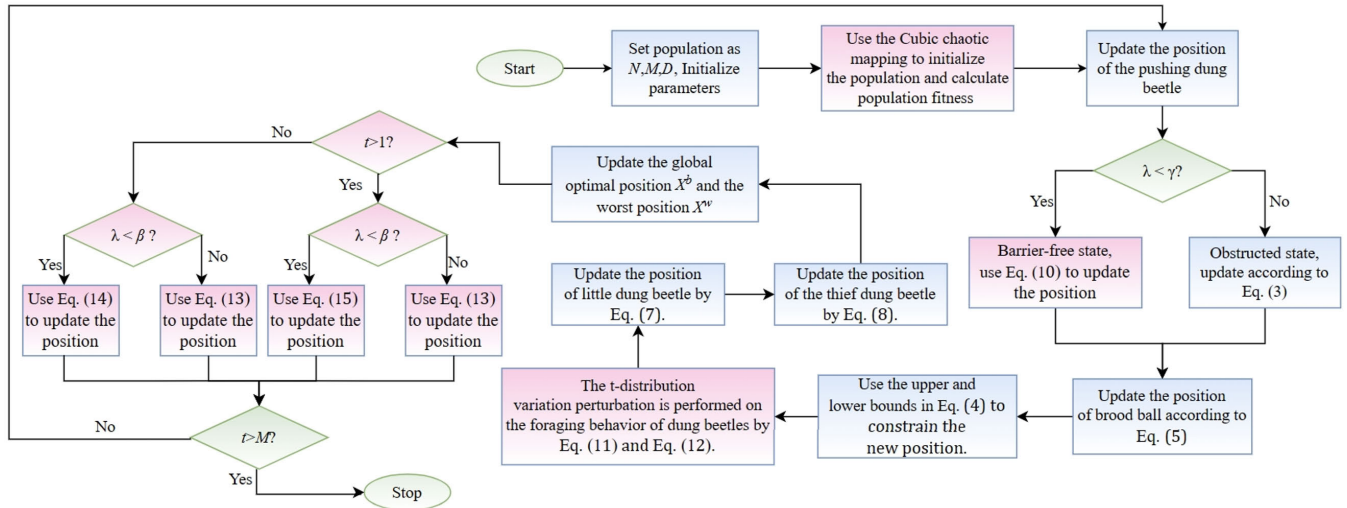


FIGURE 2. Flow chart of IDBO.

TABLE 1. Parameter configurations for competing algorithms.

Algorithms	Parameter	Value
GSA	$v, r_i$	[1,2], [0,1]
GWO	$a$	[0,2]
WOA	$a, a2, b$	[0,2], [-1,-2], 1
HHO	$E0, E1$	[-1,1], [0,2]
DMO	$peep$	2
COA	$C1, C2, \mu, \sigma$	0.2, 3, 25, 3
DBO	$k, b, \theta, \alpha$	(0, 0.2], (0, 1), [0, $\pi$ ], 1 or - 1
IDBO	$I_{l,j}, \mu, \theta$	1 or 2, 3, [0, $\pi$ ]

This set includes 29 CEC2017 test functions, each composed of different basic test functions. Specifically,  $f_1$  and  $f_3$  are unimodal functions,  $f_4$  to  $f_{10}$  are multimodal functions,  $f_{11}$  to  $f_{20}$  are hybrid functions, and  $f_{21}$  to  $f_{30}$  are composite functions. The  $f_2$  function is omitted due to its instability in higher dimensions. The search domain for the CEC2017 test function set is uniformly set to  $[-100, 100]^D$ .

The efficacy of the Improved Dung Beetle Optimizer (IDBO) algorithm was validated against six highly-cited algorithms: GSA [33], GWO [34], WOA [6], HHO [35], DMO [36], COA [37] and the original DBO algorithm [27]. Table 1 offers a detailed summary of the parameters used in these seven distinct MH algorithms. The compared algorithms are consistent with the parameter Settings in the original literature. The experimental outcomes are meticulously documented, including the average (denoted as ‘Ave’) and standard deviation (Std) for each algorithm. To enable a clear performance comparison, the best results among the 13 algorithms are emphasized in bold within the table. In these tests, the population size ( $N$ ) of each algorithm was fixed at 30, and the maximum iteration count ( $T$ ) was set to 500. Every experiment was independently conducted 30 times, with the optimum fitness value from each trial being systematically recorded.

### B. ABLATION EXPERIMENT

In this section, we conduct a thorough analysis of the impact of four proposed enhancement strategies on the DBO algorithm. These strategies encompass the utilization of Cubic chaos mapping, the introduction of a novel global exploration strategy, the incorporation of an adaptive t-distribution strategy, and the implementation of an improved population update strategy. Based on these improvements, four new algorithm variants are named: CDBO for the Cubic chaos mapping, GDBO for the new global exploration strategy, TDBO for the adaptive t-distribution strategy, and PDBO for the new improved population update strategy. According to the experimental results in Fig.3, all four strategies significantly enhance the convergence accuracy and speed of DBO, with IDBO showing particularly notable performance.

Across each benchmark function, the IDBO uniformly attains optimal or near-optimal performance, distinguished by its swift convergence and exceptional precision. While the optimization outcomes of the CDBO and the original DBO show comparable consistency, CDBO notably achieves faster initial convergence. This improvement is credited to cubic chaos mapping, which furnishes DBO with enhanced initial solutions, facilitating more rapid convergence in the initial phases of optimization. In contrast, the TDBO introduces only minimal improvements and, in some instances, degrades performance across several benchmarks. This could be attributed to the fact that, although randomness introduced by TDBO can enhance population diversity and help circumvent local optima, excessive randomness might cause the algorithm to deviate from its path, undermining its capacity to capitalize on valuable information. This, in turn, potentially diminishes its efficiency in achieving global optimality. On the other hand, both the PDBO and the GDBO significantly uplift DBO’s optimization capability,

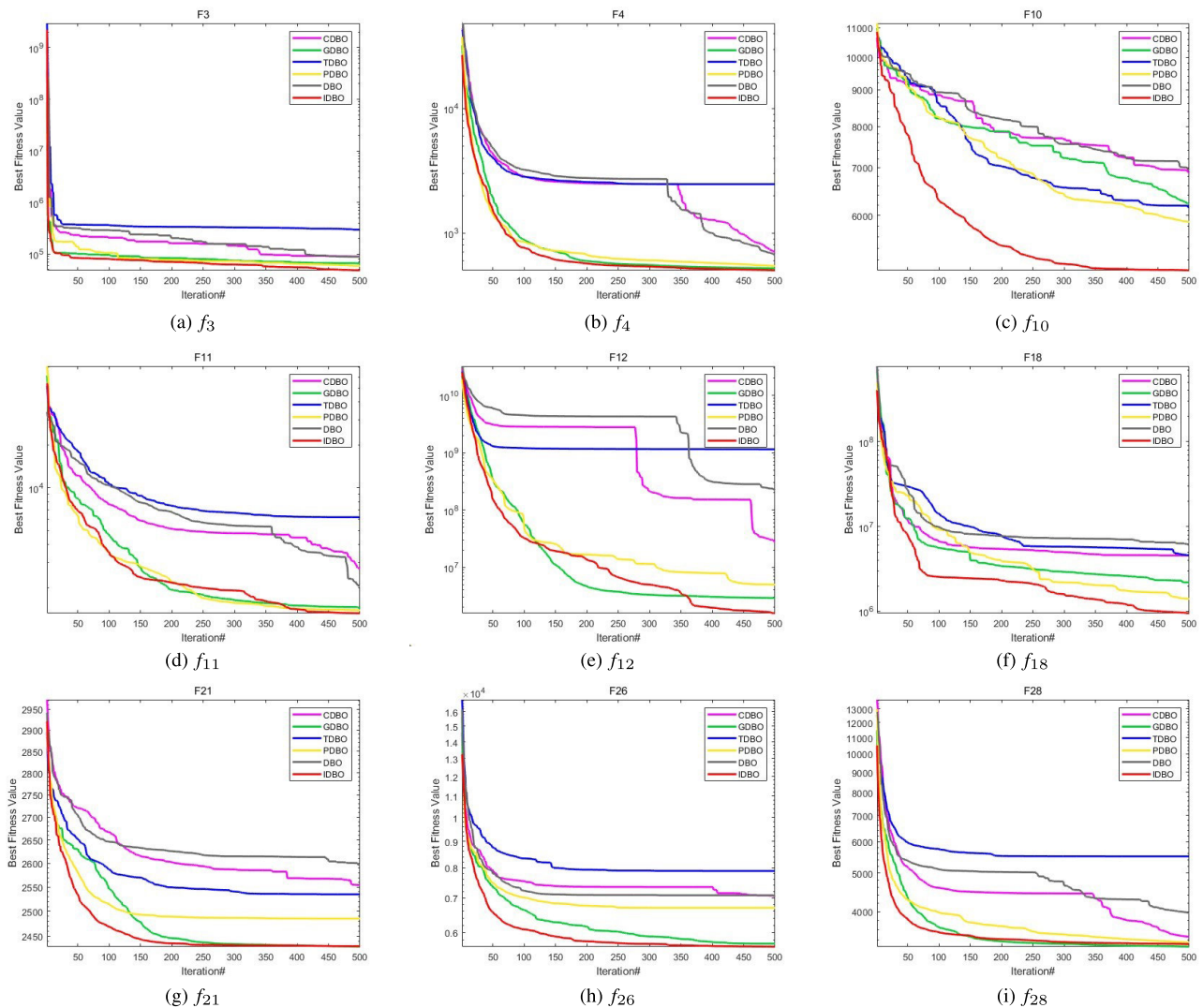


FIGURE 3. Comparison of different improvement strategies.

demonstrating steady performance across a spectrum of test functions. Notably, in complex mixed-mode functions, the performance enhancements from PDBO and GDBO are even more evident, underscoring the success of the introduced global search and population update strategies. The IDBO algorithm, encapsulating these four strategic advancements, consistently excels in optimization tasks. Overall, IDBO adeptly navigates the hurdles of local optima and premature convergence, marking a notable leap in convergence velocity and accuracy. These findings illuminate paths for the continued refinement and practical deployment of the DBO algorithm.

### C. COMPARATIVE ANALYSIS OF IDBO AND OTHER ALGORITHM

The CEC2017 series of functions is a valuable tool for simulating the complexity of real-world problems, offering valuable insights for the development of new algorithms.

In this study, we compared the algorithm proposed in this research with seven other competitive algorithms: GSA [33], GWO [34], WOA [6], HHO [35], DMO [36], COA [37] and the original DBO algorithm [27]. To ensure consistency in the experimental setup, parameters such as the number of runs, population size, test dimension, and maximum number of iterations were kept consistent with those detailed in Section IV-A. The experiments were conducted with 30 independent runs, and the best fitness values were recorded for each set of trials. Table 2 presents the best fitness average (Ave) and standard deviation (Std) obtained from 30 repeated experiments for GSA [33], GWO [34], WOA [6], HHO [35], DMO [36], COA [37] and the original DBO algorithm [27] and IDBO. A comprehensive statistical analysis highlights the superiority of the IDMO algorithm. The first line compiles the Friedman average scores of all algorithms, reflecting their performance, while the second line provides information on the overall rankings through the

final Friedman rankings. These tables prominently showcase the top results, underscoring their exceptional performance. Within each set of test functions, the algorithm with the lowest average and standard deviation is highlighted in bold, indicative of its superior performance.

In a performance comparison experiment against seven other prominent algorithms using the CEC2017 test set, IDBO emerged as the top-performing algorithm. It secured the leading position in ten out of the 30 test functions and achieved near-optimal scores on the remaining test functions, without recording the worst scores on any of the tested functions. From a statistical standpoint, the IDBO leads the ranking with an average position of 2.46 in the Friedman rankings. This result indicates that IDBO delivers superior overall performance compared to the other algorithms tested, showcasing its adaptability across diverse problem scenarios. In summary, the results affirm that the IDBO algorithm excels in various tests, both in terms of individual test function performance and the comprehensive statistical ranking across all tested functions. These findings underscore that IDBO is an efficient and dependable optimization algorithm well-suited for diverse optimization challenges.

The Wilcoxon rank sum test was conducted to evaluate the statistical significance of the IDBO in comparison to other algorithms, setting the significance threshold at 5%. A test outcome of  $p < 5\%$  indicates a statistically significant difference between IDBO and the comparative algorithm, whereas  $p \geq 5\%$  implies no significant difference. The interpretation of these results is based on the rank sum test: symbols '+', '-', and '=' represent IDBO's optimization performance as superior, inferior, or equivalent to other algorithms, respectively. The comprehensive results of the Wilcoxon rank sum test are detailed in Table 2.

The statistical results from the Wilcoxon rank sum test reveal that IDBO outperforms the 7 advanced algorithms in the CEC2017 function suite, showcasing its superior performance. These results also attest to the robustness of IDBO.

To further analyze the convergence speed and iterative process of the aforementioned algorithms, 7 different types of test functions were selected for comparison. As shown in Fig.4, IDBO outperforms other algorithms in both convergence speed and accuracy, reveals that IDBO consistently achieved the quickest convergence rate and maintained the highest level of accuracy in convergence. These results underscore IDBO's exceptional proficiency in both global exploration and local exploitation. Collectively, these findings solidify the effectiveness and superiority of IDBO as an optimization tool.

In summary, IDBO emerges as an intelligent optimization algorithm capable of consistently achieving high-quality solutions. It exhibits robust stability, rapid convergence, high precision in convergence, and an impressive capability to avoid local optima.

## V. IDBO ALGORITHM PRACTICAL ENGINEERING APPLICATION

This section is dedicated to examining the real-world application of the IDBO algorithm, specifically its utilization in three-dimensional trajectory planning for UAVs. To gauge the performance of the IDBO algorithm, simulations of UAV flight trajectories in intricate mountainous terrains were conducted. These simulations vividly illustrate the considerable potential of the IDBO algorithm in addressing challenging path planning tasks. The outcomes of the simulations validate the algorithm's proficiency in navigating complex environments, affirming its appropriateness for path planning applications.

### A. UAVS PATH PLANNING MODEL

#### 1) FLIGHT DISTANCE COST

In UAV trajectory planning, the cost associated with flight distance primarily relates to fuel consumption during the UAV's journey. Assuming the UAV achieves and maintains a constant operating speed throughout its mission, fuel consumption becomes directly proportional to the total flight distance covered by the UAV [64]. To quantify this cost, a formula has been devised that accurately reflects this relationship, considering the total distance traversed by the drone. The calculation formula for the flight distance cost is as follows:

$$f_{\text{range}} = \frac{\varepsilon}{Q_r} \sum_{i=1}^n L_i \quad (16)$$

In the equation:  $\varepsilon$  is the fuel consumption per unit flight distance,  $Q_r$  is the total amount of fuel carried by the UAVs, and  $L_i$  represents the length of the  $i$ th flight segment.

#### 2) FLIGHT ALTITUDE COST

The likelihood of a drone being affected by low temperatures increases with altitude, To manage these risks, maximum ( $h_{\text{max}}$ ) and minimum ( $h_{\text{min}}$ ) flight altitudes are set. Assuming the UAVs fly at an altitude  $h_i$ , the cost function for flight altitude can be expressed as:

$$f_{H_i} = \begin{cases} \frac{h_i - h_{\text{min}}}{h_{\text{max}} - h_{\text{min}}}, & h_{\text{min}} < h_i < h_{\text{max}} \\ \infty & , \text{ others} \end{cases} \quad (17)$$

$$f_{\text{altitude}} = \frac{1}{n} \sum_{i=1}^n f_{H_i} \quad (18)$$

#### 3) FLIGHT RISK COST

The UAVs' flight path might pass through areas with various risks, such as difficult terrain, extreme weather conditions, or military operations. The risk cost is evaluated based on the proximity of the UAVs to these risk points during flight. If a particular flight path segment  $L_i$  is divided into  $m$  sub-segments and approaches a risk point  $k$ , the risk cost function for that segment due to the risk point is calculated

**TABLE 2.** Test results for CEC 2017.

ID	Metric	GSA	GWO	WOA	HHO	DMO	COA	DBO	IDBO
CEC2017-F1	Std	1.3612E+10	3.1435E+09	4.7457E+09	3.9087E+08	4.1563E+07	1.0223E+09	2.5265E+08	<b>1.2857E+07</b>
	Ave	3.3045E+09	1.8336E+09	1.5875E+09	2.5325E+08	<b>2.3362E+07</b>	1.2226E+09	1.6476E+08	2.8083E+07
CEC2017-F3	Std	9.7678E+04	6.4089E+04	2.9345E+05	5.8274E+04	1.6488E+05	1.0587E+05	8.6821E+04	<b>4.9823E+04</b>
	Ave	1.0056E+04	1.5036E+04	7.3116E+04	<b>7.0365E+03</b>	2.5945E+04	2.4641E+04	1.4663E+04	8.9923E+03
CEC2017-F4	Std	3.9896E+03	6.5658E+02	1.3205E+03	7.2562E+02	5.7146E+02	6.2962E+02	6.9963E+02	<b>5.3425E+02</b>
	Ave	1.3356E+03	1.2477E+02	4.0708E+02	1.0396E+02	<b>2.5298E+01</b>	8.0610E+01	1.8096E+02	4.9288E+01
CEC2017-F5	Std	7.3862E+02	<b>6.3041E+02</b>	8.7490E+02	7.7226E+02	7.4190E+02	7.4696E+02	7.6262E+02	6.8647E+02
	Ave	2.2298E+01	4.0225E+01	6.8836E+01	2.7457E+01	<b>1.4770E+01</b>	5.0865E+01	7.6262E+01	3.8863E+01
CEC2017-F6	Std	6.6365E+02	6.1536E+02	6.8187E+02	6.6869E+02	<b>6.0515E+02</b>	6.5632E+02	6.5266E+02	6.4425E+02
	Ave	4.0863E+00	7.0285E+00	1.3007E+01	6.9608E+00	<b>1.0663E+00</b>	1.2625E+01	1.1052E+01	9.6114E+00
CEC2017-F7	Std	1.1133E+03	<b>9.0596E+02</b>	1.3296E+03	1.3107E+03	9.9347E+02	1.2174E+03	1.0196E+03	1.0875E+03
	Ave	7.4159E+01	5.8867E+01	9.5547E+01	6.2390E+02	<b>1.1548E+01</b>	1.1596E+02	9.5292E+01	1.0869E+02
CEC2017-F8	Std	9.6556E+02	<b>9.1536E+02</b>	1.0815E+03	9.8460E+01	1.0564E+03	9.7217E+02	1.0263E+03	9.4259E+02
	Ave	2.4269E+01	4.8445E+01	5.4765E+01	2.5314E+01	<b>1.3136E+01</b>	2.4669E+01	5.7492E+01	3.4156E+01
CEC2017-F9	Std	4.7348E+03	2.4378E+03	1.1033E+04	8.4865E+03	<b>2.3455E+03</b>	7.8832E+03	6.3044E+03	4.1694E+03
	Ave	<b>5.2696E+02</b>	9.1668E+02	3.4058E+03	1.2837E+03	5.8942E+02	1.8518E+03	1.9765E+03	1.5796E+03
CEC2017-F10	Std	5.4469E+03	<b>5.3375E+03</b>	7.5399E+03	6.3478E+03	8.6248E+03	6.3739E+03	6.3296E+03	5.3414E+03
	Ave	6.3963E+02	1.7935E+03	5.1877E+02	8.6049E+02	<b>3.7091E+02</b>	8.3818E+02	1.1336E+03	8.0245E+02
CEC2017-F11	Std	7.4578E+03	2.9987E+03	9.9704E+03	1.5458E+03	1.6904E+03	1.6755E+03	1.7657E+03	<b>1.3236E+03</b>
	Ave	1.6896E+03	1.6862E+03	3.7906E+03	1.7761E+02	1.7155E+02	2.5948E+02	4.2348E+02	<b>8.3475E+01</b>
CEC2017-F12	Std	2.3563E+09	1.0941E+08	3.9408E+08	1.0632E+08	1.9066E+07	1.3647E+07	6.7352E+07	<b>5.5969E+06</b>
	Ave	9.1477E+08	1.1063E+08	2.0880E+08	1.0145E+08	7.7494E+06	1.3836E+07	8.0315E+07	<b>6.5863E+06</b>
CEC2017-F13	Std	1.9795E+08	3.4352E+07	1.6755E+07	1.2995E+06	<b>2.9656E+04</b>	1.8794E+05	1.3796E+07	9.0147E+04
	Ave	5.1596E+08	8.5614E+07	1.9064E+07	1.3462E+06	<b>2.9012E+04</b>	1.0171E+05	3.0033E+07	3.2699E+05
CEC2017-F14	Std	1.6493E+06	5.8365E+05	1.6479E+06	1.3210E+06	<b>1.7315E+05</b>	3.5118E+05	3.7752E+05	4.8887E+05
	Ave	5.2757E+05	9.1035E+05	1.3715E+06	1.0852E+06	<b>1.2487E+05</b>	4.6028E+05	9.4463E+05	5.7136E+05
CEC2017-F15	Std	1.6446E+04	8.0374E+05	7.2819E+06	1.3375E+05	<b>8.4163E+03</b>	2.9205E+04	6.6460E+04	1.0566E+04
	Ave	<b>4.8868E+03</b>	1.4752E+06	1.2592E+07	7.3404E+04	5.6625E+03	3.0788E+04	4.7760E+04	1.1811E+04
CEC2017-F16	Std	4.0376E+03	<b>2.7269E+03</b>	4.4493E+03	3.8389E+03	3.5445E+03	3.0611E+03	3.4358E+03	2.9125E+03
	Ave	6.4569E+02	3.3178E+02	8.6057E+02	5.3168E+02	<b>2.0263E+02</b>	3.1326E+02	4.7496E+02	3.6536E+02
CEC2017-F17	Std	3.0258E+03	<b>2.1765E+03</b>	2.7962E+03	2.6845E+03	2.4688E+03	2.3434E+03	2.6948E+03	2.4978E+03
	Ave	3.1063E+02	1.7645E+02	2.3772E+02	2.4636E+02	<b>1.2245E+02</b>	2.4828E+02	2.3261E+02	2.8436E+02
CEC2017-F18	Std	3.1185E+06	1.3414E+06	1.2818E+07	3.4249E+06	6.5263E+06	3.7526E+06	3.7092E+06	<b>9.8759E+05</b>
	Ave	2.9425E+06	1.5236E+06	1.3691E+07	3.8599E+06	3.8118E+06	4.2028E+06	4.2863E+06	<b>1.1446E+06</b>
CEC2017-F19	Std	1.1247E+06	1.0455E+06	2.4961E+07	1.3008E+06	1.3469E+04	3.7499E+04	5.4985E+06	<b>1.0088E+04</b>
	Ave	1.4036E+06	1.7448E+06	2.6276E+07	9.3436E+05	1.2465E+04	5.7254E+04	1.0252E+07	<b>7.9677E+03</b>
CEC2017-F20	Std	3.0735E+03	<b>2.5166E+03</b>	2.9001E+03	2.8737E+03	2.8448E+03	2.7263E+03	2.7866E+03	2.6148E+03
	Ave	2.2925E+02	1.7535E+02	2.1158E+02	2.0772E+02	<b>9.4162E+01</b>	2.4252E+02	2.2635E+02	2.3269E+02
CEC2017-F21	Std	2.6699E+03	<b>2.4298E+03</b>	2.6596E+03	2.5871E+03	2.5336E+03	2.4841E+03	2.5649E+03	2.4636E+03
	Ave	6.1086E+01	4.5584E+01	6.2663E+01	5.2428E+01	<b>1.2355E+01</b>	4.8863E+01	4.9014E+01	4.1378E+01
CEC2017-F22	Std	7.5265E+03	5.0705E+03	7.8772E+03	7.0319E+03	6.3164E+03	4.2356E+03	5.4625E+03	<b>2.5956E+03</b>
	Ave	<b>4.6114E+02</b>	2.2256E+03	2.1145E+03	1.6962E+03	2.5414E+03	2.5090E+03	2.1862E+03	1.0477E+03
CEC2017-F23	Std	3.9915E+03	<b>2.7850E+03</b>	3.1596E+03	3.3052E+03	2.8825E+03	2.9063E+03	3.0215E+03	2.9644E+03
	Ave	1.8318E+02	3.5336E+01	1.0264E+02	1.4110E+02	<b>1.7496E+01</b>	8.7044E+01	1.0949E+02	9.0522E+01
CEC2017-F24	Std	3.9587E+03	<b>2.9966E+03</b>	3.2763E+03	3.4965E+03	3.0462E+03	3.0463E+03	3.1963E+03	3.1115E+03
	Ave	2.2665E+02	8.0905E+01	1.0585E+02	1.6835E+02	<b>1.2545E+01</b>	7.3649E+01	7.9352E+01	1.1163E+02
CEC2017-F25	Std	3.3136E+03	3.0208E+03	3.2175E+03	3.0285E+03	<b>2.9263E+03</b>	2.9941E+03	3.0014E+03	2.9469E+03
	Ave	1.5885E+02	6.5509E+01	7.0062E+01	3.6541E+01	<b>1.3119E+01</b>	5.6625E+01	6.2662E+01	3.8078E+01
CEC2017-F26	Std	8.9365E+03	<b>5.1560E+03</b>	8.5341E+03	8.3319E+03	5.9715E+03	6.9836E+03	6.8705E+03	6.0098E+03
	Ave	7.3625E+02	5.1855E+02	8.5152E+02	7.9761E+02	<b>1.4202E+02</b>	9.6369E+02	7.3163E+02	1.5365E+03
CEC2017-F27	Std	5.3847E+03	3.2845E+03	3.4809E+03	3.6101E+03	<b>3.2490E+03</b>	3.3662E+03	3.3400E+03	3.3063E+03
	Ave	3.9736E+02	4.0192E+01	1.2697E+02	1.6706E+02	<b>9.0308E+00</b>	5.5996E+01	6.3263E+01	6.3575E+01
CEC2017-F28	Std	4.5525E+03	3.4563E+03	3.8461E+03	3.4735E+03	3.3305E+03	3.3932E+03	3.5565E+03	<b>3.3196E+03</b>
	Ave	2.9739E+02	7.9878E+01	1.8568E+02	6.1459E+01	<b>2.6604E+01</b>	1.1101E+02	4.7194E+02	3.2845E+01
CEC2017-F29	Std	6.4515E+03	<b>4.0036E+03</b>	5.5266E+03	4.8516E+03	4.5309E+03	4.3462E+03	4.4762E+03	4.3763E+03
	Ave	5.0369E+02	2.3715E+02	6.5298E+02	3.6115E+02	<b>1.5400E+02</b>	2.7826E+02	3.7452E+02	2.7977E+02
CEC2017-F30	Std	2.2836E+07	1.6102E+07	6.6764E+07	1.3405E+07	3.7994E+05	1.0660E+06	4.1314E+06	<b>1.5789E+05</b>
	Ave	3.0352E+07	1.5188E+07	8.3263E+07	1.6304E+07	5.9193E+05	1.0369E+06	6.9136E+06	<b>2.3056E+05</b>
Wilcoxon(+/-/-)		3/0/26	4/0/25	0/0/29	2/0/27	6/0/23	9/0/18	6/0/23	0/29/0
Friedman Mean		6.36	3.16	6.87	5.41	3.55	3.72	4.47	<b>2.46</b>
Ranking		7	2	8	6	3	4	5	<b>1</b>

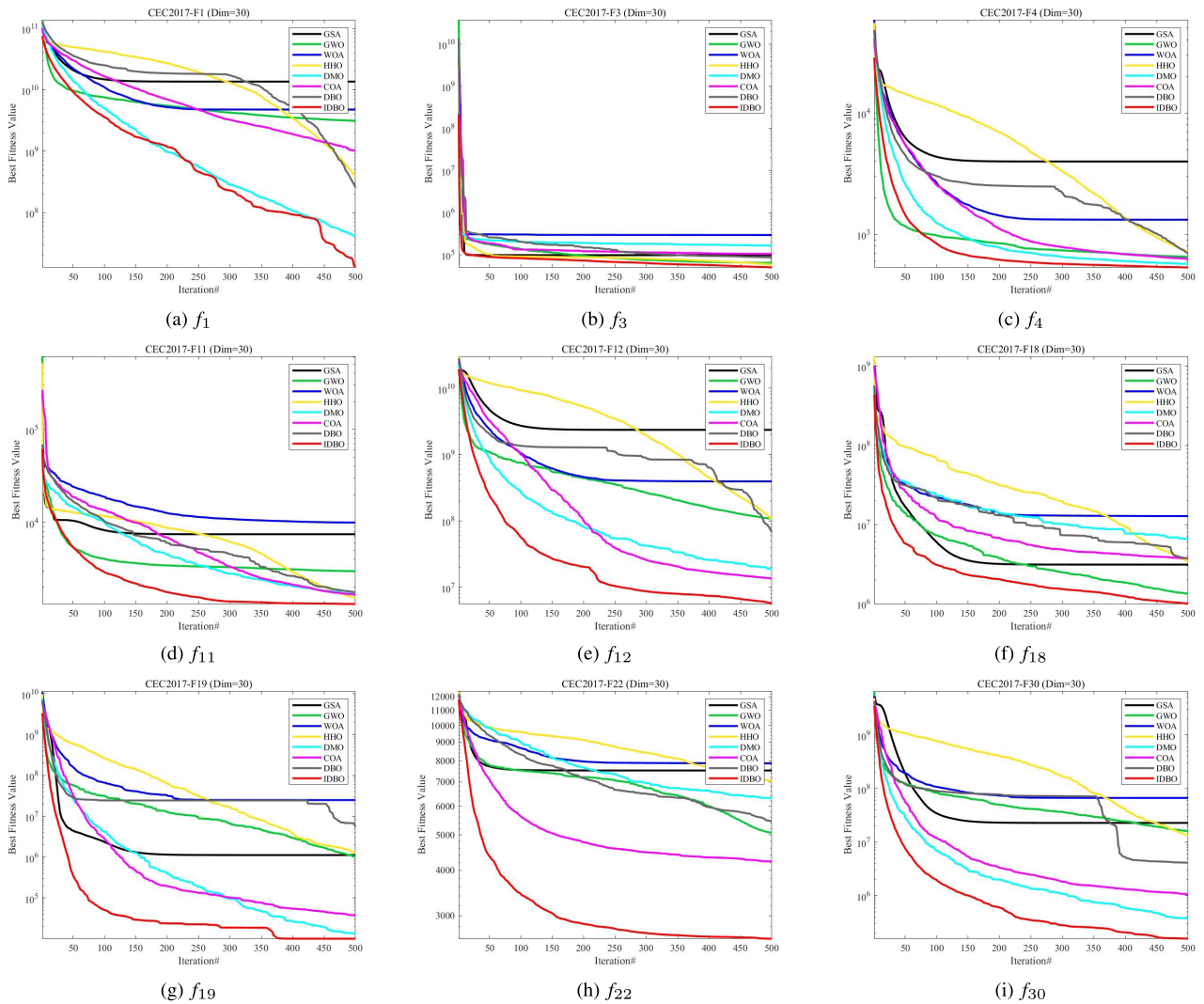


FIGURE 4. Comparison of convergence curves with different algorithms.

accordingly. The flight risk cost function for segment  $L_i$  due to risk point  $k$  can be expressed as:

$$f_{n_i,k} = \frac{1}{m} (P_k(d_{k,1}) + P_k(d_{k,2}) + \dots + P_k(d_{k,m})) \quad (19)$$

In the equation:  $n_i$  represents the  $i$ th segment  $L_i$  in the trajectory planning;  $P_k$  is the destruction probability of the UAVs by the  $k$ -th threat point;  $d_{k,m}$  is the distance from the  $k$ -th threat point to the  $m$ -th sub-segment in segment  $L_i$ . The total threat cost incurred by the UAVs's flight path planning can be expressed as:

$$f_{risk} = \frac{1}{n} \frac{1}{k} \sum_{i=1}^n \sum_{k=1}^k f_{n_i,k} \quad (20)$$

#### 4) PERFORMANCE MEASUREMENT FUNCTION OF MULTI-UAVS COLLABORATIVE PATH PLANNING

The overall trajectory planning for UAVs considers three main costs: flight distance, flight altitude, and flight risk.

Each of these costs is assigned a weight coefficient ( $\omega_1$ ,  $\omega_2$ , and  $\omega_3$ , respectively). As a result, the objective function for UAV trajectory planning is established as a weighted sum of distinct cost components. This formulation is designed to achieve an equilibrium among diverse factors to ascertain the most efficient flight path. The objective function for trajectory planning involving UAVs is defined in the following manner:

$$f = \omega_1 f_{range} + \omega_2 f_{altitude} + \omega_3 f_{risk} \quad (21)$$

### B. SIMULATION AND ANALYSIS OF UAVS 3D PATH PLANNING

#### 1) ALGORITHM APPLICATION AND EXPERIMENTAL SIMULATION

To showcase the practical application of our proposed algorithm, we performed simulation tests for 3D UAV path planning using the Matlab platform. The scenario for the

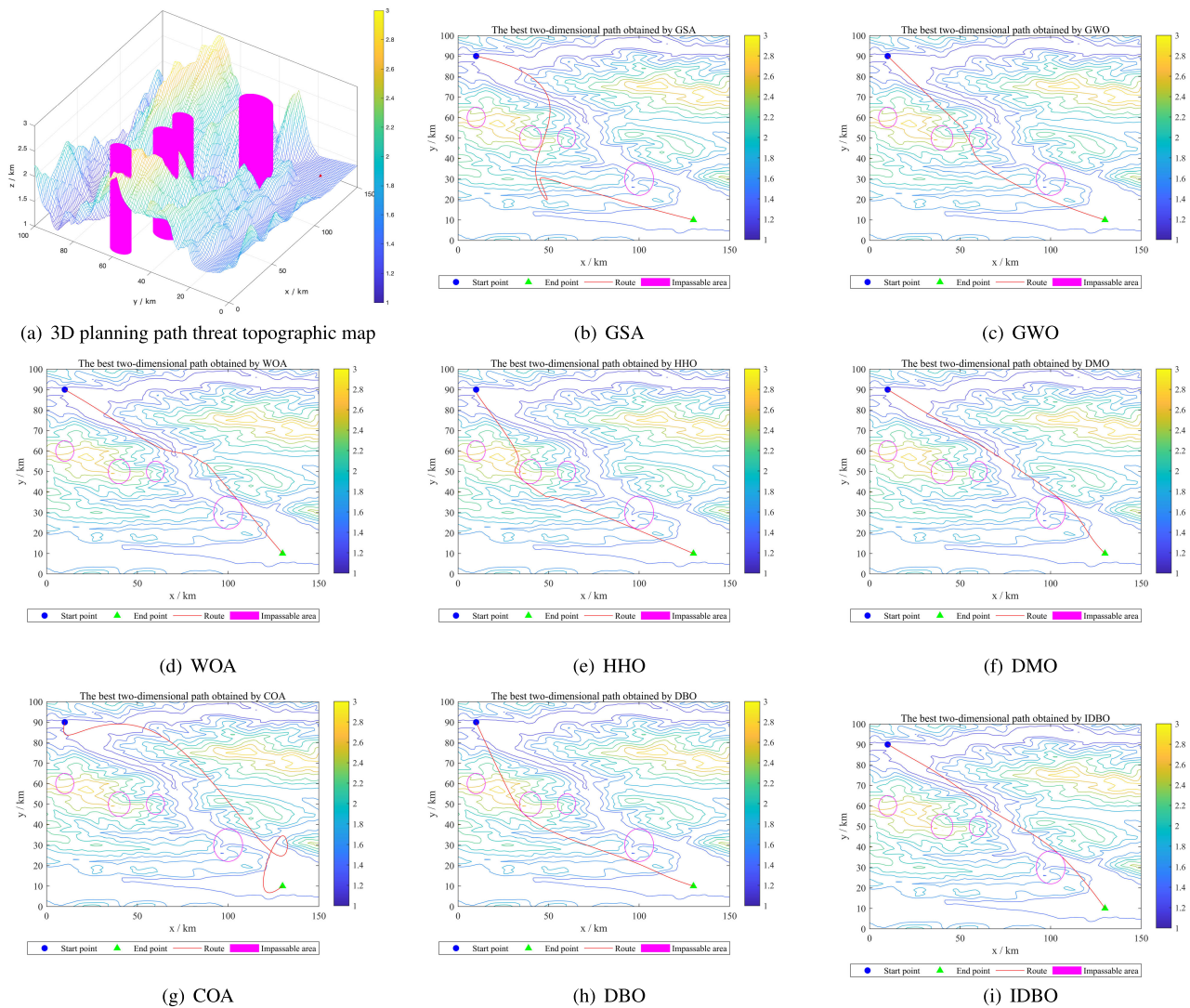


FIGURE 5. 3D topographic map and best two-dimensional path planning by each algorithm.

simulation was set in a space measuring 100 km by 150 km by 3km. The starting point of the UAVs was designated at coordinates (10, 90), with the target point at (130, 10). Table 3 in our documentation provides detailed two-dimensional coordinates for various risk areas within this scenario. In our simulations, these risk areas are visually represented in pink, as illustrated in Fig.5(a).

For comparative purposes, the original DBO algorithm was also applied to the same three-dimensional path planning problem. The parameters for DBO were established as follows: a population size (N) of 30 and a maximum iteration count (T) of 200. For the parameter configuration of the optimized IDBO algorithm, please see Section IV-A. The results from these simulations are visually depicted in our figures: Fig.5(b)-Fig.4(i) presents the outcomes for the two-dimensional track plan, and Fig.6(a)-Fig.5(h) displays the results for the three-dimensional track plan. Additionally, Fig.6(i) offers a comparative analysis of the convergence

TABLE 3. 2-D coordinate parameters of risk area.

Risky Area	Center Point	Risk Radius
1	(10,60)	5
2	(40,50)	6
3	(60,50)	5
4	(100,30)	8

curves for six different algorithms. This comprehensive set of simulations and analyses underscores the effectiveness of the IDBO algorithm in navigating complex environments and its potential advantages over other algorithms in UAV path planning.

C. ANALYSIS OF SIMULATION RESULT

Combining the paths given by different algorithms in Fig.5 and Fig.6 can be analytically concluded: The analysis of

TABLE 4. Statistics of UAVs three-dimensional path planning results.

Algorithm	Best Cost	Worst Cost	Average Cost	Standard Deviation	Initial bestfitness	Average Run Time	Wilcoxon(+/=/-)
GSA	210.9249	210.9249	210.9249	210.9249	2753.8532	210.9249	(+)
GWO	73.0417	73.5373	73.1362	0.2305	161.4434	6.7532	(=)
WOA	73.5403	119.9123	81.8179	16.8308	152.3234	6.7574	(+)
HHO	75.5641	118.7902	94.0184	18.5185	155.7235	16.9046	(+)
DMO	72.9696	73.1472	73.2133	0.0661	151.5538	12.7946	(=)
COA	114.41192	117.2672	115.8557	1.2993	127.5435	11.2699	(+)
DBO	74.0339	112.231	101.6439	16.3686	137.6232	6.8812	(+)
IDBO	73.0307	73.1808	73.0605	0.0603	117.6811	8.1894	--

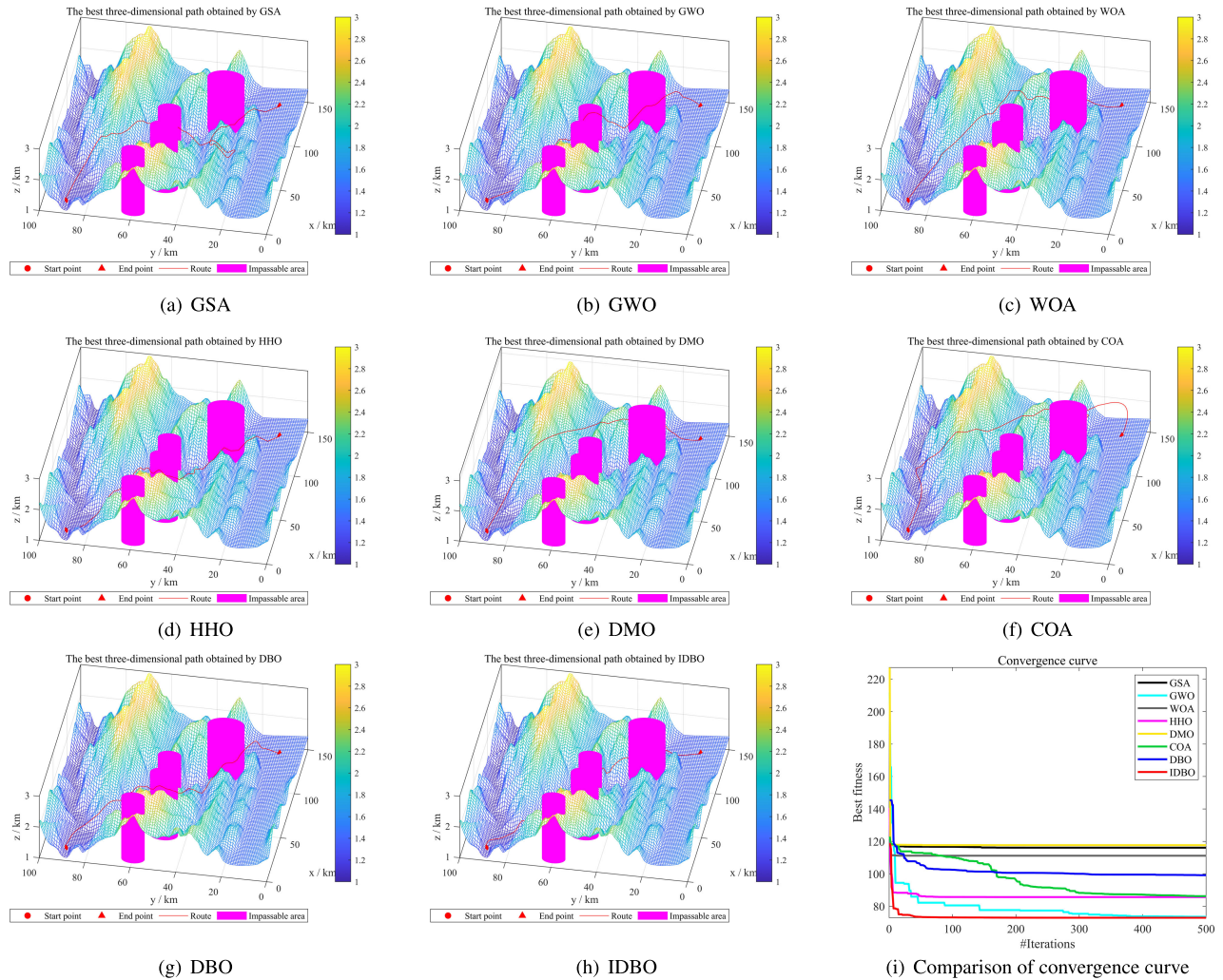


FIGURE 6. Best 3D path planning by each algorithm and comparison of convergence curve.

experimental results indicates that the flight paths generated by the DBO algorithm tend to be longer, resulting in increased fuel consumption. More critically, its flight trajectories are closer to risk zones, posing significant safety hazards and increasing the likelihood of flight accidents. The flight paths of GSA and COA algorithms not only cross risk areas but also exhibit disordered trajectories, substantially increasing flight distance and risk. Although the flight routes planned by HHO and WOA algorithms avoid risk zones, their paths

are not smooth and have significant turns, which increases the flight distance. In contrast, the optimized IDBO algorithm successfully mitigates these issues. While GWO and DMO planned routes appear similar on a 2D track plan, the 3D track plan shows that the flight altitude planned by the IDBO algorithm is lower, allowing the UAVs to fly closer to mountainous terrain. This effectively avoids risks associated with high-altitude cold and oxygen depletion. Overall, IDBO not only shortens the flight distance but also effectively

avoids risk areas with smoother flight routes, significantly reducing the risk associated with task execution.

Observing the changes in the convergence curves, the improved IDBO algorithm demonstrates the lowest initial best fitness value and the quickest convergence rate. From this, we can conclude that the improved IDBO algorithm not only starts from a more optimal position but also reaches its target solution more efficiently than the compared algorithms, highlighting its superior optimization capability. Overall, for the UAVs three-dimensional path planning, the IDBO algorithm, improved by the integration of multiple strategies, performed superiorly, further validating the achievements of the enhancement strategies proposed in this study.

Under the same testing conditions, this study conducted 30 independent simulation experiments for six algorithms. A statistical analysis is also performed on the composite cost models of these two algorithms, with the related statistical data listed in Table 4.

The analysis of eight sets of experimental data consistently demonstrates that the IDBO algorithm surpasses other algorithms across various metrics, including optimal cost, worst cost, average cost, and standard deviation. These results emphasize the exceptional optimization performance of the IDBO algorithm, especially in the domain of three-dimensional path planning, where it showcases increased stability in optimization outcomes. Regarding average running time, all other algorithms, except GSA, exhibit similar performance, with IDBO being notably efficient in terms of runtime. Wilcoxon rank sum test results show that IDBO shows a statistically significant performance improvement in most comparison algorithms, except that the P-values of GWO and DMO indicate that the performance is nearly the same as IDBO. Notably, the IDBO algorithm starts with a significantly lower initial optimal fitness when compared to other algorithms, indicating its proximity to the global optimum. This characteristic significantly diminishes the likelihood of the algorithm becoming trapped in local optima. The rapid convergence rate and advantageous initial positioning render the IDBO a dependable and stable choice for optimization tasks. This stability and reliability are crucial in practical applications. In conclusion, the IDBO algorithm has emerged as the favored choice for three-dimensional path planning in UAVs.

## VI. CONCLUSION

In this study, we conducted an exhaustive analysis of the DBO algorithm, identifying its computational challenges and limitations. To address these, we proposed and integrated four strategic enhancements: cubic chaos mapping, a novel global exploration strategy, the adaptive t-distribution perturbation technique, and a dynamic population update strategy. This integration not only augments the DBO algorithm's global search capabilities but also refines its precision during local optimization phases, leading to significantly faster convergence rates.

The efficacy of the IDBO was rigorously evaluated against advanced optimization algorithms such as GSA, GWO, WOA, HHO, DMO, COA and the original DBO algorithm. Benchmarking on the CEC2017 dataset revealed that IDBO consistently surpasses these counterparts, showcasing remarkable improvements in convergence speed, stability, and robustness. These results underscore IDBO's superior optimization performance, particularly highlighted by its application in complex real-world problems such as three-dimensional path planning for UAVs. In this study, IDBO demonstrated exceptional optimization outcomes, further substantiating the algorithm's practical applicability and potential for broader scenario applications.

The ability of IDBO to outperform existing algorithms in challenging scenarios highlights its potential as a general tool for solving a wide range of optimization problems. This adaptability makes it promising to be applied to more complex constraint problems, indicating its practical value in fields that require both high accuracy and high efficiency.

Future research will aim to further this potential. We're particularly interested in exploring the application of IDBO in diverse domains, such as logistics, healthcare, and energy management, where its optimization capabilities can be leveraged to address complex challenges. Additionally, we plan to refine IDBO's algorithmic structure to enhance its performance further and explore its integration with other computational techniques, such as machine learning models, to create hybrid approaches. This future work aims not just to push the boundaries of algorithmic optimization but also to contribute practical solutions to the pressing problems faced in various industries.

## REFERENCES

- [1] A. Chowdhury and D. De, "RGSO-UAV: Reverse glowworm swarm optimization inspired UAV path-planning in a 3D dynamic environment," *Ad Hoc Netw.*, vol. 140, Mar. 2023, Art. no. 103068.
- [2] A. A. Saadi, A. Soukane, Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, "UAV path planning using optimization approaches: A survey," *Arch. Comput. Methods Eng.*, vol. 29, no. 6, pp. 4233–4284, Oct. 2022.
- [3] M. Jones, S. Djahel, and K. Welsh, "Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey," *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–39, Nov. 2023.
- [4] J. Wang, Y. Li, R. Li, H. Chen, and K. Chu, "Trajectory planning for UAV navigation in dynamic environments with matrix alignment Dijkstra," *Soft Comput.*, vol. 26, no. 22, pp. 12599–12610, Nov. 2022.
- [5] X. Yu, N. Jiang, X. Wang, and M. Li, "A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning," *Expert Syst. Appl.*, vol. 215, Apr. 2023, Art. no. 119327.
- [6] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
- [7] X. Yang, X. Hao, T. Yang, Y. Li, Y. Zhang, and J. Wang, "Elite-guided multi-objective cuckoo search algorithm based on crossover operation and information enhancement," *Soft Comput.*, vol. 27, no. 8, pp. 4761–4778, Apr. 2023.
- [8] M. A. Elaziz, S. Lu, and S. He, "A multi-leader whale optimization algorithm for global optimization and image segmentation," *Expert Syst. Appl.*, vol. 175, Aug. 2021, Art. no. 114841.
- [9] L. Zhang, Y. Zhang, and Y. Li, "Mobile robot path planning based on improved localized particle swarm optimization," *IEEE Sensors J.*, vol. 21, no. 5, pp. 6962–6972, Mar. 2021.



- [10] F. Wei, J. Li, and Y. Zhang, "Improved neighborhood search whale optimization algorithm and its engineering application," *Soft Comput.*, vol. 27, no. 23, pp. 17687–17709, Dec. 2023.
- [11] W. Sung, H. Chung, and K. Chang, "Agricultural monitoring system based on ant colony algorithm with centre data aggregation," *IET Commun.*, vol. 8, no. 7, pp. 1132–1140, May 2014.
- [12] S. Sudhakar, V. Vijayakumar, C. S. Kumar, V. Priya, L. Ravi, and V. Subramaniaswamy, "Unmanned aerial vehicle (UAV) based forest fire detection and monitoring for reducing false alarms in forest-fires," *Comput. Commun.*, vol. 149, pp. 1–16, Jan. 2020.
- [13] J. Horyna, T. Baca, V. Walter, D. Albani, D. Hert, E. Ferrante, and M. Saska, "Decentralized swarms of unmanned aerial vehicles for search and rescue operations without explicit communication," *Auto. Robots*, vol. 47, no. 1, pp. 77–93, Jan. 2023.
- [14] S. Mohammad, M. F. M. Jusof, N. A. M. Rizal, A. A. A. Razak, A. N. K. Nasir, R. M. T. R. Ismail, and M. A. Ahmad, "Elimination-dispersal sine cosine algorithm for a dynamic modelling of a twin rotor system," in *Proc. InECCE 5th Int. Conf. Electr., Control Comput. Eng.* Singapore: Springer, 2019, pp. 167–178.
- [15] G. Amponis, T. Lagkas, K. Tsiknas, P. Radoglou-Grammatikis, and P. Sarigiannidis, "Introducing a new TCP variant for UAV networks following comparative simulations," *Simul. Model. Pract. Theory*, vol. 123, Feb. 2023, Art. no. 102708.
- [16] K. S. Suresh, R. Venkatesan, and S. Venugopal, "Mobile robot path planning using multi-objective genetic algorithm in industrial automation," *Soft Comput.*, vol. 26, no. 15, pp. 7387–7400, Aug. 2022.
- [17] X. Chai, Z. Zheng, J. Xiao, L. Yan, B. Qu, P. Wen, H. Wang, Y. Zhou, and H. Sun, "Multi-strategy fusion differential evolution algorithm for UAV path planning in complex environment," *Aerosp. Sci. Technol.*, vol. 121, Feb. 2022, Art. no. 107287.
- [18] R. K. Dewangan and P. Saxena, "Three-dimensional route planning for multiple unmanned aerial vehicles using salp swarm algorithm," *J. Experim. Theor. Artif. Intell.*, vol. 35, no. 7, pp. 1059–1078, Oct. 2023.
- [19] C. Ju, H. Ding, and B. Hu, "A hybrid strategy improved whale optimization algorithm for web service composition," *Comput. J.*, vol. 66, no. 3, pp. 662–677, Oct. 2021.
- [20] Z. Yu, Z. Si, X. Li, D. Wang, and H. Song, "A novel hybrid particle swarm optimization algorithm for path planning of UAVs," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22547–22558, Nov. 2022.
- [21] J. Chen, F. Ling, Y. Zhang, T. You, Y. Liu, and X. Du, "Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system," *Swarm Evol. Comput.*, vol. 69, Mar. 2022, Art. no. 101005.
- [22] A. Ait-Saadi, Y. Meraihi, A. Soukane, A. Ramdane-Cherif, and A. B. Gabis, "A novel hybrid chaotic Aquila optimization algorithm with simulated annealing for unmanned aerial vehicles path planning," *Comput. Electr. Eng.*, vol. 104, Dec. 2022, Art. no. 108461.
- [23] J. Sánchez-García, D. G. Reina, and S. L. Toral, "A distributed PSO-based exploration algorithm for a UAV network assisting a disaster scenario," *Future Gener. Comput. Syst.*, vol. 90, pp. 129–148, Jan. 2019.
- [24] G. Huang, Y. Cai, J. Liu, Y. Qi, and X. Liu, "A novel hybrid discrete grey wolf optimizer algorithm for multi-UAV path planning," *J. Intell. Robot. Syst.*, vol. 103, no. 3, pp. 1–18, Nov. 2021.
- [25] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [26] F. Neri, "Diversity management in memetic algorithms," in *Handbook Memetic Algorithms*. Cham, Switzerland: Springer, 2012, pp. 153–165.
- [27] J. Xue and B. Shen, "Dung beetle optimizer: A new meta-heuristic algorithm for global optimization," *J. Supercomput.*, vol. 79, no. 7, pp. 7305–7336, May 2023.
- [28] F. Zhu, G. Li, H. Tang, Y. Li, X. Lv, and X. Wang, "Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems," *Expert Syst. Appl.*, vol. 236, Feb. 2024, Art. no. 121219.
- [29] E. K. Ryu and W. Yin, *Large-Scale Convex Optimization: Algorithms & Analyses Via Monotone Operators*. Cambridge, U.K.: Cambridge Univ. Press, 2022.
- [30] H. T. Kahraman, S. Aras, and E. Gedikli, "Fitness-distance balance (FDB): A new selection method for meta-heuristic search algorithms," *Knowl.-Based Syst.*, vol. 190, Feb. 2020, Art. no. 105169.
- [31] B. Ozkaya, H. T. Kahraman, S. Duman, and U. Guvenc, "Fitness-distance-constraint (FDC) based guide selection method for constrained optimization problems," *Appl. Soft Comput.*, vol. 144, Sep. 2023, Art. no. 110479.
- [32] H. T. Kahraman, M. Kati, S. Aras, and D. A. Taşci, "Development of the natural survivor method (NSM) for designing an updating mechanism in metaheuristic search algorithms," *Eng. Appl. Artif. Intell.*, vol. 122, Jun. 2023, Art. no. 106121.
- [33] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.
- [34] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [35] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
- [36] J. O. Agushaka, A. E. Ezugwu, and L. Abualigah, "Dwarf mongoose optimization algorithm," *Comput. Methods Appl. Mech. Eng.*, vol. 391, Mar. 2022, Art. no. 114570.
- [37] H. Jia, H. Rao, C. Wen, and S. Mirjalili, "Crayfish optimization algorithm," *Artif. Intell. Rev.*, vol. 56, no. S2, pp. 1919–1979, Nov. 2023.
- [38] A. D. Boursianis, M. S. Papadopoulou, P. Diamantoulakis, A. Liopa-Tsakalidi, P. Barouchas, G. Salahas, G. Karagiannidis, S. Wan, and S. K. Goudos, "Internet of Things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: A comprehensive review," *Internet Things*, vol. 18, May 2022, Art. no. 100187.
- [39] Y. Wan, Y. Zhong, A. Ma, and L. Zhang, "An accurate UAV 3-D path planning method for disaster emergency response based on an improved multiobjective swarm intelligence algorithm," *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2658–2671, Apr. 2023.
- [40] Q. Shen, D. Zhang, M. Xie, and Q. He, "Multi-strategy enhanced dung beetle optimizer and its application in three-dimensional UAV path planning," *Symmetry*, vol. 15, no. 7, p. 1432, Jul. 2023.
- [41] W. Zilong and S. Peng, "A multi-strategy dung beetle optimization algorithm for optimizing constrained engineering problems," *IEEE Access*, vol. 11, pp. 98805–98817, 2023.
- [42] L. Li, L. Liu, Y. Shao, X. Zhang, Y. Chen, C. Guo, and H. Nian, "Enhancing swarm intelligence for obstacle avoidance with multi-strategy and improved dung beetle optimization algorithm in mobile robot navigation," *Electronics*, vol. 12, no. 21, p. 4462, Oct. 2023.
- [43] X.-R. Wei, W.-L. Bai, L. Liu, Y.-M. Li, and Z.-Y. Wang, "A hybrid dung beetle optimization algorithm with simulated annealing for the numerical modeling of asymmetric wave equations," *Appl. Geophys.*, pp. 1–15, Dec. 2023.
- [44] Z. Chang, J. Luo, Y. Zhang, and Z. Teng, "A mixed strategy improved dung beetle optimization algorithm and its application," *Tech. Rep.*, 2023.
- [45] S. Li and J. Li, "Chaotic dung beetle optimization algorithm based on adaptive t-distribution," in *Proc. IEEE 3rd Int. Conf. Inf. Technol., Big Data Artif. Intell. (ICIBA)*, vol. 3, May 2023, pp. 925–933.
- [46] H. Zhang, S. Yang, and D. Xu, "An improved optimization dung beetle algorithm," in *Proc. 4th Int. Symp. Comput. Eng. Intell. Commun. (ISCEIC)*, Aug. 2023, pp. 17–20.
- [47] R. Zhang, X. Chen, M. Li, and Y. Ding, "Multiple UAVs task assignment based on improved dung beetle optimizer," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*, Oct. 2023, pp. 779–785.
- [48] S. Jiaqi, T. Li, Z. Hongtao, L. Xiaofeng, and X. Tianying, "Adaptive multi-UAV path planning method based on improved gray wolf algorithm," *Comput. Electr. Eng.*, vol. 104, Dec. 2022, Art. no. 108377.
- [49] H. Chu, J. Yi, and F. Yang, "Chaos particle swarm optimization enhancement algorithm for UAV safe path planning," *Appl. Sci.*, vol. 12, no. 18, p. 8977, Sep. 2022.
- [50] G. Hu, J. Zhong, and G. Wei, "SaCHBA\_PDN: Modified honey badger algorithm with multi-strategy for UAV path planning," *Expert Syst. Appl.*, vol. 223, Aug. 2023, Art. no. 119941.
- [51] J.-S. Pan, J.-X. Lv, L.-J. Yan, S.-W. Weng, S.-C. Chu, and J.-K. Xue, "Golden eagle optimizer with double learning strategies for 3D path planning of UAV in power inspection," *Math. Comput. Simul.*, vol. 193, pp. 509–532, Mar. 2022.
- [52] Y. Jiang, Q. Wu, G. Zhang, S. Zhu, and W. Xing, "A diversified group teaching optimization algorithm with segment-based fitness strategy for unmanned aerial vehicle route planning," *Expert Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115690.

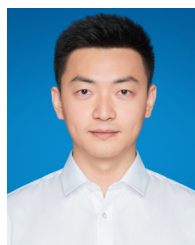
- [53] X. Zhang, X. Lu, S. Jia, and X. Li, "A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to UAV path planning," *Appl. Soft Comput.*, vol. 70, pp. 371–388, Sep. 2018.
- [54] Y. Chen, D. Pi, and Y. Xu, "Neighborhood global learning based flower pollination algorithm and its application to unmanned aerial vehicle path planning," *Expert Syst. Appl.*, vol. 170, May 2021, Art. no. 114505.
- [55] C. Yilmaz, E. Cengiz, and H. T. Kahraman, "A new evolutionary optimization algorithm with hybrid guidance mechanism for truck-multi drone delivery system," *Expert Syst. Appl.*, vol. 245, Jul. 2024, Art. no. 123115.
- [56] G. Hu, B. Du, X. Wang, and G. Wei, "An enhanced black widow optimization algorithm for feature selection," *Knowl.-Based Syst.*, vol. 235, Jan. 2022, Art. no. 107638.
- [57] J. Feng, J. Zhang, X. Zhu, and W. Lian, "A novel chaos optimization algorithm," *Multimedia Tools Appl.*, vol. 76, no. 16, pp. 17405–17436, Aug. 2017.
- [58] T. D. Rogers and D. C. Whitley, "Chaos in the cubic mapping," *Math. Model.*, vol. 4, no. 1, pp. 9–25, 1983.
- [59] J. J. Jui and M. A. Ahmad, "A hybrid metaheuristic algorithm for identification of continuous-time Hammerstein systems," *Appl. Math. Model.*, vol. 95, pp. 339–360, Jul. 2021.
- [60] M. Z. M. Tumari, M. A. Ahmad, M. H. Suid, M. R. Ghazali, and M. O. Tokhi, "An improved marine predators algorithm tuned data-driven multiple-node hormone regulation neuroendocrine-PID controller for multi-input–multi-output gantry crane system," *J. Low Freq. Noise, Vibrat. Act. Control*, vol. 42, no. 4, pp. 1666–1698, Dec. 2023.
- [61] M. Alamgeer, N. Alruwais, H. M. Alshahrani, A. Mohamed, and M. Assiri, "Dung beetle optimization with deep feature fusion model for lung cancer detection and classification," *Cancers*, vol. 15, no. 15, p. 3982, Aug. 2023.
- [62] J. B. McDonald and W. K. Newey, "Partially adaptive estimation of regression models via the generalized t distribution," *Econ. Theory*, vol. 4, no. 3, pp. 428–457, Dec. 1988.
- [63] W. Zhang and S. Liu, "Improved sparrow search algorithm based on adaptive t-distribution and golden sine and its application," *Microelectron. Comput.*, vol. 39, no. 3, pp. 17–24, 2022.
- [64] C. Huang, X. Zhou, X. Ran, J. Wang, H. Chen, and W. Deng, "Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning," *Eng. Appl. Artif. Intell.*, vol. 121, May 2023, Art. no. 105942.



**LIXIN LYU** received the B.S. degree in computer science and technology from Anhui Polytechnic University, Wuhu, China, in 2004, and the M.S. degree in computer application from Anhui University, Hefei, China, in 2010. He is currently pursuing the Ph.D. degree with the Technological University of the Philippines, Manila, Philippines.

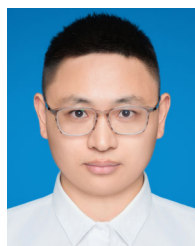
He is also a Professor with the School of Information and Artificial Intelligence, Anhui Business College. His research interests include

artificial intelligence, intelligent computing, big data technology, and robotics.



**HONG JIANG** was born in Anqing, Anhui, China, in 1994. He received the master's degree in educational technology from Anhui Normal University, Wuhu, China.

He is currently a Lecturer with the School of Information and Artificial Intelligence, Anhui Business College. His research interests include computer vision and virtual reality.



**FAN YANG** received the master's degree from Anhui Normal University, in 2014. He is currently pursuing the Ph.D. degree with the Technological University of the Philippines.

He is also a Lecturer with the School of Information and Artificial Intelligence, Anhui Business College. His research interests include machine learning, deep learning, and software engineering.

...