Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 7981670, 14 pages
https://doi.org/10.1155/2021/7981670

Research Article

A Hybrid Butterfly Optimization Algorithm for Numerical Optimization Problems

Huan Zhou[1] **Hao-Yu Cheng**[2] **Zheng-Lei Wei**[3] **Xin Zhao**[1] **An-Di Tang**[1] **and Lei Xie**[1]

[1] *Aviation Engineering School, Air Force Engineering University, Xi'an, China*
[2] *Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an, China*
[3] *China Aerodynamic Research and Development Center, Mianyang, China*

Correspondence should be addressed to Lei Xie; 310370487@qq.com

Received 1 November 2021; Accepted 22 November 2021; Published 24 December 2021

Academic Editor: Ahmed Mostafa Khalil

Copyright © 2021 Huan Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The butterfly optimization algorithm (BOA) is a swarm-based metaheuristic algorithm inspired by the foraging behaviour and information sharing of butterflies. BOA has been applied to various fields of optimization problems due to its performance. However, BOA also suffers from drawbacks such as diminished population diversity and the tendency to get trapped in local optimum. In this paper, a hybrid butterfly optimization algorithm based on a Gaussian distribution estimation strategy, called GDEBOA, is proposed. A Gaussian distribution estimation strategy is used to sample dominant population information and thus modify the evolutionary direction of butterfly populations, improving the exploitation and exploration capabilities of the algorithm. To evaluate the superiority of the proposed algorithm, GDEBOA was compared with six state-of-the-art algorithms in CEC2017. In addition, GDEBOA was employed to solve the UAV path planning problem. The simulation results show that GDEBOA is highly competitive.

1. Introduction

Optimization problems exist in all aspects of our society, including business, engineering, and science. An optimization problem is the process of finding the best value of decision variables that satisfy the maximum or minimum objective value without violating the constraints. With the development of science and technology in these days, the optimization problems we encounter have become increasingly complex. These real-world optimization problems often involve many decision variables, complex nonlinear constraints and nonconvexity, dynamic objective functions, and expensive computational costs [1, 2]. Therefore, when we solve these problems using traditional gradient-based methods, we encounter many difficulties in achieving a satisfactory solution [3]. As the field of optimization has developed, metaheuristic algorithms have become increasingly popular. It has the property of achieving an optimal or near-optimal solution in a reasonable time and does not rely on problem-specific gradient information [4]. It is therefore widely used to solve various types of optimization problems, such as task planning [5–7], feature selection [8, 9], parameter optimization [10, 11], and image segmentation [12, 13].

In the last decades, many metaheuristic algorithms have been proposed and successfully applied to different domains. These algorithms can be divided into three categories: evolutionary-based algorithms, physical-based algorithms, and swarm-based algorithms. Evolution-based algorithms are a class of algorithms that simulate the laws of evolution in nature. The genetic algorithm (GA) [14] is a widely used evolutionary-based algorithm proposed by Holland. It updates populations by simulating the natural law of survival of the fittest. With the popularity of GA and GA variants, more and more evolutionary-based algorithms are continuously being proposed, including differential evolution (DE) [15], genetic programming (GP) [16], evolutionary strategies (ES) [17], and evolutionary programming (EP) [18]. Besides these evolutionary algorithms, some novel evolutionary-based algorithms have been proposed recently, such as artificial algae algorithm (AAA) [19], biogeography-based optimization (BBO) [20], and monkey king evolutionary (MKE) [21]. Physical-based algorithms simulate the laws of physics in nature or in the universe. Simulated annealing (SA) [22] inspired by annealing phenomena in metallurgy is one of the best-known physical-based algorithms. There are other physical-based algorithms proposed, including gravitational search algorithm (GSA) [23], nuclear reaction optimization (NRO) [24], water cycle algorithm (WCA) [25], and sine cosine algorithm (SCA) [26]. Swarm-based algorithms simulate social behaviour such as self-organization and division of labour in species. The particle swarm optimization (PSO) [27]and ant colony optimization (ACO) [28] are two classic swarm-based algorithms. Inspired by these two algorithms, more and more scholars are conducting research on this subject and proposing different swarm-based algorithms. The popularity of PSO and ACO has prompted more researchers to propose new swarm-based metaheuristics. Mirjalili et al. proposed grey wolf optimizer based on the collaborative foraging of grey wolves [29]. Wang et al. proposed monarch butterfly optimization inspired by the migratory activity of monarch butterflies [30]. Inspired by the spiral foraging and parabolic foraging of tuna, Xie et al. proposed the tuna swarm optimization [31]. In addition to the above three types of algorithms, a class of human-based metaheuristics is beginning to emerge. These algorithms are inspired by the characteristics of human activity. Teaching-learning-based optimization (TLBO) [32], inspired by traditional teaching methods, is a typical example of this category of metaheuristic algorithms. Other human-based metaheuristics include social evolution and learning optimization (SELO) [33], heap-based optimizers (HBO) [34], political optimizers (PO) [35], and many others. The butterfly optimization algorithm (BOA) is a swarm-based metaheuristic algorithm proposed by Arora and Singh [36]. BOA establishes an exploitation and exploration process based on the foraging behaviour and information-sharing strategies of butterflies. Although the BOA can perform exploitation and exploration operations, the basic BOA suffers from diminished population diversity and a tendency to fall into local optimum. Meanwhile, the No free lunch theory (NFL) [37] states that no single algorithm can solve all optimization problems perfectly. These factors encourage us to further enhance and improve the performance of BOA.

Metaheuristics have a common property: they find optimal solutions by exploiting and exploring the search space. Exploitation dominates and will weaken exploration. But when exploration is enhanced, exploitation is weakened. So, we need to improve algorithm performance by balancing exploitation and exploration. The improvements to the algorithm focus on three main areas. The first one is the optimization of the algorithm's parameter settings. Fan et al. [38] adjusted the fragrance factor of BOA and proposed an adaptive fragrance factor update method to enhance the convergence of BOA. Tang et al. [39] proposed the use of chaotic mapping operators to replace the alert value of the sparrow search algorithm for the purpose of balancing exploitation and exploration. Fan et al. [32] presented a new nonlinear step-factor control parameter strategy to further enhance the global search capability of the marine predator algorithm. The second is to use some techniques from other fields to improve performance. The fractional order is an effective tool that has been used in other areas [33, 40]. Yousri et al. [34] proposed an enhanced Harris hawk optimization based on fractional-order calculus memory. Elaziz et al. [35] improved the initial population of the Harris hawk optimizer using fractional-order Gaussians and $2 \times \mathrm{mod}1$ chaotic mappings. The third is to use other operators to improve the original algorithm. Wang et al. [41] proposed a hybrid metaheuristic algorithm for butterfly and flower pollination based on a reciprocal mechanism. Houssein et al. [42] proposed a variant of the slime mould algorithm with hybrid adaptive guided differential evolution in order to overcome the disadvantages of unbalanced exploitation and exploration. Inspired by these hybrid variants, this paper proposes a BOA variant with hybrid distribution estimation strategy, GDEBOA. GDEBOA uses a Gaussian probability model to describe the distribution of dominant populations and to guide the direction of evolution, improving the performance of the basic BOA. The performance of GDEBOA was evaluated on the CEC2017 test suite and compared with seven state-of-the-art algorithms. The superiority of the proposed algorithm is verified by numerical analysis, convergence analysis, stability analysis, and statistical analysis. In addition, GDEBOA is applied to the UAV route planning problem to further validate the algorithm's ability to solve real-world optimization problems.

The remainder of this paper is organized as follows. A review of the basic BOA is presented in Section 2. Section 3 provides a detailed description of the proposed GDEBOA. In Section 4, the effectiveness of the proposed improvement strategy is verified using CEC 2017 test suite. Furthermore, GDEBOA is applied to solve the UAV route planning problem in Section 5. Finally, we summarize this work in Section 6 and offer directions for future research.

2. Butterfly Optimization Algorithm

The butterfly optimization algorithm is a swarm-based metaheuristic algorithm proposed by Arora et al. The algorithm builds a model of butterfly foraging and mating. BOA proposes three hypotheses: (1) all butterflies emit fragrance and are attracted to each other; (2) each butterfly moves randomly or towards the butterfly with the most scent; and (3) the stimulus intensity of the butterfly is determined by the landscape of the fitness function. As the butterflies move, the fragrance changes with them. All butterflies form a fragrance network, and when they do not feel the fragrance network, they fly randomly, which is called the global search phase. As the butterflies approach the butterfly with the greatest concentration of fragrance, this phase is called the local search phase. BOA solves the optimization problem through global and local search with the following mathematical model.

The fragrance of the butterfly is expressed as a function of the physical intensity of the stimulus, described as follows:

$$f_i = cI^a, \quad i = 1, 2, \ldots, NP. \tag{1}$$

where f_i represents the butterfly fragrance, c represents the sensory modality, I represents the stimulus intensity, a is a power exponent with a value from 0 to 1, and NP denotes the number of butterflies. The mathematical model of the global and local search phases of BOA is represented as follows:

$$X_i^{t+1} = X_i^t + \left(r^2 \times X_{\text{best}}^t - X_i^t\right) \times f_i, \tag{2}$$

$$X_i^{t+1} = X_i^t + \left(r^2 \times X_j^t - X_k^t\right) \times f_i, \tag{3}$$

where X_i^t denotes the position of the i^{th} butterfly in the t^{th} iteration, X_{best}^t denotes the global optimal individual, $r \in (0, 1)$ is a random number, and X_j^t and X_k^t are the j^{th} individual and the k^{th} individual selected randomly. BOA constantly executes two search strategies during the search process. Therefore, a switching probability p is introduced to control the switching of the two strategies. The pseudocode for BOA is given in Algorithm 1.

3. Proposed GDEBOA

To overcome the shortcomings of the basic butterfly optimization algorithm, a modified butterfly optimization algorithm, called GDEBOA, is proposed in this paper. The combination of the Gaussian distribution estimation algorithm and BOA provides a solution to the problem of unbalanced exploitation and exploration capabilities that exist in BOA. Here, this paper employs a Gaussian distribution estimation strategy as an alternative to the global search strategy in BOA. The GDE is used to sample the dominant population and guide the evolutionary direction of the algorithm while increasing population diversity. The improved strategies and GDEBOA are described in detail below.

3.1. Gaussian Distribution Estimation. The Gaussian distribution estimation strategy represents inter-individual relationships through a probabilistic model. The strategy uses the current dominant population to calculate the probability distribution model and generates new offspring populations based on the probability distribution model sampling, eventually obtaining the optimal solution by continuous iteration. In this paper, the distribution model is estimated using a weighted maximum likelihood estimation method, and the top one-half population that performs better is taken as the dominant population. The mathematical model of this strategy is described as follows:

$$X_i^{t+1} = \mathbf{mean} + \mathbf{y}, \quad \mathbf{y} \sim N(0, \mathbf{Cov}), \tag{4}$$

$$\mathbf{mean} = \frac{\left(X_{\text{best}} + X_{\text{mean}}^t + X_i^t\right)}{3}, \tag{5}$$

$$\mathbf{Cov}(i) = \frac{1}{NP/2} \sum_{i=1}^{NP/2} \left(X_i^{t+1} - X_{\text{mean}}^t\right) \times \left(X_i^t - X_{\text{mean}}^t\right)^T, \tag{6}$$

$$X_{\text{mean}}^t = \sum_{i=1}^{NP/2} \omega_i \times X_i^t, \tag{7}$$

$$\omega_i = \frac{\ln(0.5\text{NP} + 0.5) - \ln(i)}{\sum_{i=1}^{NP/2} (\ln(0.5\text{NP} + 0.5) - \ln(i))}, \tag{8}$$

where X_{mean}^t denotes the weighted position of the dominant population and ω denotes the weight coefficient in the dominant population in descending order of fitness values. **Cov** is the weighted covariance matrix of the dominant populations. The pseudocode and flowchart of the proposed GDEBOA are shown in Algorithm 2 and Figure 1.

4. Numerical Experiment and Analysis

In order to fully validate the superior performance of the proposed GDEBOA, the algorithm was tested using the IEEE CEC2017 single-objective test function. The CEC2017 test suite consists of 28 test functions. F1 is a unimodal function with only one global optimum solution and is used to verify the local search capability of the algorithm. F2–F8 are multimodal functions and are primarily used to test the ability of an algorithm to get outside of a local optimum. F9–F17 and F18–F28 are hybrid and composite functions, respectively, and can be used to test the potential of an algorithm to solve complex optimization problems in the real world. The definition of functions and optima is given in Table 1.

Seven state-of-the-art metaheuristics were used for comparison with GDEBOA, including artificial ecosystem-based optimization (AEO) [43], grey wolf optimizer (GWO) [29], Harris hawks optimization (HHO) [44], arithmetic optimization algorithm (AOA) [45], slime mould algorithm (SMA) [46], Manta ray foraging optimization (MRFO) [47], and pathfinder algorithm (PFA) [48]. In the CEC2017 test, the maximum number of iterations is 600, and the population size is 500. All the algorithm parameters were set to be the same as in the original literature, as shown in Table 2. All algorithms were run 51 times independently, and the experimental results were recorded as shown in Table 3. In this paper, the performance of GDEBOA was evaluated comprehensively by numerical analysis, convergence analysis, stability analysis, Wilcoxon test, and Friedman test. The experiments in this paper were conducted on a computer with an AMD R7 4800U processor and 16 GB RAM. Programming was performed using MATLAB R2016b.

The results in Table 3 show that GDEBOA performs best on the unimodal function F1. Although GDEBOA does not consistently achieve the optimal solution, it provides the best solution among the nine algorithms. Compared to BOA, the performance of GDEBOA is significantly improved, indicating that the improvement strategy is effective in enhancing the exploitation capability. In the multimodal functions F2–F8, each algorithm performs differently. GDEBOA performs best on F2, F4, and F7. GWO provides the best solution on F3, F6, and F8. SMA achieves the optimal solution on F5. Significantly, GDEBOA outperforms BOA in all multimodal functions, which indicates that

```
Generate initial population of $NP$ butterflies
Initialize parameters $c$, $a$, $p$
While $t < t_{\max}$
    For each butterfly do
        Calculate the fragrance using equation (1)
    End for
    Find the best butterfly
    For each butterfly $X_i$ do
        Generate a random number $r$ from [0,1]
        If $p < r$
            Calculate the position $X_i^{\text{new}}$ using equation (2)
        Else
            Calculate the position $X_i^{\text{new}}$ using equation (3)
        End if
        Calculate the fitness and select the better one from $[X_i^{\text{new}}, X_i]$
    End for
    Update parameter $c$
End while
Output the best position and fitness
```

ALGORITHM 1: Butterfly optimization algorithm.

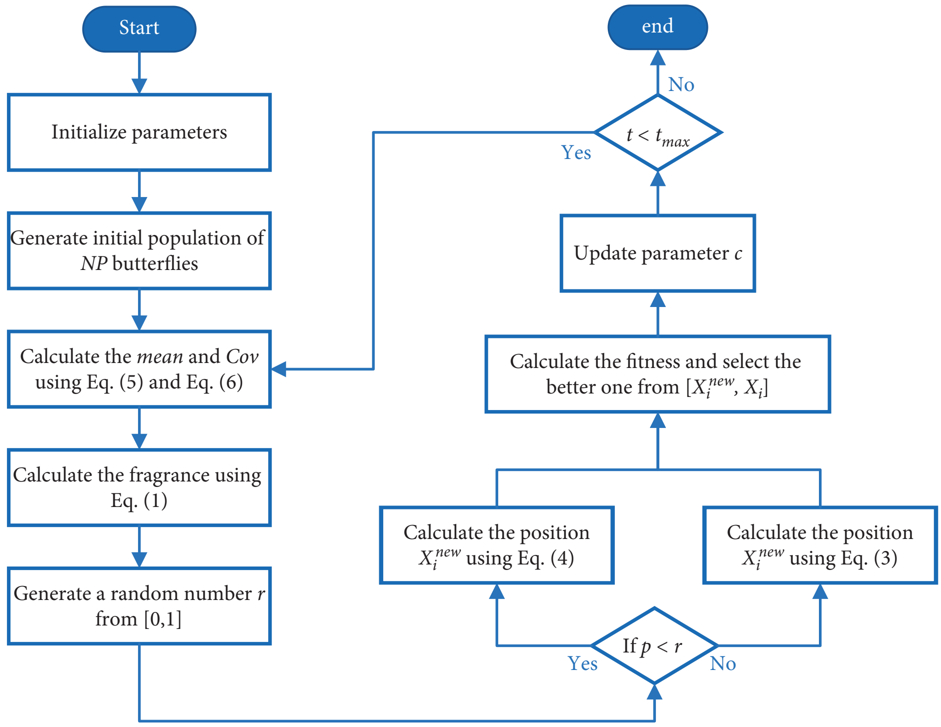

Start
Initialize parameters
Generate initial population of NP butterflies
Calculate the *mean* and *Cov* using Eq. (5) and Eq. (6)
Calculate the fragrance using Eq. (1)
Generate a random number r from [0,1]
If $p < r$
Yes
No
Calculate the position X_i^{new} using Eq. (4)
Calculate the position X_i^{new} using Eq. (3)
Calculate the fitness and select the better one from $[X_i^{new}, X_i]$
Update parameter c
$t < t_{max}$
Yes
No
end

FIGURE 1: Flowchart of GEDBOA.

GDEBOA has a strong global search capability. The improved strategy can effectively enhance BOA's ability to explore the solution space. The GDEBOA performs best in most hybrid and composite functions. Specifically, GDEBOA achieved satisfactory results in 7 of the 10 hybrid functions. The GDEBOA achieved optimal solutions on 5 of the 10 composite functions. Compared to BOA, GDEBOA only performed worse on F19. The analysis of the results for the hybrid and composite functions shows that GDEBOA has a good balance of exploitation and exploration capabilities and is able to solve complex optimization problems effectively.

Convergence speed and convergence accuracy are important indicators of an algorithm's performance. Convergence analysis provides information on how the algorithm has changed over a process of iterations. Figure 2 shows the average fitness convergence curves for F1–F28 based on the results of all algorithms solving the test suite 51 times. The

```
Generate initial population of $NP$ butterflies
Initialize parameters $c$, $a$, $p$
While $t < t_{\max}$
    Calculate the **mean** and **Cov** using equations (5) and (6)
    For each butterfly do
        Calculate the fragrance using equation (1)
    End for
    Find the best butterfly
    For each butterfly $X_i$ do
        Generate a random number $r$ from [0,1]
        If $p < r$
            Calculate the position $X_i^{\text{new}}$ using equation (4)
        Else
            Calculate the position $X_i^{\text{new}}$ using equation (3)
        End if
        Calculate the fitness and select the better one from $[X_i^{\text{new}}, X_i]$
    End for
    Update parameter $c$
End while
Output the best position and fitness
```

Algorithm 2: GDEBOA.

Table 1: Descriptions of CEC 2017 test suite.

Type	No.	Description	Fi*
Unimodal functions	1	Shifted and rotated bent Cigar function	300
	2	Shifted and rotated Rosenbrock's function	400
Unimodal functions	3	Shifted and rotated Rastrigin's function	500
	4	Shifted and rotated expanded Scaffer's F6 function	600
	5	Shifted and rotated Lunacek bi-Rastrigin function	700
	6	Shifted and rotated noncontinuous Rastrigin's function	800
	7	Shifted and rotated Levy function	900
	8	Shifted and rotated Schwefel's function	1000
Hybrid functions	9	Hybrid function 1 ($N = 3$)	1100
	10	Hybrid function 2 ($N = 3$)	1200
	11	Hybrid function 3 ($N = 3$)	1300
	12	Hybrid function 4 ($N = 4$)	1400
	13	Hybrid function 5 ($N = 4$)	1500
	14	Hybrid function 6 ($N = 4$)	1600
	15	Hybrid function 6 ($N = 5$)	1700
	16	Hybrid function 6 ($N = 5$)	1800
	17	Hybrid function 6 ($N = 5$)	1900
	18	Hybrid function 6 ($N = 6$)	2000
Composite functions	19	Composition function 1 ($N = 3$)	2100
	20	Composition function 2 ($N = 3$)	2200
	21	Composition function 3 ($N = 4$)	2300
	22	Composition function 4 ($N = 4$)	2400
	23	Composition function 5 ($N = 5$)	2500
	24	Composition function 6 ($N = 5$)	2600
	25	Composition function 7 ($N = 6$)	2700
	26	Composition function 8 ($N = 6$)	2800
	27	Composition function 9 ($N = 3$)	2900
	28	Composition function 10 ($N = 3$)	3000

results in Figure 2 show that GDEBOA has a faster convergence speed and better convergence accuracy compared to other algorithms. Specifically, GDEBOA outperformed all comparison algorithms in terms of convergence accuracy and convergence speed on 16 functions. Notably, although GDEBOA did not perform best on all functions, GDEBOA

Table 2: Algorithms used for comparative analysis and their parameter settings.

Algorithm	Parameters
AEO	No parameters
GWO	$a = 2$ (linearly decreased over iterations)
HHO	$\beta = 1.5, E_0 \in [-1, 1]$
AOA	$\mathrm{Mop}_{\max} = 1, \mathrm{Mop}_{\min} = 0.2, C = 1, \alpha = 5, \mathrm{Mu} = 0.499$
SMA	$z = 0.03$
MRFO	$S = 2$
PFA	$u_1 = -1 + 2\mathrm{rand}, u_2 = -1 + 2\mathrm{rand}$

Table 3: Statistical results of the nine comparison algorithms for CEC2017.

No.	Index	AEO	GWO	HHO	AOA	SMA	MRFO	PFA	BOA	GDEBOA
F1	Mean	9.01E − 04	1.74E + 04	1.68E + 03	6.91E + 04	4.94E − 01	6.61E + 00	4.66E + 04	3.82 E +04	7.70E − 06
	Std.	1.89E − 03	7.95E + 03	7.95E + 02	1.15E + 04	3.18E − 01	4.23E + 00	1.22E + 04	6.97 E + 03	9.81E − 07
	Rank	2	6	5	9	3	4	8	7	1
F2	Mean	6.65E + 01	1.35E + 02	1.23E + 02	7.61E + 03	8.99E + 01	6.70E + 01	9.80E + 01	9.33E + 03	5.49E + 01
	Std.	3.72E + 01	2.53E + 01	3.33E + 01	2.45E + 03	5.12E + 00	4.09E + 01	1.77E + 01	1.29E + 03	2.08E + 01
	Rank	2	7	6	8	4	3	5	9	1
F3	Mean	1.34E + 02	7.12E + 01	2.05E + 02	2.95E + 02	8.17E + 01	1.46E + 02	1.14E + 02	3.49E + 02	1.86E + 02
	Std.	3.17E + 01	2.25E + 01	3.62E + 01	3.20E + 01	1.99E + 01	4.09E + 01	3.11E + 01	2.16E + 01	1.18E + 01
	Rank	4	1	7	8	2	5	3	9	6
F4	Mean	1.67E + 01	2.28E + 00	5.62E + 01	6.21E + 01	7.35E − 01	1.05E + 01	1.47E + 01	6.63E + 01	4.03E − 02
	Std.	5.63E + 00	1.21E + 00	5.92E + 00	6.71E + 00	3.21E − 01	1.21E + 01	4.99E + 00	5.76E + 00	2.51E − 01
	Rank	6	3	7	8	2	4	5	9	1
F5	Mean	2.75E + 02	1.31E + 02	4.98E + 02	6.00E + 02	1.18E + 02	2.12E + 02	1.34E + 02	5.57E + 02	2.09E + 02
	Std.	6.77E + 01	4.45E + 01	6.57E + 01	5.66E + 01	2.40E + 01	6.09E + 01	3.12E + 01	3.17E + 01	1.14E + 01
	Rank	6	2	7	9	1	5	3	8	4
F6	Mean	1.13E + 02	7.52E + 01	1.40E + 02	2.25E + 02	9.39E + 01	1.27E + 02	9.97E + 01	2.93E + 02	1.83E + 02
	Std.	2.33E + 01	3.07E + 01	2.13E + 01	2.67E + 01	2.03E + 01	3.29E + 01	2.66E + 01	1.54E + 01	1.11E + 01
	Rank	4	1	6	8	2	5	3	9	7
F7	Mean	2.49E + 03	2.05E + 02	4.69E + 03	4.50E + 03	9.95E + 02	1.77E + 03	2.28E + 02	6.82E + 03	5.35E − 02
	Std.	9.16E + 02	1.62E + 02	8.28E + 02	7.24E + 02	1.22E + 03	1.02E + 03	1.83E + 02	8.69E + 02	1.74E − 01
	Rank	6	2	8	7	4	5	3	9	1
F8	Mean	3.37E + 03	2.64E + 03	4.35E + 03	5.51E + 03	3.04E + 03	3.41E + 03	4.98E + 03	7.33E + 03	6.94E + 03
	Std.	5.42E + 02	5.39E + 02	7.25E + 02	5.83E + 02	5.06E + 02	6.00E + 02	9.01E + 02	2.85E + 02	2.87E + 02
	Rank	3	1	5	7	2	4	6	9	8
F9	Mean	1.09E + 02	1.96E + 02	1.61E + 02	1.72E + 03	1.16E + 02	9.20E + 01	1.91E + 02	2.19E + 03	7.09E + 01
	Std.	3.10E + 01	4.04E + 01	4.86E + 01	9.74E + 02	4.33E + 01	3.46E + 01	5.28E + 01	6.72E + 02	2.60E + 01
	Rank	3	7	5	8	4	2	6	9	1
F10	Mean	3.74E + 04	2.18E + 07	7.61E + 06	6.27E + 09	1.31E + 06	9.00E + 04	1.88E + 06	2.08E + 09	3.49E + 02
	Std.	3.35E + 04	1.87E + 07	4.21E + 06	2.56E + 09	1.09E + 06	7.44E + 04	1.97E + 06	7.43E + 08	2.52E + 02
	Rank	2	7	6	9	4	3	5	8	1
F11	Mean	1.51E + 04	5.50E + 05	1.51E + 05	3.80E + 04	2.71E + 04	1.36E + 04	7.54E + 04	3.15E + 08	8.30E + 01
	Std.	1.46E + 04	3.30E + 06	9.05E + 04	1.71E + 04	2.64E + 04	1.62E + 04	4.12E + 04	2.10E + 08	2.77E + 01
	Rank	3	8	7	5	4	2	6	9	1
F12	Mean	1.40E + 02	6.15E + 04	3.82E + 04	5.72E + 04	4.71E + 04	2.01E + 03	3.00E + 04	1.19E + 05	6.29E + 01
	Std.	4.36E + 01	9.79E + 04	4.25E + 04	4.92E + 04	2.84E + 04	2.19E + 03	2.94E + 04	7.62E + 04	7.46E + 00
	Rank	2	8	5	7	6	3	4	9	1
F13	Mean	4.10E + 03	5.46E + 04	6.86E + 04	2.35E + 04	1.99E + 04	8.64E + 03	3.35E + 04	1.82E + 06	5.54E + 01
	Std.	4.49E + 03	5.70E + 04	4.86E + 04	1.22E + 04	1.57E + 04	9.15E + 03	1.77E + 04	1.46E + 06	1.94E + 01
	Rank	2	7	8	5	4	3	6	9	1
F14	Mean	1.03E + 03	6.98E + 02	1.55E + 03	1.98E + 03	8.18E + 02	8.93E + 02	1.00E + 03	3.18E + 03	1.30E + 03
	Std.	3.11E + 02	2.59E + 02	3.56E + 02	5.09E + 02	2.83E + 02	2.63E + 02	2.63E + 02	4.12E + 02	1.80E + 02
	Rank	5	1	7	8	2	3	4	9	6
F15	Mean	4.06E + 02	2.17E + 02	7.48E + 02	9.12E + 02	4.34E + 02	3.36E + 02	3.77E + 02	1.22E + 03	4.31E + 02
	Std.	1.84E + 02	1.23E + 02	2.19E + 02	2.67E + 02	1.64E + 02	2.02E + 02	1.71E + 02	2.49E + 02	7.95E + 01
	Rank	4	1	7	8	6	2	3	9	5
F16	Mean	1.70E + 04	5.57E + 05	6.90E + 05	1.29E + 06	3.75E + 05	7.66E + 04	2.75E + 05	9.60E + 05	3.68E + 01
	Std.	1.28E + 04	7.70E + 05	8.77E + 05	1.60E + 06	3.55E + 05	6.45E + 04	2.82E + 05	6.22E + 05	4.62E + 00
	Rank	2	6	7	9	5	3	4	8	1

Table 3: Continued.

No.	Index	AEO	GWO	HHO	AOA	SMA	MRFO	PFA	BOA	GDEBOA
F17	Mean	$1.94E+03$	$2.42E+05$	$1.46E+05$	$1.08E+06$	$3.00E+04$	$8.42E+03$	$4.45E+04$	$4.61E+06$	$2.79E+01$
	Std.	$3.07E+03$	$2.98E+05$	$1.42E+05$	$1.39E+05$	$2.11E+04$	$1.07E+04$	$3.91E+04$	$4.06E+06$	$2.57E+00$
	Rank	2	7	6	8	4	3	5	9	1
F18	Mean	$3.95E+02$	$3.20E+02$	$6.71E+02$	$6.94E+02$	$3.59E+02$	$3.45E+02$	$4.61E+02$	$7.29E+02$	$4.90E+02$
	Std.	$1.62E+02$	$1.03E+02$	$2.01E+02$	$1.54E+02$	$1.59E+02$	$1.33E+02$	$1.52E+02$	$9.88E+01$	$9.82E+01$
	Rank	4	1	7	8	3	2	5	9	6
F19	Mean	$3.26E+02$	$2.67E+02$	$4.06E+02$	$4.87E+02$	$2.93E+02$	$3.09E+02$	$2.90E+02$	$1.97E+02$	$3.73E+02$
	Std.	$3.26E+01$	$2.45E+01$	$3.51E+01$	$5.23E+01$	$2.17E+01$	$2.95E+01$	$2.62E+01$	$3.01E+01$	$1.21E+01$
	Rank	6	2	8	9	4	5	3	1	7
F20	Mean	$1.73E+02$	$1.49E+03$	$2.39E+03$	$5.13E+03$	$2.90E+03$	$1.00E+02$	$2.08E+02$	$4.71E+02$	$1.00E+02$
	Std.	$5.11E+02$	$1.42E+03$	$2.37E+03$	$1.21E+03$	$1.36E+03$	$7.40E-01$	$7.61E+02$	$7.76E+01$	$2.31E-04$
	Rank	3	6	7	9	8	2	4	5	1
F21	Mean	$5.41E+02$	$4.18E+02$	$7.05E+02$	$9.68E+02$	$4.35E+02$	$4.95E+02$	$4.85E+02$	$6.97E+02$	$5.28E+02$
	Std.	$4.86E+01$	$3.60E+01$	$7.35E+01$	$9.10E+01$	$1.95E+01$	$4.15E+01$	$4.09E+01$	$5.59E+01$	$1.59E+01$
	Rank	6	1	8	9	2	4	3	7	5
F22	Mean	$6.41E+02$	$5.02E+02$	$8.26E+02$	$1.14E+03$	$5.30E+02$	$5.65E+02$	$5.24E+02$	$1.10E+03$	$5.89E+02$
	Std.	$7.63E+01$	$4.93E+01$	$7.42E+01$	$1.09E+02$	$2.95E+01$	$5.19E+01$	$3.72E+01$	$1.68E+02$	$4.38E+01$
	Rank	6	1	7	9	3	4	2	8	5
F23	Mean	$3.94E+02$	$4.29E+02$	$4.11E+02$	$1.67E+03$	$3.88E+02$	$3.89E+02$	$3.97E+02$	$1.75E+03$	$3.87E+02$
	Std.	$1.47E+01$	$1.79E+01$	$1.87E+01$	$4.55E+02$	$1.69E+00$	$8.17E+00$	$1.72E+01$	$2.01E+02$	$6.50E-01$
	Rank	4	7	6	8	2	3	5	9	1
F24	Mean	$2.62E+03$	$1.69E+03$	$3.94E+03$	$6.40E+03$	$1.98E+03$	$2.71E+03$	$2.12E+03$	$5.21E+03$	$2.54E+03$
	Std.	$1.10E+03$	$2.41E+02$	$1.10E+03$	$7.22E+02$	$3.42E+02$	$1.05E+03$	$7.10E+02$	$1.49E+03$	$3.47E+02$
	Rank	5	1	7	9	2	6	3	8	4
F25	Mean	$5.92E+02$	$5.22E+02$	$6.05E+02$	$1.34E+03$	$5.11E+02$	$5.56E+02$	$5.46E+02$	$8.14E+02$	$5.01E+02$
	Std.	$3.05E+01$	$9.78E+00$	$4.00E+01$	$2.14E+02$	$1.17E+01$	$2.78E+01$	$2.85E+01$	$9.81E+01$	$1.52E+01$
	Rank	6	3	7	9	2	5	4	8	1
F26	Mean	$3.49E+02$	$5.11E+02$	$4.62E+02$	$2.95E+03$	$4.46E+02$	$3.42E+02$	$4.31E+02$	$3.28E+03$	$3.39E+02$
	Std.	$5.89E+01$	$4.00E+01$	$2.60E+01$	$6.15E+02$	$2.89E+01$	$5.27E+01$	$1.90E+01$	$3.99E+02$	$5.28E+01$
	Rank	3	7	6	8	5	2	4	9	1
F27	Mean	$9.30E+02$	$6.85E+02$	$1.32E+03$	$2.43E+03$	$7.75E+02$	$8.27E+02$	$1.03E+03$	$3.04E+03$	$9.73E+02$
	Std.	$2.09E+02$	$1.14E+02$	$2.56E+02$	$5.22E+02$	$1.83E+02$	$2.09E+02$	$2.35E+02$	$4.72E+02$	$1.13E+02$
	Rank	4	1	7	8	2	3	6	9	5
F28	Mean	$9.26E+03$	$3.77E+06$	$1.01E+06$	$1.47E+07$	$1.27E+04$	$7.34E+03$	$4.09E+05$	$3.98E+07$	$2.08E+03$
	Std.	$4.95E+03$	$2.70E+06$	$6.08E+05$	$1.01E+07$	$5.29E+03$	$3.49E+03$	$4.56E+05$	$2.31E+07$	$1.46E+02$
	Rank	3	7	6	8	4	2	5	9	1

outperformed BOA on the 27 out of 28 functions, indicating that the improvement strategy proposed in this paper better balances exploitation and exploration.

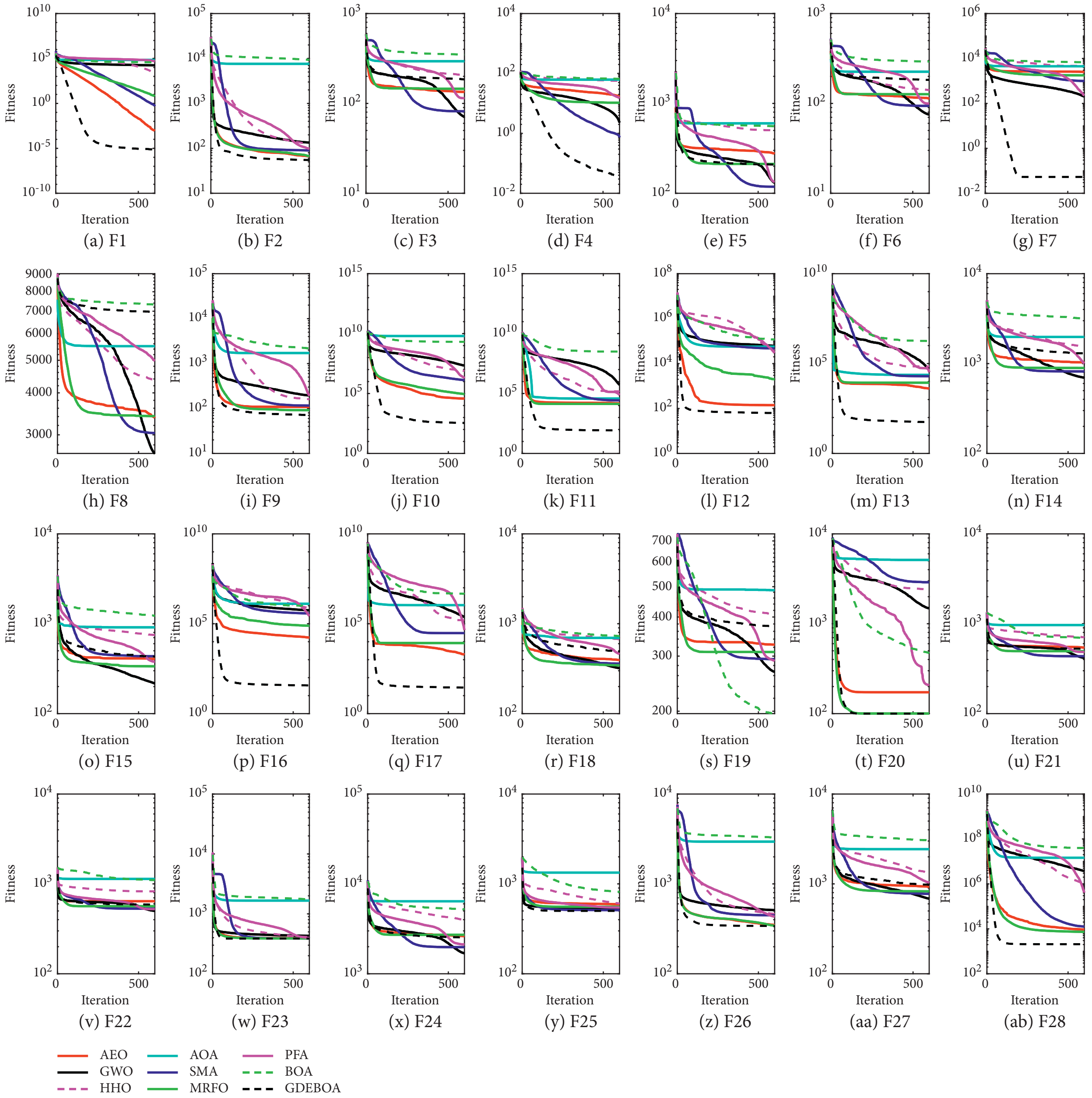
In addition, in order to analyse the distribution characteristics of GDEBOA's solutions when solving the functions, box diagrams are shown in Figure 3 based on the results of the nine algorithms solving the CEC2017 test set 51 times independently. For most of the test functions, the minimum, maximum, and median values obtained by GDEBOA are the same as the optimal solutions, which indicates that the solutions obtained by GDEBOA are more centrally distributed and more stable. Compared to BOA, GDEBOA solves the function with fewer bad values and a more concentrated distribution of solutions, indicating that GDEBOA has achieved a balance between exploitation and exploration.

Although the superiority of GDEBOA was demonstrated by comparisons of the mean and standard deviation, the literature [49, 50] demonstrates that these comparisons are not adequate. To further verify the differences between GDEBOA and other algorithms, the Wilcoxon signed-rank test was employed. Table 4 shows the results of the Wilcoxon signed-rank test with significance level $P = 0.05$. The term $R+$ indicates the extent to which GDEBOA outperforms the comparison algorithm, while $R-$ indicates the opposite. The symbol "+/=/−" indicates that GDEBOA performs better than, similar to, and worse than the comparison algorithm.

From Table 4, GDEBOA outperforms the comparison algorithm on most functions. Numerically, GDEBOA is superior to MRFO on 14 functions. GDEBOA outperforms GWO and PFA on 16 functions. Compared to HHO and AOA, GDEBOA performs better on 27 and 28 functions. In particular, GDEBOA offers better solutions than BOA on all functions, except F9. In general, GDEBOA shows a superior performance.

5. UAV Route Planning

In this section, the application of GDEBOA to UAV route planning is discussed in detail. The UAV route planning problem aims to minimize the cost of carrying out the mission, which can be considered as a multi-constraint optimization problem. The route planning model is described in detail in the following section.



(a) F1 (b) F2 (c) F3 (d) F4 (e) F5 (f) F6 (g) F7

(h) F8 (i) F9 (j) F10 (k) F11 (l) F12 (m) F13 (n) F14

(o) F15 (p) F16 (q) F17 (r) F18 (s) F19 (t) F20 (u) F21

(v) F22 (w) F23 (x) F24 (y) F25 (z) F26 (aa) F27 (ab) F28

FIGURE 2: Convergence curves of the GDEBOA and other algorithms for CEC2017.

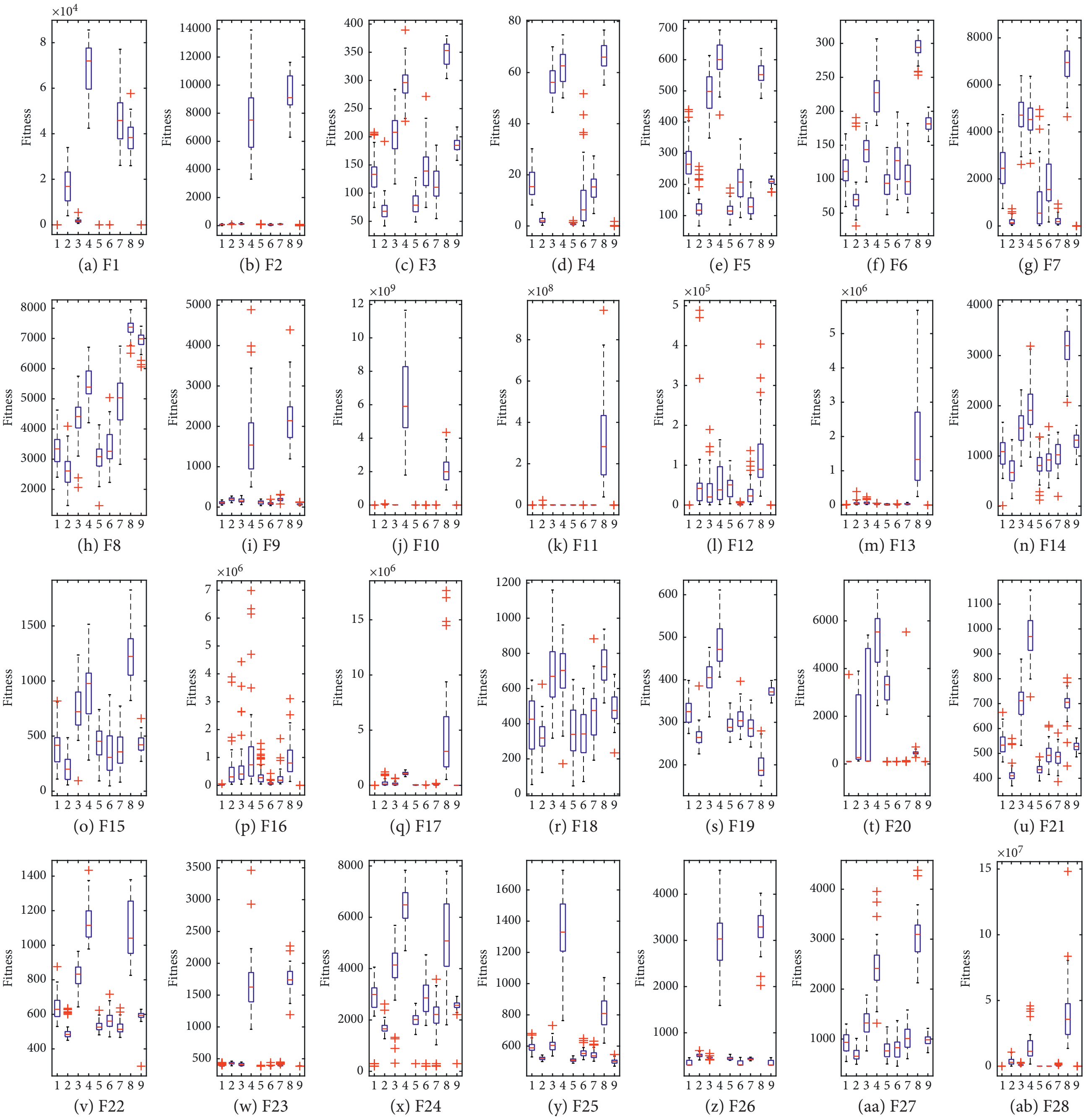
5.1. Cost Function. Considering the actual UAV flight scenario, we think that the UAV should reach the target as fast as possible while being free from threats during the flight. The cost function is described as the total of two functions and two constraints.

$$\begin{aligned} \min J &= \omega_1 \cdot \mathrm{Cost}_p + \omega_2 \cdot \mathrm{Cost}_h + \eta \cdot \mathrm{Penalty}, \\ \mathrm{Penalty} &= \sum_{i=1}^{n} c_i, \quad n = 1, 2, \\ c_i &= \begin{cases} 0, & \text{Satisfing constraints}, \\ 1, & \text{No satisfing constraints}, \end{cases} \end{aligned} \tag{9}$$

where J is the total cost function, Cost_p denotes the flight distance cost, Cost_h denotes the height cost, ω_1 and ω_2 are the weighting factors of two costs that satisfy $\omega_1 + \omega_2 = 1$, Penalty denotes the number of constraint violations, and η is the penalty function factor for converting the constrained optimization problem into an unconstrained optimization problem.

The faster the UAV reaches the target, the better it is for the mission, so the path cost is represented by the sum of the route segments.

$$\mathrm{Cost}_p = \sum_{i=1}^{NP-1} \mathrm{Dis}_i, \tag{10}$$



(a) F1 (b) F2 (c) F3 (d) F4 (e) F5 (f) F6 (g) F7

(h) F8 (i) F9 (j) F10 (k) F11 (l) F12 (m) F13 (n) F14

(o) F15 (p) F16 (q) F17 (r) F18 (s) F19 (t) F20 (u) F21

(v) F22 (w) F23 (x) F24 (y) F25 (z) F26 (aa) F27 (ab) F28

Figure 3: Boxplots analysis for CEC2017.

where Dis_i is the length of the i^{th} route segment.

Besides, it is not beneficial for UAV to fly too high to avoid the threat. Therefore, the UAV needs to maintain low height flight. The corresponding height cost function is described as follows:

$$\text{Cost}_h = \sum_{i=1}^{NP} z_i, \tag{11}$$

where z_i is the height corresponding to the i^{th} route point.

Due to the limitation of UAV's maneuverability, the turn angle and climb angle of the actual flight should be less than the theoretical maximum.

$$\begin{aligned} \text{Constraint}_1 &= \max\left|\varphi_{ij}\right| - \varphi_{\max} \le 0, \\ \text{Constraint}_2 &= \max\left|\theta_{ij}\right| - \theta_{\max} \le 0, \end{aligned} \tag{12}$$

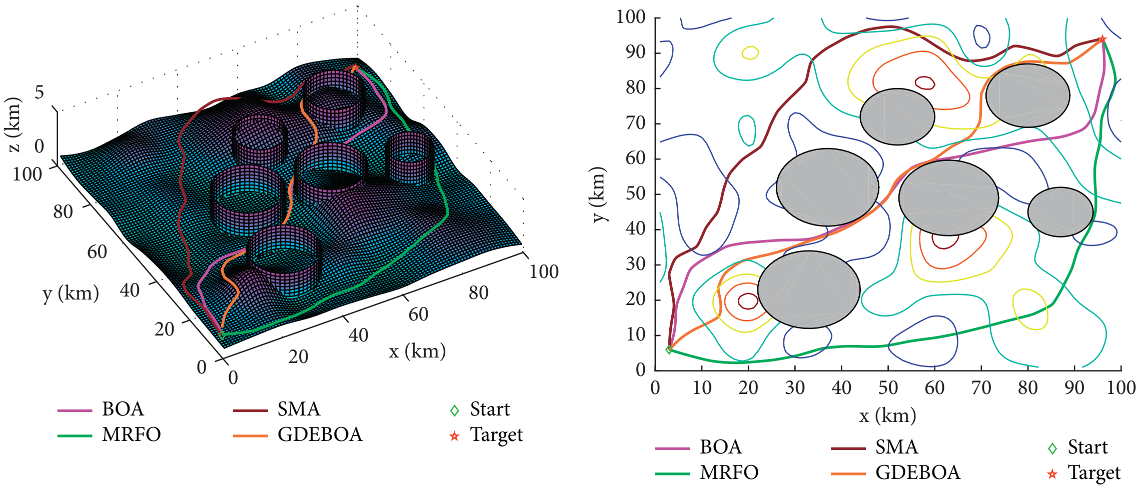
where φ_{ij} and θ_{ij} are the turn angle and climb angle between each track segment, respectively, and $\varphi_{\max}$ and $\theta_{\max}$ are the corresponding theoretical maximum values.

Table 4: Statistical results of Wilcoxon signed-rank test for CEC2017.

AEO				GWO				HHO				AOA			
P-value	*R*+	*R*−	Win	*P* value	*R*+	*R*−	Win	*P*-value	*R*+	*R*−	Win	*P* value	*R*+	*R*−	Win
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
0.071907	855	471	=	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
3.94E − 09	35	1291	−	5.46E − 10	1	1325	−	0.001534	1001	325	+	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
5.43E − 08	1243	83	+	3.94E − 09	35	1291	−	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
5.15E − 10	0	1326	−	7.35E − 10	6	1320	−	5.46E − 10	1	1325	−	6.93E − 10	1321	5	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
5.15E − 10	0	1326	−	5.15E − 10	0	1326	−	5.15E − 10	0	1326	−	5.15E − 10	0	1326	−
3.96E − 07	1204	122	+	5.15E − 10	1326	0	+	1.40E − 09	1309	17	+	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
7.35E − 10	1320	6	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
6.52E − 05	237	1089	−	1.05E − 09	12	1314	−	0.000153	1067	259	+	1.42E − 08	1268	58	+
0.276894	547	779	=	2.65E − 09	28	1298	−	2.97E − 09	1296	30	+	6.93E − 10	1321	5	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
0.004923	363	963	−	5.53E − 09	41	1285	−	2.11E − 06	1169	157	+	9.14E − 09	1276	50	+
1.27E − 08	56	1270	−	5.15E − 10	0	1326	−	8.61E − 07	1188	138	+	5.15E − 10	1326	0	+
0.205722	798	528	=	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
0.264661	782	544	=	6.15E − 10	3	1323	−	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
0.000331	1046	280	+	3.37E − 08	74	1252	−	5.15 − 10	1326	0	+	5.15E − 10	1326	0	+
0.006378	954	372	+	5.15E − 10	1326	0	+	6.93E − 10	1321	5	+	5.15E − 10	1326	0	+
0.035759	887	439	+	4.94E − 09	39	1287	−	1.53E − 07	1223	103	+	5.15E − 10	1326	0	+
5.15E–10	1326	0	+	3.03E − 08	1254	72	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
0.285261	777	549	=	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15 E − 10	1326	0	+
0.205722	528	798	=	8.77E − 10	9	1317	−	9.66E − 09	1275	51	+	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
	16/6/6				16/0/12				26/0/2				27/0/1		
SMA				MRFO				PFA				BOA			
P-value	*R*+	*R*−	Win	*P* value	*R*+	*R*−	Win	*P* value	*R*+	*R*−	Win	*P* value	*R*+	*R*−	Win
5.15E − 10	1326	0	+	5.15E–10	1326	0	+	5.15E–10	1326	0	+	5.15E–10	1326	0	+
5.15E − 10	1326	0	+	0.100931	838	488	+	5.15E–10	1326	0	+	5.15E–10	1326	0	+
5.15E − 10	0	1326	−	3.25E − 07	118	1208	−	5.46E − 10	1	1325	−	5.15E − 10	1326	0	+
7.74E − 09	1279	47	+	8.77E − 10	1317	9	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
5.15E − 10	0	1326	−	0.880784	679	647	=	6.93E − 10	5	1321	−	5.15E − 10	1326	0	+
5.15E − 10	0	1326	−	1.18E − 09	14	1312	−	5.15E − 10	0	1326	−	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
5.15E − 10	0	1326	−	5.15E − 10	0	1326	−	5.15E − 10	0	1326	−	8.25E − 08	1235	91	+
1.26E − 06	1180	146	+	0.006562	953	373	+	5.46E − 10	1325	1	+	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	6.15E − 10	1323	3	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.80E − 10	1324	2	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
2.23E − 09	25	1301	−	4.42E − 09	37	1289	−	1.69E − 07	105	1221	−	5.15E − 10	1326	0	+
0.785752	692	634	=	0.004378	359	967	−	0.052342	456	870	=	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.46E − 10	1325	1	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
1.49E − 05	201	1125	−	8.49E − 06	188	1138	−	0.174098	518	808	=	5.15E − 10	1326	0	+
5.15E − 10	0	1326	−	1.05E − 09	12	1314	−	5.15E − 10	0	1326	−	5.15E − 10	0	1326	−
5.15E − 10	1326	0	+	1.31E − 05	198	1128	−	7.32E − 09	1280	46	+	5.15E − 10	1326	0	+
5.15E − 10	0	1326	−	3.87E − 05	224	1102	−	1.25E − 07	99	1227	−	6.93E − 10	1321	5	+
1.86E − 08	63	1263	−	0.002316	338	988	−	4.64E − 08	80	1246	−	5.15E − 10	1326	0	+
9.93E − 07	1185	141	+	0.004509	966	360	+	2.94E − 07	1210	116	+	5.15E − 10	1326	0	+
1.08E − 08	53	1273	−	0.055852	867	459	=	4.37E − 05	227	1099	−	8.77E − 10	1317	9	+
0.00044	1038	288	+	6.53E − 10	1322	4	+	9.87E − 10	1315	11	+	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	0.425599	748	578	=	3.52E − 09	1293	33	+	5.15E − 10	1326	0	+
7.43E − 08	89	1237	−	7.63E − 05	241	1085	−	0.119709	829	497	=	5.15E − 10	1326	0	+
5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+	5.15E − 10	1326	0	+
	16/1/11				14/3/11				16/3/9				27/0/1		

Table 5: Threat source settings.

Threat	Type	Position (km)	Radius (km)	Height
Threat 1	Rader	(35, 20)	13	2.8
Threat 2	Missile	(35, 52)	8	2.9
Threat 3	Artillery	(52, 72)	8	3.0
Threat 4	Missile	(63, 45)	10.7	2.9
Threat 5	Rader	(78, 78)	9	3.1
Threat 6	Artillery	(87, 45)	7	3.0



(a)

(b)

Figure 4: The best path of each algorithm: (a) path in three-dimensional space; (b) path in two-dimensional space.

Table 6: Statistical results of the Friedman test.

Algorithm	Mean	B_{set}	Worst	Std.	Success (%)
BOA	9.25E + 03	2.11E + 02	4.03E + 04	1.03E + 04	43.33
MRFO	6.59E + 03	2.20E + 02	3.03E + 04	8.51E + 03	56.67
SMA	2.91E + 03	2.17E + 02	2.03E + 04	6.41E + 03	83.33
GDEBOA	2.22E + 02	1.94E + 02	2.25E + 02	5.41E + 00	100.00

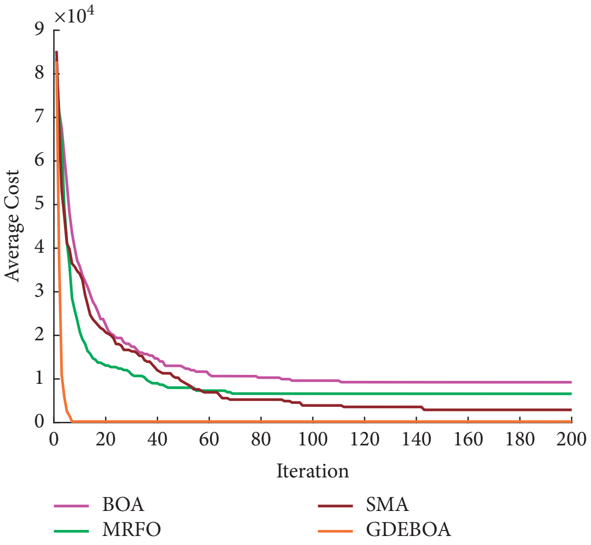


Figure 5: Convergence curves for four algorithms.

5.2. Simulation Experiment. In this subsection, we will solve the route planning model using GDEBOA and compare the results with BOA, MRFO, and SMA. All the programs were written using MATLAB R2016b and run on a Windows 10 platform with AMD R7 4700U 16 GB. For fair comparison, the parameters of each algorithm are set according to the original literature. The maximum iteration is 300, the number of populations is 50, and each algorithm is executed 30 times independently. The location of the threat is shown in Table 5. The best paths generated by MRFO, HHO, BOA, and GDEBOA are shown in Figure 4. The best values, mean, standard deviation, and success rate for 30 independent runs are shown in Table 6.

It is clear from analyzing the test results that all algorithms can give a safe flight route. This shows that the route planning model proposed in this paper is feasible. The path found by these algorithms is different as can be seen in Figure 4. Obviously, GDEBOA can find a safe path at lower cost. Also, GDEBOA can consistently find safe flight paths in 30 tests, while there are failures in the other algorithms. This indicates that GDEBOA is superior to the other three algorithms. The average cost convergence curve is plotted according to the statistical results, as shown in Figure 5. GDEBOA has faster convergence speed and better convergence accuracy.

6. Conclusions

In this paper, we propose a variant of BOA using a distribution estimation strategy, called GDEBOA, to solve the global optimization problem. The performance of BOA is enhanced by using the distribution estimation strategy to sample the evolutionary information of the dominant population and to guide the direction of individual evolution. To evaluate the effectiveness of the improved strategy and the superiority of GDEBOA, it was validated using the CEC2017 test suite. It was also compared with seven advanced algorithms through numerical analysis, convergence analysis, stability analysis, and statistical tests. The simulation results show that GDEBOA balances exploitation and exploration and is competitive with other algorithms. In addition, GDEBOA is applied to solve the UAV route planning problem. Simulation results show that GDEBOA can stably find paths with better quality, demonstrating the ability of GDEBOA to solve real-world optimization problems. Of course, there is still potential for enhancement of GDEBOA. The calculation of the covariance matrix leads to an increase in the computational cost of the algorithm. Therefore, how to reduce the cost of computing while maintaining performance is something we need to investigate further.

In future work, we plan to further apply GDEBOA to cooperative multi-UAV route planning and real-time route planning. Moreover, we plan to develop the multi-objective and binary versions of GDEBOA to solve optimization problems in other fields.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors acknowledge funding received from the following science foundations: National Natural Science Foundation of China (nos. 62101590 and 62176214) and the Science Foundation of the Shaanxi Province, China (2020JQ-481 and 2019JQ-014).

References

[1] J. Katebi, M. Shoaei-parchin, M. Shariati, N. T. Trung, and M. Khorami, “Developed comparative analysis of metaheuristic optimization algorithms for optimal active control of structures,” *Engineering Computers*, vol. 36, 2020.

[2] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, “Metaheuristic research: a comprehensive survey,” *Artificial Intelligence Review*, vol. 52, no. 5, 2019.

[3] G. Wu, W. Pedrycz, P. N. Suganthan, and R. Mallipeddi, “A variable reduction strategy for evolutionary algorithms handling equality constraints,” *Applied Soft Computing J*, vol. 37, 2015.

[4] A. Tang, H. Zhou, T. Han, and L. Xie, “A modified manta ray foraging optimization for global optimization problems,” *IEEE Access*, vol. 9, 2021.

[5] A. Di Tang, T. Han, H. Zhou, and L. Xie, “An improved equilibrium optimizer with application in unmanned aerial vehicle path planning,” *Sensors*, vol. 21, no. 5, 2021.

[6] Y. Li, T. Han, H. Zhao, and H. Gao, “An adaptive whale optimization algorithm using Gaussian distribution strategies and its application in heterogeneous UCAVs task allocation,” *IEEE Access*, vol. 7, 2019.

[7] X. Wang, H. Zhao, T. Han, H. Zhou, and C. Li, “A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem,” *Applied Soft Computing Journal*, vol. 78, 2019.

[8] M. Alweshah, S. Al Khalaileh, B. B. Gupta, A. Almomani, A. I. Hammouri, and M. A. Al-Betar, “The monarch butterfly optimization algorithm for solving feature selection problems,” *Neural Computing & Applications*, 2020.

[9] O. S. Qasim, N. A. Al-Thanoon, and Z. Y. Algamal, “Feature selection based on chaotic binary black hole algorithm for data classification,” *Chemometrics and Intelligent Laboratory Systems*, vol. 204, 2020.

[10] X. Lin and Y. Wu, “Parameters identification of photovoltaic models using niche-based particle swarm optimization in parallel computing architecture,” *Energy*, vol. 196, 2020.

[11] Q. Hao, Z. Zhou, Z. Wei, and G. Chen, “Parameters identification of photovoltaic models using a multi-strategy success-history-based adaptive differential evolution,” *IEEE Access*, vol. 8, 2020.

[12] M. Abd Elaziz, D. Yousri, M. A. A. Al-qaness, A. M. AbdelAty, A. G. Radwan, and A. A. Ewees, "A Grunwald–Letnikov based Manta ray foraging optimizer for global optimization and image segmentation," *Engineering Applications of Artificial Intelligence*, vol. 98, no. 3, 2021.

[13] A. Wunnava, M. K. Naik, R. Panda, B. Jena, and A. Abraham, "A novel interdependence based multilevel thresholding technique using adaptive equilibrium optimizer," *Engineering Applications of Artificial Intelligence*, vol. 94, 2020.

[14] H. John Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, USA, 1992.

[15] R. A. Sarker, S. M. Elsayed, and T. Ray, "Differential evolution with dynamic parameters selection for optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, 2014.

[16] J. R. Koza and J. P. Rice, "Automatic programming of robots using genetic programming," in *Proceedings of the Tenth National Conference on Artificial Intelligence*, San Jose, CA, USA, July 1992.

[17] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies–a comprehensive introduction," *Natural Computing*, vol. 1, 2002.

[18] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, 1999.

[19] S. A. Uymaz, G. Tezel, and E. Yel, "Artificial algae algorithm (AAA) for nonlinear global optimization," *Applied Soft Computing Journal*, vol. 31, 2015.

[20] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, 2008.

[21] Z. Meng and J. S. Pan, "Monkey King Evolution: a new memetic evolutionary algorithm and its application in vehicle fuel consumption optimization," *Knowledge-Based Systems*, vol. 97, 2016.

[22] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 80, 1983.

[23] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Science*, vol. 179, 2009.

[24] Z. Wei, C. Huang, X. Wang, T. Han, and Y. Li, "Nuclear reaction optimization: a novel and powerful physics-based algorithm for global optimization," *IEEE Access*, vol. 7, 2019.

[25] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm-a novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers & Structures*, vol. 110-111, 2012.

[26] S. Mirjalili, "SCA: a Sine Cosine Algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, 2016.

[27] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the ICNN'95-International Conference on Neural Networks*, Perth, WA, Australia, December 1995.

[28] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Proceedings of the 1999 Congress on Evolutionary Computation*, Washington, DC, USA, July 1999.

[29] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, 2014.

[30] G. G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing & Applications*, vol. 31, 2019.

[31] L. Xie, T. Han, H. Zhou, Z.-R. Zhang, B. Han, and A. Tang, "Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 9210050, 22 pages, 2021.

[32] Q. Fan, H. Huang, Q. Chen, L. Yao, K. Yang, and D. Huang, "A modified self-adaptive marine predators algorithm: framework and engineering applications," *Engineering Computers*, vol. 23, 2021.

[33] H. Liu, Y. Pan, S. Li, and Y. Chen, "Synchronization for fractional-order neural networks with full/under-actuation using fractional-order sliding mode control," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 7, 2018.

[34] D. Yousri, S. Mirjalili, J. A. T. Machado, S. B. Thanikanti, O. elbaksawi, and A. Fathy, "Efficient fractional-order modified Harris hawks optimizer for proton exchange membrane fuel cell modeling," *Engineering Applications of Artificial Intelligence*, vol. 100, 2021.

[35] M. Abd Elaziz, D. Yousri, and S. Mirjalili, "A hybrid Harris hawks-moth-flame optimization algorithm including fractional-order chaos maps and evolutionary population dynamics," *Advances in Engineering Software*, vol. 154, 2021.

[36] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, no. 3, 2019.

[37] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, 1997.

[38] Y. Fan, J. Shao, G. Sun, and X. Shao, "A self-adaption butterfly optimization algorithm for numerical optimization problems," *IEEE Access*, vol. 8, 2020.

[39] A. Tang, H. Zhou, T. Han, and L. Xie, "A chaos sparrow search algorithm with logarithmic spiral and adaptive step for engineering problems," *Computer Modeling in Engineering and Sciences*, vol. 100, 2021.

[40] H. Liu, S. Li, G. Lia, and H. Wang, "Robust adaptive control for fractional-order financial chaotic systems with system uncertainties and external disturbances," *Information Technology and Control*, vol. 46, 2017.

[41] Z. Wang, Q. Luo, and Y. Zhou, "Hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism for global optimization problems," *Engineering Computers*, vol. 37, 2021.

[42] E. H. Houssein, M. A. Mahdy, M. J. Blondin, D. Shebl, and W. M. Mohamed, "Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems," *Expert Systems with Applications*, vol. 174, 2021.

[43] W. Zhao, L. Wang, and Z. Zhang, "Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm," *Neural Computing & Applications*, vol. 32, no. 4, 2020.

[44] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications, Futur," *Future Generation Computer Systems*, vol. 97, 2019.

[45] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, 2021.

[46] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, 2020.

[47] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications," *Engineering Applications of Artificial Intelligence*, vol. 87, 2020.

[48] H. Yapici and N. Cetinkaya, "A new meta-heuristic optimizer: Pathfinder algorithm," *Applied Soft Computing Journal*, vol. 78, 2019.

[49] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, 2011.

[50] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization," *Journal of Heuristics*, vol. 15, no. 6, 2009.