Journal of Computers Vol. 37 No. 5, October 2026, pp. 107-129
https://doi.org/10.63367/19911599202610370500 8

A Multi-Strategy Optimized Dung Beetle Algorithm

Jiaoli Xu and Yang Liu*

School of Electronic Information Engineering, Jiangxi Polytechnic Institute,
Xinyu city, Jiangxi Province, China
xujiaoli@jxue.edu.cn, liuyang@jxue.edu.cn

Received 24 July 2025; Revised 4 February 2026; Accepted 13 February 2026

Abstract. To address the shortcomings of the Dung Beetle Optimizer (DBO) in solving complex optimization problems—namely insufficient global exploration capability, relatively slow convergence, and a tendency to become trapped in local optima—this paper proposes a multi-strategy improved DBO, termed CWSADBO, and applies it to the economic dispatch problem of microgrids. The proposed algorithm integrates Cubic chaotic initialization, a black widow behavior mechanism, a sparrow-inspired perturbation strategy, and a Cauchy–Gaussian hybrid mutation mechanism to enhance both search efficiency and solution accuracy. Specifically, the Cubic chaotic map is employed to increase the diversity of the initial population; the black widow behavior mechanism strengthens global exploration during the search process; the sparrow-inspired perturbation strategy helps the algorithm escape local optima; and the Cauchy–Gaussian hybrid mutation mechanism achieves a dynamic balance between global exploratory jumps and local fine-grained exploitation. To evaluate its performance, 16 standard benchmark functions are tested, and comparative experiments are conducted against DBO, IDBO, SSA, and SABO. The results demonstrate that CWSADBO achieves faster convergence and higher optimization accuracy on most benchmark functions, with particularly notable advantages in unimodal, multimodal, and complex multimodal cases. Moreover, its application to microgrid economic dispatch case studies confirms its ability to effectively reduce operating costs, exhibiting strong stability and engineering applicability.

Keywords: Dung Beetle Optimizer, multi-strategy optimization, chaotic initialization, black widow behavior, mutation mechanism

1 Introduction

With the high penetration of distributed renewable energy sources in microgrids, the inherent randomness and intermittency of their power output impose increasingly complex scheduling challenges on system operation [1-2]. To achieve economic and environmental objectives while ensuring power supply reliability, the microgrid energy management problem is typically formulated as a highly nonlinear optimization problem with multiple constraints and decision variables. It involves various aspects such as power output allocation among generation units, charging/discharging strategies of energy storage systems, and coordinated demand response of loads [3-4]. Therefore, developing efficient and robust optimization algorithms with strong global search capability has become crucial for solving the microgrid economic dispatch problem.

In recent years, swarm intelligence optimization algorithms have been widely applied in the field of microgrid energy management due to their advantages such as being independent of gradient information and well suited for complex nonconvex problems. These algorithms are inspired by cooperative behaviors and collective intelligence phenomena observed in biological groups such as flocks of birds, colonies of ants, and schools of fish. By simulating interactions among individuals, information sharing, and collective learning processes within a population, they enable effective search and optimization. Common bio-inspired heuristic algorithms include Particle Swarm Optimization [5], Ant Colony Optimization [6], Artificial Fish Swarm Algorithm [7], and the Bee Algorithm [8]. The Dung Beetle Optimizer (DBO) is a novel swarm intelligence optimization algorithm that searches for the optimal solution of an objective function by mimicking the process through which dung beetles locate high-quality resources in their environment. To address the drawbacks of DBO, such as slow convergence, susceptibility to local optima, and insufficient population diversity, extensive improvement studies have been conducted [9]. Reference [10] proposed the EPDBO algorithm, which enhances individual learning

* Corresponding Author

capability by introducing a replay mechanism, thereby improving solution performance in cooperative task allocation problems. However, this improvement mainly relies on reinforcement through historical information, making the search trajectory sensitive to the quality of early individuals; as a result, stability in complex multimodal problems still needs further improvement. Reference [11] applied a multi-strategy improved IDBO algorithm to parameter optimization of time convolutional networks, enhancing model prediction performance. Nevertheless, this approach focuses primarily on model parameter optimization, and its generalization capability for high-dimensional continuous constrained optimization problems remains limited. To further enhance global search capability, Reference [12] incorporated a sinusoidal search mechanism and a Cauchy–Gaussian mutation operator into DBO, improving the algorithm's ability to escape local optima. However, the superposition of multiple perturbation strategies increases parameter sensitivity and may lead to significant fluctuations during the convergence process. Reference [13] employed good-point-set initialization, a golden sine strategy, and t-distribution perturbation to improve population diversity and global exploration ability, but its exploitation capability in the later search stage is insufficient, resulting in limited convergence accuracy during fine-grained optimization. Reference [14] combined opposition-based learning with a directional shifting strategy to strengthen global guidance and improve late-stage convergence performance, yet its adaptability to complex, strongly coupled constrained problems has not been fully validated. Reference [15] utilized chaotic mapping and spiral opposition-based learning to accelerate convergence and enhance solution distribution uniformity, but premature convergence may still occur in high-dimensional search spaces. In summary, although existing improved DBO algorithms have made progress in enhancing search capability, maintaining diversity, or accelerating convergence, most methods focus on strengthening a single stage or a single mechanism. Consequently, achieving an effective balance among global exploration, local exploitation, and search stability remains challenging—an issue that is particularly pronounced in multi-constrained, highly nonlinear microgrid economic dispatch problems. Therefore, it is necessary to develop a novel improved algorithm with multi-strategy collaboration and stronger global–local search balance capability.

To address the remaining shortcomings of existing improved DBO variants in complex constrained environments—such as insufficient global exploration capability, limited convergence accuracy, and a tendency to become trapped in local optima—especially their difficulty in meeting the optimization requirements of microgrid economic dispatch characterized by multiple constraints, strong nonlinearity, and tight coupling, this paper proposes a multi-strategy hybrid improved algorithm, namely CWSADBO, and applies it to a microgrid economic dispatch model to enhance the solution performance for complex energy scheduling problems. The main contributions and innovations of this study are summarized as follows: (1) An optimization model is established incorporating constraints on distributed generation outputs, energy storage charging/discharging, power balance, and operating cost, providing an engineering-oriented basis for evaluating the proposed algorithm. (2) To overcome the premature convergence and slow convergence speed of the conventional DBO, the search performance is enhanced from multiple perspectives: Cubic chaotic initialization is adopted to improve population diversity; a black widow behavior mechanism is introduced to strengthen global exploration; a sparrow-inspired perturbation strategy is incorporated to reduce the risk of falling into local optima; and a Cauchy–Gaussian hybrid mutation mechanism is designed to balance global jumps and local fine-grained exploitation, thereby significantly improving convergence accuracy and stability. (3) Comparative experiments with DBO, IDBO, SSA, and SABO demonstrate that CWSADBO achieves superior convergence speed, solution accuracy, and robustness, and can effectively reduce the operating cost of microgrids.

2 Analysis of the Dung Beetle Optimization Algorithm

2.1 Population Initialization

The dung beetle is a common insect that feeds on and decomposes animal feces. To prevent other dung beetles from stealing their food, they mold the feces into balls and roll them away. Dung beetles use celestial cues—such as the sun, moon, and polarized light—for navigation, helping the dung ball to roll in a straight line. In complete darkness, they tend to roll in a curved or slightly circular path. Wind and uneven terrain can cause the beetle to deviate from its course [16]. When encountering obstacles, dung beetles will perform a "dance," which includes a series of rotations and pauses, to determine a new direction of movement.

The DBO algorithm is a novel swarm intelligence optimization algorithm inspired by the natural behavior of dung beetles [17]. By simulating the beetle's behaviors—such as rolling dung balls, obstacle avoidance, navigation, and burying the dung balls—a set of representative search mechanisms has been designed. The algorithm consists of five core stages: initialization, rolling behavior, navigation mechanism, burying mechanism, and individual best position update.

(1) Initialization Stage: Let the search space be d-dimensional and the population size be N. The initialization formula for each individual is:

$$x_i^{(0)} = lb + \left(ub - lb\right) \cdot rand\left(1, d\right), i = 1, 2, \cdots, N \tag{1}$$

In Equation (1), *lb* and *ub* represent the lower and upper bounds of each decision variable, respectively. rand(1, d) denotes a uniformly distributed random vector of dimension d.

(2) Global Rolling Behavior: Dung beetles use celestial cues such as the sun, moonlight, or polarized light to determine their rolling direction [18]. In the algorithm, this behavior is simulated as movement in a specific direction away from the individual with the worst historical fitness and the furthest from the swarm:

$$x_i^{(t+1)} = pX_i^{(t)} + 0.3 \cdot \left| pX_i^{(t)} - x_{worst}^{(t)} \right| + a \cdot 0.1 \cdot XX_i^{(t)}, a \in \left\{-1, 1\right\} \tag{2}$$

In Equation (2), $pX_i^{(t)}$ is the historically best solution of the current individual; $x_{\text{worst}}^{(t)}$ is the worst individual in the current population; $XX_i^{(t)}$ is the current position of the individual; a is navigation direction factor, indicating whether the environment interferes with navigation. This part implements comprehensive guidance based on the individual's bias direction and the dispersion from the worst solution, enhancing early-stage exploration ability.

(3) Orientation Jump Behavior: If a dung beetle encounters an obstacle during the ball-rolling process, it will perform a jumping behavior to change the rolling direction and redefine its path. The corresponding equation is:

$$x_i^{(t+1)} = pX_i^{(t)} + \tan\left(\theta\right) \cdot \left| pX_i^{(t)} - XX_i^{(t)} \right|, \theta \in \left(0, \pi\right) \tag{3}$$

In Equation (3), θ represents the deflection angle, which simulates a jump-based correction. This mechanism effectively prevents the path from being blocked by obstacles or falling into local optima.

(4) Reproduction Search Behavior: Dung beetles tend to lay eggs and bury dung balls in targeted areas. To simulate this local search behavior, a perturbation strategy centered around the optimal solution is introduced. First, a scaled-down search region is constructed:

$$\begin{cases} X_{new1} = bestX \cdot \left(1 - R\right) \\ X_{new2} = bestX \cdot \left(1 + R\right) \\ R = 1 - \dfrac{t}{M} \end{cases} \tag{4}$$

Individuals perform Gaussian perturbation around the new region.

$$x_j = bestX + \mathrm{N}\left(0, 1\right) \cdot \left(\frac{\left| pX_j - bestX \right| + \left| pX_j - bestXX \right|}{2} \right) \tag{5}$$

(5) Individual Best Update Mechanism: At the end of each generation, individuals update their local best positions based on their fitness values.

$$\begin{cases} f\left(x_i^{(t+1)}\right) < pFiT_i \Rightarrow pX_i = x_I^{(t+1)} \\ f\left(pX_i\right) < f_{\min} \Rightarrow bestX = pX_i \end{cases} \tag{6}$$

The update strategy in Equation (6) ensures the gradual accumulation and transfer of individual and global search information.

2.2 Analysis of the Limitations of the Original DBO Algorithm

Although the DBO algorithm features a novel concept and possesses certain exploration and exploitation capabilities, it still exhibits the following main shortcomings when applied to engineering problems and complex high-dimensional optimization tasks:

(1) Uneven Distribution of the Initial Population, the DBO algorithm uses uniform random initialization:

$$x_i^{(0)} = lb + \left(ub - lb\right) \cdot rand\left(1, d\right) \tag{7}$$

This method cannot ensure that individuals are evenly distributed across the entire search space. The population may become concentrated in local regions, resulting in poor exploration capability during the initial phase and slower convergence speed.

(2) Rigid Search Mechanism and Lack of Adaptive Regulation, the rolling behavior uses fixed coefficients:

$$x_i^{(t+1)} = pX_i + 0.3 \cdot \left| pX_i - x_{worst} \right| + a \cdot 0.1 \cdot XX_i \tag{8}$$

In Equation (8), all perturbation parameters are constants, lacking the ability to adaptively adjust search intensity based on the number of iterations. This limits the algorithm's ability to balance exploration and exploitation across different stages.

(3) Lack of Mechanism to Escape Local Optima, DBO lacks explicit mutation or perturbation mechanisms, making it prone to stagnation once individuals converge around a local optimum.

3 Design of the Improved CWSADBO Algorithm

3.1 Multi-Strategy Enhancement Methods

To overcome the aforementioned limitations of the DBO algorithm, this paper proposes a multi-strategy enhanced dung beetle optimization algorithm called CWSADBO. The main improvements are focused on the following four aspects:

(1) Cubic Chaotic Initialization (Enhancing Population Diversity): The traditional DBO algorithm typically uses uniform random initialization, which can lead to issues such as concentrated distribution of initial individuals, insufficient population diversity, and poor global exploration ability in high-dimensional search spaces. These factors negatively affect the overall performance of the algorithm.

To address this, the improved CWSADBO algorithm introduces the Cubic Chaotic Map for population initialization, aiming to enhance the balance of coverage and distribution diversity of the initial population in the search space.

Cubic mapping is a common nonlinear chaotic sequence generation mechanism. It features good ergodicity, aperiodicity, and sensitivity to initial values. Its form is as follows:

$$z_{n+1} = 4z_n^3 - 3z_n, z_0 \in \left(0,1\right) \tag{9}$$

Used to generate individual positions:

$$x_{x,j}^{(0)} = lb + \left|\sin\left(10\pi z_n\right)\right| \cdot \left(ub_j - lb_j\right) \tag{10}$$

By introducing aperiodicity and ergodicity through chaotic mapping, the spatial coverage ability of the initial population is improved.

(2) Black Widow Behavior to Enhance the Exploration Phase: In nature, black widows maintain the dominance of strong individuals in the population through cannibalism. This behavior inspires CWSADBO to design a bidirectional perturbation mechanism that enhances the rolling phase of the DBO algorithm, providing stronger directional guidance and perturbation-jump capabilities. To prevent individuals from falling into local optima, an inter-individual exclusion behavior (Black Widow Strategy) is introduced, modifying the rolling equation as follows:

$$x_j = pX_j + 0.4 \cdot \left|pX_j - x_{worst}\right| + a \cdot 0.15 \cdot XX_j, a \in \left\{-1,1\right\} \tag{11}$$

Meanwhile, the nonlinear navigation mechanism is retained:

$$x_j = pX_j + \tan\theta \cdot \left|pX_j - XX_j\right|, \theta \sim U\left(0,\pi\right) \tag{12}$$

The first part directs individuals away from the direction of the worst solution, preventing degeneration; the second part introduces large random jumps through angular nonlinear perturbations. The combination of both achieves strong exploration and long-range transfer in parallel optimization.

(3) Sparrow-Inspired Perturbation Mechanism (Preventing Local Optima Entrapment): CWSADBO incorporates the "discoverer behavior" from the Sparrow Search Algorithm to introduce exploratory perturbations, aiming to address the issue of local optima in complex problems. Inspired by the discoverer behavior, a nonlinear probability-based perturbation is introduced:

$$p_v = \frac{2\left(1-\left(t/M\right)^{1.5}\right)}{3} \tag{13}$$

In Equation (13), t denotes the current iteration number, and M represents the maximum number of iterations. This probability gradually decreases as the iterations progress, allowing the algorithm to emphasize global exploration in the early stage and local exploitation in the later stage. At each iteration, disturbance decisions are applied to individuals with relatively poor fitness in the population (the last 30% of individuals). If a random number $r \sim U(0,1)$ satisfies $r < p_i$, the exploratory search mechanism is triggered, and the perturbation method is selected according to the following criteria:

(1) Index Explosion Perturbation: When an individual's fitness is lower than the average fitness of the population, i.e., $f(x_i) > f_{mean}$, it is considered that the individual is located in a region with a weak search trend, prone to fall into local extrema. In this case, the index explosion perturbation is applied:

$$x_i = randn\left(1\right) \cdot \exp\left(\frac{worseX - pX_i}{i^2}\right) \tag{14}$$

The strategy in Equation (14) uses the worst individual to guide exponential jumps, expanding the search radius to achieve long-distance global exploration.

(2) Optimal Guidance Perturbation: When an individual's fitness is better than or close to the average fitness of the population, i.e., $f(x_i) \le f_{mean}$, it indicates that the individual has entered an optimal region. In this case, the optimal guidance perturbation is applied:

$$x_i = bestX + |\Delta_i| \cdot \left(A^{\mathrm{T}} \left(AA^{\mathrm{T}} \right)^{-1} \right) \cdot 1 \tag{15}$$

In Equation (15), $A \in \{-1, 1\}^{d \cdot 1}$ represents the direction vector. This strategy constructs the search direction through correct crossover, enabling the individual to search toward the optimal region. The population first performs local fine search to improve accuracy and stability.

(4) Cauchy-Gaussian Hybrid Mutation Mechanism (Adaptive Escape from Local Optima): To address search stagnation (i.e., when the global best solution does not change), CWSADBO triggers an adaptive mutation mechanism by combining Cauchy and Gaussian distributions to generate perturbations.

Activation Condition: The mechanism is activated when the following stagnation condition is met.

$$f_{\min}^{(t)} = f_{\min}^{(t-1)} = \cdots = f_{\min}^{(t-5)} \tag{16}$$

When convergence stagnates for multiple consecutive generations, a jump mechanism is introduced:

$$x_i = pX_i \cdot \left(1 + \beta_1 \cdot C_i + \beta_2 \cdot G_i\right) \tag{17}$$

In Equation (17), $C_i \sim Cauchy(0,1)$ has a long-tail property and is suitable for large-step jumps; $G_i \sim \mathrm{N}(0,1)$ is suitable for small-range fine-tuning.

The weighting factor varies with the number of generations:

$$\begin{cases} \beta_1 = 1 - \left(\dfrac{t}{M} \right)^2 \\ \beta_2 = \left(\dfrac{t}{M} \right)^2 \end{cases} \tag{18}$$

In the early search phase, Cauchy is mainly used to guide long-distance jumps; in the later search phase, Gaussian fine-tuning is employed to improve precision. The search granularity is adaptively controlled by β_1, β_2.

3.2 Time Complexity Analysis

In the standard DBO algorithm, the initial population is first generated randomly, producing a total of N individuals, each containing M decision variables. Therefore, the time complexity of the initialization stage is $O(NM)$. Next is the fitness evaluation stage, in which the objective function is evaluated once for each individual. Assuming the computational complexity of the objective function is C_f, the time complexity of this stage is $O(NC_f)$. In the position update stage, during each iteration, all individuals need to update their positions. Each individual involves basic operations such as addition and subtraction on M-dimensional vectors. Hence, the time complexity of this stage is $O(NM)$. Therefore, the time complexity of the original DBO algorithm in a single iteration is:

$$O_{iter-DBO} = O\left(NM\right) + O\left(NC_f\right) + O\left(NM\right) \approx O\left(NM + NC_f\right) \tag{19}$$

The overall time complexity of the algorithm after T iterations is:

$$O_{total-DBO} = O\left(T \cdot N \cdot \left(M + C_f\right)\right) \tag{20}$$

Under normal circumstances, the computational complexity of the objective function C_f is proportional to the problem dimension M, i.e., $C_f = O(M)$. Therefore, the overall time complexity of the original DBO algorithm can

be expressed as $O(TNM)$. In the proposed CWSADBO algorithm, during the Cubic chaotic initialization stage, the Cubic chaotic map is employed to generate the initial population, and the chaotic sequence is mapped into the search space. A total of N individuals are generated, each containing M decision variables; therefore, the time complexity of this stage remains $O(NM)$. In the fitness evaluation stage, similar to the original DBO algorithm, the fitness of all individuals is calculated, and the corresponding time complexity is $O(NC_f)$. In the black widow behavior search stage, black widow predation and mating behaviors are simulated to update individual positions. This process involves constant-time arithmetic operations on M-dimensional vectors, resulting in a time complexity of $O(NM)$. In the sparrow-inspired perturbation stage, a sparrow perturbation strategy is introduced for a subset of individuals to modify their positions. The perturbation operations are also applied to M-dimensional vectors, and each individual is processed at most once, leading to a time complexity of $O(NM)$. In the Cauchy–Gaussian hybrid mutation stage, Cauchy- and Gaussian-distributed hybrid perturbations are applied to individual positions. For each individual, each dimension undergoes only a limited number of random samplings and weighted computations; therefore, the time complexity of this stage is $O(NM)$. Since the constant K is much smaller than the population size N, the term $O(M+C_f)$ is dominated by $O(N\cdot(M+C_f))$. Therefore, the time complexity of the CWSADBO algorithm in a single iteration is:

$$O_{iter} = O\left(NM\right) + O\left(NCf\right) + O\left(NM\right) + O\left(NM\right) + O\left(NM\right) \approx O\left(N \cdot \left(M + C_f\right)\right) \tag{21}$$

The overall time complexity of the algorithm after T iterations is:

$$O_{total} = O\left(T \cdot N \cdot \left(M + C_f\right)\right) \tag{22}$$

Given that $C_f = O(M)$, the overall time complexity of the improved algorithm remains $O(TNM)$. Although the CWSADBO algorithm introduces Cubic chaotic initialization, a black widow behavior mechanism, a sparrow-inspired perturbation strategy, and a Cauchy–Gaussian hybrid mutation mechanism, which add several additional operations, these operations are all embedded within linear loops and involve only constant-order computations. Therefore, they do not change the asymptotic order of the algorithm's time complexity.

3.3 Improved Algorithm Testing

All experiments in this study were conducted on the same PC equipped with the Windows 11 (64-bit) operating system, an Intel® Core™ i7-12700H processor, and 32 GB of RAM. To verify the superiority and advancement of the proposed algorithm, in addition to the original DBO algorithm, the IDBO algorithm proposed in Reference [11], the Sparrow Search Algorithm proposed in Reference [19], and the Shark Algorithm Based Optimization Algorithm proposed in Reference [20] were also introduced for comparison. The population size was set to 30 and the maximum number of iterations was set to 500. The parameter settings of each algorithm are listed in Table 1.

Table 1. Parameter settings for each comparative algorithm

Algorithm	Parameter settings
CWSADBO	Black widow behavior ratio = 0.3; Sparrow anti-movement trigger probability = 0.3; Cauchy-Gaussian hybrid mutation disturbance parameters: Gaussian standard deviation = 0.1; Cauchy scale = 0.5; Hybrid ratio = 0.6
DBO	Rolling behavior ratio = 0.3; Dance behavior ratio = 0.2; Reproduction probability = 0.2; Stealing behavior step size factor = 0.1
IDBO	Number of mappings = 50; Chaotic factor = 0.95; Golden ratio = 1.618; Oscillation frequency = 0.1; Step size adjustment factor = 0.3; Local search radius = 0.1; Gaussian disturbance standard deviation = 0.1; Cauchy distribution scale parameter = 0.5
SABO	Step size control factor = 0.5; Local exploration probability = 0.3; Local disturbance resistance factor = 0.1; Local search radius = 0.05; Exploration intensity = 0.4
SSA	Discoverer ratio = 0.2; Follower ratio = 0.7; Sentinel ratio = 1; Safety threshold ST = 0.8; Sentinel behavior weight = 0.5; Discoverer behavior weight = 0.7; Follower behavior weight = 0.4

Sixteen standard benchmark test functions from Fig. 1 are used for performance testing. Among them: F1-F7 are high-dimensional unimodal functions with only one global optimum, suitable for testing the exploration capability of the algorithm; F8-F13 are high-dimensional multimodal functions with one global optimum and multiple local optima, suitable for evaluating the global search and local escape abilities; F14-F16 are fixed-dimensional multimodal functions used to verify the algorithm's ability to balance exploration and exploitation. By employing three different types of test functions, the optimization performance of the CWSADBO algorithm is thoroughly validated [21].

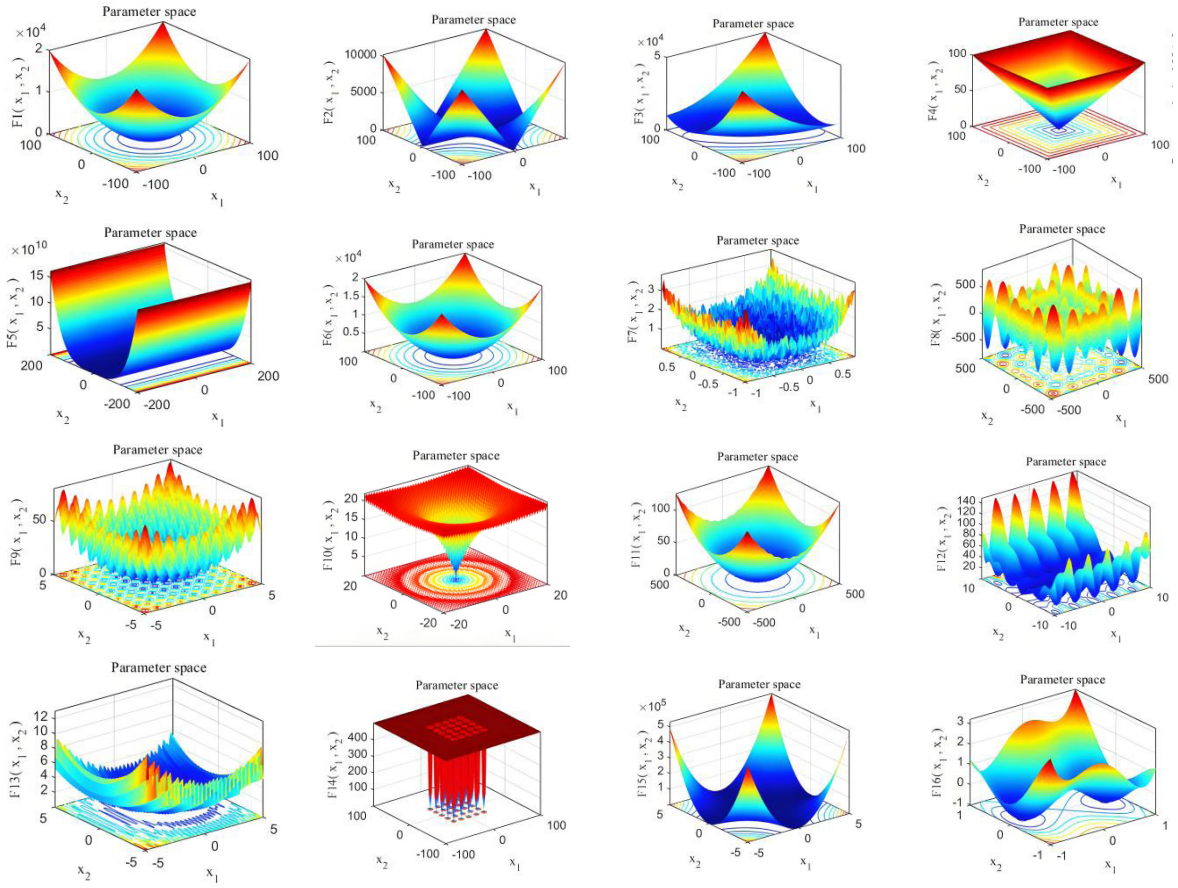


Fig. 1. Graphs of test functions F1–F16

As shown in Fig. 2, the comparison of convergence curves reveals that the CWSADBO algorithm significantly outperforms the DBO algorithm across the test functions F1 to F16. CWSADBO demonstrates faster convergence and achieves lower final fitness values on most functions, with particularly outstanding performance on F1, F2, F6, F10, and F13. In these cases, it rapidly approaches the optimal solution within the first few hundred iterations and maintains a stable convergence trend. In contrast, the DBO algorithm generally suffers from slow convergence, entrapment in local optima, and even prolonged periods without significant improvement, resulting in markedly inferior solution accuracy.

A comparison was conducted between the CWSADBO algorithm and the original DBO algorithm using three key metrics for each test function: best value, mean, and standard deviation. As shown in Table 2, CWSADBO achieved better best values, lower mean values, and smaller standard deviations on the vast majority of test functions. This demonstrates its higher convergence accuracy, stronger stability, and reduced variability.

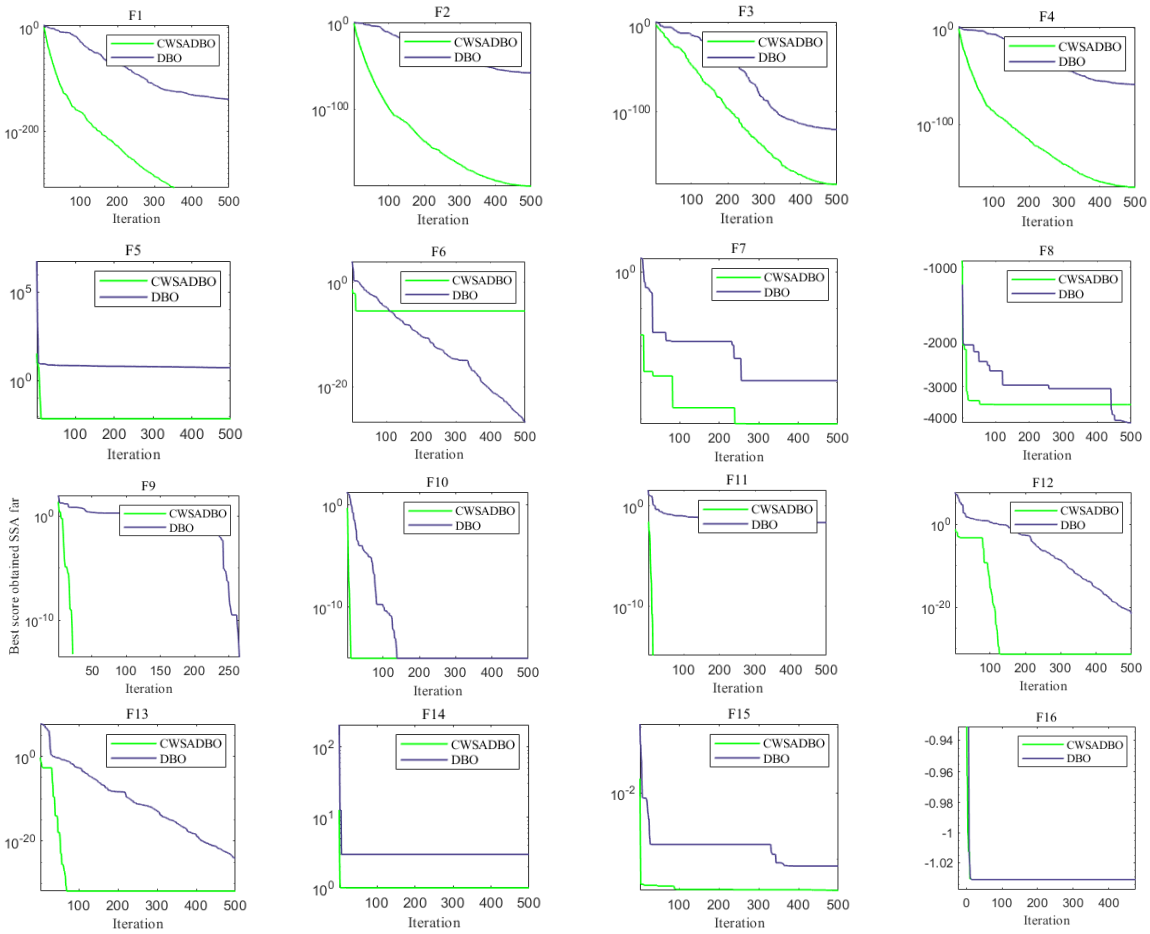


Fig. 2. Convergence Curve Comparison between DBO and CWSADBO

Table 2. Test results of DBO vs. CWSADBO

Function	Algorithm	Best value	Standard deviation	Mean value	Function	Best value	Standard deviation	Mean value
F1	CWSADBO	0.00E+00	0.00E+00	2.52E-276	F9	0.00E+00	0.00E+00	0.00E+00
F1	DBO	2.48E-159	8.70E-107	2.16E-107	F9	0.00E+00	1.63E+00	2.98E-01
F2	CWSADBO	5.94E-197	1.69E-126	3.08E-127	F10	8.88E-16	0.00E+00	8.88E-16
F2	DBO	6.79E-83	3.86E-54	7.06E-55	F10	8.88E-16	0.00E+00	8.88E-16
F3	CWSADBO	0.00E+00	0.00E+00	7.67E-183	F11	0.00E+00	0.00E+00	0.00E+00
F3	DBO	5.06E-158	3.64E-84	8.32E-85	F11	0.00E+00	5.29E-02	1.96E-02
F4	CWSADBO	6.46E-200	8.03E-136	1.47E-136	F12	4.71E-32	3.97E-08	9.34E-09
F4	DBO	3.48E-80	5.41E-49	9.89E-50	F12	2.40E-29	4.33E-22	8.33E-23
F5	CWSADBO	0.00E+00	3.72E-04	1.01E-04	F13	1.35E-32	4.10E-07	1.17E-07
F5	DBO	5.02E+00	3.89E-01	5.37E+00	F13	2.80E-28	4.55E-02	3.21E-02
F6	CWSADBO	0.00E+00	2.74E-07	8.08E-08	F14	9.98E-01	3.85E+00	2.69E+00
F6	DBO	6.16E-32	1.46E-23	3.95E-24	F14	9.98E-01	7.13E-01	1.36E+00
F7	CWSADBO	4.06E-06	5.44E-05	6.77E-05	F15	3.09E-04	4.45E-05	3.38E-04
F7	DBO	5.12E-05	4.75E-04	7.30E-04	F15	3.08E-04	3.45E-04	8.73E-04
F8	CWSADBO	-4.19E+03	6.03E+02	-3.58E+03	F16	-1.03E+00	8.38E-07	-1.03E+00
F8	DBO	-4.19E+03	5.23E+02	-3.51E+03	F16	-1.03E+00	6.12E-16	-1.03E+00

As shown in Fig. 3, the CWSADBO algorithm exhibits superior performance on most benchmark functions (e.g., F1, F2, F3, F4, F6, and F9-F13), achieving faster convergence speeds and lower final errors. These results demonstrate its strong global search capability and local exploitation ability, with particularly significant advantages in handling smooth functions and multimodal functions.

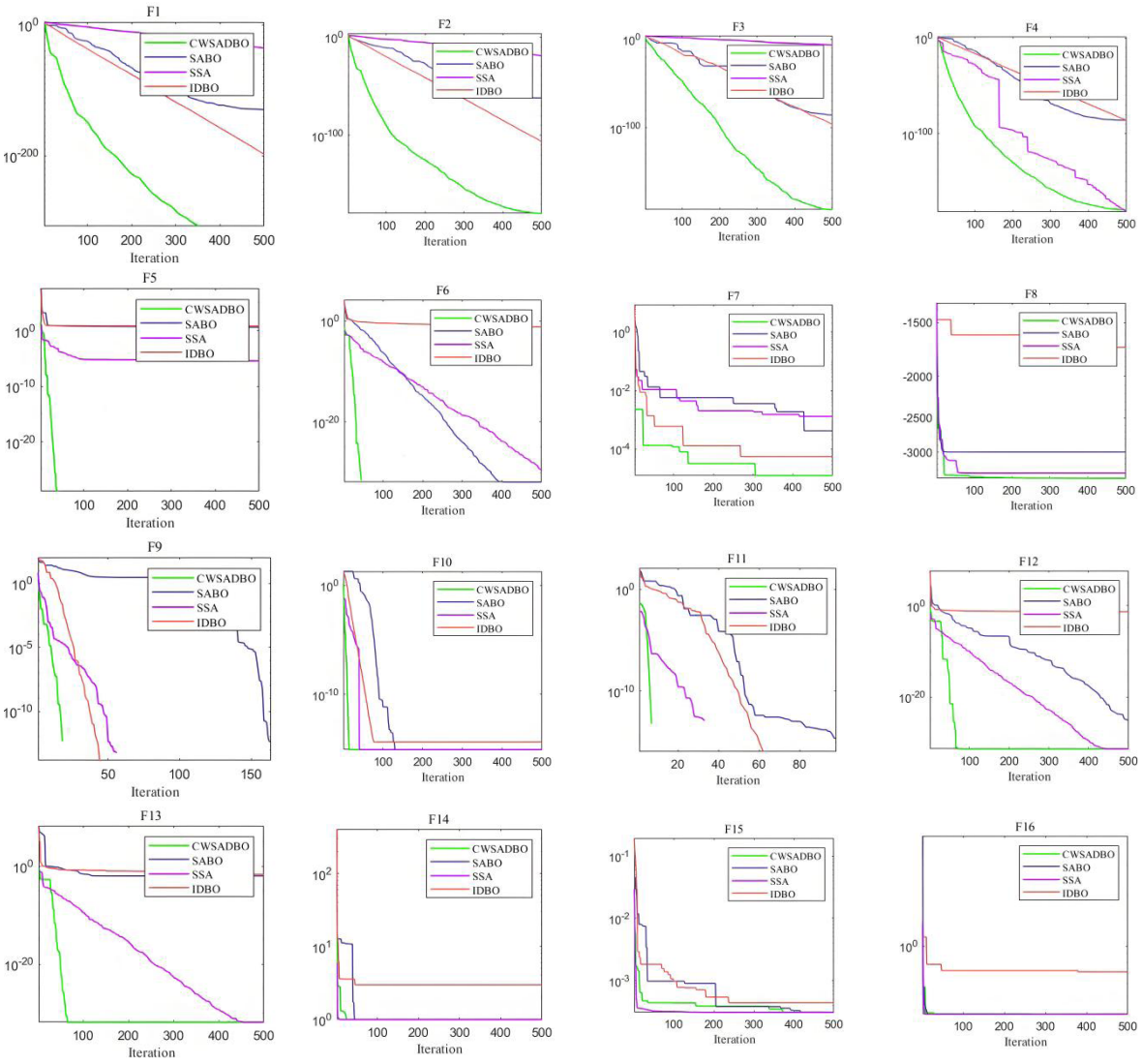


Fig. 3. Convergence curve comparison of different algorithms

The test results of the improved algorithm compared with those proposed in other studies are presented in Table 3. Overall, the CWSADBO algorithm outperforms the IDBO, SSA, and SABO algorithms in terms of the three performance indicators: best value, mean value, and standard deviation. In particular, for functions F1-F4, F6, F9-F11, and F13-F16, CWSADBO not only consistently reaches the theoretical optimal solution, but also achieves lower mean values than the other algorithms, with extremely small standard deviations. These results demonstrate its superior search accuracy and convergence stability.

Table 3. Test results of CWSADBO vs. Other algorithms

Function	Algorithm	Best value	Standard deviation	Mean value	Function	Algorithm	Best value	Standard deviation
F1	CWSADBO	0.00E+00	0.00E+00	2.52E-276	F9	0.00E+00	0.00E+00	0.00E+00
F1	IDBO	0.00E+00	0.00E+00	1.52E-96	F9	0.00E+00	0.00E+00	0.00E+00
F1	SSA	1.46E-249	5.57E-79	1.15E-79	F9	0.00E+00	0.00E+00	0.00E+00
F1	SABO	1.05E-204	0.00E+00	4.98E-199	F9	0.00E+00	0.00E+00	0.00E+00
F2	CWSADBO	5.94E-197	1.69E-126	3.08E-127	F10	8.88E-16	0.00E+00	8.88E-16
F2	IDBO	4.94E-124	2.69E-98	8.19E-90	F10	8.88E-16	0.00E+00	6.792-16
F2	SSA	3.32E-223	2.77E-39	8.72E-40	F10	8.88E-16	0.00E+00	8.88E-16
F2	SABO	2.70E-111	4.17E-109	2.13E-109	F10	8.88E-16	6.49E-16	4.32E-15
F3	CWSADBO	0.00E+00	0.00E+00	7.67E-183	F11	0.00E+00	0.00E+00	0.00E+00
F3	IDBO	2.80e-117	5.16e-75	6.56e-76	F11	0.00E+00	0.00E+00	0.00E+00
F3	SSA	1.62E-149	3.21E-34	6.02E-35	F11	0.00E+00	0.00E+00	0.00E+00
F3	SABO	3.89E-113	1.25E-79	2.28E-80	F11	0.00E+00	0.00E+00	0.00E+00
F4	CWSADBO	6.46E-200	8.03E-136	1.47E-136	F12	4.71E-32	3.97E-08	9.34E-09
F4	IDBO	4.82e-87	2.52e-84	1.22e-84	F12	3.66E-04	6.27E-02	5.93E-02
F4	SSA	0.00E+00	2.72E-29	5.03E-30	F12	4.71E-32	7.76E-34	4.76E-32
F4	SABO	1.41E-87	4.31E-84	2.27E-84	F12	9.88E-04	3.22E-02	3.94E-02
F5	CWSADBO	0.00E+00	3.72E-04	1.01E-04	F13	1.35E-32	4.10E-07	1.17E-07
F5	IDBO	6.94E+00	4.89E-01	7.45E+00	F13	1.31E-32	1.44E-33	1.66E-32
F5	SSA	2.27E-11	1.87E-05	9.31E-06	F13	1.35E-32	2.71E-33	1.56E-32
F5	SABO	6.94E+00	4.89E-01	7.45E+00	F13	5.39E-03	1.85E-01	1.78E-01
F6	CWSADBO	0.00E+00	2.74E-07	8.08E-08	F14	9.98E-01	3.85E+00	2.69E+00
F6	IDBO	0.00E+00	1.13E-04	7.37E-06	F14	9.98E-01	2.67E+00	3.29E+00
F6	SSA	0.00E+00	1.14E-32	8.53E-33	F14	9.98E-01	5.52E+00	5.61E+00
F6	SABO	3.29E-04	1.50E-01	1.02E-01	F14	9.98E-01	3.03E+00	3.21E+00
F7	CWSADBO	4.06E-06	5.44E-05	6.77E-05	F15	3.09E-04	4.45E-05	3.38E-04
F7	IDBO	5.86e-06	9.87e-05	1.11E-04	F15	3.15E-04	1.45E-05	9.05E-04
F7	SSA	6.45E-05	9.46E-04	1.15E-03	F15	3.07E-04	4.24E-05	3.15E-04
F7	SABO	6.63E-07	9.82E-05	1.06E-04	F15	3.16E-04	4.33E-03	1.96E-03
F8	CWSADBO	-4.19E+03	6.03E+02	-3.58E+03	F16	-1.03E+00	8.38E-07	-1.03E+00
F8	IDBO	-2.23E+03	1.77E+02	-1.74E+03	F16	-1.03E+00	1.72E-02	-1.02E+00
F8	SSA	-3.83E+03	2.54E+02	-3.35E+03	F16	-1.03E+00	4.97E-16	-1.03E+00
F8	SABO	-2.22E+03	1.85E+02	-1.80E+03	F16	-1.03E+00	1.42E-02	-1.02E+00

The search trajectory plots of the algorithm, as shown in the Fig. 4, demonstrate strong global search capabilities across the F1-F16 test functions. The algorithm effectively escapes local optima in multimodal and complex landscapes, successfully approaching the global optimum.

Its search paths exhibit strong directionality and coverage, showing particularly stable performance in convex functions and large-scale search spaces, with fast convergence speed and high adaptability. Moreover, in problems with narrow extreme regions or sharply defined objective functions, the algorithm maintains excellent exploration ability, reflecting its high search efficiency and flexibility.

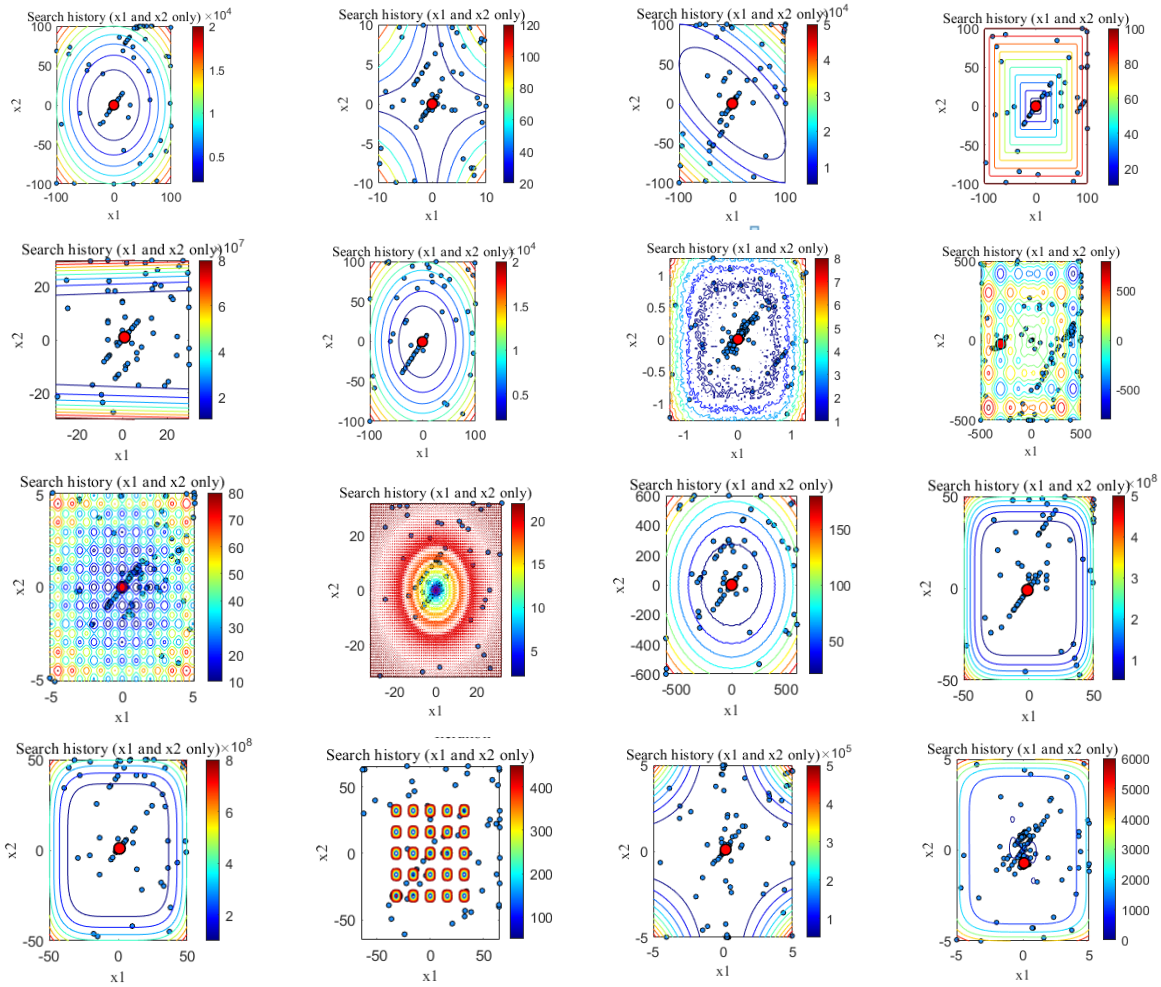


Fig. 4. CWSADBO search trajectory plot

As shown in Fig. 5, under the setting of a population size of 30 and 500 iterations, CWSABDO achieves relatively low median values on most test functions, with small interquartile ranges and only a few outliers appearing in individual cases, indicating that it can maintain good search stability even under small-population conditions. DBO exhibits larger box heights on several functions (such as F7, F9, and F11), with more dispersed result distributions and noticeable outliers, suggesting that it is prone to unstable search behavior and suffers from limited convergence performance when the population size is small. SABO shows moderate overall performance under small-population settings; on some functions (e.g., F6, F10, and F14), it presents relatively high median values and large fluctuations, implying that its global search capability is difficult to fully exploit when the population size is limited. The result distribution of SSA varies considerably across different test functions, performing acceptably on some problems but exhibiting pronounced fluctuations on more complex functions, which reflects insufficient stability. IBDO obtains relatively better median results on most functions; however, a certain degree of result dispersion can still be observed under small-population conditions, indicating that although its performance is superior to most comparison algorithms, it remains constrained by the population size.

As illustrated in Fig. 6, when the population size is increased to 80, CWSABDO exhibits low median values and compact boxplot distributions on almost all test functions, with a markedly reduced number of outliers, indicating strong robustness and consistency under large-population conditions. The performance of DBO shows a significant improvement compared with the small-population case; the box heights are substantially reduced and the median values decrease on most functions, demonstrating that enlarging the population size can effectively enhance its search accuracy and stability. SABO benefits the most from the large-population setting,

achieving notably lower median values on several functions (e.g., F6, F10, and F14) with concentrated result distributions, which reflects its good adaptability to larger population sizes. As the population size increases, the variability of SSA is alleviated to some extent; however, its median values on certain functions remain higher than those of the other algorithms, indicating relatively limited overall competitiveness. IBDO continues to deliver the best or near-best results on most test functions, characterized by small interquartile ranges and fewer outliers, demonstrating superior convergence capability and stability under large-population conditions.

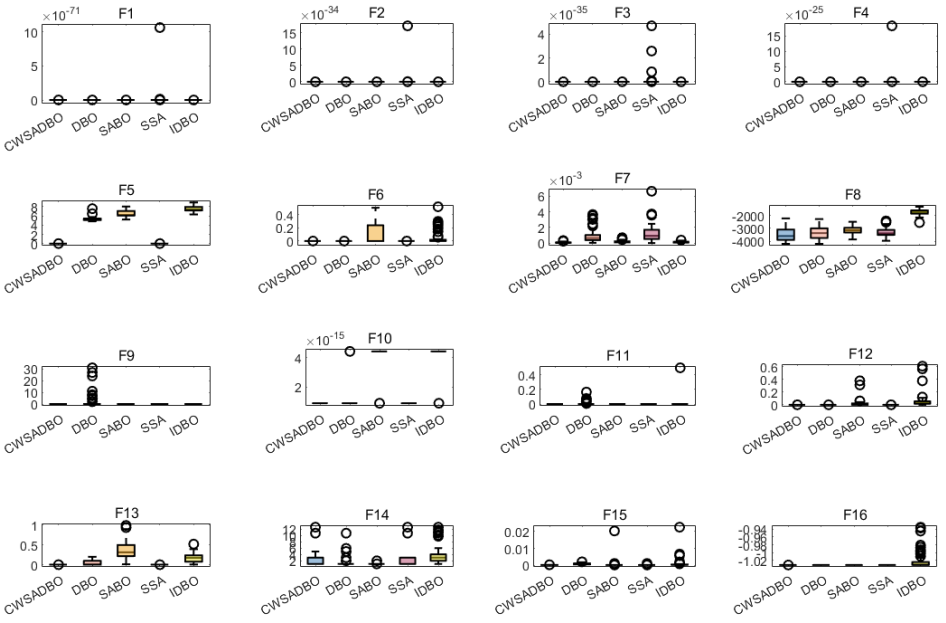


Fig. 5. Box plot with population size of 30

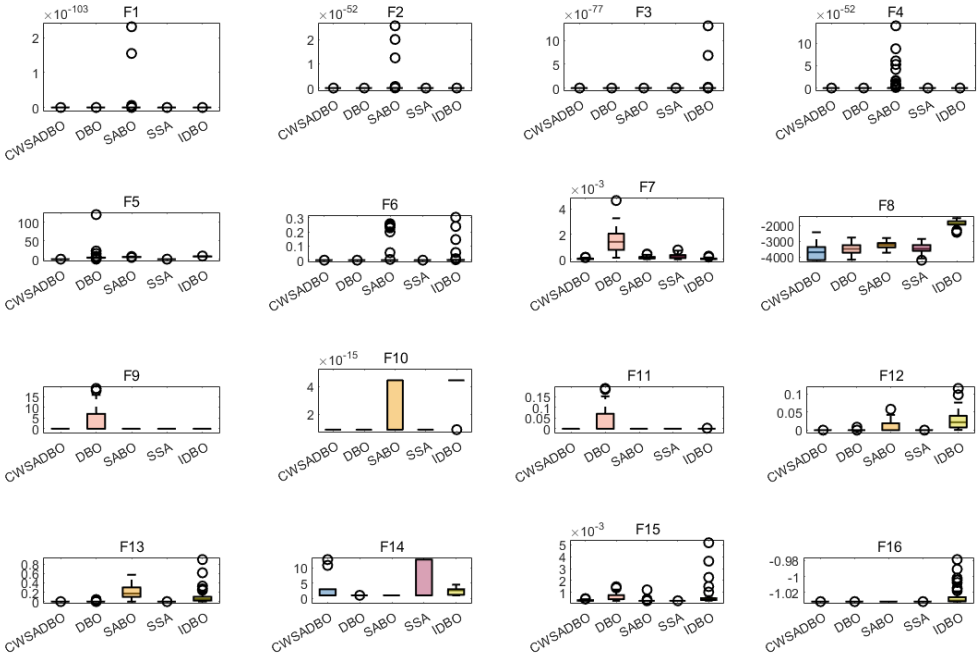


Fig. 6. Box plot with population size of 80

4 Engineering Application

4.1 Microgrid System Operation Model

A specific microgrid system is selected in this paper as a test case to model and solve its optimal scheduling problem under a typical day. The microgrid integrates various distributed energy units, including photovoltaic (PV), wind turbine (WT), diesel generator (DE), microturbine (MT), and battery energy storage system (BESS), and it is capable of bidirectional power exchange with the main grid [22]. The system supports partial islanded operation and external grid regulation capability. The operational parameters and constraints of each component are as follows:

PV and wind power are uncontrollable renewable sources, and their output is based on historical forecast data, with maximum power outputs of 50 kW and 100 kW, respectively. As they do not consume fuel or emit pollutants during operation, both operation and emission costs are considered zero. The diesel generator has a power output range of 6–30 kW, with a unit operation cost of 0.128 CNY/kWh and a ramp rate limit of 1.5 kW/min. It emits pollutants during generation, with emission factors: CO_2: 680 g/kWh, SO_2: 0.306 g/kWh, NO_x: 10.09 g/kWh. The pollutant treatment costs are CO_2: 0.023 CNY/kg, SO_2: 6 CNY/kg, NO_x: 8 CNY/kg. The microturbine has a power output range of 3–30 kW, a unit operation cost of 0.0293 CNY/kWh, and the same ramp rate limit of 1.5 kW/min. Its emission factors are CO_2: 724 g/kWh, SO_2: 1.8 g/kWh, NO_x: 0.2 g/kWh, and treatment costs are the same as for the diesel generator. The main grid power range is set to −30 to 30 kW, with positive values indicating power purchase from the grid and negative values representing power sales to the grid. The equivalent emission factors for the main grid are CO_2: 889 g/kWh, SO_2: 0.0036 g/kWh, NO_x: 1.6 g/kWh. Due to the complexity of grid energy sources, environmental costs are calculated based on unified emission factors. The operation cost of the main grid is considered zero, and economic cost is determined by the electricity price curve. The battery energy storage system (BESS) has an initial state of charge (SOC) of 50 kWh, with a capacity range of 5–150 kWh, a maximum charge/discharge power of 30 kW, and a charge/discharge efficiency of 90%. The dispatch of the storage system must satisfy SOC constraints; violations such as exceeding energy limits or frequent large power variations will incur penalties.

4.2 Characteristics of Distributed Generation and Energy Storage in the Microgrid

(1) Wind Turbine Model (WT)

The wind speed-power characteristic curve of a wind turbine (WT) is as follows [23]:

$$P_{WT}^{'} = \begin{cases} 0 & v < v_{ci} \\ a^{'}v^3 + b^{'}v^2 + c^{'}v + d^{'} & v_{ci} \le v \le v_r^{'} \\ P_r^{'} & v_r^{'} < v < v_{co}^{'} \\ 0 & v \ge v_{co}^{'} \end{cases} \tag{23}$$

In Equation 23, $P_{WT}^{'}$ represents the output power, $P_r^{'}$ is the rated power, v_{ci}, $v_r^{'}$, $v_{co}^{'}$ represent the cut-in wind speed, rated wind speed, and cut-out wind speed, respectively, and a', b', c', d' are the wind speed characteristic fitting coefficients.

(2) Photovoltaic (PV) Generation Model

In microgrid optimization scheduling, it is necessary to predict the power output of photovoltaic (PV) cells. The output power model of the PV cell is as follows [24]:

$$P_{PV}^{'} = R_{PV}^{'} q_{PV}^{'} \frac{I_T}{I_{STC}} \left[1 + a_p^{'} \left(T_C^{'} - T_{STC}^{'}\right)\right] \tag{24}$$

In Equation 24, $P_{PV}^{'}$ represents the output power of the photovoltaic system, $R_{PV}^{'}$ is the standard output power, $q_{PV}^{'}$ is the derating factor, I_T, I_{STC} are the current and standard test conditions of solar irradiance, $a_p^{'}$ is the temperature correction coefficient, and $T_C^{'}$, $T_{STC}^{'}$ are the current and standard temperature conditions, respectively.

(3) Diesel Engine Generation Model (DE)

A diesel engine is a commonly used fuel generator. During operation, it generates fuel costs, operation and maintenance costs, and pollution treatment costs. The operating cost of a diesel generator consists of three parts [25]:

$$C_{DE}\left(P_{DE}\right)=C_{DE}^{fuel}+C_{DE}^{OM}+C_{DE}^{emission} \tag{25}$$

In Equation 25, the fuel cost is:

$$C_{DE}^{fuel}=a_{DE}P_{DE}^{2}+\beta_{DE}P_{DE}+\gamma_{DE} \tag{26}$$

Operation and maintenance costs (calculated linearly based on power generation):

$$C_{DE}^{OM}=c_{DE}^{OM}\cdot P_{DE} \tag{27}$$

Pollution treatment costs (considering emissions such as SO_2, NO_x, CO_2, etc.):

$$C_{DE}^{emission}=\sum_{k}\delta_{k}^{DE}\cdot E_{K}\left(P_{DE}\right) \tag{28}$$

In the above equation, a_{DE}, β_{DE}, γ_{DE} are the fuel consumption cost coefficients, c_{DE}^{OM} is the unit operation and maintenance cost, δ_{k}^{DE} represents the cost of pollutant treatment per unit for pollutant k, and $E_k(P_{DE})$ is the emission function of pollutant k.

(4) Gas Turbine Generation Model (MT)

A gas turbine generates electricity by consuming fuel [26]. The output power of a micro gas turbine is adjustable and has a fast response speed. The operating cost of a gas turbine also consists of three parts:

$$C_{MT}\left(P_{MT}\right)=C_{MT}^{fuel}+C_{MT}^{OM}+C_{MT}^{emission} \tag{29}$$

Fuel cost (considering efficiency non-linearity):

$$C_{MT}^{fuel}=\theta_{MT}\cdot\frac{P_{MT}}{\eta_{MT}\left(P_{MT}\right)} \tag{30}$$

Operation and maintenance costs are:

$$C_{MT}^{OM}=c_{MT}^{OM}\cdot P_{MT} \tag{31}$$

Pollution treatment costs:

$$C_{MT}^{emission}=\sum_{k}\delta_{k}^{MT}\cdot E_{K}\left(P_{MT}\right) \tag{32}$$

In Equations 30-32, θ_{MT} is the fuel price per unit, $\eta_{MT}(P_{MT})$ is the fuel efficiency function of the gas turbine, c_{MT}^{OM} is the operation and maintenance cost per unit power, δ_{k}^{MT} is the pollutant treatment coefficient, and $E_K(P_{MT})$ is the pollutant emission function.

(5) Energy Storage System Model (BESS)

Batteries, as a type of energy storage device, release stored energy when renewable energy cannot meet the

load, providing a stable power source to users and ensuring the safe and stable operation of the system [27-28]. The state of charge (SOC) variation of the energy storage system is expressed as follows:

$$SOC_i = SOC_{i-1} + \eta_i \cdot P_{BESS,i} \tag{33}$$

In Equation 33, $P_{BESS,i}$ represents the charging/discharging power rate of the energy storage system at time i.

4.3 System Objective Function

(1) Operating Cost Objective Function

The optimization adjustment period is T (unit: hours), and the interaction power between the microgrid and the main grid at time t is $P_{grid}(t)$. The output power of the diesel generator, gas turbine, and energy storage system at time t is represented as $P_{DE}(t)$, $P_{MT}(t)$, $P_{ESS}(t)$, respectively. The operating costs are expressed as $C_{DE,OM}$, $C_{MT,OM}$, C_{bess}, and the purchasing/selling electricity prices are $C_{buy}(t)$, $C_{sell}(t)$, and the microgrid operation cost is the objective function.

$$f_1 = \sum_{t=1}^{T} \left(C_{grid}(t) + C_{MT}(t) + C_{DE}(t) + C_{BESS}(t) + C_{pen}(t) \right) \tag{34}$$

The expressions of each term in Equation 34 are shown as follows:

$$\begin{cases} C_{grid}(t) = \begin{cases} c_{buy}(t) \cdot P_{grid}(t), P_{grid}(t) > 0 \\ -c_{sell}(t) \cdot P_{grid}(t), P_{grid}(t) < 0 \end{cases} \\ C_{BESS}(t) = c_{bess} \cdot \left| P_{ESS}(t) \right| \\ C_{DE}(t) = c_{DE,OM} \cdot P_{DE}(t) + a_{DE} P_{DE}^2(t) + b_{DE} P_{DE}(t) + c_{DE} \\ C_{MT}(t) = c_{MT,OM} \cdot P_{MT}(t) + \dfrac{d \cdot P_{MT}(t)}{e P_{MT}^3(t) + f P_{MT}^2(t) + g P_{MT}(t) + h} \end{cases} \tag{35}$$

(2) Environmental Cost Objective Function

The pollutants are classified as nn, and the processing cost per unit of pollutant k is ck. The equipment $j \in \{grid, MT, DE\}$ has corresponding emission coefficients $\gamma_{j,k}$, and the purchasing power is defined as $P_{buy}(t) = \max(0, P_{grid}(t))$. The environmental cost objective function of the microgrid is:

$$f_2 = \sum_{t=1}^{T} \left(C_{GRID \cdot EN}(t) + C_{MT \cdot EN}(t) + C_{DE \cdot EN}(t) \right) \tag{36}$$

In Equation 36, the expressions of each subterm are as follows:

$$\begin{cases} C_{GRID \cdot EN}(t) = \sum_{k=1}^{n} c_K \cdot \gamma_{grid,k} \cdot P_{buy}(t) \\ C_{MT \cdot EN}(t) = \sum_{k=1}^{n} c_K \cdot \gamma_{MT,k} \cdot P_{MT}(t) \\ C_{DE \cdot EN}(t) = \sum_{k=1}^{n} c_K \cdot \gamma_{DE,k} \cdot P_{DE}(t) \end{cases} \tag{37}$$

(3) Comprehensive Objective Function

The objective function of the scheduling model is to minimize the total cost, which includes both operating costs and environmental protection costs. Therefore, the objective function is defined as follows:

$$Z = f_1 + f_2 \tag{38}$$

In Equation 38, z is the total cost of the microgrid, consisting of the sum of the microgrid's operating costs and environmental protection costs.

4.4 Model Constraints

(1) Energy Storage Power and SOC Constraints

The microgrid must satisfy supply and demand balance at each moment. Considering the charging and discharging efficiency of the energy storage, the following equations are established:

During energy storage charging:

$$P_{load}(t) + P_{sell}(t) + \eta_c P_{ESS}(t) = P_{WT}(t) + P_{PV}(t) + P_{DE}(t) + P_{MT}(t) + P_{buy}(t) \tag{39}$$

During energy storage discharging:

$$P_{load}(t) + P_{sell}(t) = P_{WT}(t) + P_{PV}(t) + P_{DE}(t) + P_{MT}(t) + P_{buy}(t) + \frac{P_{ESS}(t)}{\eta_d} \tag{40}$$

The dynamic energy balance and boundary conditions of the energy storage state over time are as follows:

$$\begin{cases} SOC(t) = SOC(t-1) + \begin{cases} \eta_c \cdot P_{ESS}(t), P_{ESS}(t) < 0 \\ \dfrac{1}{\eta_d} \cdot P_{ESS}(t), P_{ESS}(t) > 0 \end{cases} \\ SOC_{\min} \le SOC(t) \le SOC_{\max} \end{cases} \tag{41}$$

Diesel generator output constraints:

$$\begin{cases} P_{DE}^{\min}(t) \le P_{DE}(t) \le P_{DE}^{\max}(t) \\ \left| P_{DE}(t) - P_{DE}(t-1) \right| \le r_{DE} \end{cases} \tag{42}$$

Micro gas turbine output constraints:

$$\begin{cases} P_{MT}^{\min}(t) \le P_{MT}(t) \le P_{MT}^{\max}(t) \\ \left| P_{MT}(t) - P_{MT}(t-1) \right| \le r_{MT} \end{cases} \tag{43}$$

Main grid interaction power constraints:

$$P_{grid}^{\min}(t) \le P_{grid}(t) \le P_{grid}^{\max}(t) \tag{44}$$

In the above equation, $P_{DE}^{\max}(t)$, $P_{DE}^{\min}(t)$, $P_{MT}^{\max}(t)$, $P_{MT}^{\min}(t)$ represent the output power of the diesel generator and

micro gas turbine, respectively. r_{DE}, r_{MT} are the upper limits for the diesel generator and micro gas turbine's rated power, respectively. $P_{grid}^{\min}(t)$, $P_{grid}^{\max}(t)$ are the upper and lower limits of the grid interaction power, respectively. $SOC_{\min}$, $SOC_{\max}$ are the lower and upper limits of the energy storage state of charge, and $SOC(t)$ is the energy storage power at time t.

When the solution violates the above constraints, a penalty term will be introduced into the total cost:

$$C_{pen}(t) = f_{ele} \cdot \left|\varepsilon_{load}(t)\right| + f_{SOC} \cdot \delta_{soc}(t) + f_{DE} \cdot \left|\Delta P_{DE}(t)\right| + f_{MT} \cdot \left|\Delta P_{MT}(t)\right| \tag{45}$$

4.5 Test Results

As shown in Fig. 7, the objective function values of the five algorithms all decrease gradually with the increase in the number of iterations, indicating that each compared algorithm possesses a certain optimization capability. Among them, CWSADBO exhibits a faster convergence speed in the initial stage, rapidly reducing the objective function value within a relatively small number of iterations, and maintains stable convergence in the later stage. It ultimately achieves the lowest objective function value, demonstrating strong global exploration capability and local exploitation ability. IDBO shows the second-best optimization performance after CWSADBO, but requires a longer convergence time. In contrast, SABO, DDO, and SSA converge more slowly, and their final convergence accuracy is relatively lower, making them more prone to being trapped in inferior local optimal solutions.

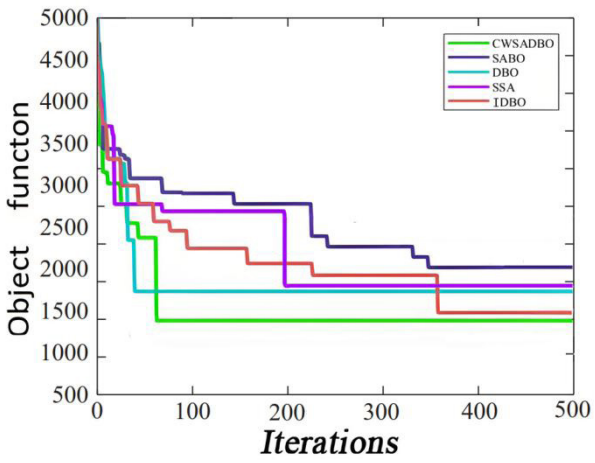


Fig. 7. Convergence diagram of algorithm solution

Table 4 presents the best value, worst value, mean value, and standard deviation of five algorithms over 30 independent test runs. As shown in Table 3, the average cost of CWSADBO is 1515.41, which is significantly superior to the other comparative algorithms. Compared with SABO, DBO, SSA, and IDBO, the average cost is reduced by 754.82, 495.06, 675.12, and 205.07, respectively, corresponding to performance improvements of 33.25%, 24.62%, 30.82%, and 11.92%. These results indicate that the proposed algorithm exhibits clear advantages in global search capability and solution accuracy, enabling it to obtain higher-quality solutions. From the distribution of the best and worst values, CWSADBO shows a relatively small gap between its best and worst solutions, indicating weaker result fluctuations across different random runs. This suggests that the algorithm has a more stable search process and is less prone to being trapped in poor local optima. Further analysis from the perspective of standard deviation demonstrates the robustness of the algorithm: the standard deviation of CWSADBO is only 13.89, which is significantly lower than that of SABO (76.48), DBO (68.87), SSA (141.48), and IDBO (88.38). Compared with the other algorithms, the result fluctuations are reduced by 81.84%, 79.83%, 90.18%, and 84.28%, respectively. This indicates that CWSADBO has stronger stability and consistency across multiple independent runs and can continuously produce high-quality solutions.

Table 4. Algorithm test results

Algorithm	Optimal solution	Worst value	Average	Standard deviation
CWSADBO	1487.64	1543.19	1515.41	13.89
SABO	2117.27	2423.19	2270.23	76.48
DBO	1872.73	2148.21	2010.47	68.87
SSA	1907.54	2473.46	2190.53	141.48
IDBO	1543.72	1897.24	1720.48	88.38

As shown in Fig. 8 to Fig. 11, when minimizing operating cost is taken as the sole objective function, the system prioritizes dispatching power generation units with the lowest unit cost to meet the load demand. Among conventional power sources, the micro gas turbine (MT) and diesel engine (DE) serve as the primary load-bearing units of the system. As illustrated in Fig. 11, the average output power of the DE is consistently maintained within the range of 20–28 kW, with almost no interruptions observed; meanwhile, the output of the MT remains stable in the 15–25 kW range and exhibits a certain degree of dynamic regulation capability. This is because both the DE and MT possess strong continuous power output capabilities and relatively low unit operation and maintenance costs. According to the modeling parameters, the unit generation cost of the DE is 0.128 CNY/kWh, while that of the MT is 0.0293 CNY/kWh. Although the MT has advantages in terms of emissions under environmental objectives, its gas-driven configuration also demonstrates exceptionally high unit cost efficiency under economic objectives.

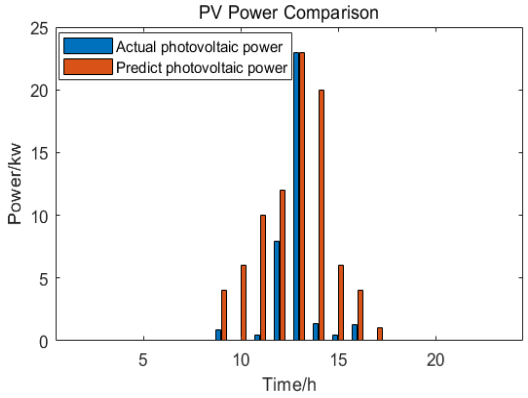


Fig. 8. Comparison diagram of photovoltaic dispatching

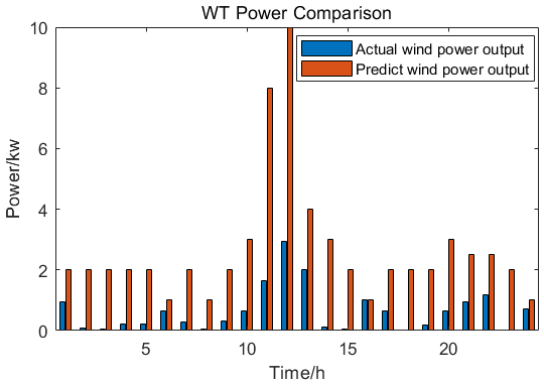


Fig. 9. Comparison diagram of wind power dispatching

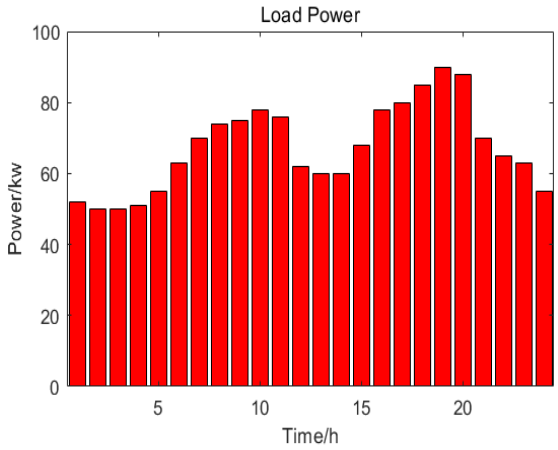


Fig. 10. Load forecasting chart

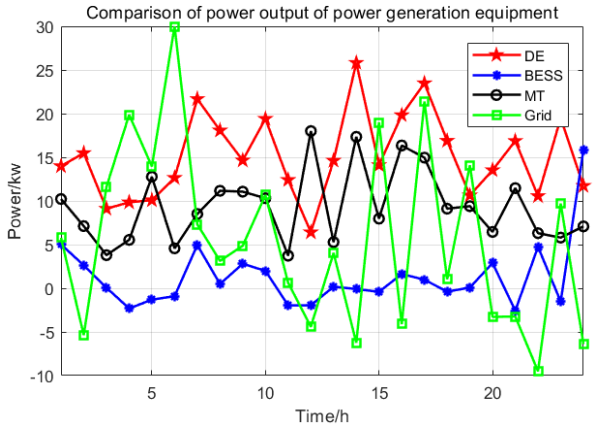


Fig. 11. Power dispatch results of DE, BESS, MT, and the Grid

To further evaluate the optimization capability of the proposed algorithm for microgrid dispatching, ablation experiments were conducted. Four comparative algorithms were designed, and each algorithm was tested over 30 independent runs. The first algorithm, DBO1, incorporates only Cubic chaotic initialization; the second, DBO2, introduces the black widow behavior mechanism; the third, DBO3, integrates a sparrow-inspired heuristic perturbation strategy; and the fourth, DBO4, employs a Cauchy–Gaussian hybrid mutation mechanism. The experimental results are presented in Table 5.

Table 5. Ablation experiment test table

Indicator	CWSADBO	DBO1	DBO2	DBO3	DBO4
Optimal solution	1487.64	1662.38	1578.41	1591.76	1558.53
Worst value	1543.19	1768.62	1669.47	1642.33	1674.28
Average	1515.41	1694.85	1648.73	1612.94	1604.02
Standard deviation	13.89	36.56	32.77	30.14	23.94

The ablation experimental results indicate that the complete CWSADBO significantly outperforms all reduced variants in terms of both optimization accuracy and stability. Its mean value is 1515.41, representing reductions of 10.59%, 8.09%, 6.05%, and 5.52% compared with DBO1–DBO4, respectively. Moreover, both the best and worst values achieved by CWSADBO are superior to those of all comparative algorithms, demonstrating that the complete combination of strategies can obtain higher-quality solutions. In terms of stability, the standard

deviation of CWSADBO is only 13.89, which is approximately 42%–62% lower than those of the ablation variants. This indicates that the proposed improvement mechanisms exhibit a synergistic enhancement effect in suppressing result fluctuations and improving the robustness of the algorithm.

5 Conclusion

This paper proposes a multi-strategy cooperative optimization algorithm termed CWSADBO. By introducing a Cubic chaotic map for population initialization, the proposed method effectively enhances the uniformity and diversity of the initial population, providing a more promising solution space for subsequent search. During the search stage, a black widow behavior mechanism is integrated to strengthen the global exploration capability of the population, while a sparrow-inspired perturbation strategy is incorporated to improve the algorithm's ability to escape local optima. Furthermore, an adaptive Cauchy–Gaussian hybrid mutation mechanism is employed to achieve a dynamic balance between long-distance exploratory jumps and fine-grained local exploitation, thereby comprehensively improving the optimization accuracy and robustness of the algorithm. The synergistic interaction among these strategies enables an effective complementary mechanism between global exploration and local exploitation, significantly enhancing the search performance in complex optimization problems. Comparative experimental results on multiple standard benchmark functions demonstrate that the proposed algorithm outperforms several existing optimization algorithms in terms of convergence speed, solution accuracy, and stability, reflecting its strong comprehensive optimization capability. Moreover, when applied to practical engineering design problems, the algorithm also exhibits good adaptability and reliability, validating its practical application potential in the field of engineering optimization. Nevertheless, CWSADBO still has certain limitations. In extremely high-dimensional complex optimization problems or in special function landscapes characterized by narrow valleys and weak gradient features, the algorithm may still encounter risks of search stagnation or reduced convergence speed, which are common issues among most swarm intelligence optimization algorithms. In addition, the algorithm involves multiple control parameters; although it generally demonstrates good robustness, it may still exhibit sensitivity to parameter settings under specific problem scenarios. Future research will focus on the following directions: (1) investigating adaptive parameter tuning strategies to enhance generalization across different problem types; and (2) exploring the application potential of CWSADBO in neural network training, complex engineering system optimization, and multi-objective optimization, as well as integrating it with other intelligent optimization mechanisms to construct more efficient hybrid optimization frameworks.

References

[1] Y.S. Liang, Z.L. Xu, H.F. Li, G. Wang, Z.J. Huang, Z.K. Li, A random optimization strategy of microgrid dispatching based on stochastic response surface method considering uncertainty of renewable energy supplies and load demands, International Journal of Electrical Power & Energy Systems 154(2023) 109408.
https://doi.org/10.1016/j.ijepes.2023.109408

[2] C. Zhao, B. Wang, Z.X. Sun, X. Wang, Optimal Configuration Optimization of Islanded Microgrid Using Improved Grey Wolf Optimizer Algorithm, Acta Energiae Solaris Sinica 43(1)(2022) 256-262.
https://doi.org/10.19912/j.0254-0096.tynxb.2020-0042

[3] H.G. Jia, Z.D. Wang, Y.Y. Yue, H. Yan, N. Yan, Q.F. Cao, Improvement of the optimization performance and solution efficiency of microgrid scheduling models, Electric Power 56(6)(2023) 107-131.
https://doi.org/10.11930/j.issn.1004-9649.202212028

[4] V.A. Suryad, S. Doolla, M. Chandorkar, Microgrids in India: possibilities and challenges, IEEE Electrification Magazine 5(2)(2017) 47-55.
https://doi.org/10.1109/MELE.2017.2685880

[5] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proc. 1995 International Conference on Neural Networks, 1995.
https://doi.org/10.1109/ICNN.1995.488968

[6] M. Dorigo, M. Birattari, T. Stützle, Ant colony optimization, IEEE Computational Intelligence Magazine 1(4)(2006) 28-39.
https://doi.org/10.1109/MCI.2006.329691

[7] F. Pourpanah, R. Wang, C.P. Lim, X.-Z. Wang, D. Yazdani, A review of artificial fish swarm algorithms: recent advances and applications, Artificial Intelligence Review 56(3)(2023) 1867-1903.

https://doi.org/10.1007/s10462-022-10214-4
[8] I.J. Jacobi, P.E. Darney, Artificial bee colony optimization algorithm for enhancing routing in wireless networks, Journal of Artificial Intelligence and Capsule Networks 3(1)(2021) 62-71.
https://doi.org/10.36548/jaicn.2021.1.006
[9] J. Xue, B. Shen, Dung beetle optimizer: a new meta-heuristic algorithm for global optimization, The Journal of Supercomputing 79(7)(2023) 7305-7336.
https://doi.org/10.1007/s11227-022-04959-6
[10] X. Zhang, W. Yue, L. Zhu, L. Wang, L. Li, Experience playback-based DBO algorithm for cooperative multi-type task allocation in sea-air heterogeneous unmanned systems, Ocean Engineering 333(2025) 121464.
https://doi.org/10.1016/j.oceaneng.2025.121464
[11] J. Fu, C. Wu, J. Wang, M.M. Haque, L. Geng, J. Meng, Lithium-ion battery SOH prediction based on VMD-PE and improved DBO optimized temporal convolutional network model, Journal of Energy Storage 87(2024) 111392.
https://doi.org/10.1016/j.est.2024.111392
[12] J. Pan, S. Li, P. Zhou, G. Yang, D. Lyu, Dung beetle optimization algorithm guided by improved sine algorithm, Computer Engineering and Applications 59(22)(2023) 92-110.
https://doi.org/10.3778/j.issn.1002-8331.2305-0021
[13] D. Zhang, Z. Wang, Y. Zhao, F. Sun, Multi-strategy fusion improved dung beetle optimization algorithm and engineering design application, IEEE Access 12(2024) 97771-97786.
https://doi.org/10.1109/ACCESS.2024.3408644
[14] Z. Wang, P. Shao, A multi-strategy dung beetle optimization algorithm for optimizing constrained engineering problems, IEEE Access 11(2023) 98805-98817.
https://doi.org/10.1109/ACCESS.2023.3313930
[15] J.X. Jia, Z.C. Xie, L.N. Ge, Multi-strategy improved dung beetle optimization algorithm for mobile robot path planning, Journal of Chongqing University of Technology (Natural Science) 39(3)(2025) 133-140.
http://clgzk.qks.cqut.edu.cn/CN/abstract/abstract7305.shtml
[16] Z. Yin, L. Zinn-Björkman, Simulating rolling paths and reorientation behavior of ball-rolling dung beetles, Journal of Theoretical Biology 486(2020) 110106.
https://doi.org/10.1016/j.jtbi.2019.110106
[17] L. Li, L. Liu, Y. Shao, X. Zhang, Y. Chen, C. Guo, H. Nian, Enhancing swarm intelligence for obstacle avoidance with multi-strategy and improved dung beetle optimization algorithm in mobile robot navigation, Electronics 12(21)(2023) 4462.
https://doi.org/10.3390/electronics12214462
[18] M. Dacke, E. Baird, B. El Jundi, E.J. Warrant, M. Byrne, How dung beetles steer straight, Annual Review of Entomology 66(1)(2021) 243-256.
https://doi.org/10.1146/annurev-ento-042020-102149
[19] J. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, Systems Science & Control Engineering 8(1)(2020) 22-34.
https://doi.org/10.1080/21642583.2019.1708830
[20] M.H. Hassan, S. Kamel, A. Selim, A. Shaheen, J. Yu, R. El-Sehiemy, Efficient economic operation based on load dispatch of power systems using a leader white shark optimization algorithm, Neural Computing and Applications 36(18)(2024) 10613-10635.
https://doi.org/10.1007/s00521-024-09612-2
[21] Q. Fan, Z. Chen, Z. Li, Z. Xia, J. Yu, D. Wang, A new improved whale optimization algorithm with joint search mechanisms for high-dimensional global optimization problems, Engineering with Computers 37(3)(2021) 1851-1878.
https://doi.org/10.1007/s00366-019-00917-8
[22] P.A. Gbadega, Y.X. Sun, JAYA algorithm-based energy management for a grid-connected micro-grid with PV-wind-microturbine-storage energy system, International Journal of Engineering Research in Africa 63(2023) 159-184.
https://doi.org/10.4028/p-du1983
[23] P. Lap-Arparat, T. Leephakpreeda, Real-time maximized power generation of vertical axis wind turbines based on characteristic curves of power coefficients via fuzzy pulse width modulation load regulation, Energy 182(2019) 975-987.
https://doi.org/10.1016/j.energy.2019.06.098
[24] A.H. Arab, B. Taghezouit, K. Abdeladim, S. Semaoui, A. Razagui, A. Gherbi, S. Boulahchiche, I.H. Mahammed, Maximum power output performance modeling of solar photovoltaic modules, Energy Reports 6(S1)(2020) 680-686.
https://doi.org/10.1016/j.egyr.2019.09.049
[25] S. Zaichenko, D. Derevianko, Comparison of the energy efficiency of synchronous power generator with spark ignition engine using different types of fuels, in: Proc. 2023 Systems, Decision and Control in Energy V, 2023.
https://doi.org/10.1007/978-3-031-35088-7_10
[26] G.E. Marin, G.R. Mingaleeva, M.S. Novoselova, A.R. Akhmetshin, Adding hydrogen fuel to the synthesis gas for the possibility of combustion in a gas turbine, International Journal of Hydrogen Energy 96(2024) 378-384.
https://doi.org/10.1016/j.ijhydene.2024.11.305

[27] H. Jain, B. Mather, A.K. Jain, S.F. Baldwin, Grid-supportive loads—a new approach to increasing renewable energy in power systems, IEEE Transactions on Smart Grid 13(4)(2022) 2959-2972.
https://doi.org/10.1109/TSG.2022.3153230
[28] M.M. Islam, T. Yu, G. Giannoccaro, Y. Mi, M.L. Scala, M.R. Nasab, Improving reliability and stability of the power systems: A comprehensive review on the role of energy storage systems to enhance flexibility, IEEE Access (12)(2024) 152738-152765.
https://doi.org/10.1109/ACCESS.2024.3476959