Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://journal.nsps.org.ng/index.php/jnsps

J. Nig. Soc. Phys. Sci. 7 (2025) 2472

Journal of the Nigerian Society of Physical Sciences

Implementing a dung beetle optimization algorithm enhanced with multi-strategy fusion techniques

Xiaojie Zhou[a,b], Majid Khan Bin Majahar Ali[a,*], Farah Aini Abdullah[a], Lili Wu[a], Ying Tian[a], Tao Li[a], Kaihui Li[a]

[a]*School of Mathematical Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia*
[b]*Weifang University of Science and Technology, Shouguang 262713, China*

Abstract

The dung beetle optimization algorithm possesses robust search and optimization capabilities. However, when it encounters complex optimization challenges, it struggles with limited accuracy, restricted global search ability, and suboptimal results from iterative optimization processes. To address these limitations, this study introduces a multi-strategy improved dung beetle optimization algorithm (CDBO). Initially, the golden sine method is implemented during the rolling phase to improve the algorithm's capacity for both late area mining and early broad exploration; Then, in order to move the population closer to the ideal location during the foraging phase, the self-spiral method of the whale optimization algorithm is adopted. In the meanwhile, the present optimal location is arbitrarily perturbed during the stealing phase by introducing the flight of Levy strategy; Ultimately, the global optimal solution is modified using the dynamic t-distribution to enhance the algorithm's capacity to eliminate the regional optimal solution. This study presents simulation tests with other intelligent optimisation algorithms on 23 test functions. The outcomes demonstrate that when the dimension is 30, the enhanced method performs optimally on at least 21 test functions. The modified method still earns the top score on 22 test functions and keeps its great search capabilities when the dimension is raised to 100. The enhanced approach is applied to address K-means clustering and engineering optimization problems to further assess its potential. The findings indicate that the improved method significantly boosts both the convergence rate and the accuracy of the optimization process.

DOI:10.46481/jnsps.2025.2472

Keywords: Dung beetle optimization algorithm, Golden sine algorithm, Self-spiral strategy, Levy flight, Adaptive t-distribution

Article History :
Received: 29 December 2024
Received in revised form: 17 February 2025
Accepted for publication: 04 March 2025
Published: 29 March 2025

© 2025 The Author(s). Published by the Nigerian Society of Physical Sciences under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

Communicated by: B. J. Falaye

1. Introduction

Optimization issues [1] deal with determining an objective function's ideal value given certain limitations, and the problem has been widely used in path planning [2], power loading [3], air forecasting [4] and so on. However, these optimization problems reflect real-life black-box problems, which in some cases are difficult to solve by conventional optimization algorithms.

Metaheuristic algorithms are more effective than traditional optimization algorithms in solving complex optimization problems that conventional methods struggle with [5–9]. Metaheuristic algorithms are mainly classified into two categories, evolutionary programming techniques such as GA [10] and population intelligence optimization algorithms such as PSO,

*Corresponding author Tel. No: +604-534-782.
Email address: majidkhanmajaharali@usm.my (Majid Khan Bin Majahar Ali)

WOA [11], GWO [12], DBO [13], NGO [14], WFO. Evolutionary algorithms simulate the process of biological evolution through genetic, mutation and crossover operations to obtain optimal solutions. Population intelligence optimization algorithms find optimal solutions by simulating group behaviour within organisms. Compared to evolutionary algorithms, population intelligent optimization algorithms are algorithmically simple, easy to implement and have no diminished search for optimality. Therefore, more and more scholars are researching on the group intelligence optimization algorithm.

As the complexity of intelligent optimization problems increases, many enhanced metaheuristic algorithms have emerged. By integrating metaheuristic algorithms with deep learning techniques, researchers have improved the adaptability and global search capabilities of these algorithms. For example, the combination of deep reinforcement learning and intelligent optimization algorithms has shown great potential in feature selection and multi-objective optimization problems. Additionally, adaptive metaheuristic algorithms have been developed to address high-dimensional optimization problems. These algorithms can dynamically adjust their search strategies based on feedback during the optimization process, thereby improving convergence speed and accuracy in complex problems.

Hybrid optimization algorithms have also become a research hotspot in recent years. They combine the strengths of various optimization techniques to avoid the shortcomings of single algorithms. Researchers have proposed more efficient hybrid strategies by combining particle swarm optimization with simulated annealing, genetic algorithms, and other techniques, widely applying them in high-dimensional and multi-objective optimization problems.

Furthermore, with the increasing computational demands, parallel computing and distributed computing methods have been introduced into intelligent optimization algorithms. This not only improves the computational efficiency of the algorithms but also enables them to handle larger-scale and more complex optimization problems. With GPU acceleration and distributed processing, the performance of optimization algorithms in handling large-scale datasets has been significantly enhanced.

DBO also suffers from limitations common to swarm intelligence optimization algorithms, such as low population diversity, slow convergence, poor convergence accuracy, and poor stability [15–22]. Therefore, many researchers have improved DBO:

An optimization approach for dung beetles based on quantum computing and multi-policy fusion was proposed by Zhu *et al.* [19].

1. Utilize the point set technique to initialize the population and ensure uniform distribution.
2. Convergence factor and dynamic equilibrium are proposed in the spawning and foraging phases. Enhance the searching ability of the algorithm.
3. Incorporate the quantum computing t-distribution strategy to aid the algorithm in avoiding entrapment in local optima.

Wang *et al.* [20] proposed an improved DBO algorithm tailored for Sinh-Cosh functions.

1. Utilize the Cosh and Sinh routines to revert to the original distribution of dung beetles, and
2. uses nonlinear augmentation to increase the algorithm's search efficiency.

The Multi-Strategy Enhanced DBO was proposed by Ye [21] and colleagues which

1. uses Latin hypercubic sampling for initialisation,
2. introduces an innovative differential variational strategy to improve the algorithm's ability to resist becoming trapped in local optima.
3. Applies dimension-by-dimension optimization techniques along with inverse learning based on the current optimal solution to increase the algorithm's accuracy.

Wang introduced an enhanced Dung Beetle Optimization Algorithm [22], that

1. combines catenary graphs and dyadic learning strategies.
2. To enhance the efficiency of the rolling phase, combines the DBO position update approach with the worldwide exploration strategy of the Osprey optimization algorithm.
3. Enhances the diversity of the population in the foraging phase by combining vertical and horizontal crossing of individuals.

However, these methods have the following shortcomings:

- Limited global search capability: SC-DBO mainly relies on the Sinh-Cosh transformation, which lacks sufficient global exploration ability in the early search stage, making it prone to getting trapped in local optima.
- Insufficient local search accuracy: Although the differential mutation strategy used in MS-DBO can optimize local search, it lacks adequate global exploration mechanisms, leading to unstable convergence accuracy in complex problems.
- Lower stability and robustness: In high-dimensional problems, the performance of these improved methods declines significantly, and they cannot guarantee optimal results in high-dimensional spaces.

To overcome these issues, we propose the CDBO algorithm, which enhances the performance of the DBO algorithm at different optimization stages by integrating various strategies (Golden Sine search, Self-Helix search, Levy flight, and Adaptive t-distribution), enabling it to maintain superior convergence performance in both low-dimensional and high-dimensional optimization problems.

The main contributions of this paper are as follows:

- During the rolling stage, the Golden Sine optimization is used to simultaneously enhance both global and local search capabilities.

- In the foraging stage, the Self-Helix Strategy is employed to improve the convergence accuracy and speed of the population towards the optimal solution.
- In the stealing stage, the Levy Flight strategy is introduced to enhance the algorithm's global search ability and reduce the risk of premature convergence.
- During the global optimum update process, the Adaptive t-distribution is utilized to optimize the convergence stability in high-dimensional problems.
- Uses CDBO to K-means clustering and engineering optimization.

This paper's basic structure is as follows. In the following section, the DBO is presented. The third part contains, the CDBO is built. Part four compares the performance of the CDBO algorithm with alternative techniques through simulation experiments on 23 test functions to verify the effectiveness of the improvements. In order to investigate the practicality of the enhanced method, the fifth section uses the CDBO algorithm to actual engineering applications, and the sixth section applies the CDBO algorithm to K-means clustering. The entire job is concluded in Section 7.

2. Dung Beetle Optimization (DBO)

Dung Beetle Optimization (DBO) primarily updates its location by mimicking the four natural actions of dung beetles: rolling, egg deposition, food searching, and theft.The four dung beetle species account for 20%, 20%, 25%, and 35% of the population, respectively.

2.1. Rolling

Equation (1) serves to update rolling dung beetles.

$$\begin{cases} x_i(t+1) = x_i(t) + \alpha \times k \times x_i(t-1) + b \times \Delta x, \\ \Delta x = |x_i(t) - X^{\omega}|, \end{cases} \tag{1}$$

where b is a constant within the interval (0,1); α can be either -1 or 1; X^{ω} represents the globally least favorable position; Δx models the variation in light intensity.

A dung beetle uses the following formula to update its place of origin when it comes across an impediment.

$$x_i(t+1) = x_i(t) + \tan\theta |x_i(t) - x_i(t-1)|, \tag{2}$$

where the symbol θ denotes the tilt angle.

2.2. Egg deposition

Female dung beetles dynamically spawn with regional locations updated below:

$$\begin{cases} Lb^* = \max(X^* \times (1-R), Lb), \\ Ub^* = \min(X^* \times (1+R), Ub). \end{cases} \tag{3}$$

The current local optimum is represented by X^* ; the lower and upper limits are denoted as Lb^* and Ub^*, respectively; and $R = 1 - t/T_{\max}$.

Given the dynamic characteristics of the spawning region, the position of the hatching dung beetles is updated as follows:

$$B_i(t+1) = X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*), \tag{4}$$

where $B_i(t)$ represents the positional data of the i-th hatching ball dung beetle at the t-th iteration; the symbols b_1 and b_2 indicate two distinct variation vectors, each with dimensions $1 \times D$.

2.3. Food searching

The following defines the limits of the little dung beetle feeding area:

$$\begin{cases} Lb^b = \max(X^b \times (1-R), Lb), \\ Ub^b = \min(X^b \times (1+R), Ub), \end{cases} \tag{5}$$

where X^b indicates the globally optimal position, while Lb^b and Ub^b correspond to the lower and upper limits, respectively. The position of the Little Dung Beetle has been revised as follows.

$$x_i(t+1) = x_i(t) + C_1 \times (x_i(t) - Lb^b) + C_2 \times (x_i(t) - Ub^b). \tag{6}$$

C_1 and C_2 denote random numbers and random vectors, respectively.

2.4. Theft

In neighbouring dung beetles, thieves have access to dung balls. The following is the amended value after each repetition.

$$x_i(t+1) = X^b + S \times g \times (|x_i(t) - X^*| + |x_i(t) - X^b|), \tag{7}$$

where g represents a randomly generated vector; S denotes a constant.

3. Adjusting the dung beetle for greater outcomes(CDBO)

3.1. CDBO algorithm flow chart

The following is the flowchart of CDBO algorithm using multiple strategies as shown in Figure 1.

3.2. Technique for golden sine optimization

The method for optimizing golden sine [23] adopts the use of sinusoidal function for iterative optimization search, at the same time, Incorporating the golden section coefficient into the update process enhances the algorithm's global search ability during the early stages of iteration, and in the later period of the iteration period, it can be sufficiently searched in the local region. The position update formula utilizing the golden sine approach in the rolling period is as follows:

$$x_i(t+1) = x_i(t) \cdot |\sin r_1| - r_2 \cdot \sin r_1 \cdot |c_1 x^b - c_2 x_i(t)|, \tag{8}$$

where r_1 is a random number for the distance travelled; r_2 is a random number for the update direction; g_1 is the golden section number, c_1 and c_2 are a constant.

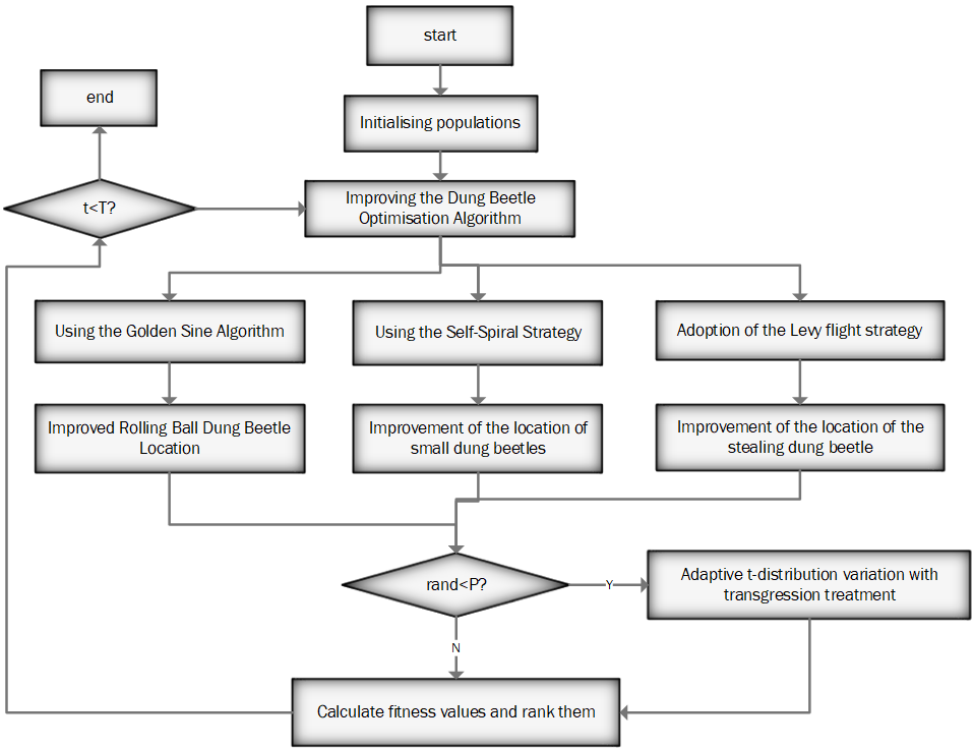

Figure 1: Flowchart of CDBO algorithm.

Table 1: Test functions.

Function	Function name	Function expression	Features	Dimension	Search range	Optimum value
F1	Shifted Sphere	$F_1 = \sum_{i=1}^{n} x_i^2$.	Single-peak	30	$[-100, 100]$	0
F2	Schwefel's 1.2	$F_2 = \sum_{i=1}^{n} i \cdot x_i^2$.	Single-peak	30	$[-10, 10]$	0
F3	Schwefel's 2.22	$F_3 = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$.	Multi-peak	30	$[-10, 10]$	0
F4	Quartic	$F_4 = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$.	Single-peak	30	$[-100, 100]$	0
F5	Generalized Schwefel's 2.26	$F_5 = \max_{1 \le i \le n} \lvert x_i \rvert$.	Single-peak	30	$[-100, 100]$	0
F6	Rosenbrock's	$F_6 = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$.	Multi-peak	30	$[-30, 30]$	0
F7	Penalized	$F_7 = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2(1 + \sin^2(3\pi x_{i+1})) + (x_n - 1)^2(1 + \sin^2(3\pi x_n))\right\} + \sum_{i=1}^{n} U(x_i, 5, 100, 4)$	Multi-peak	30	$[-50, 50]$	0
F8	Quartic with Noise	$F_8 = \sum_{i=1}^{n} i \cdot x_i^4$.	Single-peak	30	$[-1.28, 1.28]$	0
F9	Noisy Quartic	$F_9 = \sum_{i=1}^{n} i \cdot x_i^4 + rand$.	Single-peak	30	$[-1.28, 1.28]$	0
F10	Different Powers	$F_{10} = \sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$.	Single-peak	30	$[-1, 1]$	0
F11	Rastrigin	$F_{11} = \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$.	Multi-peak	30	$[-5.12, 5.12]$	0
F12	Ackley	$F_{12} = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$.	Multi-peak	30	$[-32, 32]$	0
F13	Griewank	$F_{13} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$.	Multi-peak	30	$[-600, 600]$	0
F14	Modified Schwefel	$F_{14} = \sum_{i=1}^{n}\left[(x_i - 1)^2\left(1 + \sin^2(3\pi x_{i+1})\right)\right] + \sin^2(3\pi x_1) + \lvert x_n - 1\rvert\left(1 + \sin^2(2\pi x_n)\right)$.	Multi-peak	30	$[-10, 10]$	0
F15	Sinusoidal	$F_{15} = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1 x_i\rvert$.	Multi-peak	30	$[-10, 10]$	0
F16	Sum of Cosine Waves	$F_{16} = 0.1\ln\left(0.1\sum_{i=1}^{n}\cos(5\pi x_i) - \sum_{i=1}^{n} x_i^2\right)$.	Multi-peak	30	$[-1, 1]$	0
F17	Generalized Quartic	$F_{17} = \sum_{i=1}^{n} x_i^2 + \left(0.5\sum_{i=1}^{n} i \cdot x_i\right)^2 + \left(0.5\sum_{i=1}^{n} i \cdot x_i^4\right)$.	Multi-peak	30	$[-5, 10]$	0
F18	Hybrid Composition	$F_{18} = \sum_{i=1}^{n-1}\left[0.5 + \frac{\sin^2\left(\sqrt{100x_i^2 + x_{i+1}^2}\right) - 0.5}{1 + 0.001\left(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2\right)}\right]$.	Multi-peak	30	$[-100, 100]$	0
F19	Penalized 2	$F_{19} = 0.1\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2\left(1 + \sin^2(3\pi x_{i+1})\right) + (x_n - 1)^2\left(1 + \sin^2(2\pi x_n)\right)$.	Multi-peak	30	$[-5, 5]$	0
F20	Elliptic	$F_{20} = \sum_{i=1}^{n}(10^6)^{\frac{i-1}{n-1}} x_i^2$.	Single-peak	30	$[-100, 100]$	0
F21	Gaussian	$F_{21} = (-1)^{n+1}\prod_{i=1}^{n}\cos(x_i)\exp\left(-\sum_{i=1}^{n}(x_i - \pi)^2\right)$.	Multi-peak	30	$[-100, 100]$	0
F22	Shifted Sphere with Oscillations	$F_{22} = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{n} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{n} x_i^2}$.	Single-peak	30	$[-100, 100]$	0
F23	Shifted Rosenbrock	$F_{23} = 0.5 + \frac{\sin^2\left(\sqrt{\sum_{i=1}^{n} x_i^2}\right) - 0.5}{\left(1 + 0.001\sum_{i=1}^{n} x_i^2\right)^2}$.	Single-peak	30	$[-100, 100]$	0

3.3. *Self-spiral strategy*

In foraging behaviour, dung beetles search for superiority slowly, which is not conducive to the latter step of the technique's completion. Inspired by the WOA's spiral search approach, the algorithm is enhanced, and the position update formula is presented as follows:

Table 2: Parameters of each optimisation algorithm.

Algorithm	Parameter name	Parameter symbol	Value range
GWO	Systolic expansion parameters	a	2-0 Linear reduction
	Shrinkage parameters	a	2-0 Linear reduction
WOA	Spiral constant	b	Defining Spiral Patterns
	Random number	l	Control Spiral Update Position
NGO	Prey range	d	0.1-2
	Learning rate	c	0.1-1
	Perceptual radius	r	0.1-2
DBO	Ball size	Dung_ball_size	0.1-2

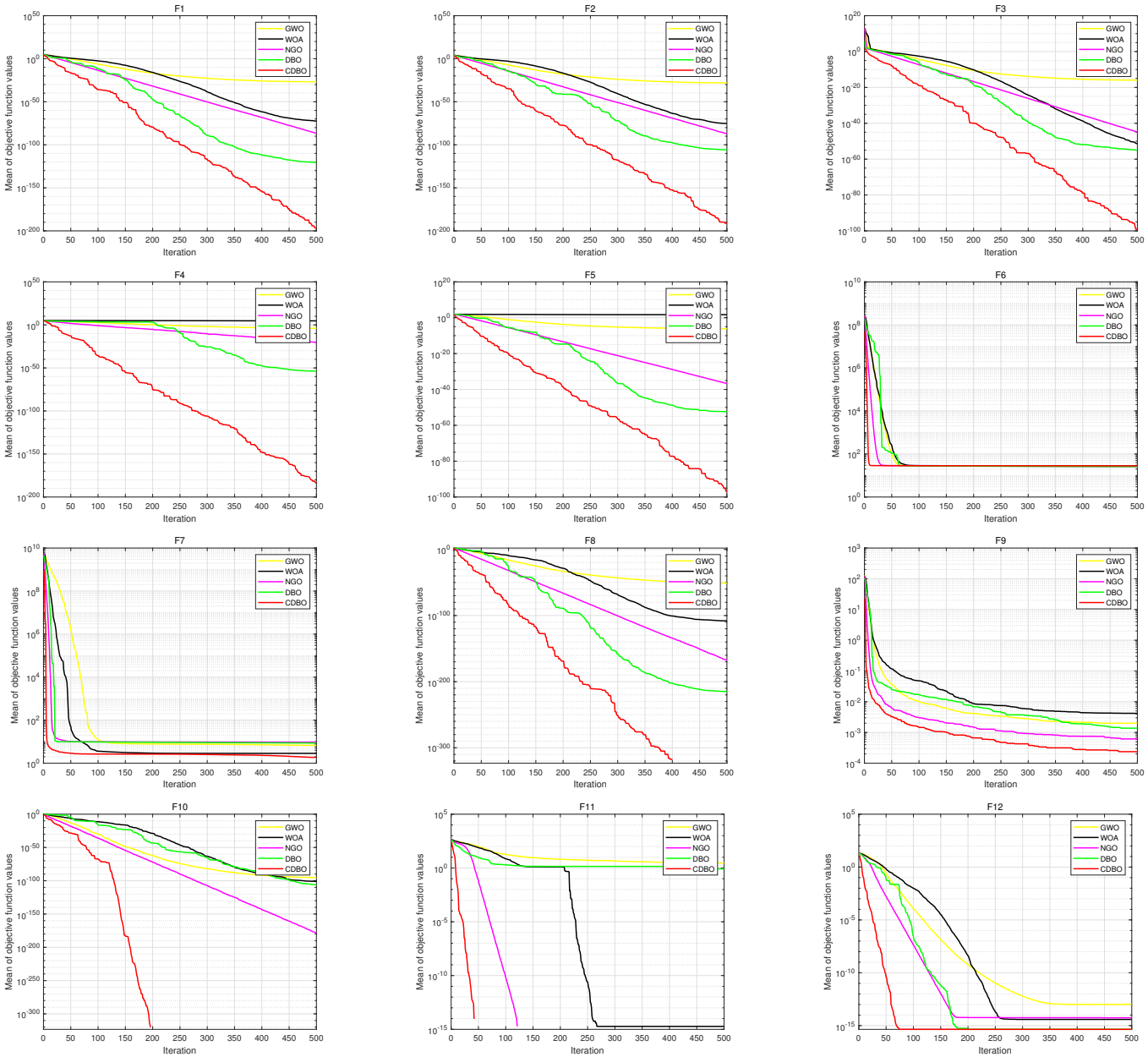

Figure 2: Comparison of convergence curves of different algorithms(30-dimension, part 1).

$$x_i(t+1) = e^{zl} \cdot \cos(2\pi l) \cdot x_i(t) + c_1(x_i(t) - Lb^b) + c_2(x_i(t) - Ub^b), \quad (9)$$

Table 3: Comparison of optimization performance of different algorithms on test functions (30-dimension, part 1).

Function	Index	GWO	WOA	NGO	DBO	CDBO
F1	average	1.19e-27	2.05e-70	8.52e-88	7.12e-119	1.55e-201
	std	1.13e-69	1.12e-69	1.84e-87	3.47e-118	0
	optimal	1.91e-29	2.52e-83	1.63e-89	1.20e-66	0
	worse	4.46e-27	6.16e-69	8.91e-87	1.89e-117	4.64e-200
F2	average	1.83e-28	1.00e-70	9.44e-89	4.04e-109	2.67e-189
	std	2.87e-28	3.11e-70	1.16e-88	1.53e-108	0
	optimal	5.77e-30	8.73e-85	8.70e-91	1.69e-159	0
	worse	1.10e-27	1.17e-69	4.76e-88	6.51e-108	8.03e-188
F3	average	1.11e-16	9.23e-52	1.70e-45	4.41e-60	1.35e-96
	std	8.57e-17	3.65e-51	3.15e-45	1.63e-59	5.14e-96
	optimal	2.37e-17	5.29e-57	1.07e-46	1.84e-79	0
	worse	4.15e-16	2.00e-50	1.74e-44	7.46e-59	2.12e-95
F4	average	1.01e-05	4.66e+04	1.29e-21	3.64e-55	1.72e-195
	std	3.40e-05	1.49e+04	5.57e-21	1.99e-43	0
	optimal	1.53e-09	1.50e+04	3.94e-28	3.60e-151	0
	worse	1.87e-04	7.06e+04	3.05e-20	1.09e-53	4.51e-194
F5	average	6.18e-07	5.89e+01	2.70e-37	8.27e-51	3.55e-103
	std	4.74e-07	2.67e+01	5.39e-37	4.52e-50	1.79e-102
	optimal	1.40e-07	6.41e+00	6.52e-39	4.50e-77	0
	worse	2.13e-06	9.19e+01	2.95e-36	2.47e-49	9.82e-102
F6	average	2.70e+01	2.79e+01	2.59e+01	2.57e+01	2.81e+01
	std	7.70e-01	4.51e-01	6.17e-01	1.79e-01	2.50e+01
	optimal	2.55e+01	2.72e+01	2.48e+01	2.54e+01	2.78e+01
	worse	2.85e+01	2.87e+01	2.73e+01	2.63e+01	2.86e+01
F7	average	7.04e-01	5.33e-01	3.15e-01	5.76e-01	3.15e-01
	std	2.04e-01	2.90e-01	2.75e-01	4.55e-01	1.98e-01
	optimal	3.53e-01	7.84e-02	1.17e-02	2.49e-04	7.58e-03
	worse	1.14e+00	1.41e+00	1.07e+00	1.58e+00	7.63e-03
F8	average	3.85e-51	7.42e-117	4.07e-169	1.87e-217	0
	std	1.37e-50	3.00e-116	0	0	0
	optimal	9.48e-55	2.34e-130	8.16e-176	8.10e-318	0
	worse	7.49e-50	1.53e-115	4.08e-168	5.63e-216	0
F9	average	1.82e-03	3.73e-03	5.81e-04	1.55e-03	2.17e-04
	std	8.85e-04	3.88e-03	3.12e-04	1.11e-03	1.86e-04
	optimal	4.53e-04	2.15e-05	1.14e-04	2.26e-04	5.30e-07
	worse	3.80e-03	1.37e-02	1.34e-03	3.93e-03	8.42e-04
F10	average	3.52e-98	7.91e-103	4.12e-180	5.38e-109	0
	std	1.84e-97	4.33e-102	0	2.94e-108	0
	optimal	1.25e-108	1.92e-136	2.51e-187	8.77e-180	0
	worse	1.00e-96	2.37e-101	5.69e-179	1.61e-107	0
F11	average	3.15e+00	1.89e-15	0	0	0
	std	3.87e+00	1.03e-14	0	5.81e+00	0
	optimal	1.13e-13	0	0	0	0
	worse	1.29e+01	5.68e-14	0	3.18e+01	0
F12	average	1.01e-13	4.11e-15	6.01e-15	4.44e-16	4.44e-16
	std	1.51e-14	2.37e-15	1.79e-15	0	0
	optimal	7.50e-14	4.44e-16	3.99e-15	4.44e-16	4.44e-16
	worse	1.46e-13	7.54e-15	7.54e-15	4.44e-16	4.44e-16

where z is a constant for the spiral equation; Random number l lies in the interval [-1,1].

3.4. Levy flying

Due to the lack of interactive behaviour between peers in the stealing behaviour, slipping into the local optimum is simple. in order to solve this problem, in the Cuckoo algorithm, we use the

Table 3: Comparison of optimization performance of different algorithms on test functions (30-dimension, part 2).

Function	Index	GWO	WOA	NGO	DBO	CDBO
F13	average	6.15e-03	7.48e-03	0	0	0
	std	1.35e-02	4.09e-02	0	0	0
	optimal	0	0	0	0	0
	worse	5.50e-02	2.24e-01	0	0	0
F14	average	2.23e+00	2.72e+01	9.55e-01	1.98e+00	1.59e+00
	std	1.55e+00	4.13e+01	5.43e-01	3.27e+00	2.17e+00
	optimal	4.23e-01	1.09e-01	2.37e-01	8.57e-03	4.65e-02
	worse	8.03e+00	1.46e+02	2.15e+00	1.59e+02	7.44e+02
F15	average	3.96e-04	6.52e-01	2.76e-46	9.73e-02	7.97e-102
	std	6.18e-04	3.57e+00	5.41e-46	4.06e-01	3.60e-101
	optimal	6.78e-17	1.77e-58	1.74e-47	3.73e-79	0
	worse	2.27e-03	1.95e+01	2.85e-45	2.09e+00	1.93e-100
F16	average	0	0	0	0	0
	std	0	0	0	0	0
	optimal	0	0	0	0	0
	worse	0	0	0	0	0
F17	average	1.71e-07	4.88e+02	1.47e-08	4.24e-26	4.77e-196
	std	367e-07	8.65e+01	4.74e-08	2.32e-25	0
	optimal	5.98e-10	3.32e+02	1.30e-11	2.72e-99	0
	worse	1.65e-06	6.86e+02	2.55e-07	1.27e+24	1.43e-194
F18	average	1.11e+01	4.38e-02	6.46e+00	1.10e+00	0
	std	5.37e-01	1.19e-01	6.50e-01	8.09e-01	0
	optimal	1.00e+01	0	4.90e+00	5.63e-04	0
	worse	1.19e+01	5.01e-01	7.70e+00	2.74e+00	0
F19	average	3.64e+00	1.29e+00	7.91e-01	2.88e+00	6.10e-01
	std	1.94e+00	1.05e+00	1.02e+00	3.78e+00	8.41e-01
	optimal	3.70e-04	1.56e-01	1.83e-04	4.15e-04	2.17e-03
	worse	7.58e+00	4.19e+00	4.14e+00	1.68e+01	3.18e+00
F20	average	5.57e-24	7.78e-67	6.22e-84	7.63e-111	2.34e-195
	std	1.09e-23	4.25e-66	8.71e-84	4.18e-110	0
	optimal	2.84e-26	2.33e-81	2.28e-86	1.15e-171	0
	worse	4.77e-23	2.33e-65	3.28e-83	2.29e-109	7.03e-194
F21	average	0	0	0	0	0
	std	0	0	0	0	0
	optimal	0	0	0	0	0
	worse	0	0	0	0	0
F22	average	1.93e-01	1.29e-01	9.98e-02	8.65e-02	1.02e-98
	std	2.53e-02	6.51e-02	4.78e-13	3.45e-02	5.06e-98
	optimal	9.98e-02	6.65e-36	9.98e-03	2.73e-54	0
	worse	1.99e-01	2.99e-01	9.98e-02	9.98e-02	2.77e-97
F23	average	3.53e-02	2.31e-02	9.71e-03	9.71e-03	0
	std	6.97e-03	2.00e-02	6.80e-14	9.45e-08	0
	optimal	9.71e-03	0	9.71e-03	9.71e-03	0
	worse	3.72e-02	7.81e-02	9.71e-03	9.71e-03	0

Levy's approach [24]. Enhance the exploration of space and enhance the overall perfect search function of the algorithm. Levy's flight strategy is as follows.

$$\text{Levy}(x) = 0.01 \times \frac{r^3 \times \sigma}{|r^4|^{(1/\varepsilon)}}. \tag{10}$$

Two random integers in this range [0,1] are r_3 and r_{34} , is a randomly generated number within the range of 0 to 2, and σ is computed as follows.

$$\sigma = \left(\frac{\Gamma(1+\xi) \times \sin(\pi\xi/2)}{\Gamma((1+\varsigma)/2) \times \xi \times 2^{((\xi-1)/2)}} \right)^{(1/\xi)}, \tag{11}$$

where $\Gamma(x) = (x-1)!$.

Table 4: Comparison of optimization performance of different algorithms on test functions (100-dimension, part 1)

Function	Index	GWO	WOA	NGO	DBO	CDBO
F1	average	1.44e-12	8.70e-73	1.81e-82	1.75e-111	5.56e-201
	std	9.55e-13	3.35e-72	7.42e-82	9.59e-111	0
	optimal	3.46e-13	7.16e-84	5.13e-85	1.08e-158	0
	worse	4.37e-12	1.64e-71	4.08e-81	5.25e-110	1.65e-199
F2	average	4.63e-13	1.66e-71	6.00e-83	1.47e-113	6.02e-197
	std	3.72e-13	5.91e-71	1.77e-82	8.05e-113	0
	optimal	8.77e-14	2.03e-84	2.25e-85	2.22e-159	8.10e-269
	worse	1.66e-12	2.94e-70	8.45e-82	4.41e-112	1.80e-195
F3	average	4.35e-08	1.83e-47	2.10e-43	2.03e-54	8.95e-102
	std	1.73e-08	9.98e-47	1.85e-43	1.11e-53	4.83e-101
	optimal	1.91e-08	1.71e-56	2.87e-44	2.34e-88	0
	worse	8.61e-08	5.46e-46	7.34e-43	6.11e-53	2.65e-100
F4	average	5.28e+02	1.03e+06	1.17e-10	7.74e-05	8.86e-195
	std	5.11e+02	2.52e+05	4.14e-10	4.24e-04	0
	optimal	1.26e+01	5.60e+05	3.37e-18	2.22e-139	0
	worse	2.23e+03	1.61e+06	2.15e+09	2.32e-03	2.64e+193
F5	average	9.88e-01	7.61e+01	3.97e-34	2.00e-47	7.07e-99
	std	1.23e+00	2.28e+01	3.68e-34	1.09e-46	3.87e-98
	optimal	8.65e-02	2.02e+01	9.59e-35	2.67e-84	0
	worse	5.57e+00	9.64e+01	1.56e+33	5.97e-46	2.12e-97
F6	average	9.80e+01	9.81e+01	9.73e+01	9.71e+01	9.76e+01
	std	5.01e-01	2.11e-01	5.91e-01	7.11e-01	2.82e-01
	optimal	9.69e+01	9.75e+01	9.62e+01	9.60e+01	9.71e+01
	worse	9.85e+01	9.84e+01	9.83e+01	9.82e+01	9.81e+01
F7	average	6.83e+00	2.85e+00	9.56e+00	8.76e+00	1.80e+00
	std	4.64e-01	1.14e+00	1.01e+00	4.44e-01	1.04e+00
	optimal	6.17e+00	8.67e+00	6.13e+00	7.61e+00	1.94e-01
	worse	8.02e+00	6.29e+00	9.92e+00	9.46e+01	4.33e+00
F8	average	4.72e-25	5.21e-102	2.20e-159	4.02e-212	0
	std	1.23e-24	2.85e-101	7.78e-159	0	0
	optimal	6.17e-27	1.41e-129	3.01e-164	2.06e-318	0
	worse	6.68e-24	1.56e-100	4.17e-158	1.20e-210	0
F9	average	7.72e-03	3.57e-03	7.23e-04	1.22e-03	2.23e-04
	std	2.32e-03	3.24e-03	3.51e-04	7.59e-04	1.51e-04
	optimal	2.93e-03	1.09e-04	1.42e-04	2.59e-04	1.44e-05
	worse	1.38e-02	1.55e-02	1.41e-03	3.00e-03	5.41e-04
F10	average	3.54e-60	1.50e-104	3.63e-180	3.80e-118	0
	std	1.82e-59	6.62e-104	0	2.08e-117	0
	optimal	1.96e-82	3.13e-130	1.06e-186	4.97e-178	0
	worse	9.99e-59	3.54e-103	3.82e-179	1.14e-116	0
F11	average	1.05e+01	7.57e-15	0	0	0
	std	6.64e+00	4.15e-14	0	0	0
	optimal	3.69e-10	0	0	0	0
	worse	3.42e+01	2.27e-13	0	0	0
F12	average	1.28e-07	3.99e-15	6.83e-15	6.80e-16	4.44e-16
	std	4.25e-08	2.08e-15	1.44e-15	9.01e-16	0
	optimal	7.23e-08	4.44e-16	3.99e-15	4.44e-16	4.44e-16
	worse	2.79e-07	1.46e-14	7.54e-15	3.99e-15	4.44e-16

3.5. Adaptive t-ditribution

For high dimensionality and high complexity objective function, in the late iteration, it is very easy to ignore the global optimal position, adaptive t-distribution [25] can be perturbed to the current position, to improve the algorithm's ability to avoid becoming trapped in local optima, and to improve the algorithm's rate of convergence and efficiency when tackling

Table 4: Comparison of optimization performance of different algorithms on test functions(100-dimension, part 2).

Function	Index	GWO	WOA	NGO	DBO	CDBO
F13	average	1.71e-03	2.29e-02	0	0	0
	std	6.58e-03	9.00e-02	0	0	0
	optimal	2.05e-13	0	0	0	0
	worse	2.85e-02	4.25e-01	0	0	0
F14	average	3.10e+01	2.78e+01	1.76e+01	2.33e+01	1.24e+01
	std	6.57e+00	5.78e+00	1.12e+01	5.01e+00	8.38e+00
	optimal	1.71e+01	2.07e+00	4.81e+00	1.46e+01	2.36e-01
	worse	4.96e+01	2.73e+02	4.35e+01	3.53e+01	2.88e+01
F15	average	3.70e-03	4.62e-52	3.03e-44	8.94e-04	1.02e-100
	std	2.48e-03	1.44e-51	4.06e-44	4.89e-33	5.09e-100
	optimal	5.23e-08	7.69e-59	2.32e-45	1.81e-84	0
	worse	9.97e-03	7.69e-51	1.57e-43	2.68e-02	2.78e-99
F16	average	6.74e-15	0	0	0	0
	std	4.26e-14	0	0	0	0
	optimal	2.84e-14	0	0	0	0
	worse	6.21e-14	0	0	0	0
F17	average	1.19e+02	1.67e+03	2.28e+01	3.15e+01	4.37e-195
	std	5.42e+01	2.23e+02	1.06e+01	9.55e+01	0
	optimal	2.77e+01	1.18e+03	8.57e+00	3.67e+01	0
	worse	2.51e+02	2.17e+03	4.45e+01	4.51e+02	1.31e+193
F18	average	4.46e+01	3.62e+01	2.59e+01	4.63e+01	0
	std	1.02e+00	7.44e-01	9.67e-01	3.31e+00	0
	optimal	4.13e+01	0	2.35e+01	1.69e-01	0
	worse	4.58e+01	3.13e+00	2.75e+01	9.92e+01	0
F19	average	5.60e+01	5.05e+00	8.71e+01	5.32e+01	4.98e+00
	std	4.22e+00	3.48e+00	1.52e+01	3.73e+01	3.56e+00
	optimal	4.67e+01	1.70e+00	5.29e+01	2.14e+01	5.18e-02
	worse	6.49e+01	1.50e+01	9.92e+01	9.39e+01	1.34e+01
F20	average	2.43e+09	5.90e-69	5.22e-79	7.28e-105	2.46e-191
	std	2.32e-09	2.65e-68	9.45e-79	3.98e-104	0
	optimal	4.51e-10	4.23e-83	6.07e-81	8.14e-160	0
	worse	1.24e-08	1.44e-67	4.53e-78	2.18e-103	2.46e-191
F21	average	0	0	0	0	0
	std	0	0	0	0	0
	optimal	0	0	0	0	0
	worse	0	0	0	0	0
F22	average	3.69e-01	1.03e-01	1.03e-01	8.86e-02	2.68e-99
	std	5.59e-02	7.18e-02	1.82e-02	3.07e-02	1.08e-98
	optimal	2.99e-01	6.22e-39	9.98e-02	8.96e-43	0
	worse	4.99e-01	2.99e-01	1.99e-01	9.98e-02	5.50e-98
F23	average	8.79e-02	1.65e-02	1.15e-02	9.71e-03	0
	std	1.98e-02	1.67e-02	6.97e-03	2.43e-07	0
	optimal	7.81e-02	0	9.71e-03	9.71e-03	0
	worse	1.26e-01	7.81e-02	3.72e-02	9.71e-03	0

Table 5: Comparison of different improved optimisation algorithms.

Algorithm	Running Phase	Food Phase	Stubborn Phase	Local Search Accuracy	Global Search Ability	High-dimensional Stability
SC-DBO	Sinh-Cosh Transformation	Standard	Standard	General	General	Decrease
MS-DBO	Standard	Differential Variation	Reinforcement Learning	High	General	Decrease
CDBO	Golden Precision	Self-tuning Search	Levy Flying	High	High	Stable

issues, the specific advances are as follows:

$$x_i^{t+1} = x_i^t + x_i^t \cdot t(iter). \tag{12}$$

Meanwhile, to reduce the time spend computing, the dynamic selection of the probability p is used to automatically regulate

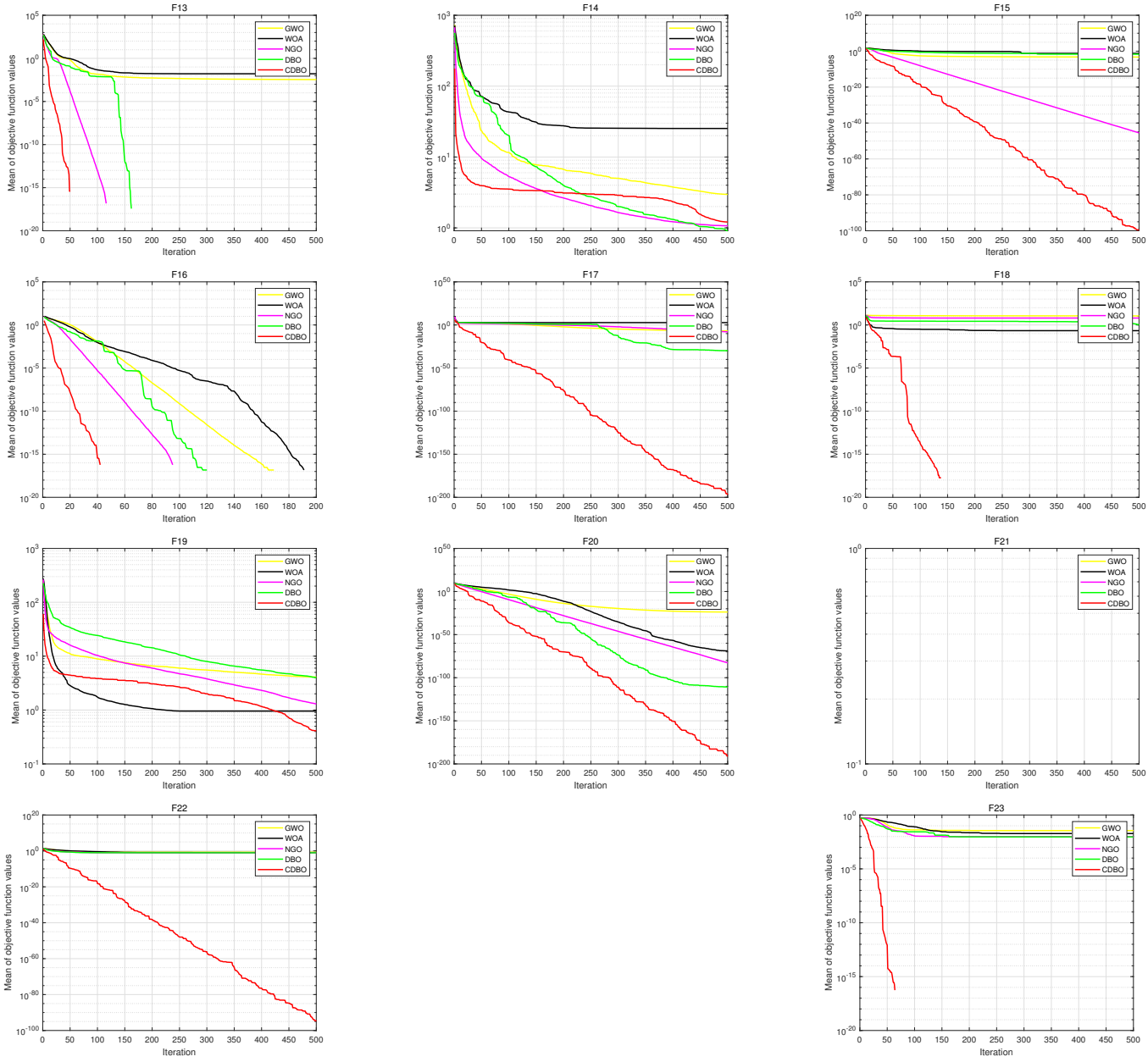

Figure 2: Comparison of convergence curves of different algorithms(30-dimension, part 2).

the use of the variance operator, which is formulated as follows:

$$p = w_1 - w_2 \cdot \frac{T - t}{T}, w_1 = 0.5, w_2 = 0.1. \tag{13}$$

The current iteration count is t , while the total number of iterations is T.

4. Experimental foundations

4.1. Test function and experimental settings

In this paper, four optimisation algorithm are employed to assess the performance of the CDBO algorithm across 23 test functions. and Table 1 gives the crucial information of the 23 test functions. Table 2 displays the parameters used by the two different algorithms. To reduce experimental randomness, the four optimisation algorithm and the CDBO algorithm were each independently run 30 times on each test function. The mean, standard deviation, best value, and worst value of the results were recorded. The programming environment is MATLABR2024a.

4.2. Analysis of experimental results

The stability, accuracy, and convergence speed of the algorithm are assessed and analyzed in the following sections.

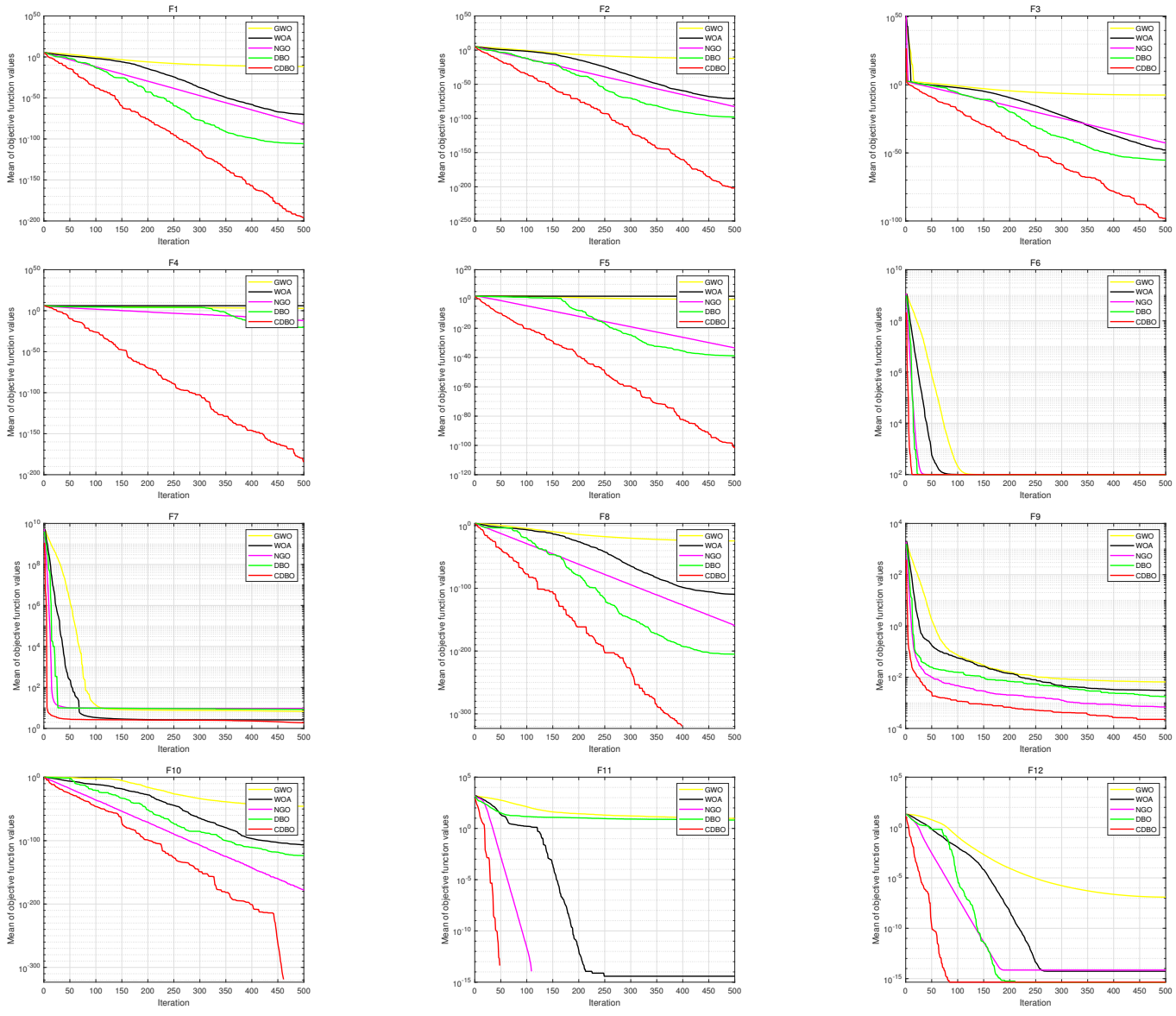

Figure 3: Comparison of convergence curves of different algorithms (100-dimension, part 1).

4.2.1. Accuracy

Tables 3 and 4 display the average, standard deviation, best, and worst values for the five approaches after thirty independent runs on the twenty-three test functions with dimensions of thirty and one hundred. respectively, where the top outcomes of the five algorithms can be seen by the marked data. The details are as follows:

In Table 3, F6 is a typical nonlinear, multi-modal function with a very narrow and curved global optimum path. The global optimum is usually located in a very narrow region, posing a significant challenge for optimization algorithms.While CDBO's global search capabilities (such as Levy flights) effectively help the algorithm escape local optima, when faced with the narrow optimization path of the Rosenbrock function, the jumps made by CDBO can be too aggressive. This leads to an inability to effectively focus on the narrow global optimum region, which affects its accuracy. In addition, the leap step size problem and insufficient local search also have an impact.

In the multi-peak function F14, NGO achieves the ideal mean and std; CDBO is the next best. On the multi-peak functions F16 and F21, all functions reach optimal values. Multi-peak functions F7 and F19 contain penalty terms for multiple local minima, and after 30 independent runs, DBO achieves the best ranking on the optimal value, but CDBO achieves the best results on the mean, std, and worst, proving the stability and robustness of the CDBO.

When the dimension is raised from 30 to 100 dimensions, Table 4 displays the corresponding simulation experiment findings. From Table 4, it can be found that the CDBO algorithm similarly achieves functionally optimal results in 21 out of 23 test functions.

When the dimension rises to 100, the test function F6 still

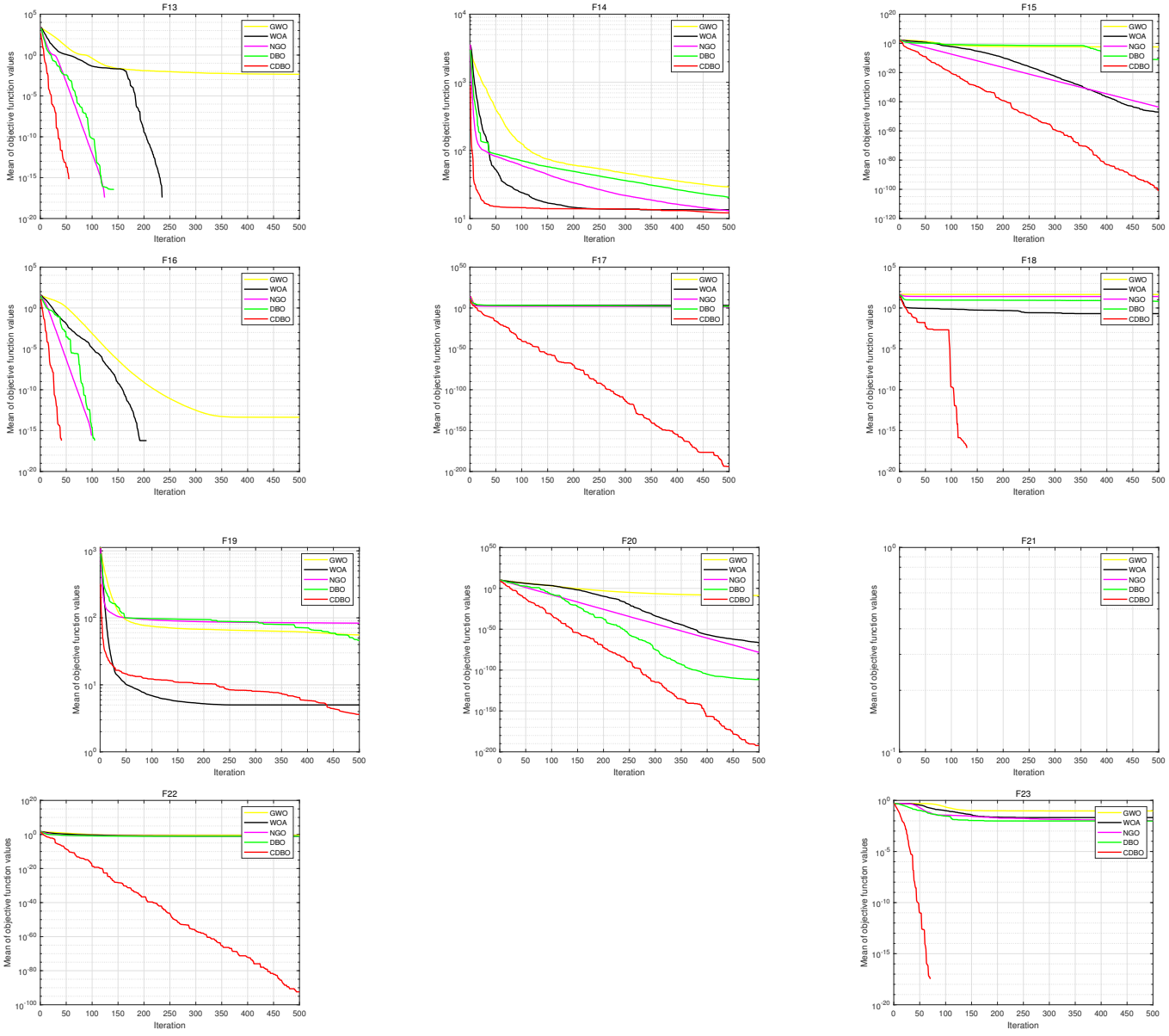

Figure 3: Comparison of convergence curves of different algorithms (100-dimension, part 2).

does not achieve optimal results. For the multi-peak function F14, the CDBO algorithm has an average global search ability when the dimension is 30, but when the dimension rises to 100 dimensions, the CDBO algorithm achieves the optimal results compared to the other algorithms, and only slightly lags behind the other intelligent optimization algorithms in terms of std. For the remaining multi-peak functions, the CDBO algorithm achieves the best ranking for all of them.

Overall, In comparison to other intelligent optimization algorithms, the CDBO algorithm's optimization ability not only does not drop when the dimension raised to 100, but it also greatly improves, showing great efficiency and efficiency capabilities. It indicates how well this algorithm's growth succeeded, which is still practical for high-dimensional functions.

4.2.2. Convergence

As shown in Figures 2 and 3, we show the convergence graphs of the five algorithms in 30 and 100 dimensions respectively to evaluate the correctness and convergence speed of the algorithms.

From Figure 2, we can get this information. When the dimension is 30, for the Multi-peaked function F6, the convergence results of all the algorithms in the last iteration are not much different, but The figure shows that the CDBO achieves rapid convergence, surpassing the other algorithms, the CDBO algorithm converges extremely fast and far exceeds the other algorithms. And it converges to the best accuracy due to the continuous exploration in the later stage. Satisfactory results have been achieved on single-peak functions.

For the multi-peak functions F7 and F12, the convergence

Table 6: Comparison of optimization performance of different improved algorithms on test functions(30-dimension, part 1).

Function	Algorithm	Mean Value	Standard Deviation	Best Value	Worst Value
F1	DBO	2.57e-101	1.41e-100	1.15e-161	7.73e-100
	SC-DBO	7.61e-116	4.15e-115	7.24e-161	2.27e-114
	MS-DBO	4.27e-100	2.34e-99	1.00e-162	1.28e-98
	CDBO	6.63e-197	0	0	1.98e-195
F2	DBO	2.64e-108	1.44e-107	2.83e-162	7.92e-107
	SC-DBO	2.68e-107	1.46e-106	3.73e-165	8.02e-106
	MS-DBO	1.04e-109	5.48e-109	4.36e-157	3.00e-108
	CDBO	6.66e-206	0	0	1.02e-204
F3	DBO	2.18e-51	1.19e-50	1.06e-84	6.54e-50
	SC-DBO	2.45e-49	1.33e-48	2.81e-83	7.30e-48
	MS-DBO	5.20e-63	2.65e-62	1.48e-86	1.45e-61
	CDBO	3.83e-99	1.50e-98	0	7.08e-98
F4	DBO	1.14e-47	6.27e-47	1.06e-147	3.43e-46
	SC-DBO	1.18e-57	6.50e-57	9.35e-175	3.56e-56
	MS-DBO	6.78e-52	3.71e-51	9.95e-152	2.03e-50
	CDBO	3.05e-192	0	0	9.16e-191
F5	DBO	3.93e-54	2.07e-53	2.61e-77	1.13e-52
	SC-DBO	2.17e-54	1.16e-53	5.04e-77	6.41e-53
	MS-DBO	1.54e-53	8.43e-53	2.57e-84	4.61e-52
	CDBO	1.28e-100	5.78e-100	0	3.15e-99
F6	DBO	2.57e+01	3.06e-01	2.53e+01	2.70e+01
	SC-DBO	2.58e+01	2.68e-01	2.53e+01	2.67e+01
	MS-DBO	2.57e+01	2.36e-01	2.52e+01	2.62e+01
	CDBO	2.80e+01	1.84e-01	2.78e+01	2.86e+01
F7	DBO	4.71e-01	3.94e-01	1.95e-04	1.59e+00
	SC-DBO	7.45e-01	4.75e-01	2.19e-03	1.76e+00
	MS-DBO	3.48e-01	4.00e-01	2.68e-04	1.59e+00
	CDBO	3.48e-01	1.86e-01	1.27e-02	7.39e-01
F8	DBO	3.96e-217	0	0	1.19e-215
	SC-DBO	2.71e-223	0	5.84e-308	8.15e-222
	MS-DBO	1.92e-232	0	0	5.45e-231
	CDBO	0	0	0	0
F9	DBO	1.13e-03	8.84e-04	7.85e-05	3.44e-03
	SC-DBO	1.15e-03	9.32e-04	1.88e-04	4.20e-03
	MS-DBO	1.17e-03	8.72e-04	1.78e-04	3.99e-03
	CDBO	1.98e-04	1.89e-04	1.52e-05	7.35e-04
F10	DBO	2.16e-113	1.18e-112	6.28e-187	6.48e-112
	SC-DBO	2.11e-109	1.15e-108	7.65e-219	6.32e-108
	MS-DBO	5.16e-126	2.08e-125	8.29e-191	1.09e-124
	CDBO	0	0	0	0
F11	DBO	3.75e+00	1.01e+01	0	3.58e+01
	SC-DBO	9.33e-01	2.94e+00	0	1.39e+01
	MS-DBO	8.62e-01	4.72e+00	0	2.58e+01
	CDBO	0	0	0	0

accuracy of CDBO algorithm is comparable with other algorithms, but In contrast to other algorithms, the CDBO method converges more quickly; for the multi-peak function F14, the CDBO algorithm converges fast in the early stage, and the algorithm exists a local excavation potential in the late stage, and the result of the search for the optimal result is in the second place; for the multi-peak function F19, the CDBO algorithm exists a high number of local minima . As can be seen from the figure, when iterating to four-fifths, the local mining ability of the CDBO algorithm gradually becomes stronger, further exerting the effect of algorithmic improvement, and not falling into the local optimum while guaranteeing higher accuracy, indicating that The pre-global search and the later growth of the optimality seeking ability are well-balanced in the CDBO algorithm; for the multi-peak function F21, there is no function image on the convergence curve graph, as the individual algorithms attained

Table 6: Comparison of optimization performance of different improved algorithms on test functions(30-dimension, part 2).

Function	Algorithm	Mean Value	Standard Deviation	Best Value	Worst Value
F12	DBO	4.44e-16	0	4.44e-16	4.44e-16
	SC-DBO	4.44e-16	0	4.44e-16	4.44e-16
	MS-DBO	4.44e-16	0	4.44e-16	4.44e-16
	CDBO	4.44e-16	0	4.44e-16	4.44e-16
F13	DBO	9.85e-04	5.39e-03	0	2.95e-02
	SC-DBO	0	0	0	0
	MS-DBO	3.42e-03	1.87e-02	0	1.02e-01
	CDBO	0	0	0	0
F14	DBO	1.72e+00	3.11e+00	1.24e-02	1.68e+01
	SC-DBO	1.16e+00	1.71e+00	1.64e-02	9.03e+00
	MS-DBO	2.54e+00	4.57e+00	1.21e-02	2.32e+01
	CDBO	1.34e+00	1.97e+00	1.01e-02	6.10e+00
F15	DBO	1.76e-04	4.47e-04	7.68e-82	1.90e-03
	SC-DBO	1.99e-03	9.69e-03	6.36e-79	5.32e-02
	MS-DBO	9.73e-05	3.60e-04	3.69e-81	1.89e-03
	CDBO	1.93e-102	9.31e-102	0	5.10e-101
F16	DBO	0	0	0	0
	SC-DBO	0	0	0	0
	MS-DBO	0	0	0	0
	CDBO	0	0	0	0
F17	DBO	1.61e-32	8.82e-32	4.50e-132	4.83e-31
	SC-DBO	7.65e-33	4.16e-32	1.15e-102	2.28e-31
	MS-DBO	2.75e-29	1.50e-28	3.36e-97	8.25e-28
	CDBO	2.99e-196	0	0	8.97e-195
F18	DBO	1.28e+00	1.11e+00	5.63e-04	2.90e+00
	SC-DBO	1.67e+00	8.74e-01	2.80e-02	2.90e+00
	MS-DBO	5.47e+00	1.47e+00	1.55e+00	7.93e+00
	CDBO	0	0	0	0
F19	DBO	3.70e+00	5.99e+00	7.29e-04	2.64e+01
	SC-DBO	3.50e+00	3.58e+00	2.05e-04	1.19e+01
	MS-DBO	2.47e+00	2.84e+00	1.46e-04	1.00e+01
	CDBO	6.43e-01	8.61e-01	4.63e-03	2.61e+01
F20	DBO	1.41e-101	7.77e-101	5.55e-180	4.25e-100
	SC-DBO	4.49e-109	2.46e-108	5.97e-158	1.34e-107
	MS-DBO	3.16e-109	1.73e-108	5.68e-157	9.48e-108
	CDBO	5.59e-196	0	0	1.67e-194
F21	DBO	0	0	0	0
	SC-DBO	0	0	0	0
	MS-DBO	0	0	0	0
	CDBO	0	0	0	0
F22	DBO	9.32e-02	2.53e-02	2.50e-74	9.98e-02
	SC-DBO	9.32e-02	2.53e-02	5.62e-79	9.98e-02
	MS-DBO	9.15e-02	2.56e-02	4.17e-65	9.98e-02
	CDBO	7.38e-99	3.97e-98	0	2.17e-97
F23	DBO	9.71e-03	1.18e-07	9.71e-03	9.71e-03
	SC-DBO	9.71e-03	3.68e-07	9.71e-03	9.71e-03
	MS-DBO	9.39e-03	1.77e-03	5.55e-17	9.71e-03
	CDBO	0	0	0	0

the function's optimal value of 0 at the start of the iteration; for other multi-peak functions, The CDBO algorithm strikes an acceptable balance between the first pre-global search and the subsequent growth of the optimal outcomes seeking abilities.

When the dimension is raised from 30 to 100 dimensions, the convergence curves for each algorithm are displayed in Figure 3. this illustrates the CDBO algorithm's strong performance across all 23 test functions and further supports its

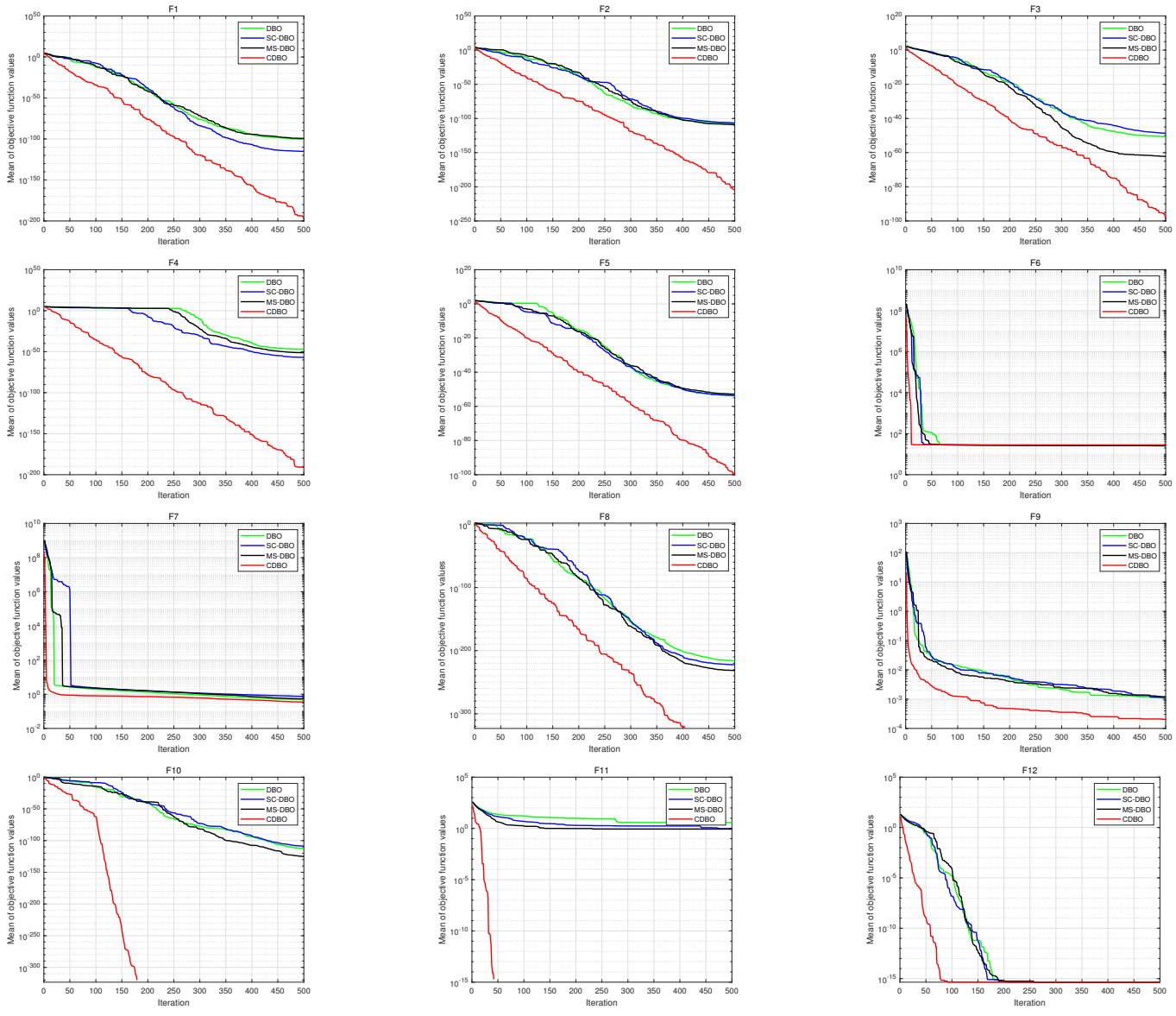

Figure 4: Comparison of convergence curves of different improved algorithms(30-dimension, part 1).

ability to conduct global searches and refine solutions in high-dimensional space.

For the Multi-peaked function F7, the CDBO algorithm quickly finds the function near the optimal value with no more than 50 iterations, and in the late stage of the function, it also finds better results compared to other algorithms. For other single-peaked functions, the CDBO algorithm finds the function's ideal value and produces very good search results, verifying the enhanced algorithm's better search performance.

On the multi-peak function F19, In the last iteration, the CDBO algorithm greatly outperforms the WOA algorithm by capitalizing on its success, and in the late iteration, it keeps searching for exploitation to further search for the optimal value of the function and find the optimal value of the function, showcasing the enhanced algorithm's exploration capability in the final iteration. For the multi-peak function F21, the image does not have a convergence curve as the algorithms locate the function's optimal value at the start of the iteration. For the multi-peak functions F9 and F12, although the final results of the individual algorithms' search for the optimum are not very different, it is clear from the graph that Compared to the other approaches, the CDBO algorithm can find the function's ideal value with the shortest iterations and converge more rapidly.

4.3. Performance comparison of CDBO with SC-DBO and MS-DBO

In this section, we compare the performance of three optimization algorithms—CDBO, SC-DBO, and MS-DBO—across several critical factors that influence their effectiveness in high-dimensional optimization tasks. The comparison is presented in terms of the dynamic phases, complexity, local and global search abilities, as well as their

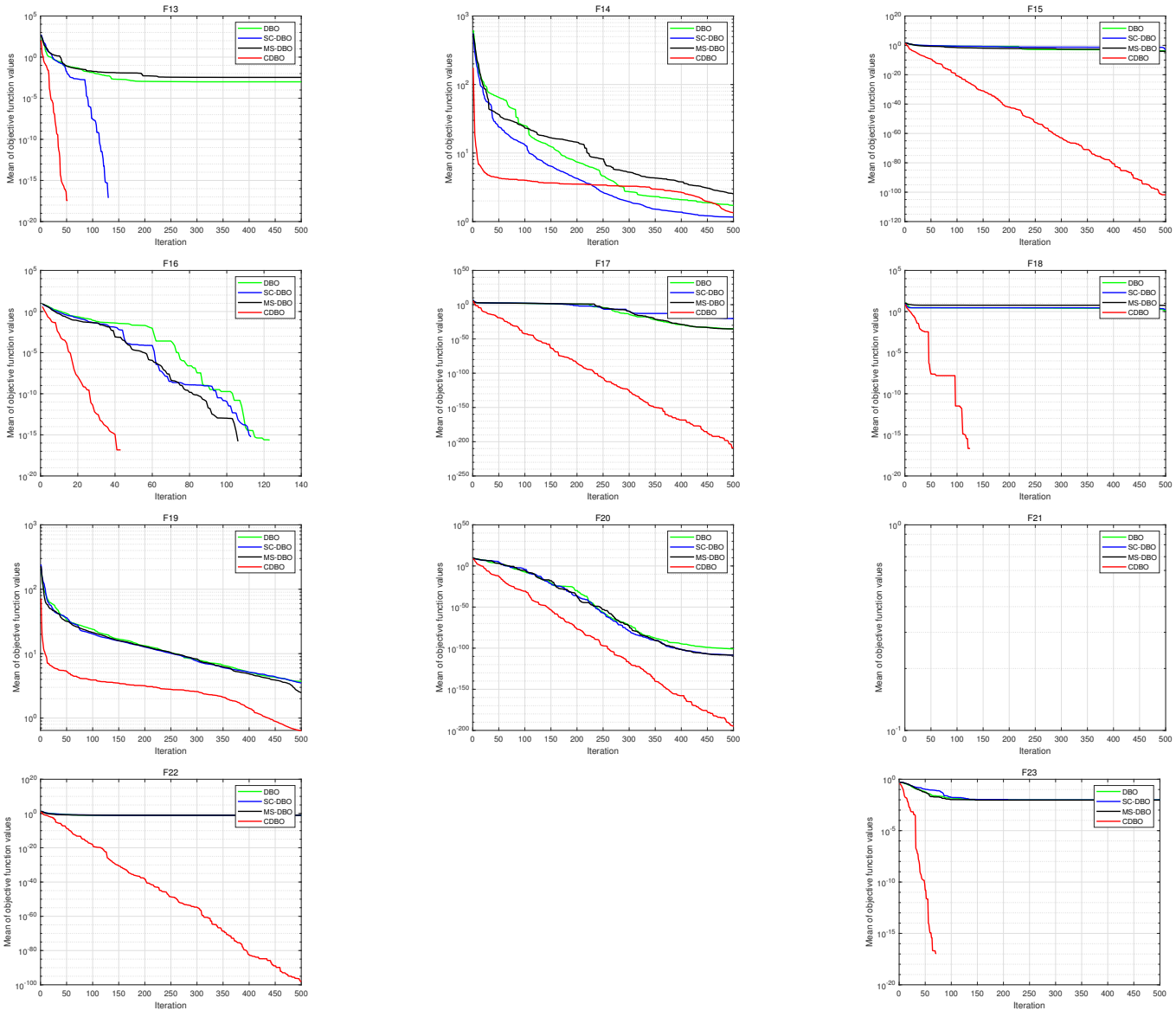

Figure 4: Comparison of convergence curves of different improved algorithms (30-dimension, part 2).

performance in high-dimensional stability. Table 5 summarizes the key characteristics and performance metrics for each method.

As shown in the Tables 6 and 7, SC-DBO is characterized by its standard complexity and search abilities, while MS-DBO exhibits high global search ability, especially in situations requiring inverse learning. On the other hand, CDBO, which employs levy flights in its dynamic phase, offers superior stability and search precision, making it particularly suitable for high-dimensional problems.

1. Performance Table Analysis(Table 6 and 7): The tables list various metrics for each algorithm, such as mean value, standard deviation, best value, and worst value across different test functions. Here are some key trends:
    - Mean Values:The CDBO algorithm generally shows the lowest mean values across the functions in both 30D and 100D, indicating better overall performance compared to SC-DBO and MS-DBO. In some cases, such as functions F8, F12, and F15 (both 30D and 100D), CDBO demonstrates exceptionally low mean values, suggesting that it consistently finds better solutions compared to the other methods.
    - Best Values:CDBO frequently achieves the best values in many test functions, particularly in high-dimensional settings (100D), suggesting superior convergence to the optimal solution compared to SC-DBO and MS-DBO.MS-DBO and SC-DBO perform competitively in certain test cases, but they do not consistently match CDBO in finding the best values.
    - Worst Values:The worst values across all methods

Table 7: Comparison of optimization performance of different improved algorithms on test functions(100-dimension, part 1).

Function	Algorithm	Mean Value	Standard Deviation	Best Value	Worst Value
F1	DBO	3.17e-112	1.53e-111	1.74e-171	8.38e-111
	SC-DBO	6.69e-111	2.72e-110	4.98e-166	1.37e-109
	MS-DBO	1.89e-111	1.03e-110	7.70e-169	5.67e-110
	CDBO	8.02e-196	0	0	2.40e-196
F2	DBO	6.42e-110	3.51e-109	1.52e-163	1.92e-108
	SC-DBO	2.64e-113	1.45e-112	1.93e-167	7.94e-112
	MS-DBO	2.58e-115	1.10e-114	3.30e-155	5.84e-114
	CDBO	1.53e-194	0	0	4.60e-193
F3	DBO	4.20e-57	1.72e-56	1.48e-84	8.76e56
	SC-DBO	2.76e-59	1.36e-58	1.14e-78	7.48e-58
	MS-DBO	1.88e-57	1.01e-56	6.96e-83	5.58e-56
	CDBO	8.91e-98	3.40e-97	0	1.48e-96
F4	DBO	1.06e-24	5.84e-24	3.17e-124	3.20e-23
	SC-DBO	4.54e-41	2.49e-40	1.17e-135	1.36e-39
	MS-DBO	3.39e-76	1.85e-75	8.51e-140	1.01e-74
	CDBO	4.11e-201	0	0	1.23e-199
F5	DBO	7.54e-45	4.12e-44	1.77e-72	2.26e-43
	SC-DBO	1.94e-50	1.06e-49	8.66e-74	5.83e-49
	MS-DBO	4.73e-51	2.31e-50	1.92e-78	1.26e-49
	CDBO	1.58e-101	7.79e-101	0	4.27e-100
F6	DBO	9.71e+01	6.47e-01	9.60e+01	9.83e+01
	SC-DBO	9.70e+01	6.30e-01	9.60e+01	9.82+e+01
	MS-DBO	9.72e+01	7.69e-01	9.58e+01	9.82+e+01
	CDBO	9.76e+01	2.65e-01	9.72e+01	9.81e+01
F7	DBO	8.58e+00	7.32e-01	5.10e+00	9.23e+00
	SC-DBO	8.89e+00	3.81e-01	7.99e+00	9.62e+00
	MS-DBO	8.75e+00	4.33e-01	7.93e+00	9.52e+00
	CDBO	1.73e+00	7.58e-01	1.84e-01	3.73e+00
F8	DBO	1.67e-215	0	0	5.01e-4
	SC-DBO	3.84e-226	0	0	1.15e-224
	MS-DBO	2.34e-225	0	0	7.04e-224
	CDBO	0	0	0	0
F9	DBO	1.37e-03	1.52e-03	3.38e-05	5.79e-03
	SC-DBO	1.20e-03	9.82e-04	3.49e-05	3.81e-03
	MS-DBO	1.47e-03	1.34e-03	1.54e-04	5.25e-03
	CDBO	2.88e-04	2.80e-03	1.24e-05	1.18e-03
F10	DBO	9.41e-119	4.53e-118	6.93e-211	2.47e-117
	SC-DBO	2.54e-126	1.39e-125	2.04e-204	7.63e-125
	MS-DBO	3.55e-123	1.92e-122	3.86e-194	1.05e-121
	CDBO	0	0	0	0
F11	DBO	9.40e+00	3.77e+01	0	1.86e+02
	SC-DBO	0	0	0	0
	MS-DBO	0	0	0	0
	CDBO	0	0	0	0

generally indicate that MS-DBO and SC-DBO tend to perform worse in certain test cases, particularly in functions with higher complexity (e.g., F5, F16).

- Standard Deviation:In both 30D and 100D, CDBO tends to show smaller standard deviations in several functions, indicating that it is more stable in terms of convergence.

2. Convergence Curve Analysis(Figures 4 and 5): The convergence curves further reinforce the findings from the tables:

- CDBO:In both the 30D and 100D cases, CDBO shows fast convergence, often reaching the optimal or near-optimal solution in fewer iterations compared to SC-DBO and MS-DBO. The curves for CDBO tend to drop sharply, indicating that it reaches low error levels quickly and remains stable

Table 7: Comparison of optimization performance of different improved algorithms on test functions(100-dimension, part 2).

Function	Algorithm	Mean Value	Standard Deviation	Best Value	Worst Value
F12	DBO	4.44e-16	0	4.44e-16	4.44e-16
	SC-DBO	4.44e-16	0	4.44e-16	4.44e-16
	MS-DBO	4.44e-16	0	4.44e-16	4.44e-16
	CDBO	4.44e-16	0	4.44e-16	4.44e-16
F13	DBO	0	0	0	0
	SC-DBO	0	0	0	0
	MS-DBO	0	0	0	0
	CDBO	0	0	0	0
F14	DBO	2.09e+01	5.60e+00	1.02e+01	3.31e+01
	SC-DBO	2.13e+01	6.06e+00	1.20e+01	3.37e+01
	MS-DBO	2.41e+01	7.34e+00	1.28e+01	4.47e+01
	CDBO	1.30e+01	8.21e+00	1.28e-01	2.66e+01
F15	DBO	3.24e-04	1.77e-03	4.28e-80	9.74e-03
	SC-DBO	2.42e-04	1.32e-03	7.54e-80	7.27e-03
	MS-DBO	1.28e-03	4.95e-03	2.03e-81	2.23e-02
	CDBO	1.00e-103	3.66e-103	0	1.53e-102
F16	DBO	0	0	0	0
	SC-DBO	0	0	0	0
	MS-DBO	0	0	0	0
	CDBO	0	0	0	0
F17	DBO	4.84e+01	1.29e+02	1.44e-27	4.93ee+02
	SC-DBO	5.94e+00	2.29e+01	1.44e-27	1.18e+02
	MS-DBO	1.02e+01	2.93e+01	4.61e-53	1.33e+02
	CDBO	2.35e-199	0	0	6.97e-198
F18	DBO	5.28e+00	3.43e+00	4.65e-01	9.92e+00
	SC-DBO	8.02e+00	3.03e+00	4.12e-01	9.92e+00
	MS-DBO	2.36e+01	3.22e+00	1.10e+01	2.73e+01
	CDBO	0	0	0	0
F19	DBO	4.64e+01	3.90e+01	1.00e+00	9.34e+01
	SC-DBO	8.78e+01	4.11e+00	7.83e+01	9.81e+01
	MS-DBO	8.64e+01	3.87e+00	7.67e+01	9.46e+01
	CDBO	4.23e+00	3.87e+00	2.18e-01	1.32e+01
F20	DBO	7.34e-110	3.84e-109	1.18e-161	2.10e-108
	SC-DBO	2.64e-102	9.55e-102	3.81e-160	4.75e-101
	MS-DBO	3.25e-105	1.78e-104	1.68e-162	9.76e-104
	CDBO	5.44e-191	0	0	1.59e-189
F21	DBO	0	0	0	0
	SC-DBO	0	0	0	0
	MS-DBO	0	0	0	0
	CDBO	0	0	0	0
F22	DBO	9.32e-02	2.53e-02	1.15e-63	9.98e-02
	SC-DBO	7.99e-02	4.06e-02	1.57e-49	9.98e-02
	MS-DBO	9.65e-02	1.82e-02	2.17e-21	9.98e-02
	CDBO	4.23e-101	1.80e-100	0	9.44e-100
F23	DBO	9.71e-03	2.52e-07	9.71e-03	9.71e-03
	SC-DBO	9.71e-03	1.36e-07	9.71e-03	9.71e-03
	MS-DBO	9.71e-03	4.73e-07	9.71e-03	9.71e-03
	CDBO	0	0	0	0

after that.

- SC-DBO and MS-DBO:In both dimensional settings, SC-DBO and MS-DBO show more gradual convergence compared to CDBO. For some functions, the convergence rate is slower, and the curves are flatter, indicating that they require more iterations to approach optimal solutions. Particularly in high-dimensional problems (100D), MS-DBO and

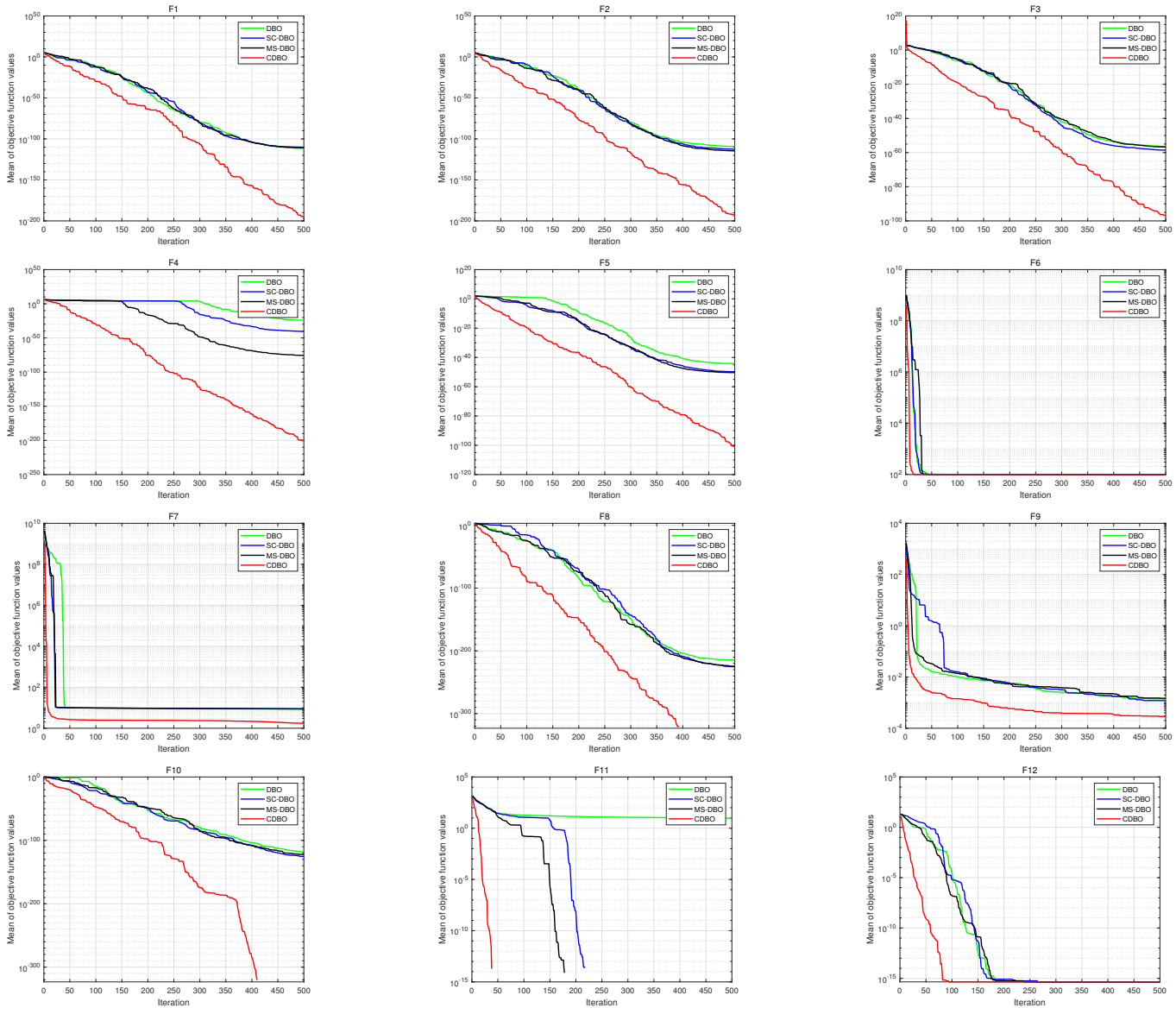

Figure 5: Comparison of convergence curves of different improvement methods (100-dimension, part 1).

Table 8: Optimization results for compression spring design.

Algorithm	d	D	N	Optimal value
SSA[17]	0	1	4	0.0141
MSA[17]	0	1	5	0.0150
BOA[17]	0	0	12	0.0142
WFO	0	1	14	0.0499
DBO	0	1	4	0.0144
CDBO	0	0	15	0.0134

SC-DBO struggle with achieving fast convergence in complex functions, while CDBO outperforms them in both speed and stability.

- High-Dimensional Performance: As the dimensionality increases from 30D to 100D, all algorithms exhibit slower convergence. However, CDBO stands out by maintaining better performance across a wide range of functions. This suggests that CDBO may be better equipped to handle the curse of dimensionality due to its stable search dynamics and effective optimization strategies.
- CDBO's Strengths: Based on both the tables and convergence curves, CDBO proves to be the most robust algorithm in both 30D and 100D spaces. Its ability to find optimal solutions more consistently and its stability across multiple test functions suggest that it may be more suitable for complex, high-dimensional optimization problems. The use of Levy flights likely contributes to its ability to explore the solution space more efficiently.
- SC-DBO and MS-DBO: While these algorithms

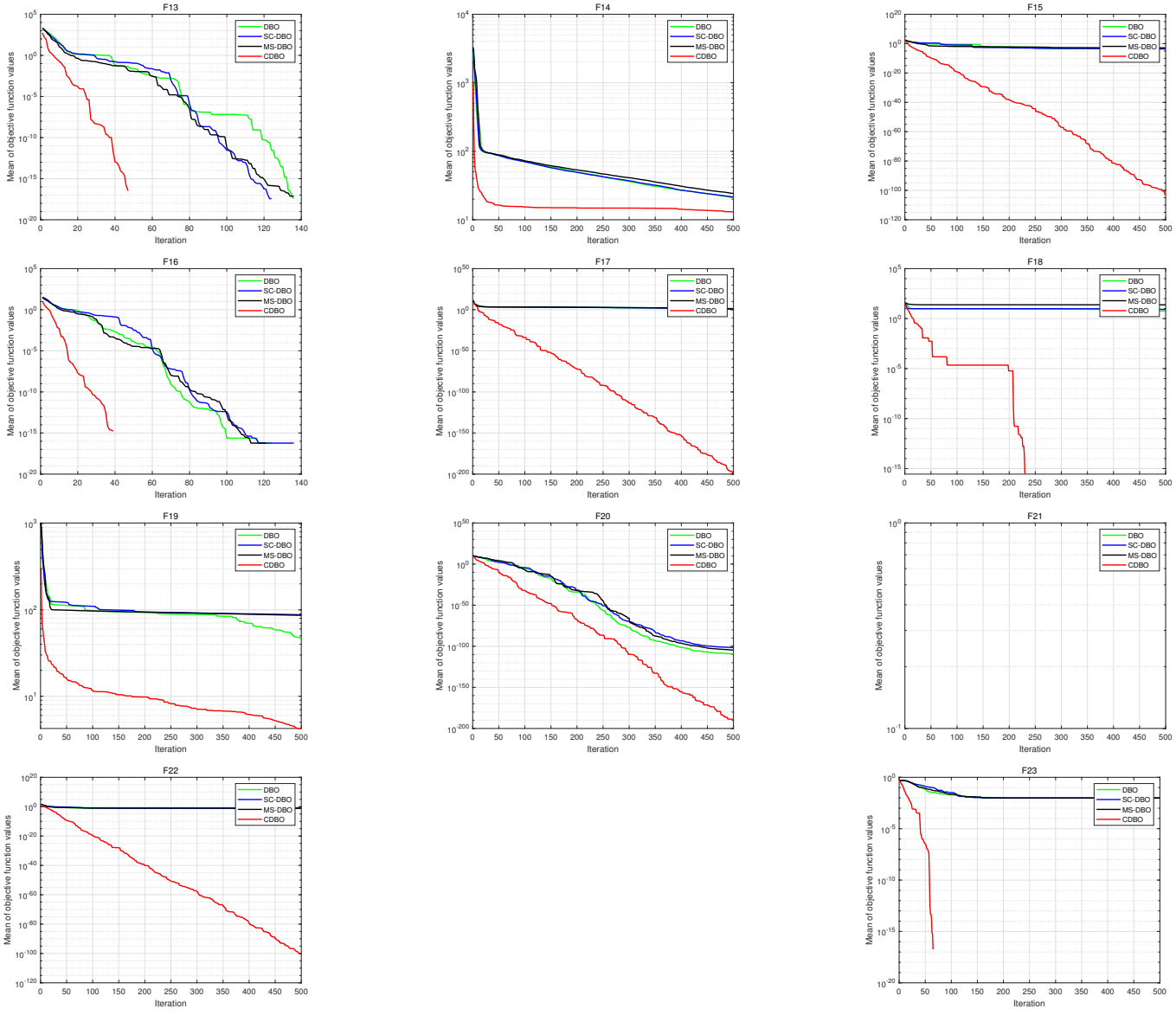

Figure 5: Comparison of convergence curves of different improved algorithms(100-dimension, part 2).

Table 9: Experimental table of data sets.

Data set	Data type	Sample size	Sample dimension	Number of classes
Iris	UCI	150	4	3
glass	UCI	214	9	6
ionosphere	UCI	351	34	2
sonar	UCI	208	60	2
wpbc	UCI	198	34	2
vote	UCI	435	16	2
heart	UCI	303	14	2

also show competitive performance, particularly in lower-dimensional cases, they struggle more with the high-dimensional tasks. SC-DBO performs adequately in terms of convergence but lacks the consistency and stability of CDBO. MS-DBO shows potential but doesn't match CDBO's rapid convergence and stable performance, especially in 100D.

Table 10: Experimental table of data sets.

Data set	index	GA	PSO	WOA	DBO	CDBO
Iris	avg	0.811	0.7766	0.9232	0.803	0.741
	std	8.93e-04	1.25e-02	6.34e-02	1.17e-16	1.17e-03
glass	avg	1.069e+01	1.274e+01	1.021e+01	1.008e+01	6.543
	std	5.61e-02	2.39e-02	2.91e-01	0	0
ionosphere	avg	1.176	1.099	1.113	1.022	0
	std	1.83e-02	6.70e-03	2.15e-02	0	0
sonar	avg	1.408	1.336	1.462	1.352	1.305
	std	7.5e-03	1.16e-02	1.08e-02	2.34e-16	0
wpbc	avg	1.027	1.109	1.064	9.022e-01	0
	std	2.41e-02	2.02e-02	1.90e-04	0	0
vote	avg	1.093	1.099	1.147	1.114	0
	std	3.33e-02	2.53e-02	1.00e-02	2.34e-16	0
heart	avg	1.105	1.129	1.054	1.108	1.054
	std	2.57e-04	8.40e-03	2.96e-02	2.34e-16	0

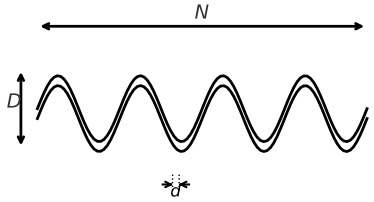

Figure 6: Design problem in compression spring.

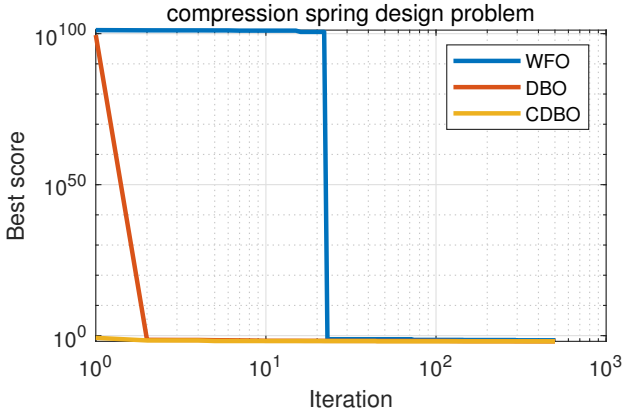

Figure 7: Convergence curve of the algorithm in compression spring design.

5. Engineering applications

Figure 6 illustrates the compression spring design problem[26], which includes independent variables: wire diameter (d), average coil diameter (D), and the effective number of coils (N). The goal is to minimize the weight of the spring. This problem involves optimizing the objective function and constraints to find the best design solution through the combination of these variables.

Here is an illustration of the mathematical model used for developing a compression spring:

$$x = [x_1, x_2, x_3] = [D\ d\ N]. \tag{14}$$

purposeful function:

$$f(x) = (x_3 + 2)x_2x_1^2. \tag{15}$$

restrictive condition:

$$\begin{cases} g_1(x) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \le 0, \\ g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0, \\ g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0, \\ g_4(x) = \frac{x_1+x_2}{1.5} - 1 \le 0. \end{cases} \tag{16}$$

Range of values

$$0.05 \le x_1 \le 2,\ 0.25 \le x_2 \le 1.3,\ 2 \le x_3 \le 15. \tag{17}$$

Three intelligent optimization algorithms SSA, MSA, and BOA are applied to solve the aforementioned engineering problems, with results compared to those from reference [17]. Table 8 presents the detailed outcomes of these comparison experiments. The data reveal that CDBO consistently achieves the best results among the algorithms.

Figure 7 shows the convergence speed of different optimization algorithms (WFO, DBO, CDBO) in the compression spring design problem. By comparing the best scores of these algorithms as a function of iteration number, the performance of each algorithm can be clearly observed.

The CDBO algorithm performs the best in terms of convergence speed and stability. As the number of iterations increases, the CDBO algorithm quickly reaches a lower best score, demonstrating its advantage in efficient optimization. Although the DBO algorithm converges faster in the early stages, its convergence speed is slower compared to CDBO, and its final score is higher than that of CDBO. The performance of the WFO algorithm is relatively poor, with slower convergence,

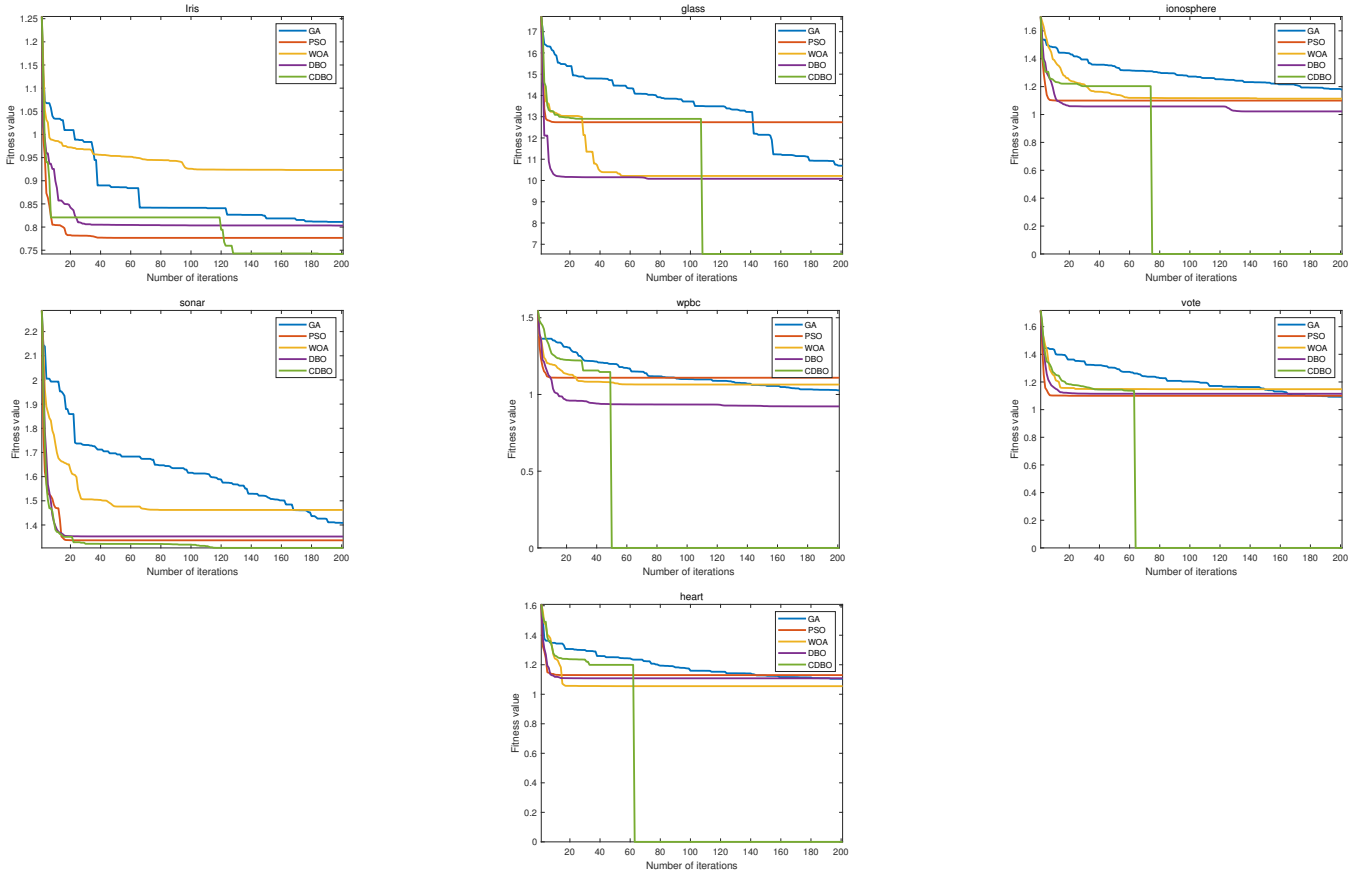

Figure 8: Convergence curves for each algorithm applied to K-Means clustering.

and its best score does not improve significantly over longer iterations.

These results indicate that CDBO not only converges quickly when solving such engineering optimization problems but also maintains a high solution quality, further validating its strong performance in complex engineering problems.

6. Apply CDBO to K-means

6.1. K-means

Davidson Boulding Index (DBI) is a commonly used internal evaluation index in clustering [27], the main idea is that the intra-class distance is minimized while the inter-class distance is maximized, calculated as follows:

$$DBI = \frac{1}{K}\sum_{i=1}^{k}\max_{i\neq j}\left(\frac{\bar{C}_i + \bar{C}_j}{\sqrt{(w_i - w_j)^2}}\right), \tag{18}$$

where C_i denotes the mean value in a class and w_i denotes the location of the centroid of the class, so the smaller the index of DBI, the better the clustering effect.

To validate the various clustering performance metrics of CDBO in this study, the classical datasets Iris, glass, ionosphere, sonar, wpbc, vote, and heart on the UCI dataset (http://archive.ics.uci.edu/ml/) are selected for the experiments, and the data details are shown in Table 9.

6.2. Analysis of clustering results

This simulation experiment uses a relatively objective internal evaluation index DBI, and introduces GA, PSO, WOA, DBO and CDBO for comparison, at the same time, For comparison testing, the number of clusters (k value) in the CDBO-K-means method is known, the experiment findings are displayed in Table 10 and the algorithm will be performed independently for 10 times and iterated 100 times, capturing the mean and standard deviation. In addition, in order to compare the convergence speed of each algorithm, the convergence curves of each dataset are plotted and the outcomes are displayed in Figure 8.

As shown in Table 10, for datasets with medium-to-low complexity and low dimensionality (Iris, glass, vote, heart), the clustering performance of CDBO-K-means is significantly better than that of other algorithms, with the exception of the glass dataset, where it did not surpass the others. This supports the effectiveness of the improved algorithm. For datasets with high complexity and high dimensionality (ionosphere, sonar, wpbc), CDBO achieves optimal convergence accuracy and stability, resulting in a superior clustering effect.

As shown in the convergence curve in Figure 8, it can be seen that better convergence accuracy along with faster conver-

gence are features of the CDBO-K-means method, better clustering effect, and better stability, which can effectively improve the clustering effect.

7. Conclusion

For the original DBO, this paper proposes to improve the DBO algorithm based on several strategies.

- The golden sine selection approach is given at the rolling step. Sync the global search phase of the algorithm during the pre-iteration phase with the local mining capability during the late-iteration phase.
- Utilize the self-helix structure found in the WOA during the foraging stage. Boost the algorithm's accuracy and rate of convergence.
- Introduce Levy flight for stealing behaviour. Broaden the search range and avoid premature maturity.
- The globally optimal solution is finally modified using a dynamic t-distribution to avoid the technique from reaching a local optimum.

Simulation experiments were conducted using 23 test functions to confirm the effectiveness of the CDBO. Based on the experimental results, the CDBO algorithm improves its global searching ability and broadens its search range during the pre-iteration stage. It also enhances its ability to survive local limits and expands its exploratory ability over the late-iteration stage. And when the dimension rises from 30 to 100, the CDBO algorithm still maintains the superior search performance, and results prove its effectiveness, and rationality of the improvement. Subsequently, CDBO was applied to engineering optimization and K-means clustering, and from the experimental results, CDBO can solve various optimization problems well to some extent. However, CDBO has shortcomings in high-dimensional computation and scalability issues, therefore, in the subsequent work, the focus will be on further enhancing the performance of the algorithm, improving its efficiency, and exploring the application of CDBO to more practical problems.

Data availability

The link to the dataset used in this study is provided here: http://archive.ics.uci.edu/ml/.

References

[1] R. Salgotra, P. Sharma, S. Raju & A. H. Gandomi, "A contemporary systematic review on meta-heuristic optimization algorithms with their MATLAB and Python code reference", Archives of Computational Methods in Engineering **31** (2024) 1749. https://doi.org/10.1007/s11831-023-10030-1.

[2] J. He & L. Fu, "Robot path planning based on improved dung beetle optimizer algorithm", J. Braz. Soc. Mech. Sci. Eng. **46** (2024) 235. https://doi.org/10.1007/s40430-024-04768-3.

[3] J. Liu, L. Cong, Y. Xia, G. Pan, H. Zhao & Z. Han, "Short-term power load prediction based on DBO-VMD and an IWOA-BILSTM neural network combination model", Dianli Xitong Baohu Yu Kongzhi Power Syst. Prot. Control **52** (2024) 123. https://doi.org/10.19783/j.cnki.pspc.231402.

[4] X. Yi & Y. Liu, *Multi-model ensemble air quality prediction with dung beetle optimization*, presented at the 2023 5th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), 2023, pp. 338–342. https://doi.org/10.1109/MLBDBI60823.2023.10482164.

[5] D. Xu, Z. Li & W. Wang, "An ensemble model for monthly runoff prediction using least squares support vector machine based on variational modal decomposition with dung beetle optimization algorithm and error correction strategy", J. Hydrol. **629** (2024) 130558. https://doi.org/10.1016/j.jhydrol.2023.130558.

[6] Q. Yuan, L. Wu, Y. Huang, Z. Guo & N. Li, "Water-body detection from spaceborne SAR images with DBO-CNN", IEEE Geosci. Remote Sens. Lett. **20** (2023) 4013405. https://doi.org/10.1109/LGRS.2023.3325939.

[7] S. Gao, Z. Li, Y. Zhang, S. Zhang & J. Zhou, "Remaining useful life prediction method of rolling bearings based on improved 3 and DBO-HKELM", Meas. Sci. Technol. **35** (2024) 106101. https://iopscience.iop.org/article/10.1088/1361-6501/ad52b5/meta.

[8] D. Zhang, C. Zhang, X. Han & C. Wang, "Improved DBO-VMD and optimized DBN-ELM based fault diagnosis for control valve", Meas. Sci. Technol. **35** (2024) 075103. https://doi.org/10.1088/1361-6501/ad3be0.

[9] W. Cao, Z. Liu, H. Song, G. Li & B. Quan, "Dung beetle optimized fuzzy PID algorithm applied in four-bar target temperature control system", Appl. Sci.-Basel **14** (2024) 4168. https://doi.org/10.3390/app14104168

[10] J. Sharma & R. S. Singhal, *Comparative research on genetic algorithm, particle swarm optimization and hybrid GA-PSO*, Proc. 2nd Int. Conf. Comput. Sustain. Global Dev. (INDIACOM), 2015, pp. 110–114. https://ieeexplore.ieee.org/document/7100231.

[11] W. Tong, "A hybrid algorithm framework with learning and complementary fusion features for whale optimization algorithm", Sci. Program. **2020** (2020) 5684939. https://doi.org/10.1155/2020/5684939.

[12] S. Gupta & K. Deep, "Enhanced leadership-inspired grey wolf optimizer for global optimization problems", Eng. Comput. **36** (2020) 1777. https://doi.org/10.1007/s00366-019-00795-0.

[13] J. Xue & B. Shen, "Dung beetle optimizer: a new meta-heuristic algorithm for global optimization", J. Supercomput. **79** (2023) 7305. https://doi.org/10.1007/s11227-022-04959-6.

[14] M. Dehghani, S. Hubalovsky & P. Trojovsky, "Northern goshawk optimization: a new swarm-based algorithm for solving optimization problems", IEEE Access **9** (2021) 162059. https://ieeexplore.ieee.org/document/9638618?denied=.

[15] X. Kuang, B. Yang, H. Ma, W. Tang, H. Xiao & L. Chen, "Multi-strategy improved dung beetle optimization algorithm", Comput. Eng., Mar. 2024, Advance online publication, Accessed: Jun. 12, 2024. [Online]. https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CAPJ&dbname=CAPJLAST&filename=JSJC2024022900B.

[16] L. Wang & L. Gu, "Improved dung beetle optimization algorithm with multi-strategy", Comput. Syst. Appl. **33** (2024) 224. https://doi.org/10.15888/j.cnki.csa.009397.

[17] J. Pan, S. Li, P. Zhou, G. Yang & D. Lv, "Dung beetle optimization algorithm guided by improved sine algorithm", Comput. Eng. Appl. **59** (2023) 92.

[18] R. Yu, Z. Xu, X. Chen & L. Jiang, "Research on multi-strategy improvement of dung beetle optimization algorithm", Journal of Nanning Normal University. https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CAPJ&dbname=CAPJLAST&filename=GXSZ20240416004.

[19] F. Zhu, G. Li, H. Tang, Y. Li, X. Lv & X. Wang, "Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems", Expert Syst. Appl. **236** (2024) 121219. https://doi.org/10.1016/j.eswa.2023.121219.

[20] X. Wang, Y. Wei, Z. Guo, J. Wang, H. Yu & B. Hu, "A Sinh-Cosh-enhanced DBO algorithm applied to global optimization problems", Biomimetics **9** (2024) 271. https://doi.org/10.3390/biomimetics9050271.

[21] M. Ye, H. Zhou, H. Yang, B. Hu & X. Wang, "Multi-strategy improved dung beetle optimization algorithm and its applications", Biomimetics **9** (2024) 291. https://doi.org/10.3390/biomimetics9050291.

[22] X. Wang, H. Kang, Y. Shen, X. Sun & Q. Chen, "An improved dung beetle optimization algorithm for high-dimension optimization and its engineering applications", Symmetry-Basel **16** (2024) 586. https://doi.org/10.

3390/sym16050586.
[23] J. Zhang & J. S. Wang, "Improved whale optimization algorithm based on nonlinear adaptive weight and golden sine operator", IEEE Access **8** (2020) 77013. https://ieeexplore.ieee.org/document/9076166.
[24] R. Rajabioun, "Cuckoo optimization algorithm", Appl. Soft Comput. **11** (2011) 5508. https://doi.org/10.1016/j.asoc.2011.05.008.
[25] D. T. Cassidy, "Describing n-day returns with Student's t-distributions", Phys. Stat. Mech. Appl. **390** (2011) 2794. https://doi.org/10.1016/j.physa.2011.03.019.
[26] Q. Li, H. Shi, W. Zhao & C. Ma, "Enhanced dung beetle optimization algorithm for practical engineering optimization", Mathematics **12** (2024) 7. https://doi.org/10.3390/math12071084.
[27] H. Ismkhan, "I-k-means-plus: An iterative clustering algorithm based on an enhanced version of the k-means", Pattern Recognit. **79** (2018) 402. https://doi.org/10.1016/j.patcog.2018.02.015.