Advanced Bamboo Forest Growth Optimization Algorithm for Optimization of Water Pump Scheduling for Water Distribution Network

Jeng-Shyang Pan[1,3], *An-Ning Zhang*[2], *Shu-Chuan Chu*[2,4*], *Qing Feng*[2], *Yu-Chung Huang*[5]

[1] *School of Artificial Intelligence, Nanjing University of Information Science and Technology, China*
[2] *College of Computer Science and Engineering, Shandong University of Science and Technology, China*
[3] *Department of Information Management, Chaoyang University of Technology, Taiwan*
[4] *Information Engineering College, Guangzhou City Construction College, China*
[5] *Department of Electrical Engineering, National University of Tainan, Taiwan*
jengshyangpan@gmail.com, zhanganning@sdust.edu.cn, scchu0803@gmail.com,
202082060009@sdust.edu.cn, kisan7119@gmail.com

Abstract

Metaheuristic optimization algorithms are problem-solving techniques inspired by natural or abstract phenomena and find wide applications across various fields. The bamboo forest growth optimization algorithm (BFGO) is a meta-heuristic optimization algorithm inspired by the growth characteristics of bamboo forests. It has a strong convergence ability and therefore can effectively solve high-dimensional complex optimization problems. However, for multi-modal problems and composite problems, BFGO still has the disadvantage of easily falling into local optimality. In response to the above shortcomings, this article proposes a bamboo forest growth optimization algorithm based on the clustering strategy (CSBFGO). Clustering grouping and the Cauchy mutation strategy are introduced in the algorithm to improve the convergence and diversity of the algorithm. The CEC2017 benchmark test set was used to test and analyze its optimization accuracy and convergence ability to verify the effectiveness of the CSBFGO algorithm. In addition, to verify the performance of the CSBFGO algorithm in solving practical problems, integrating the CSBFGO algorithm is applied to optimize pump scheduling in water distribution networks. The experimental results show that the CSBFGO algorithm has good performance in solving practical problems.

Keywords: Bamboo forest growth optimization, Clustered grouping, Cauchy mutation, Pump scheduling, Water distribution networks

1 Introduction

Optimization problems widely exist in social fields such as production scheduling, mathematical science, and engineering optimization [1]. Essentially, they are applied to search the optimal solution of a given problem under certain constraints to achieve the optimal performance of a certain indicator [2]. Usually, optimization algorithms can be used to efficiently solve optimization problems. Firstly, existing problems are abstracted into concrete models, and then specific methods are designed to obtain the optimal solution of the model, providing the best solution for decision-makers [3]. With the continuous development of technology and the deepening of research, the importance of optimization algorithms has become increasingly prominent, but at the same time, challenges are also coming one after another [4]. Optimization problems are becoming more and more complex, and data also exhibit more complex high-dimensional nonlinear characteristics. For this reason, researchers are constantly committed to the study of optimization algorithms to increase their applicability to different optimization problems [5].

The current optimization algorithms are mainly divided into two categories: accurate algorithms and approximation algorithm [6]. The precise algorithm uses mathematical methods to solve optimization problems and obtain a unique global optimal solution. This type of algorithm usually has a fixed structure and parameters, suitable for solving small-scale optimization problems, and can ensure the accuracy of the optimal solution [7]. However, its solution results strongly depend on the initial values, and in large-scale problems, it will seriously consume computational power and easily lead to "combinatorial explosion" problems [8]. The approximation algorithm does not have the above problems when solving large-scale complex optimization problems. Although this type of algorithm cannot guarantee sufficient accuracy of the optimal solution, it can converge the problem to an approximate value of the optimal solution in polynomial time [9]. And the randomness of its optimization process can provide more possibilities for the algorithm to jump out of local optima in multi-modal functions [10-11].

As a kind of approximation algorithm, the meta-heuristic search (HMS) algorithm performs random searches by simulating biological activities in nature, using information from multiple heuristic functions, and

---

*Corresponding Author: Shu-Chuan Chu; Email: scchu0803@gmail.com

DOI: https://doi.org/10.70003/160792642026052703002

dynamically adjusting the search direction according to the weight of each function, in order to find the optimal excellent solution [12]. HMS is mainly divided into five categories: First, algorithms based on evolution theory, which form iterative evolution algorithms by simulating the mechanism of natural selection and genetic evolution, such as genetic algorithm (GA) [13], differential evolution (DE) [14] and quasi-affine transformation evolution (QUATRE) [15]. Second, it is based on biologically inspired algorithms, which constructs global optimization algorithm process by simulating the special behavior of biological population, such as particle swarm optimization (PSO) [16], tumbleweed optimization based chaotic (CTOA) [17], moth flame optimization (MFO) [18] and Pelican optimization algorithm (POA) [19]. Third, algorithms based on human behavior, which are heuristic algorithms constructed by simulating human teaching, social, and emotional behavior, such as teaching learning based optimization (TLBO) [20], brain storm optimization algorithms (BSO) [21] and student psychology based optimization algorithm (SPBO) [22]. Fourth, based on physical and chemical algorithms, algorithm models are constructed by simulating physical rules and chemical reactions in the universe, such as simulated annealing (SA) [23], water cycle algorithm (WCA) [24] and gradient-based optimizer (GBO) [25]. Fifth, it is a music-based algorithm. The pioneer in this field is the Harmony Search (HS) [26].

The workflow of HMS algorithm is mainly divided into three parts: guidance mechanism, search operator and update mechanism. Simply put, the purpose of the guidance mechanism is to select candidate solutions from the population that guide individuals to perform the search process. The search operator is modeled by simulating natural laws to perform exploration and development tasks, while the update mechanism is to select survivors in the population. Or, that is, selecting the next generation population [27]. In recent years, researchers have proposed many new improvement strategies for the guidance mechanism and update mechanism. Kahraman et al. proposed a natural survivor method (NSM) as a population update method, based on the NSM score as the population update criterion [28]. Kahraman et al. proposed a new selection method based on fitness-distance balance (FDB) to solve the problem of insufficient algorithm diversity caused by premature convergence in MHS algorithm [29]. Ozkaya et al. proposed a fitness distance constraint (FDC) selection method and a dynamic guidance mechanism to improve the overall performance of the algorithm [30]. Kahraman et al. proposed a dynamic adaptive distance balancing (d FDB) selection method and applied it to the directional overcurrent relays (DOCRs) coordination problem [31]. Kahraman et al. proposed an AFDB-SFS algorithm to solve the optimal power flow (OPF) problem [32].

The optimization algorithm for bamboo forest growth is inspired by the unique growth laws of bamboo forests, and uses the random growth model of bamboo forests as a mathematical basis to model the growth process of bamboo forests [33]. BFGO has its unique advantages compared with other meta-heuristic optimization algorithms. Firstly, the BFGO algorithm groups populations and replaces information exchange between individuals with information exchange between groups, fully considering the impact of all individuals in the population on the solution; Secondly, in the algorithm, dynamic grouping, elite pool, and dimension flipping strategies are introduced based on the growth characteristics of bamboo forests to simulate the concept of bamboo whips and the phenomenon of bamboo whip interlocking in bamboo forests. Not only does it increase the possibility of communication between populations, but it can also guide the algorithm in the direction of optimization and enhance its ability to jump out of local optima.

However, according to the “no free lunch” theorem, no optimization algorithm can perfectly solve any problem [34]. Research in recent years has tended to use different optimization strategies to reduce the disadvantages of algorithms. e.g., Hu et al. used multi-surrogate assisted optimization to reduce the evaluation cost and balance the global search and local search capabilities of the algorithm [35]. Lei et al. replaced the updating method of ant pheromones from constant to clustering method to ensure the integrity of clustering as much as possible [36]. Wang et al. added a parallel communication strategy to slime mould algorithm (SMA) to improve the efficiency of the algorithm [37]. Du et al. added a chaotic mapping strategy to PSO and applied the strategy to local populations to improve the diversity of the algorithm [38]. Zhang et al. optimized the convergent evolution behavior in phasmatodea population evolution algorithms and introduced a restart strategy to balance the diversity and convergence of the algorithm [39]. Later, Wu et al. introduced chaotic mapping into phasmatodea population, and the performance of the algorithm was significantly improved [40]. Chen et al. proposed a uniform non-inertial velocity update method to update the population in PSO, and then introduced an adaptive elite selection strategy to enhance the competitiveness of the algorithm [41].

BFGO can effectively handle complex high-dimensional problems. However, for multi-modal problems and composite problems, the iterative process of the algorithm will converge prematurely, making it difficult to jump out of the local optimal solution, and the final solution result will be quite different from the actual optimal solution. Therefore, improving the bamboo forest growth optimization algorithm has certain research significance. In response to the above shortcomings, this study proposes two improvement strategies. The main contributions are presented in the following aspects:

1. This study proposes a bamboo forest growth optimization algorithm based on the clustering strategy (CSBFGO) to solve the problem that the BFGO is prone to falling into local optimality. First, the algorithm uses cluster grouping to enhance the diversity of solutions in the early groups of the algorithm by expanding the differences between different groups. Secondly, the Cauchy mutation strategy is introduced to generate a survival algorithm of the fittest for mutated individuals, guiding the algorithm to jump out of the local optimum.

2. In order to verify the effectiveness of the CSBFGO algorithm, its optimization accuracy and convergence ability were tested and analyzed using the CEC2017 benchmark test set.

3. The CSBFGO method is integrated into the optimization problem of pump scheduling in water distribution pipe networks to verify the performance of the CSBFGO algorithm in solving practical problems.

Section 2 will introduce the inspiration sources and mathematical models of the classic bamboo forest growth optimization algorithm. Section 3 will describe the improvement strategy and algorithm proposed in this study. Section 4 will focus on discuss the experimental data of CSBFGO algorithm under CEC2017 testing. Section 5 will introduce the optimization problem of water pump scheduling in the water distribution network, design the coding method and fitness function of the problem. Finally, in Section 6, the experimental analysis is summarized and conclusions are drawn.

2 Preparations

2.1 Inspiration

As a herbaceous plant, bamboo has the height of a tree, which is because it takes several years to extend its roots in the soil, and then only takes one month to achieve explosive growth [42]. Therefore, during the growth process of bamboo, the long-term extension stage of the root underground can correspond to the global search stage in the optimization algorithm, while the short-term explosive growth stage of bamboo height can correspond to the local search stage in the optimization algorithm. The extension of the root is actually the process of nutrient storage and expansion of the bamboo whip, the Underground stem of bamboo [43]. In this process, the height of bamboo grows slowly or even does not change. The explosive growth of bamboo height is a key characteristic of bamboo, which enables it to better adapt to forest environments [44].

The characteristics of the bamboo forest are shown in Figure 1. Among them, bamboo whip nodes are numerous and dense, capable of randomly growing and extending in various directions. Each growth point has a bamboo shoot tip that can penetrate almost any soil to continue growing. A bamboo forest is composed of multiple bamboo whips, and bamboo belonging to one whip is grouped.

The measured growth curve of bamboo height is shown in Figure 2 [45]. From the figure, it can be seen that the growth of bamboo forests is divided into two stages: the first stage is the underground extension stage of bamboo whips, where the height of the bamboo changes slightly, while the bamboo whips are expanding their underground territories, corresponding to the global search process in the optimization algorithm. The second stage is the explosive growth stage of bamboo height, which quickly reaches a certain height and gradually fades away, corresponding to the local search process in the optimization algorithm.

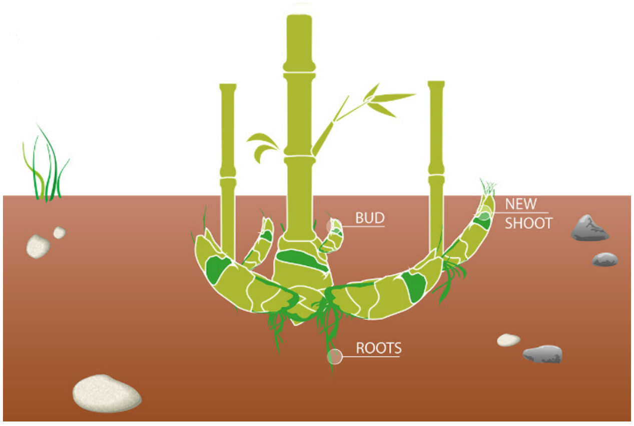


Figure 1. Description of bamboo forest characteristics

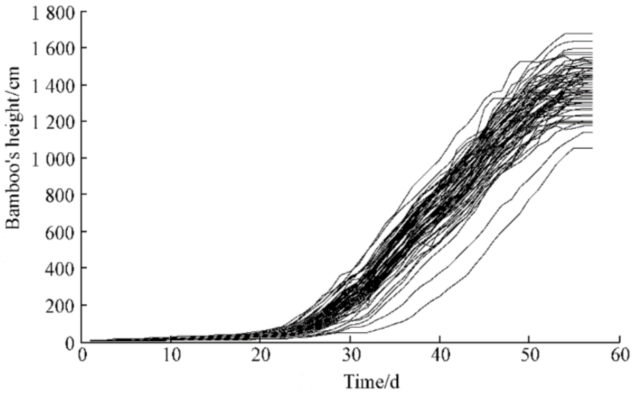


Figure 2. The growth curve of bamboo height

2.2 Mathematical Model of the BFGO Algorithm

The corresponding relationship between the growth process of bamboo forests and optimization problems is shown in Table 1. The algorithm is divided into two stages: global search and local search, corresponding to the underground extension stage of the bamboo whip and the explosive growth stage of the bamboo height, respectively.

Table 1. Correspondence between bamboo forest growth process and optimization problems

The growth process of bamboo forests	The optimization problem
The bamboo	The feasible solution
The bamboo forest	The search space
The extension trend of bamboo whips	The global search direction
The growth trend of bamboo height	The local search direction
The growth time	The iteration times

2.2.1 The Global Search Stage

Bamboo whips expand their occupation area by randomly extending underground to absorb more nutrients and store energy in the soil. The direction of the all-round extension of the bamboo whip depends on three factors: the optimal individual of the population, the optimal individual of the bamboo whip, and the central individual of the bamboo whip. Therefore, the updating of individuals at this stage also depends on the direction of group cognitive items, bamboo whip memory items, and bamboo forest central items, respectively. The cosine values of the current individual in these three directions are shown in Eq.1, Eq.2, and Eq.3. The updates of individuals at this stage are shown in Eq.4.

$$\cos\beta = \frac{\overrightarrow{X^t}\cdot\overrightarrow{X_G^t}}{\left|\overrightarrow{X^t}\right|\left|\overrightarrow{X_G^t}\right|} \tag{1}$$

$$\cos\gamma = \frac{\overrightarrow{X^t}\cdot\overrightarrow{X_{P(G)}^t}}{\left|\overrightarrow{X^t}\right|\left|\overrightarrow{X_G^t}\right|} \tag{2}$$

$$cos\xi = \frac{\overrightarrow{X^t}\cdot\overrightarrow{X_{C(G)}^t}}{\left|\overrightarrow{X^t}\right|\left|\overrightarrow{X_G^t}\right|} \tag{3}$$

$$X^{t+1} = \begin{cases} X_G^t + q\times\left(r_1\times X_G^t - X^t\right)\times\cos\beta, r_2 < 0.4, \\ X_{P(g)}^t + q\times\left(r_1\times X_{P(g)}^t - X^t\right)\times\cos\gamma, 0.4\le r_2 < 0.7, \\ X_{C(g)}^t + q\times\left(r_1\times X_{C(g)}^t - X^t\right)\times\cos\xi, else \end{cases} \tag{4}$$

Among them, β, γ and ξ represent the direction of the current individual in the population optimal individual, bamboo whip optimal individual, and bamboo whip center individual, respectively. t represents the current number of iterations, and G represents the number of bamboo whips. X^t represents the current individual, X_G^t represents the optimal individual in the population, $X_{p(g)}^t$ represents the optimal individual on the first bamboo whip, and $X_{C(g)}^t$ represents the central individual on the second bamboo whip. r_1 is a random number on the interval [0,2], and r_2 is a random number on the interval [0,1]. q is a decreasing coefficient, as shown in Eq.5.

$$q = 2 - \frac{t}{T} \tag{5}$$

Among them, T is the maximum number of iterations. As the number of iterations increases, the value of q decreases from 2 to 1, which will continuously affect the rate of convergence of the algorithm.

2.2.2 The Local Search Stage

During the explosive growth stage of bamboo height, only a small portion of unearthed bamboo shoots can grow into bamboo. These bamboo shoots that have the opportunity to grow will receive sufficient energy and grow rapidly in a short period of time. And bamboo shoots that have no hope of growing up can only perish in the survival of the fittest in nature. In combination with the Stochastic process model of bamboo height growth, the update of individual position at this stage is shown in Eq.6.

$$X_{temp}^t = \begin{cases} X^t + XD\times\Delta H & if\ rand\le 0.5, \\ X^t - XD\times\Delta H & else \end{cases} \tag{6}$$

Among them, X_{temp}^t represents the temporary matrix. After updating the individuals, the survival of the fittest will be carried out, and the individuals in X_{temp}^t who are better than the original population will be selected to join the iteration process, while the rest will be eliminated. XD represents the relationship between the distance from the current individual to the central individual of the bamboo whip and the distance from the optimal individual of the bamboo whip to the central individual, as shown in Eq.7.

$$XD = 1 - \left|\frac{X^t - X_{C(g)+1}^t}{X_G^t - X_{C(g)+1}^t}\right| \tag{7}$$

where ΔH represents the difference between the cumulative of bamboo growth between the two growth iterations, as shown in Eq.8.

$$\Delta H = \frac{Q_{(t)} - Q_{(t-1)}}{X_G^t - X^t} \tag{8}$$

where $Q_{(t)}$ represents the cumulative of the generation of bamboo height growth, as shown in Eq.9.

$$Q_{(t)} = X_G^T\times e^{-d_1}\times e^{\frac{b}{\lambda\times t^{\lambda}}} \tag{9}$$

Among them, the range of d_1 is [-1,1]. Both b and λ are parameters of the growth curve, while are equivalent to a random environmental factor for bamboo growth. In Eq.7, the smaller the value of XD, the smaller the difference between the current individual and the optimal individual, making it suitable for searching within a small range around the current individual. The larger the value of XD, the greater the difference between the current individual and the optimal individual. It is difficult to find a better solution within a small range near the current individual, and it is suitable to expand the range to find solutions that are farther away.

2.3 The Implementation of the BFGO Algorithm

The optimization process of BFGO algorithm includes three stages: population initialization, global search, and local search.

2.3.1 The Population Initialization Stage

In the initialization stage, the population is divided into groups based on the characteristics of bamboo whips in the bamboo forest. After initialization, the generated individuals are evenly grouped based on their fitness values to ensure that the quality of the individuals in each group is equal.

2.3.2 The Global Search Stage

In the global search stage, individuals are updated according to Eq.1-Eq.3. When the optimal individual for each group in the search space is not updated, the default population is in a lagging state. By initiating dynamic grouping, individuals are disrupted and re grouped to awaken information exchange between populations, ensuring the randomness of each group and increasing the diversity of individuals in the population.

2.3.3 The Local Search Stage

In the local search stage, update the individuals according to Eq.6-Eq.9. At the same time, simulating the growth phenomenon of nutrient transfer through the intricate intersection of bamboo whips. The elitist library and dimension flipping ideas were introduced into the algorithm to avoid getting stuck in local optima.

The essence of the elite pool idea is to set an elite pool variable in the algorithm to store individuals who have been eliminated but still have good fitness values. The idea of dimension flipping is to extract a random number of individuals from the elite pool that is less than the number of elites when the algorithm has not undergone global optimal updates for 50 consecutive generations. Flip and exchange elements in the random dimension to achieve individual updates.

3 Proposed Improvement Algorithm

3.1 Improvement Strategy

3.1.1 Cauchy Mutation

Meta-heuristic optimization algorithms have achieved good results by applying Gaussian variational strategies to improve the convergence accuracy of the algorithms. Both Gaussian and Cauchy distributions are a common distribution in probability theory and mathematical statistics [46-47]. The one-dimensional Gaussian distribution density function is shown in Eq.10:

$$f(x)=\frac{1}{\sigma\sqrt{2\Pi}}\times e^{\frac{(x-\mu)^2}{2\sigma^2}},-\infty<x<+\infty \qquad \textbf{(10)}$$

where $\sigma = 1$ and $\mu = 0$ for the standard Gaussian distribution. The probability density function of the one-dimensional standard Cauchy distribution is shown in Eq.11:

$$f(x)=\frac{1}{\Pi}\times\frac{t}{t+x^2},-\infty<x<+\infty \qquad \textbf{(11)}$$

The probability density function curves of the standard Gaussian distribution and Cauchy distribution are shown in Figure 3. From the figure, it can be seen that the Cauchy distribution is flatter than the Gaussian distribution, decreases to zero more slowly and has smaller peak at the center than the Gaussian distribution. From the random variable generating function theorem, the random variable generating function of the Cauchy distribution is shown in Eq.12:

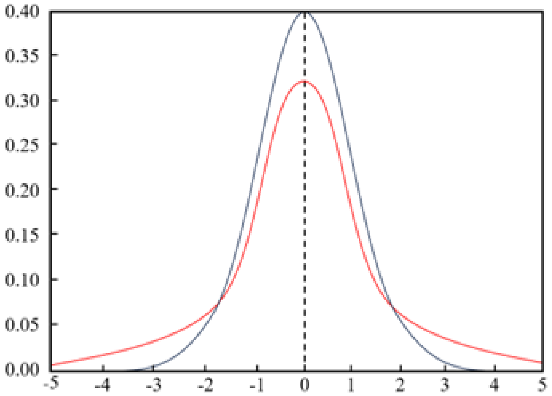


Figure 3. Probability density function curves of standard Gaussian distribution and standard Cauchy distribution

$$\eta = \tan[(\xi - 0.5)\Pi] \qquad \textbf{(12)}$$

where ξ is a random variable subject to uniform distribution on [0,1]. To improve the global search ability of the algorithm, the diversity of the population is increased through the use of Cauchy variation. The peak of the Cauchy distribution function at the origin is small but the distributions at the two ends are long. Using the Cauchy variation can enable the variation to reduce the constraints of the local optima of the bamboo individuals and make it easier to jump out of the local optimum. In this study, the algorithm is enabled to focus on the search for the global optimum by taking advantage of the small peak of the Cauchy distribution. The Cauchy variation of an individual is added in the local search phase, and the current individual is mutated using the update formula shown in Eq.13.

$$X_{newbest} = X_{temp} \times Cauchy(0,1) \qquad \textbf{(13)}$$

where $Cauchy(0,1)$ is the standard Cauchy distribution. In the Cauchy variation strategy, the Cauchy variation operator is used to derive a new solution by perturbing the variation operation on the bamboo individuals to improve the defect of the algorithm falling into the local area.

3.1.2 Clustering Grouping

Analyzing the classical bamboo forest growth optimization algorithm, it is not difficult to find that the dynamic grouping strategy of the algorithm is flexible in the conversion between uniform grouping and disrupted grouping, but in the late stage of the algorithm's optimization search, the impact of these two groupings on the update of individuals does not differ much, because a large number of times of disrupted grouping also has a definite rule. Disorganized grouping transforms into aggregated grouping, where good, moderate, and crossover individuals are grouped separately. In this manner, the four best individuals and the four center of gravity individuals are obtained from the four groups with large gaps, thereby guiding the algorithm to learn not only from the good individuals but also from the poor individuals appropriately. Aggregate grouping enables individuals to learn and update more comprehensively, increases the diversity of solutions, and thus increases the possibility of the algorithm jumping out of the local optimum.

3.2 Description and Implementation of the Improved Algorithm

The pseudo code of the CSBFGO algorithm is shown in Algorithm 1.

4 Performance Testing and Analysis of Improved Algorithm (CSBFGO)

To test the performance of the proposed CSBFGO algorithm, seven classic metaheuristic optimization algorithms were selected for comparative analysis, namely the particle swarm optimization algorithm (PSO), whale optimization algorithm (WOA), sine cosine algorithm (SCA), seagull optimization algorithm (SOA), sparrow search algorithm (SSA), advanced phasmatodea population evolution algorithms (APPE), and BFGO. Perform 50 runs on unimodal functions, multi-modal functions, hybrid functions, and combination functions, and compare the average of the 50 optimal values obtained. Analyze the optimization accuracy and convergence ability of the algorithm to validate the effectiveness of the CSBFGO algorithm in solving optimization problems.

4.1 Experimental Design

The CEC2017 benchmark function was chosen as the testing function, and the experimental suite includes two unimodal functions, seven simple multi-modal functions, ten hybrid functions, and ten composition functions [49]. Detailed information about these functions is provided in Table 2, where $f_1 - f_2$ represent unimodal functions, $f_3 - f_9$ denote simple multi-modal functions, $f_{10} - f_{19}$ denote hybrid functions, and $f_{20} - f_{29}$ signify composition functions. The parameter settings for the comparison algorithms in the experiment are outlined in Table 3. To ensure fair comparison, the max iterations for each algorithm were set to $maxIter = 10000*Dimensions$, and all experimental results were averaged over 30 independent runs for each test function. All experimental results were averaged over 30 independent runs for each test function.

Algorithm 1. CSBFGO

```
Input: runs: Number of runs; T: Number of iterations; f : Fitness founction; N: Population size;
Output: $X^t_G$: Optimal individual position; $f(X^t_G)$: Optimal individual fitness value; Converage: Storage variable of the optimal value in the iteration;
Initialize the number of bamboo whips $G$ individual positions, $count = 0$, calculate individual fitness values and group them evenly;
while $t \le T$ do
    for $g$ to $G$ do
        for $i$ to $\frac{N}{G}$ do
            Perform global search to update individual positions in the population according to Eq.1-Eq.5;
            if $X^t_{p(G)}$ is not updated then
                $count = count + 1$;
            end if
            if $count = G$ then
                Start dynamic grouping, reuse aggregate grouping and update the group's optimal individual and central individual;
            end if
    According to Eq.6-Eq.9, local search is performed to save temporary individuals, and the population is updated by comparing and selecting the best ones.
        The eliminated individuals undergo Cauchy mutation operation to generate mutant individuals to participate in the population optimization process;
            if $t > \frac{2T}{3} \| Converage(t) == Converage(t-50)$ then
            Start population communication and randomly select individuals from the elite pool to implement the dimension flipping strategy.
            end if
        end for
    end for
end while
```

4.2 Optimization Accuracy Analysis

The average and standard deviation of the optimal values obtained by the CSBFGO algorithm and seven comparison algorithms, each run 30 times in the 100-dimensional space, are displayed in Table 4 and Table 5. The choice of 100 dimensions is justified because most algorithms exhibit excellent convergence capabilities in lower dimensional spaces, diminishing the significance of comparing convergence scenarios in such spaces. Convergence curves in high-dimensional spaces more accurately reflect the convergence abilities of algorithms, and the final line indicates the number of times the CSBFGO algorithm outperformed the others. Unimodal function f_2 is excluded due to its instability at high latitudes.

From Table 4, it can be observed that in the 100 dimensional space, the CSBFGO algorithm reaches its optimal performance in 8 functions: $f_4, f_8, f_{10} - f_{11}, f_{19}, f_{22}, f_{27} - f_{28}$, the BFGO algorithm reaches its optimal performance in 5 functions, the SOA algorithm achieves optimal performance in 5 function. The PSO algorithm achieves optimal performance in 6 functions, while the APPE and SSA algorithm does so in 2 function. Conversely, the CSBFGO algorithm significantly outperformed other algorithms in obtaining the optimal number of solutions among 28 functions. In addition, the CSBFGO algorithm outperformed the PSO algorithm 19 times, the WOA algorithm 26 times, the SCA algorithm 26 times, the SOA algorithm 23 times, the SSA algorithm 25 times, the APPE algorithm 23 times, and the BFGO algorithm 19 times across 29 functions, surpassing half of the total instances. This highlights its significant superiority over the other seven algorithms, showcasing excellent optimization accuracy and strong competitiveness. From Table 4, it can be noted that in the 100-dimensional space, the CBFGO algorithm achieves 1 optimal among 3 multimodal functions and 2 optimal among 10 hybrid functions. Obtained 3 optima out of 10 composition functions, CSBFGO achieved the best value in 8 test functions, but at least 19 functions won when compared with each function individually, which proves that CSBFGO is more robust, especially in high-dimensional problems. At the same time, it also shows that the CSBFGO algorithm has good global and local search capabilities in 100-dimensional space. Although it ranks second in performance for hybrid and combination functions, overall, the CSBFGO algorithm still has the most optimal functions in 28 functions, which shows that the algorithm has superior performance. In addition, its performance is better when compared separately, which shows that it is more competitive than other algorithms.

Table 2. Details of the CEC2017 benchmark suite

Function type	Number	Function name	Minimum value
Unimodal function	1	Shifted and Rotated Bent Cigar Function	100
	2	Shifted and Rotated Zakharov Function	200
Multi-modal function	3	Shifted and Rotated Rosenbrock’s Function	300
	4	Shifted and Rotated Rastrigin’s Function	400
	5	Shifted and Rotated Expanded Scaffer’s F6 Function	500
	6	Shifted and Rotated Lunacek Bi Rastrigun Function	600
	7	Shifted and Rotated Non-Continuous Rastrigin’s Function	700
	8	Shifted and Rotated Levy Function	800
	9	Shifted and Rotated Schwefel’s Function	900
Hybrid function	10	Hybrid Function 1 (N=3)	1000
	11	Hybrid Function 1 (N=3)	1100
	12	Hybrid Function 1 (N=3)	1200
	13	Hybrid Function 1 (N=4)	1300
	14	Hybrid Function 1 (N=4)	1400
	15	Hybrid Function 1 (N=4)	1500
	16	Hybrid Function 1 (N=5)	1600
	17	Hybrid Function 1 (N=5)	1700
	18	Hybrid Function 1 (N=5)	1800
	19	Hybrid Function 1 (N=6)	1900
Combination function	20	Composition Function 1 (N=3)	2000
	21	Composition Function 2 (N=3)	2100
	22	Composition Function 3 (N=4)	2200
	23	Composition Function 4 (N=4)	2300
	24	Composition Function 5 (N=5)	2400
	25	Composition Function 6 (N=5)	2500
	26	Composition Function 7 (N=6)	2600
	27	Composition Function 8 (N=6)	2700
	28	Composition Function 9 (N=3)	2800
	29	Composition Function 10 (N=3)	2900

Table 3. Parameter settings for related algorithms

Algorithms	Year	Parameter	
PSO	1995	$l_1 = 2, l_2 = 2, \mu = 0.7$	
WOA	2016	$b_1 = 1, pc \in [0,1], l \in [-1,1]$	
SCA	2016	$a_2 = 2$	Number of populations (N): 100;
SOA	2019	$\mu_1 = 1, \mu_2 = 1, f_c = 2$	Number of iterations (T): 1000;
SSA	2020	$ST = 0.6, PR = 0.5, R_2 \in [0,1]$	Solution space dimension (D): 100
APPE	2022	$\alpha = 0.99$	
BFGO	2023	$G = 5, b \in [0,2], \lambda = 2$	

To verify the statistically significant difference between the CSBFGO and the compared algorithms, the Wilcoxon rank sum test and the Friedman test were employed for testing, and the results are shown in Table 6. According to the results of the Wilcoxon signed-rank test, when the p value exceeds 0.05, it indicates that there is no statistically significant difference between the algorithms for pairwise comparison. The data in Table 6 reveals that most p-values are lower than 0.05, which means that the optimization performance of the CSBFGO is significantly different from other algorithms. Delving into the results of the Friedman test, the p-values derived from this statistical test are displayed, and they are placed in the rightmost column of Table 6, providing a clear and direct reference. This outcome demonstrates that in multiple independent experiments, the CSBFGO algorithm has exhibited superior performance, reflecting the overall performance advantage of the algorithm. To demonstrate the convergence of the algorithm on different problems, four distinct types of problems were selected from the CEC2017 benchmark suite to assess the convergence ability of the algorithm. The solution space dimensions were set to 30, 50, and 100. Specifically, F1 (unimodal), F7 (multi-modal), F19 (hybrid), and F26 (composition) test functions were chosen from the suite, and the test results are depicted in Figure 4. It can be observed from the convergence curve in the figure that the convergence and diversity of the CSBFGO have significantly improved compared to the BFGO, and optimal results can be achieved in most convergence graphs.

Table 4. Comparison of mean results of various algorithms in CEC2017 benchmark function

Founction	PSO	WOA	SCA	SOA	SSA	APPE	BFGO	CSBFGO
f1	2.5817E+09	3.1941E+09	1.6980E+11	1.1947E+11	1.5526E+04	2.5273E+09	3.0531E+09	1.5781E+09
f3	4.2124E+05	8.4467E+05	3.5068E+05	2.7677E+05	6.1736E+05	1.3423E+05	2.0506E+05	2.2085E+05
f4	1.4746E+03	1.9262E+03	3.0871E+04	1.2924E+04	3.8907E+04	1.5270E+03	1.4461E+03	1.2393E+03
f5	9.3925E+02	1.5528E+03	1.9193E+03	1.5422E+03	1.9910E+03	1.4570E+03	1.3936E+03	1.3809E+03
f6	6.8002E+02	6.8599E+02	6.9326E+02	6.7565E+02	7.0316E+02	7.9238E+02	6.7105E+02	6.7163E+02
f7	1.6920E+03	3.4084E+03	3.5347E+03	3.1499E+03	3.6791E+03	2.8892E+03	2.9275E+03	2.8622E+03
f8	1.8405E+03	1.9712E+03	2.2725E+03	1.8255E+03	2.3784E+03	1.8391E+03	1.8585E+03	1.8159E+03
f9	1.9379E+04	5.1984E+04	7.3382E+04	4.8019E+04	7.7241E+04	2.3528E+04	2.8507E+04	3.3782E+04
f10	2.4757E+04	2.3994E+04	3.1726E+04	2.3672E+04	3.0374E+04	3.0083E+04	1.9476E+04	1.8376E+04
f11	1.4104E+04	5.0179E+04	1.0507E+05	5.1912E+04	1.4855E+05	1.7766E+04	1.2723E+04	1.1771E+04
f12	1.2536E+08	1.5675E+09	6.4905E+10	3.3159E+10	1.6816E+09	1.6317E+08	8.7437E+08	8.7007E+08
f13	1.0819E+06	2.9334E+06	1.1033E+10	4.0519E+09	1.6702E+06	1.23450E+06	8.8722E+05	2.0563E+06
f14	3.4135E+06	6.1978E+06	2.7506E+07	7.8987E+06	2.5368E+07	3.6216E+06	9.0640E+05	1.0379E+06
f15	3.6178E+05	1.9968E+06	3.3179E+09	1.3791E+09	6.6175E+05	1.8255E+05	8.8232E+04	1.7826E+05
f16	5.8376E+03	1.1528E+04	1.2896E+04	7.7198E+03	1.3712E+04	9.1932E+03	7.3679E+03	7.2988E+03
f17	4.9080E+03	8.1228E+03	1.4805E+04	7.0847E+03	1.6600E+04	4.2351E+03	6.3142E+03	6.5078E+03
f18	8.5748E+06	4.6938E+06	4.7687E+07	5.6814E+06	5.1497E+06	4.4636E+06	1.4555E+06	1.5399E+06
f19	6.3940E+06	2.8918E+07	2.8186E+09	9.5264E+08	5.5787E+07	6.6310E+06	6.1275E+06	6.0559E+06
f20	5.9335E+03	6.3946E+03	7.6724E+03	5.8030E+03	7.1694E+03	6.2416E+03	5.9701E+03	5.8202E+03
f21	3.9325E+03	4.1248E+03	3.9691E+03	3.4397E+03	4.1892E+03	4.1241E+03	3.9112E+03	3.8138E+03
f22	2.9398E+04	2.6706E+04	3.4188E+04	2.6322E+04	3.2997E+04	3.3261E+04	2.3026E+04	2.1763E+04
f23	5.1953E+03	4.9058E+03	4.8832E+03	3.8255E+03	5.8250E+03	5.6231E+03	5.1663E+03	5.0221E+03
f24	6.7963E+03	6.0123E+03	6.7321E+03	4.4050E+03	8.2198E+03	7.0491E+03	6.7718E+03	6.7421E+03
f25	3.9050E+03	4.2469E+03	1.7559E+04	1.0429E+04	1.8768E+03	4.2157E+03	3.8488E+03	3.6807E+03
f26	2.6057E+04	3.1438E+04	3.4881E+04	1.7418E+04	3.7344E+04	3.0182E+04	2.5918E+04	2.4038E+04
f27	3.7219E+03	4.9066E+03	6.9827E+03	4.1115E+03	8.7086E+03	3.8123E+03	4.7696E+03	3.3473E+03
f28	3.9324E+03	5.1129E+03	2.1238E+04	2.1353E+04	2.3252E+04	4.1674E+03	4.1923E+03	3.8193E+03
f29	9.7952E+03	1.5833E+04	1.7507E+04	1.1645E+04	2.9620E+04	1.2123E+04	1.0154E+04	1.0606E+04
Win/All	19/28	26/28	26/28	23/28	25/28	23/28	19/28	

Table 5. Comparison of standard deviation results of various algorithms in CEC2017 benchmark function

Founction	PSO	WOA	SCA	SOA	SSA	APPE	BFGO	CSBFGO
f1	7.8952E+07	1.1061E+09	7.8299E+09	1.1773E+10	4.7776E+09	1.2314E+10	1.3391E+09	3.1296E+09
f3	6.3563E+04	2.2432E+05	3.4162E+04	1.8633E+04	1.0590E+05	2.4243E+05	2.3283E+04	1.7997E+04
f4	1.5753E+02	2.5041E+02	4.3646E+03	3.1908E+03	2.9466E+03	2.3241E+04	1.7904E+02	2.5615E+02
f5	1.5066E+02	8.7952E+01	5.5658E+01	8.8234E+01	2.3980E+01	6.2351E+01	6.5646E+01	6.2530E+01
f6	3.9433E+00	9.6708E+00	5.1604E+00	5.9503E+00	1.9480E+00	1.0235E+00	4.4739E+00	6.0347E+00
f7	1.2668E+02	1.5550E+02	1.3865E+02	8.2056E+01	5.9259E+01	2.3211E+02	1.8975E+02	2.4885E+02
f8	7.4665E+01	1.0557E+02	6.1323E+01	7.0868E+01	3.6395E+01	9.1241E+01	8.9544E+01	9.0562E+01
f9	1.2391E+04	8.3455E+03	6.5735E+03	6.3679E+03	2.5982E+03	6.5231E+03	3.8551E+03	5.6729E+03
f10	4.6424E+03	1.7403E+03	5.2542E+02	1.5467E+03	5.2323E+02	2.1419E+03	1.9434E+03	2.2082E+03
f11	5.7594E+03	2.1996E+04	1.6168E+04	7.0529E+03	7.8675E+03	2.5235E+03	3.7951E+03	4.9351E+03
f12	1.3232E+08	5.0764E+08	7.2038E+09	6.6858E+09	6.7693E+09	2.1543E+09	4.2583E+08	5.0015E+08
f13	4.2879E+06	2.1879E+06	1.7831E+09	1.6529E+09	1.8097E+09	5.2123E+08	2.4819E+06	5.8884E+06
f14	1.7657E+06	2.4234E+06	1.4814E+07	3.6298E+06	5.1254E+06	2.1231E+06	3.4784E+05	4.8645E+05
f15	1.7973E+06	4.3342E+06	7.8566E+08	9.7669E+08	9.0015E+08	4.8451E+06	3.7586E+04	3.3668E+05
f16	7.9658E+02	1.2581E+03	7.8545E+02	1.0757E+03	5.4329E+02	1.1324E+03	1.0716E+03	1.1134E+03
f17	4.2507E+02	9.7684E+02	2.8171E+03	3.3879E+03	2.6182E+04	2.8213E+03	6.0726E+02	9.9510E+02
f18	3.0507E+06	1.8488E+06	2.2593E+07	2.8891E+06	1.3327E+07	3.4542E+06	5.7624E+05	7.7325E+05
f19	5.5045E+03	1.6827E+07	7.2630E+08	4.8723E+08	6.8439E+08	6.1215E+04	5.4171E+06	6.1875E+06
f20	1.3426E+03	6.3145E+02	3.1884E+02	7.2291E+02	3.4569E+02	6.2311E+02	5.5972E+02	5.5475E+02
f21	7.8999E+01	1.8911E+02	7.3724E+01	1.0419E+02	6.0448E+01	1.5370E+02	1.9907E+02	2.5460E+02
f22	4.3308E+03	1.4993E+03	3.5186E+02	1.7887E+03	1.0186E+03	2.1588E+03	2.2272E+03	1.9929E+03
f23	6.2842E+01	2.6042E+02	1.0894E+02	8.7824E+01	2.8100E+02	4.4121E+02	3.5626E+02	3.7241E+02
f24	9.6243E+01	3.7034E+02	2.4758E+02	1.0753E+02	5.6435E+02	6.2342E+02	7.1626E+02	6.8615E+02
f25	1.2656E+02	1.5621E+02	2.0914E+03	1.3589E+03	1.1324E+03	4.3212E+03	8.8360E+01	1.3099E+02
f26	6.6125E+02	3.5663E+03	1.7344E+03	1.0277E+03	1.4869E+03	4.2319E+03	6.9786E+03	8.8615E+03
f27	5.5971E+01	6.0396E+02	2.7174E+02	1.9224E+02	5.6698E+02	6.6542E+02	6.4009E+02	3.6848E+02
f28	1.4038E+02	3.6070E+02	1.6322E+03	8.1319E+02	1.0104E+03	7.2311E+02	2.7165E+02	2.9476E+02
f29	6.2590E+02	1.8035E+03	3.2678E+03	1.3584E+03	4.6451E+03	4.3249E+04	1.3102E+03	1.4686E+03

Table 6. Comparison results of each algorithm with Wilcoxon sign rank test and Friedman test

Fun	PSO	WOA	SCA	SOA	SSA	APPE	BFGO	**p (Friedman)**
f1	1.0487E-02	1.0479E-03	2.9137E-11	2.9137E-11	2.9137E-11	1.9217E-02	2.5101E-02	3.3521E-40
f3	5.9082E-03	2.9082E-11	2.9082E-11	1.5578E-10	2.9082E-11	4.1219E-03	3.1463E-02	3.4537E-38
f4	1.9319E-04	1.1737E-09	3.0199E-11	3.0199E-11	3.0199E-11	1.2739E-04	1.8916E-04	8.4653E-40
f5	3.1589E-03	6.5183E-09	3.0199E-11	1.1023E-08	3.0199E-11	3.0199E-02	5.1060E-01	5.9445E-38
f6	2.0199E-01	4.3106E-08	6.6955E-11	7.6171E-03	3.0199E-11	3.2139E-01	6.3088E-01	1.4517E-36
f7	2.9468E-02	3.3990E-10	5.9235E-11	6.4359E-07	2.9468E-11	2.9468E-02	2.7716E-01	3.2344E-39
f8	2.9358E-01	3.7421E-07	2.9358E-11	9.9410E-01	3.0199E-11	3.0199E-02	7.7272E-02	5.7457E-37
f9	6.5757E-05	8.1366E-10	2.7218E-11	4.2155E-09	3.0180E-11	4.1821E-05	2.5301E-04	8.54531E-38
f10	1.3101E-05	2.3753E-10	2.7218E-11	8.9878E-11	3.0180E-11	2.2341E-03	5.1854E-02	6.4564E-35
f11	1.7047E-02	2.7218E-11	2.7218E-11	2.7218E-11	3.0199E-11	1.2343E-02	1.6687E-01	1.4533E-38
f12	1.2382E-03	5.3618E-07	2.8003E-11	2.8003E-11	3.0199E-11	5.4329E-02	9.2344E-01	7.4681E-41
f13	6.5757E-04	1.2769E-05	2.7218E-11	2.7218E-11	3.0199E-11	4.7619E-03	2.8126E-02	5.1186E-40
f14	3.1660E-08	5.4843E-11	2.7218E-11	7.3811E-11	3.0199E-11	3.3251E-05	4.1191E-01	3.8646E-37
f15	1.0594E-06	8.1491E-09	2.8628E-11	2.8628E-11	3.0199E-11	4.3244-04	5.9424E-02	2.5657E-40
f16	7.1676E-02	3.0103E-11	2.7218E-11	1.9991E-01	3.0199E-11	8.2571E-02	9.7052E-01	5.5444E-38
f17	5.0392E-03	6.6408E-07	2.7218E-11	6.3009E-01	3.0199E-11	6.1287E-02	4.4641E-01	6.5432E-38
f18	3.0967E-03	6.2969E-10	2.8003E-11	1.7451E-09	3.0199E-11	3.9421E-02	7.9584E-01	5.5977E-37
f19	2.9358E-02	1.0473E-07	2.9358E-11	2.9358E-11	3.0199E-11	3.9341E-02	3.4783E-01	4.54598E-41
f20	2.3291E-01	6.3630E-04	2.7218E-11	4.4537E-01	1.6132E-10	6.8311E-01	4.2039E-01	3.8798E-21
f21	2.9082E-02	3.1096E-05	5.8118E-04	4.0604E-09	8.1975E-07	2.2121E-02	5.9424E-02	4.3452E-33
f22	9.1904E-03	1.9591E-10	2.7218E-11	1.5554E-09	3.0199E-11	2.3526E-03	4.0589E-02	7.3143E-34
f23	1.9358E-01	2.3956E-01	1.6660E-01	2.9358E-11	2.0338E-09	2.1423E-01	2.0621E-01	2.5763E-34
f24	1.8003E-01	7.0010E-05	7.9551E-01	2.8003E-11	1.4110E-09	5.2130E-01	9.2344E-01	2.3430E-34
f25	1.7162E-08	3.3286E-11	2.7218E-11	2.7218E-11	3.0199E-11	7.4521E-07	9.5299E-07	7.9769E-40
f26	9.7204E-02	3.2306E-05	9.9724E-08	3.8668E-04	5.5727E-10	1.0129E-02	2.7719E-01	6.8745E-36
f27	2.4171E-05	2.1574E-10	2.7218E-11	3.2201E-09	3.0199E-11	7.4653E-08	3.8202E-10	4.4870E-39
f28	1.2013E-02	3.0084E-11	2.7201E-11	2.7201E-11	3.0199E-11	8.5422E-04	1.1674E-05	1.1087E-38
f29	3.3286E-02	1.4650E-10	3.6798E-11	1.8129E-03	3.0199E-11	1.0227E-02	1.2597E-01	3.3421E-39

4.3 Stability Analysis

The purpose of this section is to demonstrate and verify the stability and efficiency of the BFGO and CSBFGO algorithms on the benchmark test functions. In this process, 28 functions in the CEC2017 test suite are used for simulation experiments. The max iterations for each algorithm were set to *maxIter* = 10000**Dimensions*. Each algorithm runs M times independently on each test function and calculates the fitness value. In this study, finding a feasible solution is used as the evaluation criterion, and the average fitness of the two algorithms in each test function is used as the feasible solution of each function. Based on the research in [48] as a reference, the success rate (SR), average number of fitness evaluations (AFEs), and average calculation durations (ACDs) of the two algorithms are calculated for each problem.

(1) Success rate (SR): The SR value is calculated based on the number of successful runs and the total number of runs. The calculation formula is as follows:

$$SR = \frac{m}{M} \times 100\% \tag{14}$$

where m is the number of successful runs and M is the total number of runs.

(2) Average number of fitness evaluations (AFEs): In stability analysis, when the algorithm finds a feasible solution, it stops searching and records the fitness evaluation value at that moment. These fitness evaluation values are calculated based on the search that successfully found a feasible solution in M independent runs. Define a vector FEs_m that contains the number of fitness evaluations made by searches that successfully found a feasible solution, where m represents the number of successful searches.Using this data, the AFEs are calculated as follows:

$$AFEs = \frac{1}{m}\sum_{i=1}^{m} FEs\left[i\right] \tag{15}$$

where FEs represents the number of fitness evaluations performed by the search that successfully found a feasible solution.

(3) Average calculation durations (ACDs): The value of the ACDs parameter is calculated based on the number of successful searches the algorithm has done to find a feasible solution. The ACD calculation is as follows:

$$ACDs = \frac{1}{m}\sum_{i=1}^{m} Cal\_Duration\left[i\right] \tag{16}$$

where m is the number of successful runs.

This work requires setting a feasible solution for each problem. In this study, we select the mean of each run of BFGO and CSBFGO as the standard for feasible solutions, and take the average of the two algorithm means as the feasible solution. The feasible solutions of the 28 test functions in the CEC2017 test suite are shown in Table 7. Building upon these feasible solutions, we then meticulously calculated and compared the Success Rate (SR), Average Fitness Evaluations (AFEs), and Average Convergence Distance (ACD) values for both BFGO and CSBFGO across the suite's 28 test functions. The ensuing data, presented in Table 8, provides a clear and quantitative insight into the performance of the algorithms. Notably, the CSBFGO algorithm exhibits a marked improvement in stability over the BFGO algorithm. This enhancement in stability is not only statistically significant but also reinforces the robustness of the algorithm we've proposed, underscoring its efficacy and reliability in tackling optimization problems.

Table 7. Possible solutions to 28 problems in the CEC 2017 benchmark suite

f1	f3	f4	f5	f6	f7	f8
2.3156E+09	2.1296E+05	1.3427E+03	1.3873E+03	6.7134E+02	2.8949E+03	1.8372E+03
f9	f10	f11	f12	f13	f14	f15
3.1145E+04	1.8926E+04	1.2247E+04	8.7222E+08	1.4718E+06	9.7215E+05	1.3325E+05
f16	f17	f18	f19	f20	f21	f22
7.3334E+03	6.4110E+03	1.4977E+06	6.0917E+06	5.8952E+03	3.8625E+03	2.2395E+04
f23	f24	f25	f26	f27	f28	f28
5.0942E+03	6.7570E+03	3.7648E+03	2.4978E+04	4.0585E+03	4.0058E+03	1.0380E+04

Table 8. SR, MFE, and MST performance of BFGO and CSBFGO algorithms

Fun	SR		AFEs		ACDs	
	BFGO	CSBFGO	BFGO	CSBFGO	BFGO	CSBFGO
f1	23.46%	100.0%	2.2432E+04	1.8673E+04	4.8419E+00	2.1276E+00
f3	83.26%	65.45%	6.8141E+03	8.3241E+03	1.7274E+00	1.0322E+00
f4	92.82%	77.23%	8.2135E+03	9.3242E+03	1.7913E+00	1.0576E+00
f5	95.43%	100.0%	5.6124E+04	4.5632E+04	1.5282E+01	6.0070E+00
f6	90.25%	87.02%	1.1342E+04	1.4565E+04	3.6899E+00	2.6666E+00
f7	74.42%	96.42%	1.0241E+04	8.3242E+03	2.9586E+00	1.2515E+00
f8	70.62%	90.16%	1.1231E+04	8.1341E+03	2.9179E+00	1.0763E+00
f9	100.0%	88.57%	9.8214E+03	9.4124E+03	2.8072E+00	1.3445E+00
f10	53.52%	85.54%	3.8132E+05	3.1058E+05	9.4232E+01	4.2214E+01
f11	83.91%	96.15%	9.5153E+03	9.5258E+03	2.2296E+00	1.1460E+00
f12	61.92%	82.18%	4.8351E+04	4.0523E+04	1.1489E+01	5.4212E+00
f13	100.0%	23.86%	3.2931E+04	4.7421E+04	7.3558E+00	5.8769E+00
f14	92.26%	72.35%	1.7435E+04	1.9126E+04	4.8543E+00	2.7381E+00
f15	100.0%	45.12%	1.8921E+04	2.2234E+04	4.6025E+00	2.7088E+00
f16	80.20%	92.58%	2.0844E+05	1.5625E+05	5.4190E+01	1.9981E+01
f17	87.81%	77.82%	1.9265E+04	2.3588E+04	6.3472E+00	3.9909E+00
f18	60.23%	64.88%	2.3562E+04	2.2163E+04	6.0003E+00	2.9009E+00
f19	71.21%	72.82%	2.2214E+04	2.1598E+04	1.3387E+01	8.5602E+00
f20	64.71%	66.92%	3.8354E+05	3.9236E+05	1.2928E+02	7.0566E+01
f21	89.26%	88.54%	2.8234E+05	2.6241E+05	9.9621E+01	6.2433E+01
f22	79.82%	84.23%	2.6274E+05	2.0371E+05	1.0258E+02	5.0779E+01
f23	61.92%	72.12%	3.2523E+05	2.9021E+05	1.2547E+02	7.8946E+01
f24	92.58%	98.21%	1.2313E+04	1.0222E+04	5.3571E+00	2.9712E+00
f25	82.12%	87.69%	9.8214E+03	8.2414E+03	3.6131E+00	2.3027E+00
f26	87.19%	88.62%	2.7345E+05	2.3345E+05	1.2808E+02	7.6140E+01
f27	40.72%	100.0%	1.8244E+04	6.1442E+03	8.1671E+00	2.7208E+00
f28	31.73%	100.0%	1.6452E+05	1.2341E+05	6.6257E+01	4.1199E+01
f29	92.24%	60.34%	1.1023E+03	1.6371E+03	3.9785E-01	4.3694E-01

5 Optimization and Analysis of Water Pump Scheduling in the Water Distribution Network

5.1 Problem Description

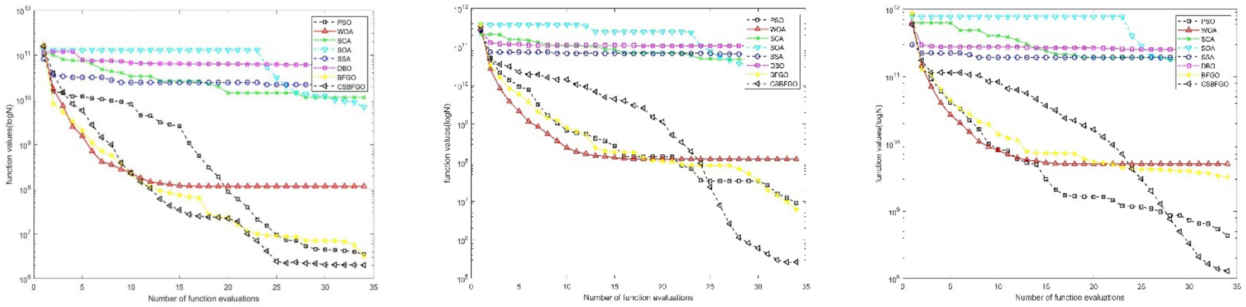
Metaheuristic algorithms are ideally suited for addressing real-world challenges due to their robustness, parallelism, and adaptability, making them widely applicable across diverse practical scenarios [50-51]. Specifically, their application in resolving distribution dilemmas within water supply networks has garnered significant interest. Currently, most water distribution departments rely solely on experience for decision-making and scheduling, resulting in significant energy consumption in the pipeline network, which undoubtedly leads to serious economic and energy losses [52]. In order to solve this problem, extensive research has focused on developing energy-saving and consumption reduction technologies, and optimizing the scheduling of water pumps is an essential part of energy-saving and consumption reduction [53]. Pump group optimization scheduling is a complex objective optimization problem with numerous variables, which can only be solved by establishing linear or nonlinear models to find the optimal energy-saving and consumption-reduction plan [54]. In order to quickly obtain the optimal solution, meta-heuristic optimization algorithms are typically employed [55].

This study tests the capability of the CSBFGO algorithm to solve real-world optimization problems and utilizes the CSBFGO algorithm to optimize water pump scheduling. By taking the speed ratio of the water pump as the independent variable and the minimum power distribution cost of the water pump as the objective function, hydraulic simulation and calculation of the water distribution network are conducted through joint programming using the EPANET toolbox and MATLAB, achieving real-time optimization of water pump scheduling to minimize electrical energy consumption.

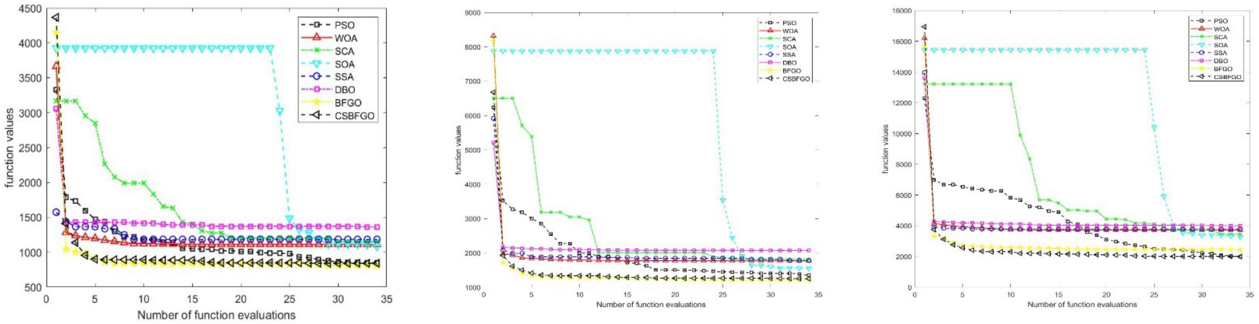
5.1.1 Objective Function

The minimum energy consumption cost of water pumps refers to a function that aims to minimize the power distribution cost of each water pump, as shown in Eq.14:

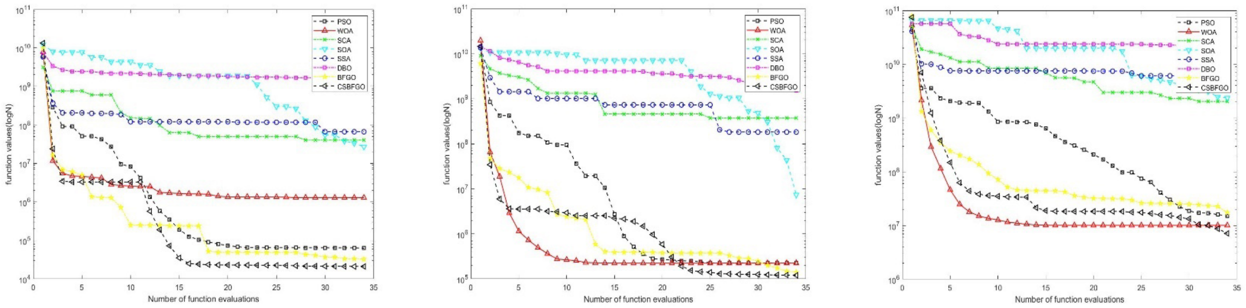
$$min W_t = \omega_t \sum_{i=1}^{N_p} \frac{KQ_{it} h_{it}}{\eta_{it}} \times X_{it} \qquad \textbf{(17)}$$



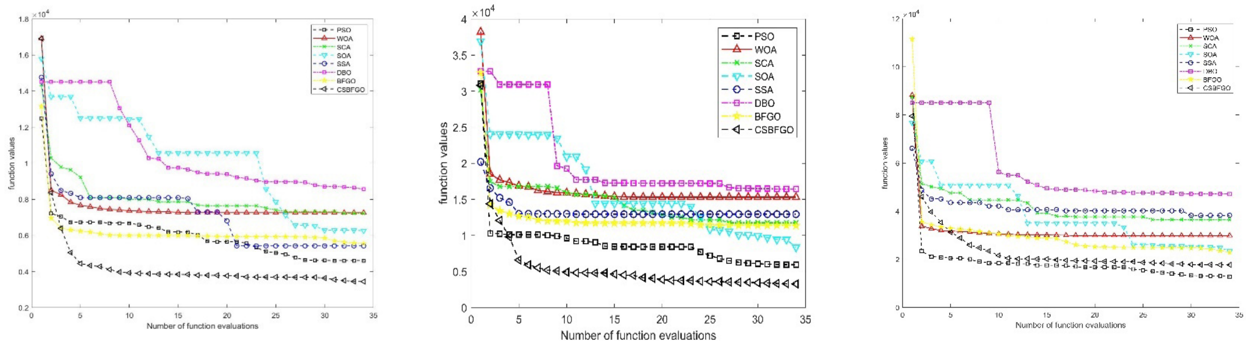
F1 (Unimodal) D = 30 F1 (Unimodal) D = 50 F1 (Unimodal) D = 100



F7 (Unimodal) D = 30 F7 (Unimodal) D = 50 F7 (Unimodal) D = 100



F19 (Unimodal) D = 30 F19 (Unimodal) D = 50 F19 (Unimodal) D = 100



F26 (Unimodal) D = 30 F26 (Unimodal) D = 50 F26 (Unimodal) D = 100

Figure 4. Convergence curves of algorithms for unimodal, multi-modal, hybrid and composition function in CEC2017 (D=30, 50, 100)

Where W_t represents the power distribution cost of each water pump at time t. ω_t represents the unit electricity cost at time t. N_t represents the number of water pumps. Q_{it} and h_{it} are the flow rate and outlet pressure of water pump i at time t, respectively. K is a constant with a value of 0.01019. η_{it} represents the efficiency of water pump i. X_{it} is the speed ratio of the water pump. Its value is 0 when the water pump is turned off and 1 when running at full load.

5.1.2 Constraints

The calculation of the objective function must not only satisfy the continuity equation and energy conservation equation, but also meet the following three constraint conditions.

The outlet flow constraint, speed ratio constraint, efficiency constraint, and monitoring point pressure constraint of the water pump should respectively meet Eq.15 to Eq.17.

$$q_{min,i} \le q_i \le q_{max,i} \tag{18}$$

$$\eta_{min,i} \le \eta_i \le \eta_{max,i} \tag{19}$$

$$H_{min,i} \le H_i \le H_{max,i} \tag{20}$$

Among them, $q_{min,i}$ and $q_{max,i}$ are the minimum and maximum flow rates of water pump i. $\eta_{min,i}$ and $\eta_{max,i}$ are the minimum and maximum efficiency of water pump i and $H_{min,i}$ and $H_{max,i}$ are the minimum and maximum pressures of water pump i.

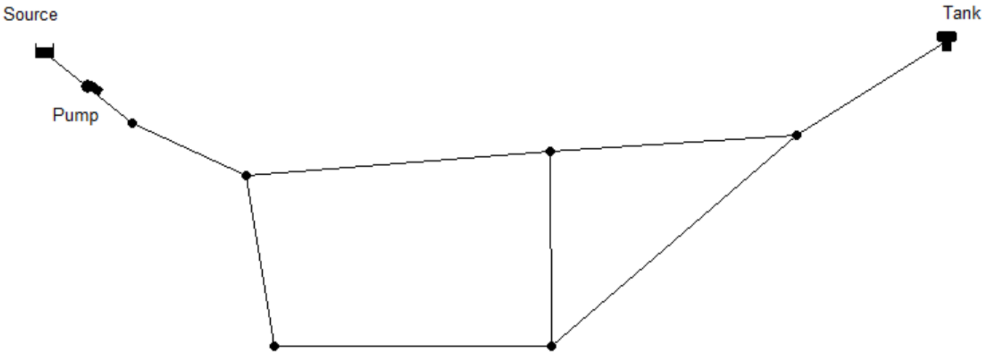


Figure 5. Real water distribution networks case 1

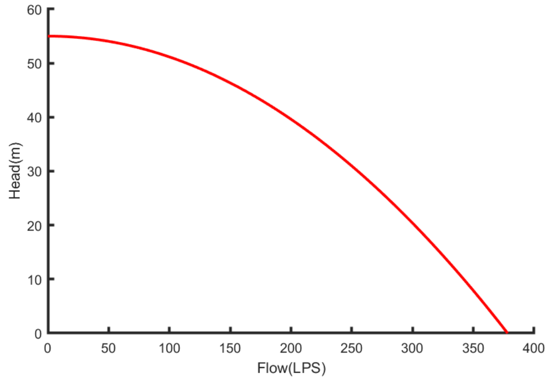


Figure 6. Characteristic curve

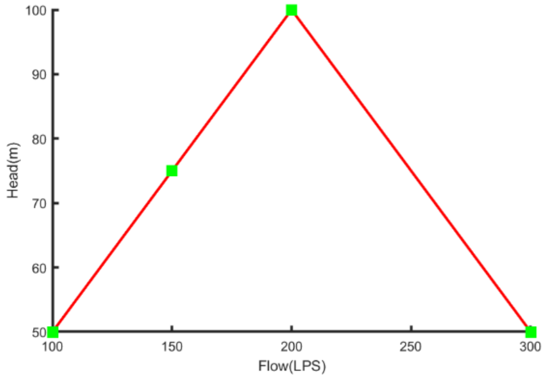


Figure 7. Efficiency curve

5.2 Solution Process

Using the CS algorithm for optimization scheduling, the steps are as follows: *a*. Set the time to t, write a program in MATLAB, randomly generate N bamboo, which represent, N water pump speed ratios, and input them into the OWA toolbox for hydraulic calculation. Calculate the power consumption cost of the water pump through Eq.14; *b*. Update the position of bamboo individuals through global and local search stages, perform hydraulic calculations again, and compare with the limiting conditions of Eq.15 to Eq.17. If it meets the requirements, calculate the electricity cost of the water pump through Eq.14. If it does not meet the requirements, discard it; *c*. Repeat iterative calculations until the electricity bill approaches the optimal solution; *d*. Set time $t = t + 1$, return step a to repeat the operation until time t is greater than the maximum time.

5.3 Case Testing and Analysis

Using a manually constructed simple water distribution network and a real water distribution network as examples, this study verifies the practicality of the proposed pump scheduling optimization method.

5.3.1 Real Water Distribution Networks in Case 1

The pipeline network includes a water source reservoir, which is pumped up by a water pump and transported to a pipeline network with two loops. As shown in Figure 5, the water distribution network comprises 8 nodes, 9 pipelines, 1 water pump, and 1 water plant.

The water pump characteristic curve represents the relationship between the head and flow of the water pump, where the water pump operates at its rated speed, as shown in Figure 6. The pump efficiency curve indicates that the pump efficiency (Y, expressed as a percentage) is a function of the pump flow (X, expressed in flow units), as shown in Figure 7.

The default electricity fee is charged based on a gradient, with 8:00-11:00 and 18:00-23:00 designated as peak hours; 11:00-18:00 and 7:00-8:00 as off-peak hours; and 23:00-7:00 as the low-usage period. The charging standards are detailed in Table 9.

5.3.2 Real Water Distribution Networks in Case 2

The pipeline network includes a water source reservoir, which is pumped up by a water pump and transported to a pipeline network consisting of three rings. As shown in Figure 8, the water distribution network comprises 11 nodes, 12 pipelines, 1 water pump, and 1 water plant. The characteristic curve of the water pump is shown in Figure 9, and the pump efficiency curve is shown in Figure 7. The default electricity charging standards, as referenced in Table 9, are charged according to gradients.

Table 9. Electricity fee charging rules

Period of time	$Cost\left(\frac{yuan}{kW\cdot h}\right)$
Peak	1.29
Flat peak	0.84
Low valley	0.42

5.4 Experimental Analysis

Meta-heuristic optimization algorithms were used in the water distribution network to optimize the scheduling of water pumps at different time periods, and comparisons were made. Calculate the sum of the energy consumption of the water pump and the leakage cost for each partition at each moment, and perform a unified operation 10 times. Compare the average power consumption cost, minimum power consumption cost, and variance of the 10 operation results. Table 10 shows a comparison of the electricity consumption costs optimized by the algorithm for real water distribution Networks Case 1. Table 11 shows the speed ratio of the water pump corresponding to the minimum electricity consumption cost optimized by each algorithm over a period of 12 hours.

From Table 10, it can be seen that under the full power operation of the water pump, the electricity consumption cost reached 1086.9 yuan. After using the meta heuristic optimization algorithm for scheduling optimization, the electricity consumption cost can reach $\left[\frac{1}{10},\frac{7}{10}\right]$ of the total cost, greatly saving electricity consumption. Additionally, the CSBFGO algorithm outperforms other algorithms in terms of average, minimum, and variance of power consumption costs, saving over 900 yuan compared to full power operation.

Table 12 shows a comparison of the power consumption costs for real water distribution Networks Case 2. The speed ratio of the water pump corresponding to the minimum power consumption cost optimized by each algorithm over a 12-hour period is shown in Table 13.

From Table 12, it can be seen that under the full power operation of the water pump, the electricity consumption cost reached 37056.9 yuan, and the use of meta heuristic optimization algorithm for scheduling optimization greatly saved electricity consumption. In addition, the CSBFGO algorithm outperforms other algorithms in terms of average, minimum, and variance of power consumption costs, and can save at least 30000 yuan compared to full power operation.

6 Conclusion

This article proposes two improvement strategies to address the shortcomings of bamboo forest growth optimization algorithms, which often fall into local optima. By integrating aggregation grouping and Cauchy mutation strategies into the algorithm, the diversity of solutions within the population is enhanced, thereby improving the algorithm's ability to escape local optima. In tests conducted with the CEC2017 benchmark, the CSBFGO algorithm demonstrated commendable optimization accuracy and convergence capabilities, effectively addressing the limitations of the BFGO algorithm. Additionally, the CSBFGO algorithm was applied to optimize the scheduling of water pumps in a water distribution network, illustrating its effectiveness in real-world applications. The experimental results highlighted the algorithm's ability to significantly reduce power consumption throughout the pipe network, achieving notable optimization outcomes in practical settings.

Looking ahead, future research could focus on optimizing both the time and space complexities of the BFGO algorithm. For instance, exploring the integration of Compact technology, which uses Gaussian distribution to efficiently represent the population, could enhance memory management. Moreover, to reduce algorithmic runtime, it is crucial to streamline the update strategy and refine the algorithm's internal update mechanisms. By adopting these various approaches, while maintaining the algorithm's convergence and diversity, it is possible to reduce both execution time and memory usage. This concerted effort will make the algorithm more competitive in addressing real-world challenges.

Table 10. Comparison of the electricity consumption costs for real water distribution networks case 1

Cost	PSO	WOA	SCA	SOA	APPE	BFGO	CSBFGO
Electricity consumption for full power operation	1086.9	1086.9	1086.9	1086.9	1086.9	1086.9	1086.9
Average power consumption cost	110.1	184.8	193.5	139.2	167.3	117.9	87.7
Minimum power consumption cost	107.9	105.9	141.5	88.5	98.3	83.7	77.9
Variance	53.8	63.6	46.8	53.5	56.9	54.8	5.2

Table 11. The speed ratio of the water pump for real water distribution networks case 1

Time (hours)	PSO	WOA	SCA	SOA	APPE	BFGO	CSBFGO
1	0.630	0.566	0.566	0.596	0.670	0.643	0.634
2	0.693	0.769	0.385	0.720	0.650	0.713	0.682
3	0.690	0.619	0.665	0.498	0.506	0.463	0.482
4	0.711	0.432	0.642	0.539	0.609	0.458	0.492
5	0.509	0.680	0.572	0.722	0.610	0.653	0.549
6	0.406	0.741	0.643	0.687	0.607	0.598	0.595
7	0.742	0.540	0.669	0.621	0.707	0.657	0.627
8	0.887	0.485	0.929	0.440	0.697	0.544	0.540
9	0.908	0.662	0.431	0.513	0.584	0.501	0.535
10	0.705	0.497	0.725	0.533	0.715	0.659	0.504
11	0.762	0.719	0.564	0.713	0.571	0.487	0.630
12	0.640	0.512	0.630	0.000	0.589	0.378	0.434

Table 12. Comparison of the electricity consumption costs for real water distribution networks case 2

Cost	PSO	WOA	SCA	SOA	APPE	BFGO	CSBFGO
Electricity consumption for full power operation	37056.9	37056.9	37056.9	37056.9	37056.9	37056.9	37056.9
Average power consumption cost	2709.0	6644.8	4807.5	3731.6	2271.7	2450.2	2070.1
Minimum power consumption cost	1539.4	5499.9	4069.4	2269.9	1495.1	1557.7	1401.1
Variance	1021.5	1164.9	513.7	876.8	7238.3	731.8	846.4

Table 13. The speed ratio of the water pump for real water distribution networks case 2

Time (hours)	PSO	WOA	SCA	SOA	APPE	BFGO	CSBFGO
1	0.792	0.724	0.824	0.743	0.757	0.736	0.735
2	0.727	0.707	0.713	0.703	0.730	0.700	0.700
3	0.817	0.700	0.960	0.734	0.719	0.701	0.779
4	0.827	0.709	0.808	0.708	0.770	0.778	0.724
5	0.817	0.746	0.717	0.387	0.754	0.725	0.000
6	0.702	0.746	0.112	0.192	0.772	0.998	0.000
7	0.256	0.268	0.000	0.000	0.223	0.270	0.055
8	0.439	0.745	0.000	0.000	0.734	0.000	0.122
9	0.265	0.746	0.000	0.000	0.342	0.112	0.000
10	0.583	0.746	0.007	0.000	0.342	0.921	0.000
11	0.943	0.746	0.000	0.000	0.000	0.539	0.511
12	0.422	0.313	0.000	0.000	0.731	0.000	0.321

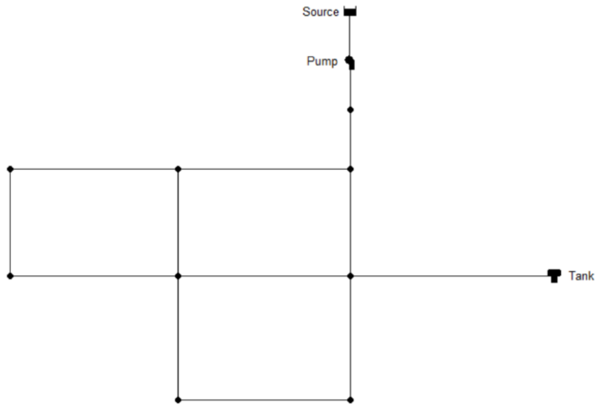


Figure 8. Real water distribution networks case 2

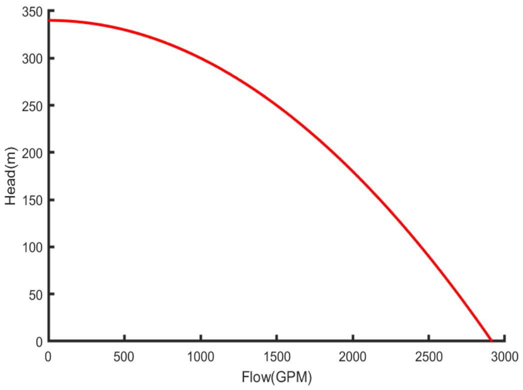


Figure 9. Characteristic curve

References

[1] S. N. Sivanandam, S. N. Deepa, Genetic algorithm optimization problems, *Introduction to genetic algorithms*, Springer, Berlin, Heidelberg, 2008, pp. 165-209.

[2] E. Osaba, E. Villar-Rodriguez, J. Del Ser, A. J. Nebro, D. Molina, A. LaTorre, P. N. Suganthan, C. A. C. Coello, and F. Herrera, A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems, *Swarm and Evolutionary Computation*, Vol. 64, No. 100888, July, 2021.

[3] D. Wang, D. Tan, and L. Liu, Particle swarm optimization algorithm: An overview, *Soft computing*, Vol. 22, No. 2, pp. 387-408, January, 2018.

[4] M. S. Shaikh, C. Hua, M. Hassan, S. Raj, M. A. Jatoi and M. M. Ansari, Optimal parameter estimation of overhead transmission line considering different bundle conductors with the uncertainty of load modeling, *Optimal control applications and methods*, Vol. 43, No. 3, pp. 652-666, May/June, 2022.

[5] O. W. Khalid, N. A. M. Isa, and H. A. M. Sakim, Emperor penguin optimizer: A comprehensive review based on state-of-the-art meta-heuristic algorithms, *Alexandria Engineering Journal*, Vol. 63, pp. 487-526, January, 2023.

[6] M. Lukasiewycz, M. Gla, F. Reimann, and J. Teich, Opt4j: A modular framework for meta-heuristic optimization, Proceedings of the 13th annual conference on Genetic and evolutionary computation, Dublin, Ireland, 2011, pp. 1723-1730.

[7] S. Guindon, and O. Gascuel, A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood, *Systematic biology*, Vol. 52, No. 5, pp. 696-704, October, 2003.

[8] T. Mancini, M. Maratea and F. Ricca, Experimental evaluation of algorithms for solving problems with combinatorial explosion, *Ai Communications*, Vol. 29, No. 2, pp. 245-247, April, 2015.

[9] D. S. Hochba, Approximation algorithms for np-hard problems, *ACM Sigact News*, Vol. 28, No. 2, pp. 40-52, June, 1997.

[10] M. S. Shaikh, C. Hua, M. A. Jatoi, M. M. Ansari, A. A. Qader, Application of grey wolf optimisation algorithm in parameter calculation of overhead transmission line system, *IET Science, Measurement & Technology*, Vol. 15, No. 2, pp. 218-231, March, 2021.

[11] D. Stilck Frana, and R. Garcia-Patron, Limitations of optimization algorithms on noisy quantum devices, *Nature Physics*, Vol. 17, No. 11, pp. 1221-1227, November, 2021.

[12] M. S. Shaikh, C. Hua, S. Raj, S. Kumar, M. Hassan, M. M. Ansari and M. A. Jatoi, Optimal parameter estimation of 1-phase and 3-phase transmission line for various bundle conductor's using modified whale optimization algorithm, *International Journal of Electrical Power & Energy Systems*, Vol. 138, Article No. 107893, June, 2022.

[13] A. M. Fadel, A. S. Abdel-Khalik, and R. A. Hamdy, Genetic algorithm based parameter estimation of six-phase induction machine sequence circuits, *2021 22nd International Middle East Power Systems Conference (MEPCON)*, Assiut, Egypt, 2021, pp. 233-238.

[14] S. Das, and P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE Transactions on Evolutionary Computation*, Vol. 15, No. 1, pp. 4-31, February, 2011.

[15] Z. Meng, J.-S. Pan, and H. Xu, Quasi-affine transformation evolutionary (quatre) algorithm: A cooperative swarm based algorithm for global optimization, *Knowledge-Based Systems*, Vol. 109, pp. 104-121, October, 2016.

[16] M. S. Shaikh, S. Raj, R. Babu, S. Kumar and K. Sagrolikar, A hybrid moth–flame algorithm with particle swarm optimization with application in power transmission and distribution, *Decision Analytics Journal*, Vol. 6, Article No. 100182, March, 2023.

[17] T.-Y. Wu, A. Shao, and J.-S. Pan, Ctoa: Toward a chaotic-based tumbleweed optimization algorithm, *Mathematics*, Vol. 11, No. 10, Article No. 2339, May, 2023.

[18] M. S. Shaikh, S. Raj, M. Ikram and W. Khan, Parameters estimation of AC transmission line by an improved moth flame optimization method, *Journal of Electrical Systems and Information Technology*, Vol. 9, No. 1, Article No. 25, December, 2022.

[19] P. Trojovský and M. Dehghani, Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications, *Sensors*, Vol. 22, No. 3, Article No. 855, February, 2022.

[20] R. V. Rao, V. J. Savsani and D. P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, *Computer-aided design*, Vol 43, No. 3, pp. 303-315, March, 2011.

[21] S. Cheng, Q. Qin, J. Chen and Y. Shi, Brain storm optimization algorithm: a review, *Artificial Intelligence Review*, Vol. 46, No. 4, pp. 445-458, December, 2016.

[22] B. Das, V. Mukherjee and D. Das, Student psychology

based optimization algorithm: A new population based optimization algorithm for solving optimization problems, *Advances in Engineering software*, Vol. 146, Article No. 102804, August, 2020.

[23] D. Bertsimas, and J. Tsitsiklis, Simulated annealing, *Statistical science*, Vol. 8, No. 1, pp. 10-15, February, 1993.

[24] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, Water cycle algorithm-a novel metaheuristic optimization method for solving constrained engineering optimization problems, *Computers & Structures*, Vol. 110-111, pp. 151-166, November, 2012.

[25] I. Ahmadianfar, O. Bozorg-Haddad and X. Chu, Gradient-based optimizer: A new metaheuristic optimization algorithm, *Information Sciences*, Vol. 540, pp. 131-159, November, 2020.

[26] Z. W. Geem, J. H. Kim, G. V. Loganathan, A new heuristic optimization algorithm: harmony search, *Simulation*, Vol. 76, No. 2, pp. 60-68, February, 2001.

[27] U. Guvenc, S. Duman, H. T. Kahraman, S. Aras, and M. Katı, Fitness–distance balance based adaptive guided differential evolution algorithm for security-constrained optimal power flow problem incorporating renewable energy sources, *Applied Soft Computing*, Vol. 108, Article No. 107421, September, 2021.

[28] H. T. Kahraman, M. Katı, S. Aras, D. A. Taşci, Development of the Natural Survivor Method (NSM) for designing an updating mechanism in metaheuristic search algorithms, *Engineering Applications of Artificial Intelligence*, Vol. 122, Article No. 106121, June, 2023.

[29] H. T. Kahraman, S. Aras and E. Gedikli, Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms, *Knowledge-Based Systems*, Vol. 190, Article No. 105169, February, 2020.

[30] B. Ozkaya, H. T. Kahraman, S. Duman, and U. Guvenc, Fitness-Distance-Constraint (FDC) based guide selection method for constrained optimization problems, *Applied Soft Computing*, Vol. 144, Article No. 110479, September, 2023.

[31] H. T. Kahraman, H. Bakir, S. Duman, M. Katı, S. Aras and U. Guvenc, Dynamic FDB selection method and its application: modeling and optimizing of directional overcurrent relays coordination, *Applied Intelligence*, Vol. 52, No. 5, pp. 4873-4908, March, 2022.

[32] S. Duman, H. T. Kahraman, M. Kati, Economical operation of modern power grids incorporating uncertainties of renewable energy sources and load demand using the adaptive fitness-distance balance-based stochastic fractal search algorithm, *Engineering Applications of Artificial Intelligence*, Vol. 117, Article No. 105501, January, 2023.

[33] S.-C. Chu, Q. Feng, J. Zhao, and J.-S. Pan, BFGO: Bamboo forest growth optimization algorithm, *Journal of Internet Technology*, Vol. 24, No. 1, pp. 1-10, January, 2023.

[34] J.-W. Kang, H.-J. Park, J.-S. Ro, and H.-K. Jung, A strategy-selecting hybrid optimization algorithm to overcome the problems of the no free lunch theorem, *IEEE Transactions on Magnetics*, Vol. 54, No. 3, pp. 1-4, March, 2018.

[35] P. Hu, J.-S. Pan, S.-C. Chu, and C. Sun, Multi-surrogate assisted binary particle swarm optimization algorithm and its application for feature selection, *Applied Soft Computing*, Vol. 121, Article No. 108736, May, 2022.

[36] X. J. Lei, S. Wu, G. E. Liang, A. D. Zhang, Clustering ppi data based on ant colony optimization algorithm, *Chinese Journal of Electronics*, Vol. 22, No. 1, pp. 118-123, January, 2013.

[37] H.-J. Wang, J.-S. Pan, T.-T. Nguyen, and S. Weng, Distribution network reconfiguration with distributed generation based on parallel slime mould algorithm, *Energy*, Vol. 244, Article No. 123011, April, 2022.

[38] Y. X. Du, Y. H. Chen, Time optimal trajectory planning algorithm for robotic manipulator based on locally chaotic particle swarm optimization, *Chinese Journal of Electronics*, Vol. 31, No. 5, pp. 906-914, September, 2022.

[39] A.-N. Zhang, S.-C. Chu, P.-C. Song, H. Wang, and J.-S. Pan, Task scheduling in cloud computing environment using advanced phasmatodea population evolution algorithms, *Electronics*, Vol. 11, No. 9, Article No. 1451, May, 2022.

[40] T.-Y. Wu, H. Li, and S.-C. Chu, CPPE: An improved phasmatodea population evolution algorithm with chaotic maps, *Mathematics*, Vol. 11, No. 9, Article No. 1977, May, 2023.

[41] L. Kang, R. S. Chen, W. Cao, Y. C. Chen, Non-inertial opposition-based particle swarm optimization and its theoretical analysis for deep learning applications, *Applied Soft Computing*, Vol. 88, Article No. 106038, March, 2020.

[42] R. J. Soreng, P. M. Peterson, K. Romaschenko, G. Davidse, J. K. Teisher, L. G. Clark, P. Barber, L. J. Gillespie, and F. O. Zuloaga, A worldwide phylogenetic classification of the poaceae (gramineae) ii: An update and a comparison of two 2015 classifications, *Journal of Systematics and evolution*, Vol. 55, No. 4, pp. 259-290, July, 2017.

[43] X. Song, C. Peng, G. Zhou, H. Gu, Q. Li, and C. Zhang, Dynamic allocation and transfer of non-structural carbohydrates, a possible mechanism for the explosive growth of moso bamboo (phyllostachys heterocycla), *Scientific reports*, Vol. 6, No. 1, Article No. 25908, May, 2016.

[44] G. Jin, P.-F. Ma, X. Wu, L. Gu, M. Long, C. Zhang, and D.-Z. Li, New genes interacted with recent whole-genome duplicates in the fast stem growth of bamboos, *Molecular biology and evolution*, Vol. 38, No. 12, pp. 5752-5768, December, 2021.

[45] Y. Shi, E. Liu, G. Zhou, Z. Shen, S. Yu, and others, Bamboo shoot growth model based on the stochastic process and its application, *Scientia Silvae Sinicae*, Vol. 49, No. 9, pp. 89-93, September, 2013.

[46] R. A. Krohling, and L. dos Santos Coelho, Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 36, No. 6, pp. 1407-1416, December, 2006.

[47] M. Wang, J.-S. Wang, X.-D. Li, M. Zhang, and W.-K. Hao, Harris hawk optimization algorithm based on cauchy distribution inverse cumulative function and tangent flight operator, *Applied Intelligence*, Vol. 52, No. 10, pp. 10999-11026, August, 2022.

[48] H. T. Öztürk, H. T. Kahraman, Meta-heuristic search algorithms in truss optimization: Research on stability and complexity analyses, *Applied Soft Computing*, Vol. 145, Article No. 110573, September, 2023.

[49] G. Wu, R. Mallipeddi, and P. N. Suganthan, Problem definitions and evaluation criteria for the cec 2017 competition and Special Session on constrained single objectve real-parameter optimization, Technical Report, September, 2017.

[50] M. S. Shaikh, C. Hua, M. A. Jatoi, M. M. Ansari and A. A. Qader, Parameter estimation of AC transmission line considering different bundle conductors using flux

linkage technique, *IEEE Canadian Journal of Electrical and Computer Engineering*, Vol. 44, No. 3, pp. 313-320, Summer, 2021.

[51] M. S. Shaikh, M. M. Ansari, M. A. Jatoi, Z. A. Arain and A. A. Qader, Analysis of underground cable fault techniques using MATLAB simulation, *Sukkur IBA Journal of Computing and Mathematical Sciences*, Vol. 4, No. 1, pp. 1-10, July, 2020.

[52] M. A. Mabrok, A. Saad, T. Ahmed, and H. Alsayab, Modeling and simulations of water network distribution to assess water quality: Kuwait as a case study, *Alexandria Engineering Journal*, Vol. 61, No. 12, pp. 11859-11877, December, 2022.

[53] T. Luna, J. Ribau, D. Figueiredo, and R. Alves, Improving energy efficiency in water supply systems with pump scheduling optimization, *Journal of cleaner production*, Vol. 213, pp. 342-356, March, 2019.

[54] R. Menke, E. Abraham, P. Parpas, and I. Stoianov, Exploring optimal pump scheduling in water distribution networks with branch and bound methods, *Water Resources Management*, Vol. 30, No. 14, pp. 5333-5349, November, 2016.

[55] S. M. S. Abkenar, S. D. Stanley, C. J. Miller, D. V. Chase, and S. P. McElmurry, Evaluation of genetic algorithms using discrete and continuous methods for pump optimization of water distribution systems, *Sustainable Computing: Informatics and Systems*, Vol. 8, pp. 18-23, December, 2015.

Biographies

Jeng-Shyang Pan received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is currently the Professor of Shandong University of Science and Technology. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section and Tainan Chapter Chair of IEEE Signal Processing Society. His current research interest includes the information hiding, artificial intelligence and wireless sensor networks.

An-Ning Zhang received his B.S. degree from Shandong University of Aeronautics, China, in 2021. He is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. His recent research interests include swarm intelligence and cloud computing.

Shu-Chuan Chu received a Ph.D. degree in 2004 from the School of Computer Science, Engineering and Mathematics, Flinders University of South Australia. She joined Flinders University in December 2009 after 9 years at the Cheng Shiu University, Taiwan. She has been a Research Fellow and Associate Professor in the College of Science and Engineering of Flinders University, Australia since December 2009. Currently, she is a Research Fellow with a Ph.D. advisor in the College of Computer Science and Engineering of Shandong University of Science and Technology from September 2019. She also serves as an editorial board member for Engineering Applications of Artificial Intelligence (EAAI), Journal of Internet Technology (JIT) and Research Reports on Computer Science (RRCS). Her research interests are mainly in swarm intelligence, intelligent computing, and wireless sensor networks.

Qing Feng received her B.S. degree from Qingdao Binhai University, China, in 2020. She is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. Her recent research interests include swarm intelligence, clustering, and FPGA.

Yu-Chung Huang received BS and MS degrees in Chemical Engineering from the National Cheng Kung University, Taiwan in 1981 and 1983 respectively. He was a lecturer (1985 – 1989) and an associate professor (1989 – 1991) at the Southern Taiwan University of Science and Technology. After that, he founded Energy Management System Co. LTD and served as a general manager (1991 – 1995) and a chairman (1995 - 2013) until he left the company in 2013. Meanwhile, he was also a member of the Taiwan BSMI (Bureau of Standards, Metrology and Inspection) Standards Committee. Currently, he is a doctoral student in Electrical Engineering at the National University of Tainan, Taiwan. He is also a chairman of Taiwan Smart Water Co, LTD, a vice chairman of Energy Management System Co. LTD, as well as a consultant to several companies in China. He is currently interested in finding water leakage using hydraulic model in water supply network.