

An Improved Attraction-Repulsion Optimization Algorithm for Global Optimization and Engineering Design Problems

Fang Feng^{1*} , Mingjiang Cai^{2,4} , Hung-Wei Li³ , Kuan-Ching Li^{3*} 

¹ Guizhou Provincial Key Laboratory of Computing and Network Convergence, School of Information, Guizhou University of Finance and Economics, Guiyang (China)

² Nanjing District Bureau of Commerce, Guiyang (China)

³ Department of Computer Science and Information Engineering, Providence University, Taichung (Taiwan)

⁴ College of Public Management, Guizhou University of Finance and Economics, Guiyang (China)

* Corresponding author: fengf15@mail.gufe.edu.cn (F. Feng), kuancli@pu.edu.tw (K. C. Li).

Received 25 October 2024 | Accepted 6 May 2026 | Published 28 August 2026



ABSTRACT

Attraction-Repulsion Optimization Algorithm (AROA) is a recently proposed meta-heuristic algorithm known for its simplicity, ease of implementation, and robustness. However, AROA may converge to local optima when applied to complex optimization problems. To address this limitation, we propose an enhanced version called the Differential Cauchy Tangent Attraction-Repulsion Optimization Algorithm (DCTAROA). First, we propose a mutation operator based on a tangent flight mutation strategy and a dimension decision mechanism using the inverse cumulative distribution function of the Cauchy distribution. The tangent flight mutation enhances the local search capability and accelerates convergence, while the dimension decision strategy of the Cauchy distribution inverse cumulative function increases population diversity and improves exploration efficiency. Subsequently, we integrate Differential Evolution (DE) as a local search mechanism to strengthen the global optimization performance of AROA. To evaluate the proposed algorithm, we compare it with 15 state-of-the-art algorithms on 29 CEC2017 benchmark functions across various dimensions. Experimental results demonstrate that DCTAROA outperforms the compared algorithms in terms of solution accuracy, stability, convergence speed, and statistical significance based on the Wilcoxon rank-sum test. Furthermore, we apply DCTAROA to three practical engineering design problems. The results confirm that DCTAROA effectively explores the search space and yields competitive solutions, thereby validating its practical applicability.

KEYWORDS

Attraction-Repulsion Optimization Algorithm, Cauchy Distribution Inverse Cumulative Function, Differential Evolution, Dimension Decision Strategy, Tangent Flight Mutation Operator.

DOI: 10.9781/ijimai.2026.2220
eLocator: e2220

I. INTRODUCTION

IN recent decades, numerous meta-heuristic and evolutionary algorithms (MAS) have been developed to solve complex optimization problems [1]–[11]. These algorithms are widely adopted in both academic and industrial applications due to their versatility [12]–[14].

A key advantage of MAS is their straightforward implementation. Unlike traditional optimization methods, MAS do not require the objective or constraint functions to possess specific analytical properties. They initiate the search process with a randomly generated population and iteratively update solutions based on predefined rules.

Meta-heuristic algorithms are often inspired by natural evolution, swarm behaviors of insects and birds [15], [16], mathematical laws [17], physical principles [18], [19], and human behavior [20], [21].

A fundamental aspect of MAS is the balance between exploration and exploitation [22]. Excessive exploration can lead to poor solution quality, while over-exploitation increases the risk of local convergence. Therefore, achieving an effective balance is crucial for obtaining high-quality solutions.

The “No Free Lunch” (NFL) theorem states that no single algorithm is universally optimal for all optimization problems [23], [24]. This implies that algorithm performance varies across problem types, motivating the continuous development of new and improved methods.

Please cite this article as:

F. Feng, M. Cai, H.-W. Li, K.-C. Li. An improved Attraction-Repulsion Optimization Algorithm for Global Optimization and Engineering Design Problems, *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 10, no. 1, e2220, 2026, <http://doi.org/10.9781/ijimai.2026.2220>

Existing improvements to MAs generally fall into two categories: hybridization of different algorithms and modification of algorithmic components. Hybrid approaches combine strengths from multiple algorithms. For example, Yildiz et al. integrated the Grasshopper Optimization Algorithm (GOA) with the Nelder–Mead simplex method to enhance local search [25]. Kamboj et al. proposed a hybrid Harris Hawks–Sine Cosine Algorithm (hHHO-SCA) that demonstrated strong performance in engineering design [16]. Bao et al. developed a hybrid HHO-DE algorithm for multi-level color image segmentation [16].

Component-based improvements often involve parameter adjustment, initialization strategies, or update mechanisms. For instance, opposition-based learning (OBL) was used to improve initial population quality in the Grey Wolf Optimizer [26] and in evolutionary algorithms [27]. Chaotic mapping replaced linear parameter reduction in the Chaotic Grasshopper Optimization Algorithm [28]. Adaptive parameter strategies were introduced to enhance convergence [29], [30]. Gaussian mutation and Lévy flight were incorporated to maintain population diversity [31].

Recently, fitness-distance balance (FDB) and its variants have been proposed to refine selection mechanisms [32]–[35], leading to algorithms such as FDB-SOS [32], FDB-AEO [36], and dFDB-ARO [37]. Elite evolutionary strategies [38] and quantum-inspired multi-strategy enhancements [39] have also been developed to improve optimization performance.

Inspired by natural attraction–repulsion phenomena, Cymerys et al. recently proposed the Attraction-Repulsion Optimization Algorithm (AROA) [40]. AROA introduces a search strategy in which candidate solutions move in the search space based on the quality of the solutions in their neighborhood and the best candidate solution. The candidate solutions are managed by local search operators based on modified Brownian motion, trigonometric functions, random selection, and memory forms. Despite its superior performance, it exhibits limitations in handling discrete problems and may struggle to converge closely to the global optimum.

To address these limitations, this study proposes an enhanced variant called the Differential Cauchy Tangent Attraction-Repulsion Optimization Algorithm (DCTAROA). The main contributions of this work are summarized as follows:

- (1) Proposal of a novel mutation operator (CT): The proposed CT operator integrates a tangent flight strategy with a dimension decision mechanism based on the inverse cumulative distribution function of the Cauchy distribution, which enhances local search accuracy,
- (2) Development of the DCTAROA algorithm: The proposed algorithm integrates the new CT mutation operator with a Differential Evolution (DE) strategy to systematically improve the core AROA framework, balancing exploration and exploitation.
- (3) Comprehensive performance validation: The algorithm's efficacy is rigorously evaluated against 15 state-of-the-art optimization algorithms using the IEEE CEC 2017 benchmark suite. Statistical analysis via the Wilcoxon rank-sum test confirms the significant performance advantages of DCTAROA.
- (4) Demonstration of practical applicability: The proposed algorithm is successfully applied to solve three constrained engineering design problems, confirming its robustness for real-world optimization challenges.

The remainder of this paper is organized as follows: Section II describes the original AROA algorithm, Section III introduces the proposed DCTAROA, Section IV presents experimental results and discussions, and Section V concludes the study and suggests future work.

II. ATTRACTION-REPULSION OPTIMIZATION ALGORITHM

The Attraction-Repulsion Optimization Algorithm (AROA) models the attraction–repulsion phenomena observed in nature [40]. In this algorithm, candidate solutions navigate the search space based on the fitness quality of neighboring solutions and the current global best candidate. Solutions with higher fitness attract others, while those with lower fitness are repelled. Therefore, the candidate's position depends on the attraction–repulsion operator, attraction to the optimal solution, memory, influence of the population random solution, and a set of search space operators responsible for local motion.

A. Initialization

The initial positions of all agents in population X are generated randomly, defined as:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (1)$$

The position of the i -th agent x_i is initialized using (2):

$$x_i = x_{\min} + \text{rand}(0, 1) \cdot (x_{\max} - x_{\min}) \quad (2)$$

Here, $\text{rand}(0, 1)$ is a random number uniformly distributed in $[0, 1]$, and $x_{\max}$ and $x_{\min}$ define the upper and lower bounds of the search space, respectively.

B. Attraction and Repulsion

The first component influencing candidate positions is the fitness information from other population members, which depends on the distance between members. The vector n_i is added to the position of the candidate solution, computed as:

$$n_i = \frac{1}{n} \sum_{j=1}^k c \cdot (x_j - x_i) \cdot I(d_{i,j}, d_{i,\max}) \cdot s(f_i, f_j) \quad (3)$$

where f denotes the objective function vector, c is a step size, and n is the population size.

To update candidate positions, a distance matrix DM is constructed:

$$DM = \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,n} \\ d_{2,1} & d_{2,2} & \dots & d_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n,1} & d_{n,2} & \dots & d_{n,n} \end{bmatrix} \quad (4)$$

In this matrix, $d_{i,j}$ represents the Euclidean distance between agents i and j . The squared Euclidean distance $d^2(x_i, x_j)$ is given by:

$$d^2(x_i, x_j) = \sum_{k=1}^D (x_i^k - x_j^k)^2 \quad (5)$$

Let D define the solution space's topological dimension in the given optimization framework, where x_i , x_j denote the position vectors of the i -th and j -th agents respectively. The distance matrix DM computationally encodes pairwise spatial separations, with $d_{i,\max}$ specifying the maximum Euclidean distance between the i -th agent and its population counterparts. Based on $d_{i,j} \in D$ and $d_{i,\max}$, the vector n_i is incorporated into the candidate's position. This vector represents the effect of the attraction–repulsion mechanism, considering the k nearest individuals of the i -th solution.

I is the function to evaluate the impact of the solution, computed as:

$$I(d_{i,j}, d_{i,\max}) = 1 - \frac{d_{i,j}}{d_{i,\max}} \quad (6)$$

The direction function s is given by:

$$s(f_i, f_j) = \begin{cases} 1, & f_i > f_j \\ 0, & f_i = f_j \\ -1, & f_i < f_j \end{cases} \quad (7)$$

where f_i and f_j denote the fitness values of the i -th and j -th agents, respectively.

The number of neighbors k is determined as:

$$k = \lfloor (1 - \frac{t}{t_{\max}}) \cdot n \rfloor + 1 \quad (8)$$

Here, t and $t_{\max}$ denote the current and maximum iteration numbers, respectively. $t_{\max}$ is given by:

$$t_{\max} = \lceil \frac{fes_{\max} - n}{2n} \rceil \quad (9)$$

$fes_{\max}$ denotes the number of fitness function evaluations.

C. Attraction by the Best Solution

The second factor that affects the position of the candidate solution is the attraction of the best solution b_i , formulated as:

$$b_i = \begin{cases} c \cdot m \cdot (x_{\text{best}} - x_i), & r_1 \geq p_1 \\ c \cdot m \cdot (a_1 \odot x_{\text{best}} - x_i), & r_1 < p_1 \end{cases} \quad (10)$$

where a_1 is a random vector in $[0, 1]$, c is the step size, x_{best} is the current best solution, r_1 is a random number in $[0, 1]$, and p_1 is a probability threshold.

The balance factor m is calculated as:

$$m = \frac{1}{2} \left(\frac{\exp(z) - 1}{\exp(z) + 1} + 1 \right) \quad (11)$$

with z defined by:

$$z = 18 \cdot \frac{t}{t_{\max}} - 4 \quad (12)$$

D. Local Search Operators

The third component involves local search operators selected via a probability switch p_2 . Three operators are employed:

1. Modified Brownian motion operator r_B :

$$r_B = u_1 \odot N(0, fr_1 \cdot (1 - \frac{t}{t_{\max}}) \cdot (x_{\max} - x_{\min})) \quad (13)$$

u_1 represents a binary vector, N represents a randomly generated number vector with normal distribution, and fr_1 represents a small number. When random values are greater than the threshold 1, Elements of this vector are set to 1. The elements with smaller values are set to 0. Consequently, only randomly selected elements of the decision vector will be updated.

2. Trigonometric function operator r_{tri} , selected via a roulette wheel strategy:

$$r_{tri} = \begin{cases} fr_2 \cdot u_2 \cdot (1 - \frac{t}{t_{\max}}) \cdot \sin(2r_5\pi) \odot |a_2 \odot x_w - x_i|, & r_4 < 0.5 \\ fr_2 \cdot u_2 \cdot (1 - \frac{t}{t_{\max}}) \cdot \cos(2r_5\pi) \odot |a_2 \odot x_w - x_i|, & r_4 \geq 0.5 \end{cases} \quad (14)$$

Here, fr_2 is a multiplier, u_2 is a binary vector with threshold tr_2 ,

r_4 and r_5 are random numbers in $[0, 1]$, a_2 represents a vector random numbers between 0 and 1. Solution x_w refers to the roulette wheel's selection method. In AROA, the local search operator adopts the roulette wheel selection in SCA, and the optimal solution selection adopts the greedy selection method.

3. Random position operator r_R :

$$r_R = u_3 \odot (2 \cdot a_3 - o) \odot (x_{\max} - x_{\min}) \quad (15)$$

where a_3 is a random vector, u_3 is a binary vector with threshold tr_3 , and o is a unit vector.

Here, the probability switch p_2 determines which of the three random local search operators to use. The position update is obtained using (16), r_2 and r_3 are random numbers ranging from 0 to 1, r_i represents the action of local search operators:

$$r_i = \begin{cases} r_B, & r_3 > 0.5 \cdot \frac{t}{t_{\max}} + 0.25 \\ r_{tri}, & r_3 \leq 0.5 \cdot \frac{t}{t_{\max}} + 0.25, \quad r_2 < p_2 \\ r_R, & r_2 \geq p_2 \end{cases} \quad (16)$$

The candidate's position is then updated as:

$$x_i(t) = x_i(t-1) + n_i + b_i + r_i \quad (17)$$

Here, The vector x_i at the t iteration is written as $x_i(t)$, while its value at the $(t-1)$ iteration is denoted by $x_i(t-1)$.

E. Population-Based Operators

Population-based operators affect all candidates and are computed as:

$$x_i = \begin{cases} x_i + c_f \cdot (u_4 \odot (a_4 \odot (x_{\max} - x_{\min}) + x_{\min})), & r_6 < e_f \\ x_i + (e_f \cdot (1 - r_7) + r_7)(x_{r8} - x_{r9}), & r_6 \geq e_f \end{cases} \quad (18)$$

where e_f is a probability threshold, r_6 and r_7 are random values between 0 and 1, r_8 and r_9 are indexes of randomly selected individuals, u_4 is a binary vector with threshold $1 - e_f$, a_4 represents a vector of randomly generated numbers, the coefficient c_f is given by:

$$c_f = \left(1 - \frac{t}{t_{\max}} \right)^3 \quad (19)$$

Finally, the selection of the optimal solution adopts the greedy selection method, formulated as:

$$x_i(t) = \begin{cases} x_i(t), & f(x_i(t)) < f(x_i(t-1)) \\ x_i(t-1), & f(x_i(t)) \geq f(x_i(t-1)) \end{cases} \quad (20)$$

Here, $f(x_i(t))$ and $f(x_i(t-1))$ denote the fitness values of $x_i(t)$ and $x_i(t-1)$, respectively.

The complete procedure of AROA is summarized in Algorithm 1.

III. PROPOSED DCTAROA

This section describes the proposed improvements to the Attraction-Repulsion Optimization Algorithm (AROA) to enhance its search performance. The enhanced version introduces a refined strategy to coordinate multiple search operators and improve solution quality.

The differential evolution algorithm is a widely used evolutionary algorithm for global optimization in continuous search spaces, known for its simplicity, intuitive operation, and strong convergence properties. The DE/best/1 operator, in particular, exhibits relatively fast convergence, which can enhance the exploitation capability of hybrid algorithms [41]. In practice, DE is often employed as a local

Algorithm 1. Pseudo-code of the original AROA**Require:**

Population size N , fitness evaluations $f_{es_{max}}$, problem dimension D , search space bounds x_{max} and x_{min} , fitness function $f(\cdot)$

Ensure:

Best solution x_{best} and its fitness f_{best}

```

1: Calculate  $t_{max}$  using (9)
2: for  $i = 1$  to  $N$  do
3:   Initialize  $x_i$  using (2)
4: end for
5: for  $i = 1$  to  $N$  do
6:   Compute fitness  $f(x_i)$ 
7: end for
8: Identify  $x_{best}$  and  $f_{best}$ 
9: for  $t = 1$  to  $t_{max}$  do
10:  Compute distance matrix  $DM$  using (4)
11:  for  $i = 1$  to  $N$  do
12:    Compute  $d_{i,max}$  using (5)
13:    Select  $k$  nearest neighbors using (8)
14:    for  $j = 1$  to  $k$  do
15:      Compute attraction-repulsion using (3)
16:    end for
17:    Compute  $b_i$  using (10)
18:    Compute  $r_i$  using (16)
19:    Update  $x_i(t)$  using (17)
20:  end for
21:  for  $i = 1$  to  $N$  do
22:    if  $f_{es_{max}}$  reached then
23:      Break
24:    end if
25:    Compute fitness  $f(x_i)$ 
26:  end for
27:  Update  $x_{best}$  and  $f_{best}$ 
28:  Apply population-based operators using (18)
29:  for  $i = 1$  to  $N$  do
30:    if  $f_{es_{max}}$  reached then
31:      Break
32:    end if
33:    Compute fitness  $f(x_i)$ 
34:  end for
35:  Update  $x_{best}$  and  $f_{best}$ 
36:  Apply greedy selection using (20)
37: end for
38: return  $x_{best}, f_{best}$ 

```

search mechanism to improve the performance of hybrid algorithms, such as those combining DE with genetic algorithms [42], sine-cosine algorithms [43], and artificial bee colony algorithms [44].

The tangent flight function provides a strong exploration capability due to its periodic nature and range from $-\infty$ to $+\infty$, which helps maintain a balance between exploration and exploitation [45]. The inverse cumulative function of the Cauchy distribution enhances the algorithm's search mechanism, mitigates deficiencies in local search, and stabilizes the search process [46].

Inspired by these observations, this paper proposes a new mutation operator based on the tangent flight operator and a dimension decision strategy derived from the inverse cumulative function of the Cauchy

distribution. This mutation operator, combined with the DE algorithm, enhances the AROA algorithm, resulting in a new heuristic algorithm named DCTAROA. The following sections describe these components in detail.

A. Tangent Flight Operator Mutation

Inspired by the mathematical tangent function, Zhao et al. proposed the Tangent Search Algorithm (TSA) [45]. TSA uses models based on tangent functions to move solutions toward improved positions. The tangent flight function provides strong exploration capability, its periodic variation between $-\infty$ and $+\infty$ helps maintain a balance between exploration and exploitation. In TSA, movement equations are controlled by a global step size of the form " $step \cdot \tan(\theta)$," where the tangent function acts as a flight function analogous to Lévy flight.

The tangent flight operator G is defined as:

$$G = \tan(v \cdot \pi / 2) \quad (21)$$

Let v be a $1 \times D$ random vector whose components are independently and uniformly distributed on $[0, 1]$, and let D be the dimension of the function:

$$v = \text{rand}(1, D) \quad (22)$$

The step size of tangent flight is typically small with many steps, occasionally including larger steps, which contributes to balancing exploration and exploitation. To incorporate information from the current population, two positions are randomly selected, and their difference interacts with the tangent flight to form a tangent flight mutation operator O :

$$O = G \cdot (x_{\text{rand},1}^D - x_{\text{rand},2}^D) \quad (23)$$

Here, $x_{\text{rand},1}^D$ and $x_{\text{rand},2}^D$ represent the positions of two randomly selected individuals, O represents the tangent flight mutation operator, and each operator contains D dimensions.

Integrating the tangent flight operator enhances the search mechanism and convergence speed, mitigates local search limitations, stabilizes the search process, and facilitates faster identification of optimal solutions.

B. Dimension Decision Logic Based on the Inverse Cumulative Function of the Cauchy Distribution

The Cauchy distribution, also known as the Cauchy-Lorentz distribution, is a continuous probability distribution [47] that lacks a defined mathematical expectation. When a random variable X follows this distribution, its probability density function is given by:

$$H(x; a, b) = \frac{1}{\pi} \left[\frac{b}{(x - a)^2 + b^2} \right] \quad (24)$$

where a represents the peak position, while b determines the half-width at half maximum. When $a = 0$ and $b = 1$, it is called the standard Cauchy distribution, with probability density function:

$$H(x; 0, 1) = \frac{1}{\pi} \left[\frac{1}{x^2 + 1} \right] \quad (25)$$

The cumulative distribution function is:

$$H(x; a, b) = \frac{1}{\pi} \arctan \left(\frac{x - a}{b} \right) + \frac{1}{2} \quad (26)$$

In the context of cumulative distribution functions, the Cauchy distribution can be derived using its inverse function. Additionally, random numbers that follow a uniform distribution can be generated using the inverse transformation method. The inverse function of the cumulative distribution function for the Cauchy distribution is:

$$H^{-1}(p; a, b) = a + b \cdot \tan\left(\pi \left(p - \frac{1}{2}\right)\right) \quad (27)$$

Let p be a $1 \times D$ random vector whose components are independently and uniformly distributed on $[0, 1]$:

$$p = \text{rand}(1, D) \quad (28)$$

In this work, $a = 0$ and $b = 0.01$ to avoid excessive jumps during the search process. (27) is adopted in this paper. Cauchy inverse cumulative distribution is introduced into AROA, and the data of some dimensions on the position vector can be randomly mutated so that the mutated position vector still retains the superior characteristics of the original position. The dimension decision logic determines whether a mutation occurs in each dimension:

$$k = \text{Judge}^D > \theta \quad (29)$$

Here, Judge^D is a D -dimensional random vector, θ is a threshold determining whether the current solution is modified, and k indicates which dimensions undergo mutation.

The dimension decision logic mutation strategy of the Cauchy inverse cumulative distribution is:

$$C = k \cdot H^{-1}(D) \quad (30)$$

where C is the mutation operator and H^{-1} is obtained from (27).

C. CT Mutation Operator

This subsection introduces a new mutation operator, termed the CT operator, which combines the tangent flight mutation operator with the dimension decision strategy based on the inverse cumulative function of the Cauchy distribution. After executing AROA, DE operations are applied to the population, and the current positions of the candidate solutions are stored as x_{old} . Subsequently, a new position x_{new} is generated by adding the CT operator to the current position x_{old} . Then, sort the fitness values of x_{old} and x_{new} in ascending order, and select the top N fitness values corresponding to x'_i as the final candidate solution. This process is defined as:

$$x_{new} = x_{old} + O \cdot C \quad (31)$$

$$x(t+1) = x'_i, \quad i = 1, \dots, N \quad (32)$$

where x_{old} is the old position after executing AROA and DE, x_{new} is the mutated position, x'_i ($i = 1, \dots, N$) are the top N positions after sorting, $x(t+1)$ is the candidate position at the $(t+1)$ -th iteration, O and C are given by (23) and (30), respectively.

D. Differential Evolution Algorithm

The Differential Evolution algorithm is a population-based stochastic search algorithm that simulates natural evolution through mutation, crossover, and selection, guiding the population toward global optima [48]. The main operations in DE are as follows.

1. Mutation Operation

Mutation in DE involves randomly selecting three distinct indices from the population of size N , corresponding to solution vectors X_{r1} , X_{r2} , and X_{r3} , to generate a new mutation solution V_i . Two common mutation operators are:

DE/best/1:

$$V_i = X_{best} + L(X_{r1} - X_{r2}) \quad (33)$$

DE/current-to-best/1:

$$V_i = X_i + L(X_{best} - X_{r1}) + L(X_{r2} - X_{r3}) \quad (34)$$

where X_{best} is the best solution vector, X_i represents the target

vector of the i -th individual at the t -th iteration, and L is a scaling factor in $[0, 1]$. This paper uses DE/best/1 to accelerate convergence while maintaining population diversity.

2. Crossover Operation

Crossover generates a crossover vector U_i from the mutant vector V_i and the target vector X_i :

$$U_{ij} = \begin{cases} V_{ij}, & \text{if } \text{rand}(0, 1) \leq CR \text{ or } j = j_{\text{rand}} \\ X_{ij}, & \text{otherwise} \end{cases} \quad (35)$$

where CR is the crossover rate, D is the problem dimension, and j_{rand} is a randomly chosen index from 1 to D . Here, U_{ij} , V_{ij} , and X_{ij} denote the j -th component of the crossover vector, mutation vector, and current target vector for the i -th individual at the t -th iteration, respectively.

3. Selection Operation

Selection chooses between X_i and U_i based on their fitness values $f(\cdot)$:

$$X_i(t+1) = \begin{cases} U_i, & \text{if } f(U_i) \leq f(X_i) \\ X_i, & \text{otherwise} \end{cases} \quad (36)$$

Here, $X_i(t+1)$ represents the solution vector of the i -th individual at the $(t+1)$ -th iteration.

E. Proposed DCTAROA

This subsection describes DCTAROA, an enhanced version of AROA that integrates the CT mutation operator and the mutation, crossover, and selection operations of DE.

After executing AROA, DE operations are applied to the population, and the current positions of the candidate solutions are stored as x_{old} . Subsequently, two search agents are randomly selected, and their difference is subjected to CT mutation, which combines the tangent flight mutation operator with the dimension decision strategy based on the inverse cumulative function of the Cauchy distribution. The mutated position is added to the current position to generate a new candidate x_{new} . Finally, the fitness values of the old and new candidates are evaluated, and the top N candidates are selected as the updated positions. The complete procedure of DCTAROA is shown in Algorithm 2.

IV. EXPERIMENT RESULTS AND DISCUSSION

A. Benchmark Functions and Experimental Setup

All experiments were conducted on a computer server with an Intel Core i7-8700 3.19GHz CPU and 16GB of memory, using MATLAB. The CEC2017 benchmark suite [49] provides a challenging test set that effectively simulates real-world optimization problems. It consists of 29 functions, categorized into four groups: unimodal (F1, F3), multimodal (F4-F10), hybrid (F11-F20), and composite (F21-F30). Note that F2 was excluded from the CEC2017 competition due to its unstable behavior [50].

To evaluate the proposed DCTAROA algorithm, we performed qualitative analysis, ablation experiment, key parameter analysis, quantitative analysis, convergence analysis, and significance analysis on the CEC2017 suite. The algorithm was also applied to three practical engineering design problems. For quantitative analysis, convergence analysis, and significance analysis, DCTAROA was tested against seven single optimization algorithms: AROA (2024), EGO (2024) [51], RIME (2023) [52], CDO (2023) [53], AVOA (2021) [54], GTO (2021) [55], and LFD (2020) [56]; and seven advanced hybrid optimization algorithms: FDB-SDO (2020) [57], DFDB-MRFO (2021) [33], FDB-SFS

Algorithm 2. Pseudo-code of DCTAROA

Require:

Population size N , fitness evaluations $f_{es_{max}}$, problem dimension D , search space bounds x_{max} and x_{min} , fitness function $f(\cdot)$.

Ensure:

Best solution x_{best} and its fitness f_{best} .

- 1: Calculate t_{max} using (9).
- 2: **for** $i = 1$ to N **do**
- 3: Initialize x_i using (2).
- 4: **end for**
- 5: **for** $i = 1$ to N **do**
- 6: Compute fitness $f(x_i)$
- 7: **end for**
- 8: Identify x_{best} and f_{best}
- 9: **for** $t = 1$ to t_{max} **do**
- 10: Compute distance matrix DM using (4)
- 11: Execute AROA steps (lines 11–34 of Algorithm 1).
- 12: Apply DE to generate x_{old} .
- 13: Apply CT to generate x_{new} .
- 14: Sort x_{old} and x_{new} by fitness, select top N as x'_i .
- 15: Update x_{best} and f_{best} .
- 16: **end for**
- 17: **return** x_{best} , f_{best}

(2021) [58], EESHHO (2021) [38], CTHHO (2022) [59], EAPSO (2023) [60], and QMESSA (2023) [39].

The maximum number of iterations was set to 500. The average value (*avg*) and standard deviation (*std*) of the best solution over 20 independent runs were recorded for quantitative analysis. Convergence behavior was analyzed using average convergence curves, and statistical significance was assessed with the Wilcoxon rank-sum test [61] at a 5% significance level. The parameters of all the comparison algorithms come from the original literature in the study.

B. Qualitative Evaluation

This subsection presents a qualitative analysis of DCTAROA regarding convergence behavior, population diversity, and the balance between exploration and exploitation. Representative functions from each CEC2017 class were selected for illustration: F3 (unimodal); F5, F8, F9 (multimodal); F12, F17 (hybrid); and F22, F26, F27 (composite).

1. Convergence Behavior Analysis

The convergence characteristics of DCTAROA were examined through the search trajectory of the agents, the convergence curve, the average fitness values of all individuals, and the trajectory of the first dimension. Fig. 1 and Fig. 2 illustrate these aspects.

The three-dimensional function plots (first column) show the smooth landscape of unimodal functions and the numerous local optima in multimodal and hybrid functions. The search history graphs (second column) indicate that agents explore promising regions early and converge near the global optimum later, demonstrating an effective balance between exploration and exploitation. For example, in F26, DCTAROA consistently focused on the left region of the solution space, leading to optimal results.

The average fitness history (third column) shows a decreasing trend, and the first-dimension trajectory (fourth column) transitions from fluctuation to stability, reflecting the shift from global exploration to local exploitation. The convergence curves (fifth column) confirm that DCTAROA achieves fast convergence speed and high solution accuracy across different function types.

2. Population Diversity Analysis

Population diversity is crucial for avoiding premature convergence in meta-heuristic algorithms [62]. We compared the diversity of AROA and DCTAROA using the inertial matrix I_c (with its dimension D_{max} set to 100), defined as:

$$I_c(t) = \sqrt{\sum_{i=1}^N \sum_{D=1}^{D_{max}} (x_{iD}(t) - c_D(t))^2} \quad (37)$$

where $c_D(t)$ is the dispersion of the population from its mass center c in each iteration:

$$c_D(t) = \frac{1}{N} \sum_{i=1}^N x_{iD}(t) \quad (38)$$

Here, $x_{iD}(t)$ represents the value of D -th dimension of the i -th search agent at the t -th iteration.

The results in Fig. 3 show that DCTAROA maintains higher population diversity than AROA throughout the iteration process. This indicates a wider exploration of the search space, reducing the risk of stagnation in local optima and enhancing the potential for finding global solutions.

3. Exploration and Exploitation Analysis

The balance between exploration (global search) and exploitation (local search) is vital for meta-heuristic performance. We quantified this balance using the following metrics [63]:

$$Exploration(\%) = \frac{Div(t)}{Div_{max}} \times 100 \quad (39)$$

$$Exploitation(\%) = \frac{|Div(t) - Div_{max}|}{Div_{max}} \times 100 \quad (40)$$

Here, $Exploration(\%)$ and $Exploitation(\%)$ represent the percentage of global search and local search activity, respectively, during the optimization process. Div_{max} denotes the maximum diversity observed throughout the entire iteration process [63], where $Div(t)$ represents the dimension-wise diversity measurement:

$$Div(t) = \frac{1}{D_{max}} \sum_{D=1}^{D_{max}} \frac{1}{N} \sum_{i=1}^N |median(x_D(t)) - x_{iD}(t)| \quad (41)$$

The exploration-exploitation balance is visualized in Fig. 4. Although the transition between the two phases occurs at different points for various test functions, DCTAROA consistently maintains a higher exploration rate in early iterations and gradually increases exploitation toward later stages. This dynamic balance helps the algorithm avoid premature convergence and local optima entrapment across different optimization landscapes.

C. Ablation Experiment

To evaluate the individual contributions of the differential evolution (DE) operator, tangent flight mutation (Tan), and Cauchy distribution-based dimension decision strategy (Cau), we conducted an ablation study. Eight algorithm variants were tested on the 29 CEC2017 functions: AROA (baseline), DEAROA (AROA+DE), TanAROA (AROA+Tan), CauAROA (AROA+Cau), CTAROA (AROA+CT), DCAROA (AROA+DE+Cau), DTAROA (AROA+DE+Tan), and DCTAROA (AROA+DE+CT).

Each algorithm was run 20 times with a population size of 30 and dimension 30. The average (*avg*) and standard deviation (*std*) of the best solutions are summarized in Table I. The final ranking, based on the average performance across all functions, places DCTAROA first, followed by DCAROA. These results confirm that integrating all three components yields the most effective optimizer.

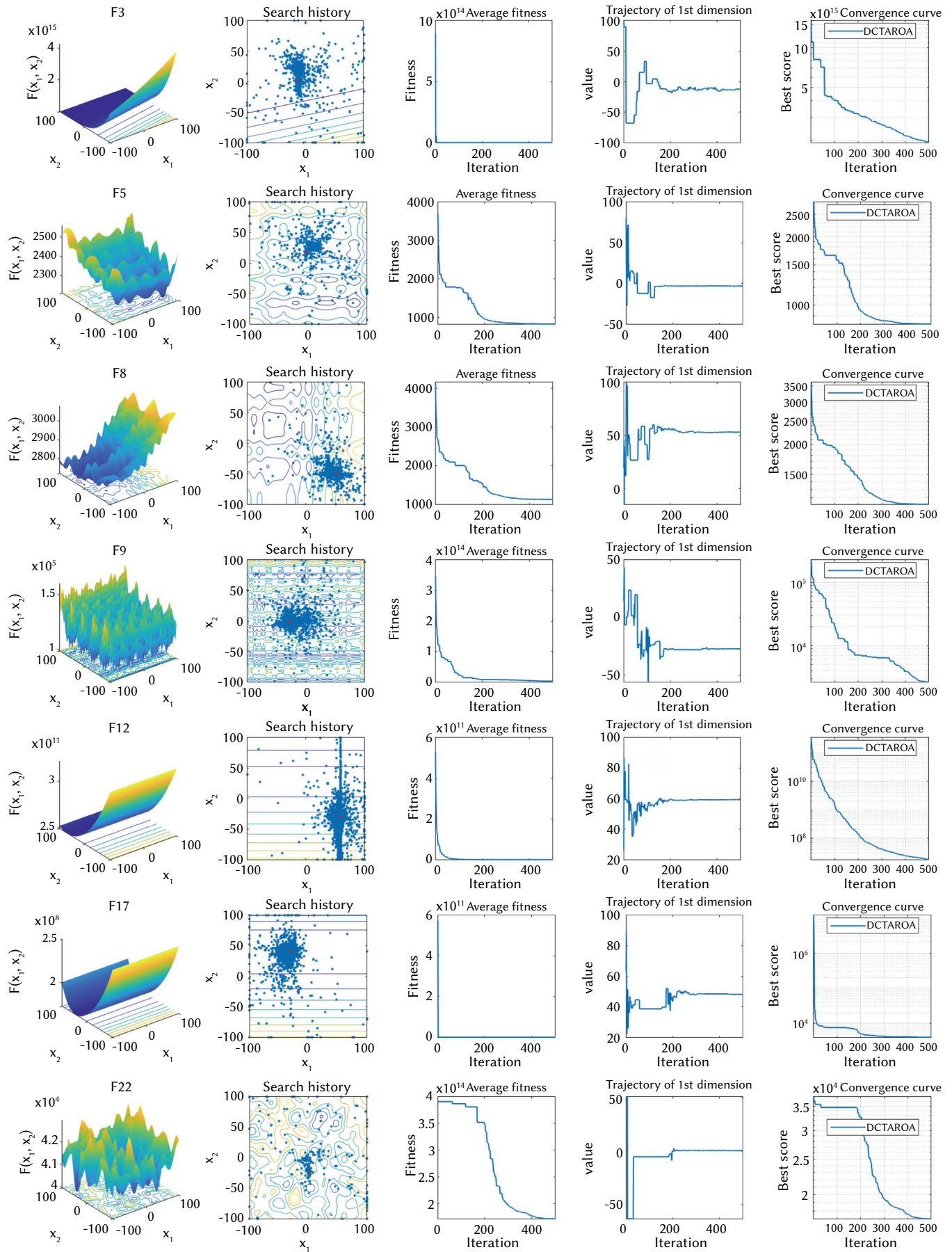


Fig. 1. Convergence behavior of DCTAROA on selected CEC2017 functions (F3, F5, F8, F9, F12, F17, F22).

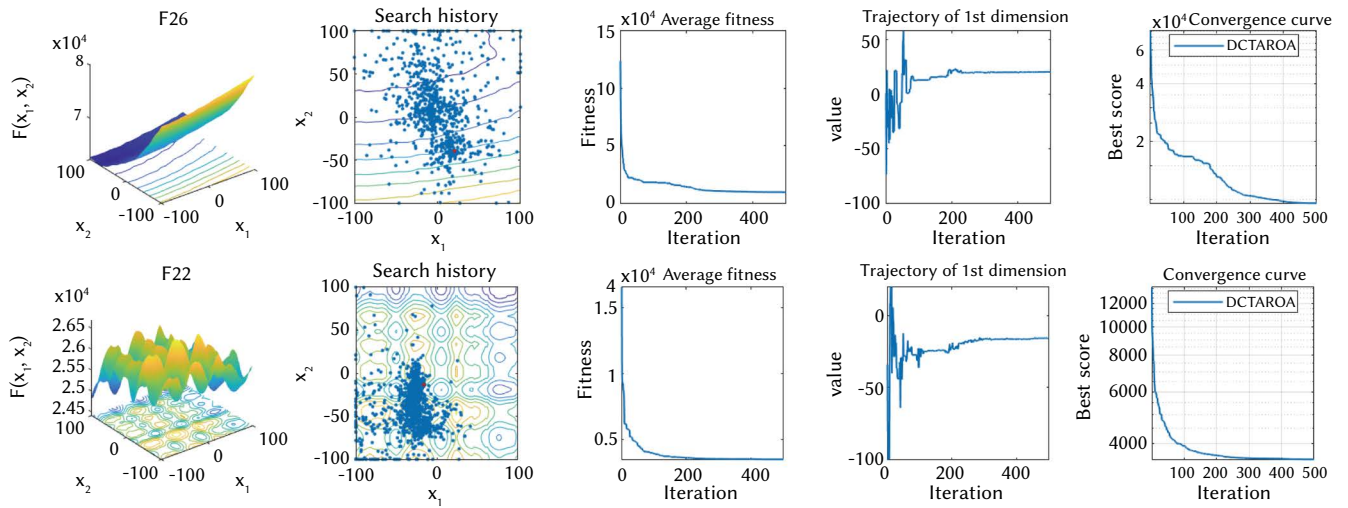


Fig. 2. Convergence behavior of DCTAROA on selected CEC2017 functions (F26, F27).

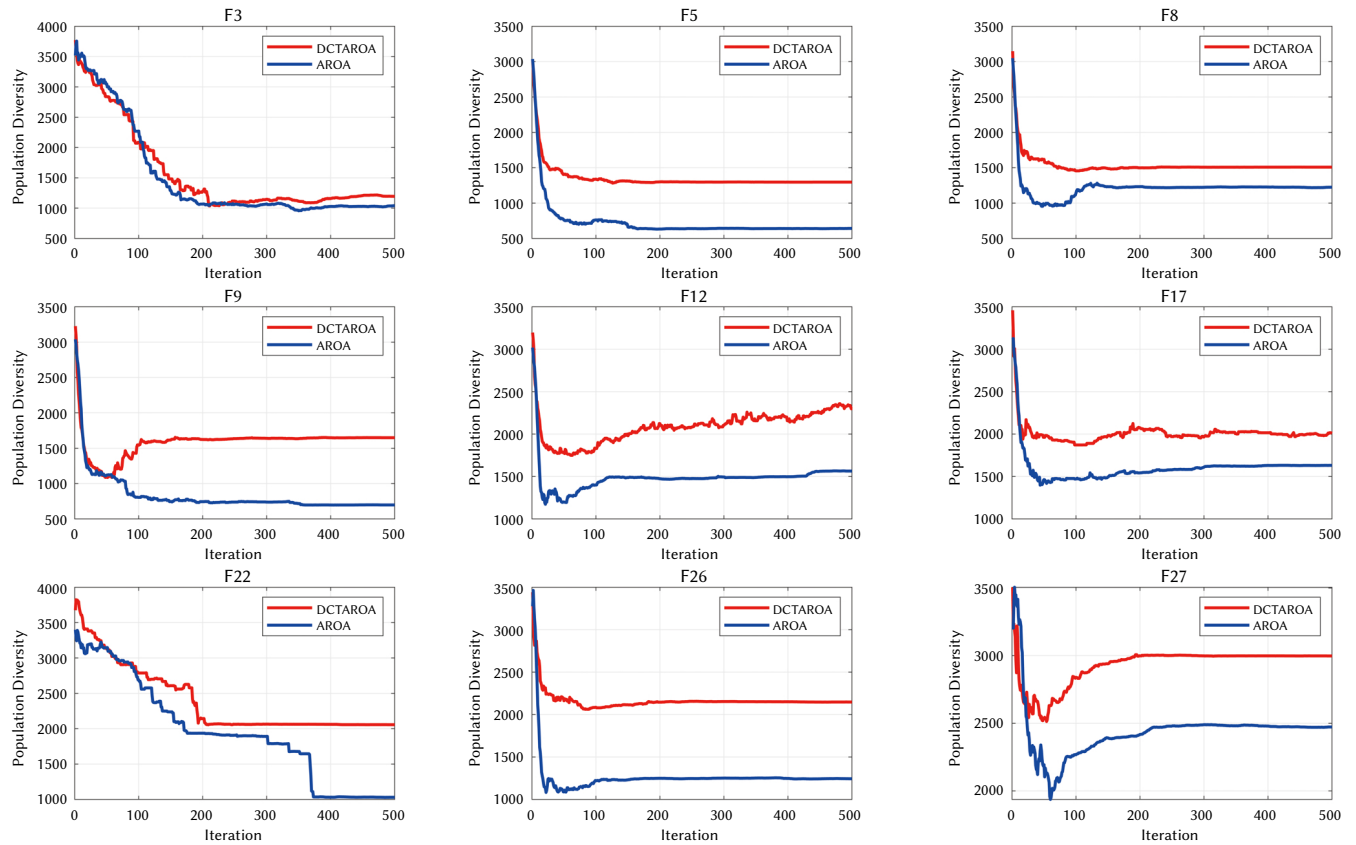


Fig. 3. Population diversity comparison between AROA and DCTAROA.

D. Quantitative Analysis

1. Comparison With Single Optimization Algorithms

DCTAROA was compared with seven single optimization algorithms (AROA, EGO, RIME, CDO, AVOA, GTO, LFD) on the CEC2017 suite at dimensions 30, 50, and 100. Detailed results (*avg* and *std* for all 29 functions) are provided in Tables II-IV.

Overall, DCTAROA achieved the best average ranking (rank 1) across all dimensions. In terms of the number of functions where it obtained the best *avg* value, DCTAROA outperformed others: 25/29 (30D), 26/29 (50D), and 24/29 (100D). Regarding solution stability

(*std*), DCTAROA also provided the best *std* value for the majority of functions: 21/29 (30D), 13/29 (50D), and 13/29 (100D). These results demonstrate DCTAROA's superior optimization performance and robustness across varying problem dimensions.

2. Comparison With Advanced Hybrid Optimization Algorithms

DCTAROA was further compared with seven advanced hybrid algorithms (FDB-SDO, DFDB-MRFO, FDB-SFS, EESHHO, CTHHO, EAPSO, QMESSA). Comprehensive results for dimensions 30, 50, and 100 are presented in Tables V-VII.

DCTAROA again secured the top average rank (1) in all dimensional settings. It obtained the best *avg* value for 26/29 (30D), 27/29 (50D), and

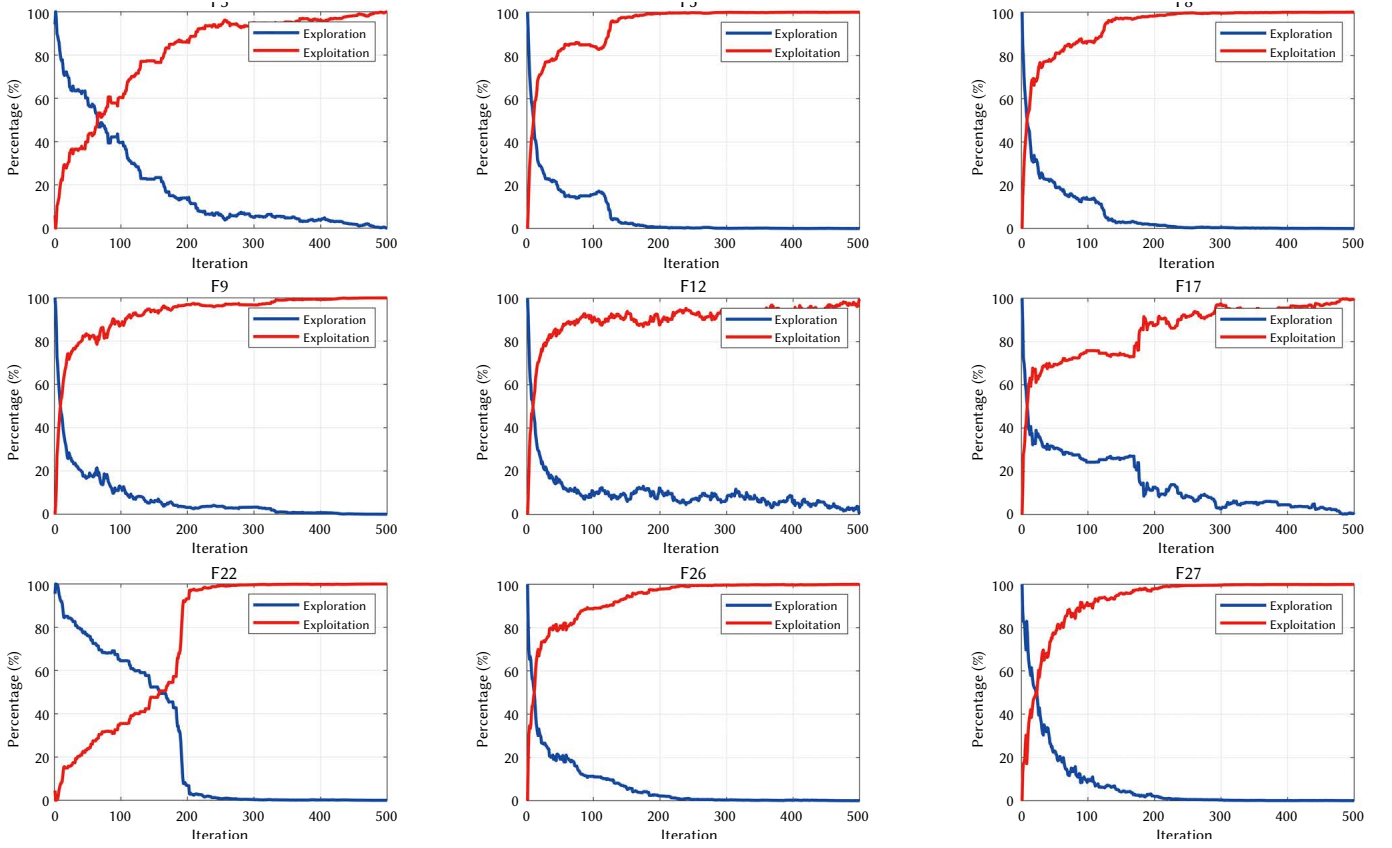


Fig. 4. Exploration and exploitation percentages of DCTAROA.

TABLE I. RESULTS OF ABLATION ALGORITHMS

Criteria	DCTAROA	DCAROA	DTAROA	CTAROA	CauAROA	TanAROA	DEAROA	AROA
# Best Avg ($\uparrow$)	11	5	4	0	0	1	7	1
# Best Std ($\downarrow$)	8	5	5	0	0	1	7	3
Avg Rank ($\downarrow$)	2.28	2.90	2.93	6.76	6.83	6.31	2.97	5.03
Final Rank	1	2	3	7	8	6	4	5

$\uparrow$: higher is better; $\downarrow$: lower is better. # Best Avg: number of functions where the algorithm achieves the highest mean value. # Best Std: number of functions where the algorithm achieves the lowest standard deviation. Avg Rank: average ranking across all 29 functions based on mean values.

TABLE II. COMPARISON OF OPTIMIZATION RESULTS OF EIGHT SINGLE OPTIMIZATION ALGORITHMS OBTAINED FOR THE 30-DIMENSIONAL OF CEC2017 TEST SUITE

Criteria	DCTAROA	AROA	EGO	RIME	CDO	AVOA	GTO	LFD
# Best Avg ($\uparrow$)	25	3	1	0	0	0	0	0
# Best Std ($\downarrow$)	21	4	3	0	1	0	0	0
Avg Rank ($\downarrow$)	1.14	1.93	7.10	3.66	7.41	4.97	3.59	6.21
Final Rank	1	2	7	4	8	5	3	6

$\uparrow$: higher is better; $\downarrow$: lower is better. # Best Avg: number of functions where the algorithm achieves the highest mean value. # Best Std: number of functions where the algorithm achieves the lowest standard deviation. Avg Rank: average ranking across all 29 functions based on mean values.

26/29 (100D) functions. In terms of *std*, it ranked first for 24/29 (30D), 22/29 (50D), and 16/29 (100D) functions, confirming its consistent superiority over state-of-the-art hybrid optimizers.

E. Convergence Analysis

1. Comparison With Single Optimization Algorithms

Average convergence curves for representative functions (F1, F5, F8, F9, F12, F15, F20, F23, F24, F30) at dimensions 30, 50, and 100 are shown in Fig. 5-7 (Appendix). DCTAROA consistently demonstrates faster convergence speed and reaches lower fitness values compared to all seven single-algorithm competitors (AROA, EGO, RIME, CDO, AVOA, GTO, LFD). This indicates more efficient exploration and exploitation throughout the search process.

2. Comparison With Advanced Hybrid Optimization Algorithms

Convergence curves comparing DCTAROA with seven advanced hybrid algorithms on the same representative functions are presented in Fig. 8-10 (Appendix). DCTAROA maintains a competitive edge, exhibiting faster initial descent and superior final accuracy across all dimensions, which underscores its enhanced search capabilities relative to other high-performance hybrid methods.

F. Significance Analysis

1. Comparison With Single Optimization Algorithms

The Wilcoxon rank-sum test (5% significance level) was used to assess the statistical significance of performance differences between DCTAROA and each single optimization algorithm. The *p*-values

TABLE III. COMPARISON OF OPTIMIZATION RESULTS OF EIGHT SINGLE OPTIMIZATION ALGORITHMS OBTAINED FOR THE 50-DIMENSIONAL OF CEC2017 TEST SUITE

Criteria	DCTAROA	AROA	EGO	RIME	CDO	AVOA	GTO	LFD
# Best Avg (↑)	26	1	2	0	0	0	0	0
# Best Std (↓)	13	9	3	0	4	0	0	0
Avg Rank (↓)	1.10	2.03	6.83	3.69	7.55	4.83	3.79	6.17
Final Rank	1	2	7	4	8	5	3	6

↑: higher is better; ↓: lower is better. # Best Avg: number of functions where the algorithm achieves the highest mean value. # Best Std: number of functions where the algorithm achieves the lowest standard deviation. Avg Rank: average ranking across all 29 functions based on mean values.

TABLE IV. COMPARISON OF OPTIMIZATION RESULTS OF EIGHT SINGLE OPTIMIZATION ALGORITHMS OBTAINED FOR THE 100-DIMENSIONAL OF CEC2017 TEST SUITE

Criteria	DCTAROA	AROA	EGO	RIME	CDO	AVOA	GTO	LFD
# Best Avg (↑)	24	3	2	0	0	0	0	0
# Best Std (↓)	13	6	3	0	6	0	1	0
Avg Rank (↓)	1.17	1.97	7.03	4.07	7.17	4.59	3.72	6.28
Final Rank	1	2	7	4	8	5	3	6

↑: higher is better; ↓: lower is better. # Best Avg: number of functions where the algorithm achieves the highest mean value. # Best Std: number of functions where the algorithm achieves the lowest standard deviation. Avg Rank: average ranking across all 29 functions based on mean values.

TABLE V. COMPARISON OF OPTIMIZATION RESULTS OF EIGHT ADVANCED OPTIMIZATION ALGORITHMS OBTAINED FOR THE 30-DIMENSIONAL OF CEC2017 TEST SUITE

Criteria	DCTAROA	FDB-SDO	DFDB-MRFO	FDB-SFS	EESHHO	CTHHO	EAPSO	QMESSA
# Best Avg (↑)	26	0	1	1	0	0	0	1
# Best Std (↓)	24	0	2	0	1	0	0	2
Avg Rank (↓)	1.31	3.69	3.90	2.62	6.41	7.48	4.62	5.97
Final Rank	1	3	4	2	7	8	5	6

↑: higher is better; ↓: lower is better. # Best Avg: number of functions where the algorithm achieves the highest mean value. # Best Std: number of functions where the algorithm achieves the lowest standard deviation. Avg Rank: average ranking across all 29 functions based on mean values.

TABLE VI. COMPARISON OF OPTIMIZATION RESULTS OF EIGHT ADVANCED OPTIMIZATION ALGORITHMS OBTAINED FOR THE 50-DIMENSIONAL OF CEC2017 TEST SUITE

Criteria	DCTAROA	FDB-SDO	DFDB-MRFO	FDB-SFS	EESHHO	CTHHO	EAPSO	QMESSA
# Best Avg (↑)	27	0	1	1	0	0	0	0
# Best Std (↓)	22	0	2	2	1	0	0	2
Avg Rank (↓)	1.10	4.62	3.93	2.55	6.55	7.14	3.79	6.31
Final Rank	1	5	4	2	7	8	3	6

↑: higher is better; ↓: lower is better. # Best Avg: number of functions where the algorithm achieves the highest mean value. # Best Std: number of functions where the algorithm achieves the lowest standard deviation. Avg Rank: average ranking across all 29 functions based on mean values.

TABLE VII. COMPARISON OF OPTIMIZATION RESULTS OF EIGHT ADVANCED OPTIMIZATION ALGORITHMS OBTAINED FOR THE 100-DIMENSIONAL OF CEC2017 TEST SUITE

Criteria	DCTAROA	FDB-SDO	DFDB-MRFO	FDB-SFS	EESHHO	CTHHO	EAPSO	QMESSA
# Best Avg (↑)	26	0	2	1	0	0	0	0
# Best Std (↓)	16	0	3	5	1	2	1	1
Avg Rank (↓)	1.14	5.24	3.38	3.14	6.52	6.69	3.66	6.24
Final Rank	1	5	3	2	7	8	4	6

↑: higher is better; ↓: lower is better. # Best Avg: number of functions where the algorithm achieves the highest mean value. # Best Std: number of functions where the algorithm achieves the lowest standard deviation. Avg Rank: average ranking across all 29 functions based on mean values.

for all 29 functions across dimensions 30, 50, and 100 are reported in Tables XIII- XV (Appendix).

A p -value ≤ 0.05 indicates a significant difference. For 30 dimensions, DCTAROA showed significantly better performance than AROA on 21 out of 29 functions. Against all other single algorithms (EGO, RIME, CDO, AVOA, GTO, LFD), DCTAROA's superiority was statistically significant for nearly all functions. This trend held and became even more pronounced at higher dimensions (50 and 100). The results provide strong statistical evidence that DCTAROA outperforms the compared single optimizers.

2. Comparison With Advanced Hybrid Optimization Algorithms

Similarly, Wilcoxon test results comparing DCTAROA with the seven advanced hybrid algorithms are detailed in Tables XVI-XVIII

(Appendix). The number of functions with non-significant differences ($p > 0.05$) was low: 16/203 (30D), 12/203 (50D), and 7/203 (100D). For the vast majority of comparisons, DCTAROA's performance advantage was statistically significant. As dimensionality increased, the significance of these differences further strengthened, confirming DCTAROA's robust competitiveness against advanced hybrid optimizers.

G. Parameter Sensitivity Analysis

The dimension decision strategy introduced in (29) involves a key probability factor θ . To determine its optimal setting, we tested DCTAROA with θ values ranging from 0 to 0.9 (in steps of 0.1), denoted as DCTAROA0 to DCTAROA9. The experimental settings are listed in Table VIII. The average rankings across all CEC2017 functions (Table IX) show that DCTAROA7 ($\theta = 0.7$) achieved the highest rank. Therefore, $\theta = 0.7$ was adopted for all other experiments in this study.

TABLE VIII. EXPERIMENTAL DESIGN FOR THE PROBABILITY FACTOR PARAMETER θ

Function	θ	D	t_{max}	N	Repeat times
DCTAROA0	0	30	500	30	20
DCTAROA1	0.1	30	500	30	20
DCTAROA2	0.2	30	500	30	20
DCTAROA3	0.3	30	500	30	20
DCTAROA4	0.4	30	500	30	20
DCTAROA5	0.5	30	500	30	20
DCTAROA6	0.6	30	500	30	20
DCTAROA7	0.7	30	500	30	20
DCTAROA8	0.8	30	500	30	20
DCTAROA9	0.9	30	500	30	20

TABLE IX. AVERAGE RANKING VALUE

Metric	DCTAROA0	DCTAROA1	DCTAROA2	DCTAROA3	DCTAROA4	DCTAROA5	DCTAROA6	DCTAROA7	DCTAROA8	DCTAROA9
# Best Avg ($\uparrow$)	0	1	2	3	2	2	2	5	0	2
# Best Std ($\downarrow$)	3	1	2	3	1	1	4	2	2	2
Avg Rank ($\downarrow$)	5.72	6.00	5.34	5.28	5.34	5.17	5.34	4.55	6.07	6.17
Final Rank	7	8	4	3	4	2	4	1	9	10

$\uparrow$: higher is better; $\downarrow$: lower is better. # Best Avg: number of functions where the algorithm achieves the highest mean value. # Best Std: number of functions where the algorithm achieves the lowest standard deviation. Avg Rank: average ranking across all 29 functions based on mean values.

TABLE X. RESULTS OF DCTAROA VERSUS OTHER MAs IN THE SPRING DESIGN PROBLEM

Algorithm	Optimum variables			Optimum weight
	d	D	N	
DCTAROA	5.1685E-02	3.5662E-01	1.1295E+01	1.2665238E-02
AROA	5.1566E-02	3.5388E-01	1.1493E+01	1.2670217E-02
AOA	5.2903E-02	3.8579E-01	1.3806E+01	1.5117860E-02
RIME	6.6212E-02	8.3217E-01	3.3821E+00	1.6894848E-02
CDO	5.1210E-02	3.4424E-01	1.3120E+01	1.3161161E-02
AVOA	5.5129E-02	4.5576E-01	8.7465E+00	1.3209919E-02
GTO	5.0905E-02	3.3932E-01	1.2737E+01	1.2721412E-02
HHO	5.9962E-02	6.0099E-01	5.1587E+00	1.4165056E-02

H. Experiments on Practical Engineering Problems

The effectiveness of DCTAROA was further validated on three constrained engineering design problems: tension/compression spring design, welded beam design, and five-stage cantilever beam design. The mathematical formulations of these problems are provided in the Appendix.

1. Tension/Compression Spring Design Problem

The main objective of the Tension/compression spring design problem is to optimize the coil diameter (D), the wire diameter (d), and the number of coils (N) to minimize the weight of the tension/compression spring. Table X compares the best solutions found by DCTAROA and seven other algorithms. DCTAROA achieved the lowest weight (1.2665238E-02), demonstrating its capability in handling constrained optimization.

2. Welded Beam Design Problem

The objective is to minimize the fabrication cost of a welded beam by optimizing four variables: weld thickness (h), length (l), height (t), and rebar thickness (b). As shown in Table XI, DCTAROA obtained the lowest cost (1.6927683E+00), outperforming all other compared algorithms.

3. Five-Stage Cantilever Beam Design Problem

The third engineering design problem addressed in this investigation is the cantilever beam design problem, which focuses on minimizing production costs. This problem involves five hollow square blocks of constant thickness, and height (or width), namely x_1, x_2, x_3, x_4 , and x_5 . The optimization results in Table XII show that DCTAROA found the

lightest design (weight = 1.3399564E+00), confirming its effectiveness in solving this type of engineering problem.

V. CONCLUSIONS AND FUTURE DIRECTIONS

In this work, an improved attraction-repulsion optimization algorithm, termed DCTAROA, is proposed. The integration of differential evolution (DE) and attraction-repulsion optimization (AROA) enables a smooth transition between exploration and exploitation. The proposed mutation operator enhances population diversity and spatial search capability, thereby increasing the likelihood of reaching the global optimum and improving the overall performance of DCTAROA.

DCTAROA was evaluated against seven single optimization algorithms (AROA, EGO, RIME, CDO, AVOA, GTO, LFD) and seven advanced hybrid algorithms (FDB-SDO, DFDB-MRFO, FDB-SFS, EESHOO, CTHHO, EAPSO, QMESSA) using 29 IEEE benchmark functions. Performance was assessed in terms of average results, standard deviations, and statistical rankings.

The experimental results demonstrate that DCTAROA achieves superior performance compared to all tested algorithms. It attained the highest average ranking and the lowest standard deviation of the best solution across 30-, 50-, and 100-dimensional problems.

In comparison with single optimization algorithms, DCTAROA obtained the best average results for the majority of test functions. Similarly, when compared with advanced hybrid algorithms, DCTAROA consistently outperformed the alternatives across all three dimensions. Statistical tests confirm that these differences are significant.

TABLE XI. OPTIMIZATION RESULTS OF DCTAROA VERSUS OTHER MAs IN THE WELDED BEAM DESIGN PROBLEM

Algorithm	Optimum variables				Optimum weight
	h	l	t	b	
DCTAROA	2.0573E-01	3.2349E+00	9.0366E+00	2.0573E-01	1.6927683E+00
AROA	2.0572E-01	3.2347E+00	9.0377E+00	2.0573E-01	1.6928945E+00
AOA	2.0096E-01	5.2511E+00	9.5382E+00	2.4830E-01	2.3181602E+00
PSO	2.1375E-01	3.1653E+00	8.8918E+00	2.1377E-01	1.7219111E+00
CDO	1.9228E-01	3.5233E+00	9.4736E+00	2.0928E-01	1.8120559E+00
AVOA	1.9341E-01	3.6004E+00	9.0505E+00	2.0715E-01	1.7319088E+00
GTO	2.0975E-01	3.1786E+00	9.0037E+00	2.0975E-01	1.7128308E+00
HHO	2.1195E-01	4.3673E+00	8.5272E+00	2.5793E-01	2.0238356E+00

TABLE XII. CALCULATIONS OF DCTAROA VERSUS OTHER MAs IN CANTILEVER BEAM DESIGN PROBLEM

Algorithm	Optimum variables					Optimum weight
	x_1	x_2	x_3	x_4	x_5	
DCTAROA	6.0160E+00	5.3092E+00	4.4943E+00	3.5015E+00	2.1527E+00	1.3399564E+00
AROA	6.0249E+00	5.3078E+00	4.4943E+00	3.4954E+00	2.1517E+00	1.3399897E+00
AOA	8.8638E+00	9.3093E+00	1.1480E+01	5.9319E+00	8.4927E+00	2.7504522E+00
PSO	6.0143E+00	5.3062E+00	4.4968E+00	3.5049E+00	2.1519E+00	1.3399821E+00
CDO	6.0713E+00	5.4328E+00	4.5790E+00	3.5026E+00	2.2337E+00	1.3615215E+00
AVOA	5.9702E+00	5.3288E+00	4.5040E+00	3.5117E+00	2.1665E+00	1.3404234E+00
GTO	6.0181E+00	5.3052E+00	4.4943E+00	3.4971E+00	2.1600E+00	1.3400175E+00
HHO	5.9690E+00	5.2867E+00	4.5446E+00	3.5270E+00	2.2168E+00	1.3443511E+00

A balance and diversity analysis was conducted on selected functions (e.g., F3, F5, F8, F9, F12, F17, F22, F26, F27). The results indicate that DCTAROA maintains a dynamic equilibrium between exploration and exploitation, which aids in avoiding local optima and premature convergence.

DCTAROA was also applied to three engineering design problems: tension/compression spring design (optimal weight: 0.012665238), welded beam design (optimal cost: 1.6927683), and a five-stage cantilever beam design (optimal weight: 1.3399564). In all cases, DCTAROA produced competitive or superior solutions compared to classical heuristic algorithms, confirming its practical applicability and potential for cost reduction.

Despite its competitive performance across benchmark functions and engineering design problems, DCTAROA has several limitations. First, the integration of differential evolution with attraction-repulsion optimization introduces additional computational overhead, particularly for high-dimensional problems (e.g., 100 dimensions), where the mutation and selection steps increase the per-iteration complexity. Second, the performance of DCTAROA is sensitive to the control parameters of both DE (scaling factor and crossover rate) and AROA (attraction and repulsion coefficients); suboptimal parameter tuning may degrade convergence behavior. These limitations highlight opportunities for further improvement and should be considered when applying DCTAROA to new problem domains.

Future work will focus on incorporating advanced strategies such as adversarial learning, federated learning [64], [65], and lens-imaging learning to further enhance convergence speed and search accuracy. Additionally, a binary version of DCTAROA will be developed for applications in imbalanced classification, medical diagnosis, network security [66]–[68], and feature selection.

APPENDIX

The appendix contains the comprehensive numerical results, and problem formulations in the main text. All figures and tables below are provided as supporting material.

A. Engineering Problem Formulations

1. Tension/Compression Spring Design Problem

Consider:

$$\vec{x} = [x_1, x_2, x_3] = [d, D, N]$$

Minimize:

$$f(\vec{x}) = (x_3 + 2)x_2x_1^2$$

Subject to:

$$g_1(\vec{x}) = 1 - \frac{x_3^3x_2}{71785x_1^4} \leq 0$$

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5180x_1^2} \leq 0$$

$$g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

Variable range:

$$0.05 \leq x_1 \leq 2.00$$

$$0.25 \leq x_2 \leq 1.30$$

$$2.00 \leq x_3 \leq 15.0$$

2. Welded Beam Design Problem

Consider:

$$\vec{x} = [x_1, x_2, x_3, x_4] = [h, l, t, b]$$

Minimize:

$$f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

Subject to:

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \leq 0$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \leq 0$$

TABLE XIII. RESULTS OF WILCOXON RANK-SUM TEST BETWEEN DCTAROA AND SINGLE OPTIMIZATION ALGORITHMS ON CEC2017 TEST FUNCTION FOR 30-DIMENSIONAL (APPENDIX)

	AROA	EGO	RIME	CDO	AVOA	GTO	LFD
F1	6.796E-08	6.796E-08	6.796E-08	6.796E-08	2.218E-07	6.796E-08	6.796E-08
F3	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F4	5.075E-01	6.796E-08	1.600E-05	6.796E-08	6.917E-07	4.166E-05	6.796E-08
F5	9.127E-07	6.796E-08	1.065E-07	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F6	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F7	5.874E-06	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F8	9.278E-05	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F9	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F10	3.235E-01	6.796E-08	5.629E-04	6.796E-08	1.918E-07	1.235E-07	6.796E-08
F11	3.382E-04	6.796E-08	6.796E-08	6.796E-08	6.796E-08	1.065E-07	6.796E-08
F12	9.173E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	2.960E-07	6.796E-08
F13	2.503E-01	6.796E-08	9.173E-08	6.796E-08	5.227E-07	4.388E-02	6.796E-08
F14	9.173E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F15	1.997E-04	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F16	1.625E-03	6.796E-08	1.918E-07	6.796E-08	9.173E-08	1.803E-06	6.796E-08
F17	3.235E-01	6.796E-08	6.015E-07	6.796E-08	1.431E-07	1.235E-07	1.431E-07
F18	3.793E-01	6.796E-08	6.796E-08	6.796E-08	1.235E-07	7.577E-06	6.796E-08
F19	6.868E-04	6.796E-08	3.939E-07	6.796E-08	7.898E-08	1.201E-06	6.796E-08
F20	3.057E-03	6.796E-08	1.235E-07	6.796E-08	6.796E-08	2.218E-07	6.796E-08
F21	3.705E-05	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F22	2.733E-01	6.796E-08	4.166E-05	7.898E-08	8.597E-06	1.058E-02	3.416E-07
F23	6.040E-03	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F24	3.057E-03	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F25	1.349E-03	6.796E-08	6.796E-08	6.796E-08	6.796E-08	8.597E-06	6.796E-08
F26	1.075E-01	6.796E-08	1.201E-06	6.796E-08	1.600E-05	1.014E-03	6.796E-08
F27	4.680E-05	7.113E-03	9.173E-08	6.796E-08	6.796E-08	6.796E-08	1.058E-02
F28	7.557E-01	6.796E-08	6.015E-07	6.796E-08	6.796E-08	5.227E-07	6.796E-08
F29	3.966E-03	6.796E-08	7.898E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F30	5.609E-01	6.796E-08	6.796E-08	6.796E-08	6.796E-08	2.925E-05	6.796E-08

$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \leq 0$$

$$g_4(\vec{x}) = x_1 - x_4 \leq 0$$

$$g_5(\vec{x}) = P - P_C(\vec{x}) \leq 0$$

$$g_6(\vec{x}) = 0.125 - x_1 \leq 0$$

$$g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0$$

Variable range:

$$0.1 \leq x_1 \leq 2, 0.1 \leq x_2 \leq 10, 0.1 \leq x_3 \leq 10, 0.1 \leq x_4 \leq 2$$

where

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \tau' = \frac{P}{\sqrt{2}x_1x_2}$$

$$\tau'' = \frac{MR}{J}, M = P(L + \frac{x_2}{2})$$

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2}$$

$$J = 2 \left\{ \sqrt{2}x_1x_2 \left[\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2 \right] \right\}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4x_3^3}, \delta(\vec{x}) = \frac{6PL^3}{Ex_3^3x_4}$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right)$$

$$P = 60001b, L = 14 \in \dots \delta_{max} = 0.25 \in \dots$$

$$E = 30 \times 10^6 psi, G = 12 \times 10^6 psi$$

$$\tau_{max} = 13600 psi, \sigma_{max} = 30000 psi$$

3. Five-stage Cantilever Beam Design Problem

Consider:

$$\vec{x} = [x_1, x_2, x_3, x_4, x_5]$$

Minimize:

$$f(\vec{x}) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5)$$

Subject to:

$$g(\vec{x}) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0$$

Variable range:

$$0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$$

CREDiT AUTHORSHIP CONTRIBUTION STATEMENT

Fang Feng: Conceptualization, Investigation, Software, Validation, Methodology, Writing-original draft, Writing-review & editing, Funding acquisition, Supervision.

Mingjiang Cai: Writing-review & editing.

Hung-Wei Li: Writing-review & editing.

Kuan-Ching Li: Supervision, Writing-review & editing.

DATA STATEMENT

All data sets used are publicly available <https://github.com/P-N-Suganthan/CEC2017-BoundConstrained>

TABLE XIV. RESULTS OF WILCOXON RANK-SUM TEST BETWEEN DCTAROA AND SINGLE OPTIMIZATION ALGORITHMS ON CEC2017 TEST FUNCTION FOR 50-DIMENSIONAL (APPENDIX)

	ARO	EGO	RIME	CDO	AVOA	GTO	LFD
F1	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F3	1.431E-07	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F4	2.564E-02	6.796E-08	1.657E-07	6.796E-08	6.796E-08	7.898E-08	6.796E-08
F5	3.939E-07	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F6	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F7	5.227E-07	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F8	8.597E-06	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F9	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F10	6.168E-01	6.796E-08	5.874E-06	6.796E-08	2.062E-06	1.600E-05	6.796E-08
F11	7.898E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F12	5.874E-06	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F13	2.977E-01	6.796E-08	6.796E-08	6.796E-08	6.796E-08	5.227E-07	6.796E-08
F14	2.733E-01	6.796E-08	1.065E-07	6.796E-08	6.796E-08	7.406E-05	6.796E-08
F15	4.570E-01	6.796E-08	6.796E-08	6.796E-08	1.235E-07	6.868E-04	6.796E-08
F16	2.503E-01	6.796E-08	1.807E-05	6.796E-08	2.563E-07	1.415E-05	7.898E-08
F17	4.679E-02	6.796E-08	1.576E-06	6.796E-08	1.918E-07	2.960E-07	9.173E-08
F18	8.103E-02	6.796E-08	6.796E-08	6.796E-08	6.796E-08	3.048E-04	6.796E-08
F19	7.643E-02	6.796E-08	1.065E-07	6.796E-08	1.657E-07	7.712E-03	6.796E-08
F20	1.719E-01	6.796E-08	1.803E-06	6.796E-08	1.918E-07	7.577E-06	6.796E-08
F21	1.116E-03	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F22	6.168E-01	6.796E-08	6.917E-07	6.796E-08	6.796E-08	3.069E-06	6.796E-08
F23	6.917E-07	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F24	5.896E-05	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F25	3.987E-06	6.796E-08	1.431E-07	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F26	3.605E-02	6.796E-08	6.796E-08	6.796E-08	3.416E-07	1.415E-05	6.796E-08
F27	3.987E-06	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	1.201E-06
F28	4.539E-07	6.040E-03	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.015E-07
F29	2.393E-01	6.796E-08	7.898E-08	6.796E-08	6.796E-08	7.898E-08	6.796E-08
F30	3.372E-02	6.796E-08	6.796E-08	6.796E-08	6.796E-08	1.201E-06	6.796E-08

TABLE XV. RESULTS OF WILCOXON RANK-SUM TEST BETWEEN DCTAROA AND SINGLE OPTIMIZATION ALGORITHMS ON CEC2017 TEST FUNCTION FOR 100-DIMENSIONAL (APPENDIX)

[illegible]

TABLE XVI. RESULTS OF WILCOXON RANK-SUM TEST BETWEEN DCTAROA AND ADVANCED OPTIMIZATION ALGORITHMS ON CEC2017 TEST FUNCTION FOR 30-DIMENSIONAL (APPENDIX)

	FDB-SDO	DFDB-MRFO	FDB-SFS	EESHHO	CTHHO	EAPSO	QMESSA
F1	6.796E-08	2.616E-01	9.209E-04	6.796E-08	6.796E-08	1.953E-03	6.796E-08
F3	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F4	4.680E-05	5.792E-01	5.609E-01	3.499E-06	1.667E-02	8.181E-01	6.674E-06
F5	3.939E-07	6.796E-08	3.069E-06	6.796E-08	6.796E-08	1.431E-07	6.796E-08
F6	6.796E-08	6.796E-08	3.069E-06	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F7	1.235E-07	1.657E-07	2.563E-07	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F8	2.596E-05	6.796E-08	7.577E-06	6.796E-08	6.796E-08	3.416E-07	6.796E-08
F9	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F10	2.471E-04	1.159E-04	6.554E-01	9.748E-06	6.015E-07	6.674E-06	5.166E-06
F11	7.898E-08	1.576E-06	2.690E-06	6.796E-08	7.898E-08	2.563E-07	6.796E-08
F12	9.173E-08	1.803E-06	3.382E-04	6.796E-08	6.796E-08	1.610E-04	1.065E-07
F13	5.250E-01	9.246E-01	4.679E-02	6.168E-01	2.218E-07	7.972E-01	9.091E-02
F14	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F15	6.796E-08	9.173E-08	6.917E-07	6.796E-08	6.796E-08	1.235E-07	1.235E-07
F16	4.155E-04	5.896E-05	6.557E-03	4.539E-07	1.431E-07	4.680E-05	6.917E-07
F17	1.929E-02	5.629E-04	5.115E-03	4.539E-07	1.065E-07	9.127E-07	9.173E-08
F18	1.806E-01	1.657E-07	3.750E-04	1.235E-07	9.173E-08	9.173E-08	1.235E-07
F19	6.610E-05	5.255E-05	4.155E-04	7.406E-05	1.657E-07	7.406E-05	1.037E-04
F20	2.690E-06	7.898E-08	3.382E-04	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F21	9.127E-07	9.173E-08	6.015E-07	1.201E-06	1.201E-06	1.657E-07	6.796E-08
F22	7.113E-03	7.205E-02	9.620E-02	6.610E-05	2.356E-06	1.159E-04	8.357E-04
F23	6.796E-08	7.898E-08	2.062E-06	6.796E-08	6.796E-08	1.065E-07	6.796E-08
F24	1.918E-07	1.657E-07	2.960E-07	6.796E-08	6.796E-08	1.657E-07	6.796E-08
F25	6.796E-08	6.787E-02	2.596E-05	6.796E-08	6.796E-08	1.349E-03	5.874E-06
F26	1.227E-03	1.201E-06	1.136E-01	5.874E-06	1.610E-04	4.155E-04	5.792E-01
F27	6.796E-08	1.065E-07	1.047E-06	6.796E-08	7.898E-08	1.047E-06	1.235E-07
F28	1.235E-07	1.234E-02	1.444E-04	9.173E-08	1.918E-07	4.112E-02	6.015E-07
F29	6.674E-06	4.680E-05	1.436E-02	9.173E-08	1.235E-07	5.227E-07	6.796E-08
F30	6.796E-08	6.220E-04	8.292E-05	6.796E-08	6.796E-08	5.255E-05	6.796E-08

TABLE XVII. RESULTS OF WILCOXON RANK-SUM TEST BETWEEN DCTAROA AND ADVANCED OPTIMIZATION ALGORITHMS ON CEC2017 TEST FUNCTION FOR 50-DIMENSIONAL (APPENDIX)

	FDB-SDO	DFDB-MRFO	FDB-SFS	EESHHO	CTHHO	EAPSO	QMESSA
F1	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F3	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F4	6.796E-08	3.639E-03	1.014E-03	6.796E-08	1.657E-07	3.057E-03	9.173E-08
F5	1.415E-05	7.898E-08	3.416E-07	6.796E-08	6.796E-08	1.047E-06	6.796E-08
F6	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F7	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F8	1.431E-07	6.796E-08	7.898E-08	6.796E-08	6.796E-08	3.939E-07	6.796E-08
F9	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F10	2.218E-07	4.680E-05	3.057E-03	7.406E-05	8.597E-06	3.750E-04	2.356E-06
F11	6.796E-08	6.796E-08	1.431E-07	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F12	6.796E-08	4.703E-03	8.103E-02	6.796E-08	6.796E-08	6.011E-02	6.796E-08
F13	1.104E-05	4.388E-02	1.136E-01	1.251E-05	6.796E-08	3.235E-01	2.471E-04
F14	1.481E-03	1.235E-07	2.471E-04	6.796E-08	6.796E-08	9.173E-08	6.796E-08
F15	2.748E-02	8.392E-01	7.972E-01	4.112E-02	1.065E-07	4.735E-01	7.113E-03
F16	3.639E-03	4.703E-03	1.436E-02	1.047E-06	2.960E-07	1.444E-04	7.577E-06
F17	2.302E-05	5.227E-07	3.336E-03	1.431E-07	9.173E-08	3.987E-06	9.173E-08
F18	1.415E-05	1.235E-07	2.356E-06	6.796E-08	6.796E-08	9.173E-08	6.796E-08
F19	9.091E-02	6.011E-02	2.184E-01	7.712E-03	6.796E-08	8.585E-02	3.048E-04
F20	1.047E-06	2.563E-07	5.896E-05	1.065E-07	6.796E-08	1.918E-07	6.796E-08
F21	7.948E-07	9.173E-08	9.748E-06	6.796E-08	6.796E-08	3.939E-07	6.796E-08
F22	1.807E-05	9.278E-05	7.643E-02	3.069E-06	2.563E-07	6.389E-02	5.896E-05
F23	6.796E-08	6.796E-08	9.173E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F24	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F25	6.796E-08	2.690E-06	1.657E-07	6.796E-08	9.173E-08	3.048E-04	6.796E-08
F26	6.796E-08	1.000E+00	8.103E-02	2.218E-07	6.796E-08	6.796E-08	5.560E-03
F27	6.796E-08	6.796E-08	1.918E-07	6.796E-08	6.796E-08	5.227E-07	6.796E-08
F28	6.796E-08	7.948E-07	9.173E-08	6.796E-08	6.796E-08	7.577E-06	6.796E-08
F29	2.218E-07	6.796E-08	2.596E-05	6.796E-08	6.796E-08	1.657E-07	6.796E-08
F30	6.796E-08	1.431E-07	1.576E-06	6.796E-08	6.796E-08	4.320E-03	6.796E-08

TABLE XVIII. RESULTS OF WILCOXON RANK-SUM TEST BETWEEN DCTAROA AND ADVANCED OPTIMIZATION ALGORITHMS ON CEC2017 TEST FUNCTION FOR 100-DIMENSIONAL (APPENDIX)

	FDB-SDO	DFDB-MRFO	FDB-SFS	EESHHO	CTHHO	EAPSO	QMESSA
F1	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F3	9.173E-08	6.796E-08	3.939E-07	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F4	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F5	6.796E-08	6.796E-08	1.065E-07	6.796E-08	6.796E-08	2.218E-07	6.796E-08
F6	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F7	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F8	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	9.173E-08	6.796E-08
F9	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F10	6.796E-08	1.610E-04	4.680E-05	1.657E-07	7.898E-08	5.227E-07	1.657E-07
F11	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F12	6.796E-08	7.948E-07	1.065E-07	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F13	6.796E-08	3.235E-01	5.652E-02	6.796E-08	6.796E-08	5.255E-05	6.796E-08
F14	1.235E-07	9.173E-08	4.539E-07	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F15	2.218E-07	7.353E-01	7.353E-01	1.376E-06	6.796E-08	2.227E-02	1.431E-07
F16	2.563E-07	1.600E-05	6.557E-03	6.796E-08	6.917E-07	3.605E-02	2.218E-07
F17	5.896E-05	5.255E-05	2.390E-02	1.918E-07	6.796E-08	4.540E-06	1.235E-07
F18	1.794E-04	3.293E-05	3.966E-03	2.218E-07	3.416E-07	9.127E-07	5.227E-07
F19	6.796E-08	9.892E-01	9.892E-01	6.796E-08	6.796E-08	1.136E-01	6.796E-08
F20	9.127E-07	8.597E-06	1.349E-03	1.657E-07	1.235E-07	6.015E-07	1.235E-07
F21	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F22	6.796E-08	1.444E-04	3.705E-05	6.796E-08	6.796E-08	3.966E-03	7.898E-08
F23	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F24	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F25	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F26	6.796E-08	1.600E-05	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F27	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F28	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08
F29	6.796E-08	3.939E-07	6.015E-07	7.898E-08	6.796E-08	1.803E-06	6.796E-08
F30	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08	6.796E-08

DECLARATION OF CONFLICTS OF INTEREST

We have no conflict of interest to declare.

ACKNOWLEDGMENT

This work was supported by Guizhou Provincial Science and Technology Project, under Grant NO.QKH-Basic-ZK[2022]General018.

REFERENCES

- [1] G. Hu, B. Du, X. Wang, G. Wei, "An enhanced black widow optimization algorithm for feature selection," *Knowledge-Based Systems*, vol. 235, p. 107638, 2022, doi: <https://doi.org/10.1016/j.knosys.2021.107638>
- [2] V. Rajinikanth, S. Kadry, R. G. Crespo, E. Verdú, "A Study on RGB Image Multi-Thresholding using Kapur/Tsallis Entropy and Moth-Flame Algorithm," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 7, no. 2, 2021, doi: <https://doi.org/10.9781/ijimai.2021.11.008>
- [3] S. Foroughi, J. K. Hamidi, M. Monjezi, M. Nehring, "The integrated optimization of underground stope layout designing and production scheduling incorporating a non-dominated sorting genetic algorithm (NSGA-II)," *Resources Policy*, vol. 63, p. 101408, 2019, doi: <https://doi.org/10.1016/j.resourpol.2019.101408>
- [4] M. Mahalakshmi, G. Hariharan, G. Brindha, "An efficient wavelet-based optimization algorithm for the solutions of reaction-diffusion equations in biomedicine," *Computer Methods and Programs in Biomedicine*, vol. 186, p. 105218, 2020, doi: <https://doi.org/10.1016/j.cmpb.2019.105218>
- [5] R. Kuo, M. F. Luthfiansyah, N. A. Masrurroh, F. E. Zulvia, "Application of improved multi-objective particle swarm optimization algorithm to solve disruption for the two-stage vehicle routing problem with time windows," *Expert Systems with Applications*, vol. 225, p. 120009, 2023, doi: <https://doi.org/10.1016/j.eswa.2023.120009>
- [6] S. Yang, X. Gu, Y. Liu, R. Hao, S. Li, "A general multi-objective optimized wavelet filter and its applications in fault diagnosis of wheelset bearings," *Mechanical Systems and Signal Processing*, vol. 145, p. 106914, 2020, doi: <https://doi.org/10.1016/j.ymssp.2020.106914>
- [7] J. Wang, C. Zhan, S. Li, Q. Zhao, J. Liu, Z. Xie, "Adaptive variational mode decomposition based on Archimedes optimization algorithm and its application to bearing fault diagnosis," *Measurement*, vol. 191, p. 110798, 2022, doi: <https://doi.org/10.1016/j.measurement.2022.110798>
- [8] T. Acarer, "Energy-Aware Path Planning by Autonomous Underwater Vehicle in Underwater Wireless Sensor Networks for Safer Maritime Transportation," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 8, no. 7, 2024, doi: <https://doi.org/10.9781/ijimai.2024.08.003>
- [9] K. K. Verma, B. M. Singh, "Deep Multi-Model Fusion for Human Activity Recognition Using Evolutionary Algorithms," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 7, no. 2, 2021, doi: <https://doi.org/10.9781/IJIMAI.2021.08.008>
- [10] J. Henriques, F. Caldeira, "A Model for Planning TELCO Work-Field Activities Enabled by Genetic and Ant Colony Algorithms," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 7, no. 6, 2022, doi: <https://doi.org/10.9781/ijimai.2022.08.011>
- [11] Q. Ding, X. Xu, "Improved GWO Algorithm for UAV Path Planning on Crop Pest Monitoring," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 7, no. 5, 2022, doi: <https://doi.org/10.9781/ijimai.2022.07.002>
- [12] C. Liang, Z. Diao, X. Wang, Y. Huo, K. Li, D. He, W. Liang, "FedAHPiP: Federated Learning with Adaptive Hot Parameter Identification and Personalized Anchoring for multi-agent collaboration," *Journal of Industrial Information Integration*, vol. 51, p. 101087, 2026.
- [13] S. Zhou, K. Li, Y. Chen, C. Yang, W. Liang, A. Y. Zomaya, "TrustBCFL: Mitigating data bias in IoT through blockchain-enabled federated learning," *IEEE Internet of Things Journal*, vol. 11, no. 15, pp. 25648–25662, 2024.
- [14] C. Liang, W. Liang, Y. Chen, K. Li, A. Y. Zomaya, "FedTCTF: Tensor Completion-Based Federated Learning for Device Heterogeneity," *IEEE Transactions on Sustainable Computing*, vol. 10, no. 6, pp. 1227–1239, 2025.
- [15] S. Mirjalili, A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016, doi: <https://doi.org/10.1016/j.advengsoft.2016.01.008>

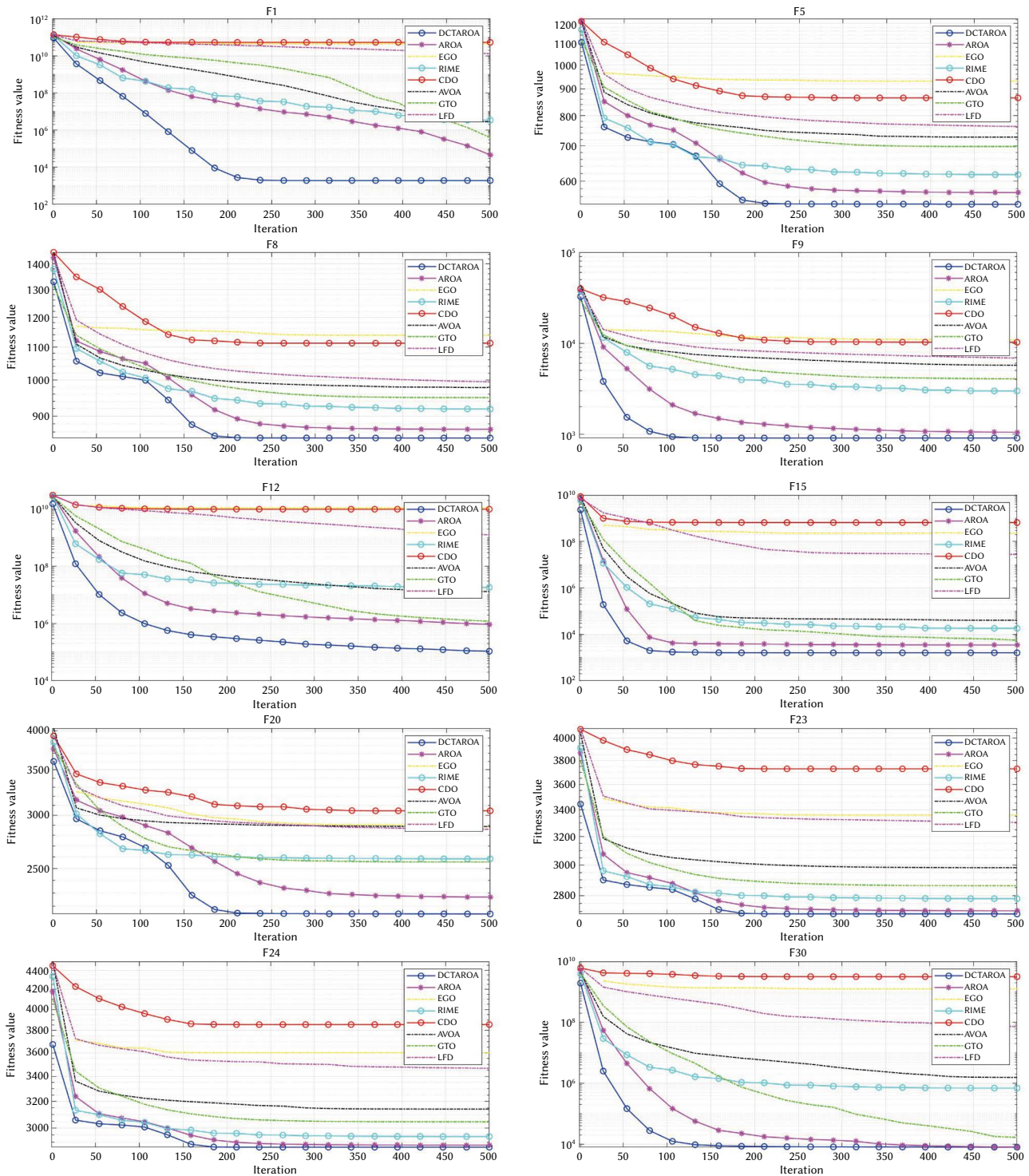


Fig. 5. Average convergence curves of eight single optimization algorithms on representative CEC2017 functions (30-dimensional) (Appendix).

- [16] X. Bao, H. Jia, C. Lang, "A novel hybrid harris hawks optimization for color image multilevel thresholding segmentation," *IEEE Access*, vol. 7, pp. 76529–76546, 2019, doi: <https://doi.org/10.1109/ACCESS.2019.2921545>
- [17] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A. H. Gandomi, "The arithmetic optimization algorithm," *Computer methods in applied mechanics and engineering*, vol. 376, p. 113609, 2021, doi: <https://doi.org/10.1016/j.cma.2020.113609>
- [18] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied intelligence*, vol. 51, pp. 1531–1551, 2021, doi: <https://doi.org/10.1007/s10489-020-01893-z>
- [19] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, "Equilibrium

- optimizer: A novel optimization algorithm," *Knowledge-based systems*, vol. 191, p. 105190, 2020, doi: <https://doi.org/10.1016/j.knsys.2019.105190>
- [20] R. V. Rao, V. J. Savsani, D. Vakharia, "Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems," *Information sciences*, vol. 183, no. 1, pp. 1–15, 2012, doi: <https://doi.org/10.1016/j.ins.2011.08.006>
- [21] Y. Li, W. Liang, K. Xie, D. Zhang, S. Xie, K. Li, "Lightnests: quick and accurate neural sequential tensor completion via meta learning," in *IEEE INFOCOM 2023-IEEE conference on computer communications*, 2023, pp. 1–10, IEEE doi = <https://doi.org/10.1109/INFOCOM53939.2023.10228967>
- [22] A. Malik, Y. Tikhmarine, D. Souag-Gamane, O. Kisi, Q. B. Pham, "Support vector regression optimized by meta-heuristic algorithms for

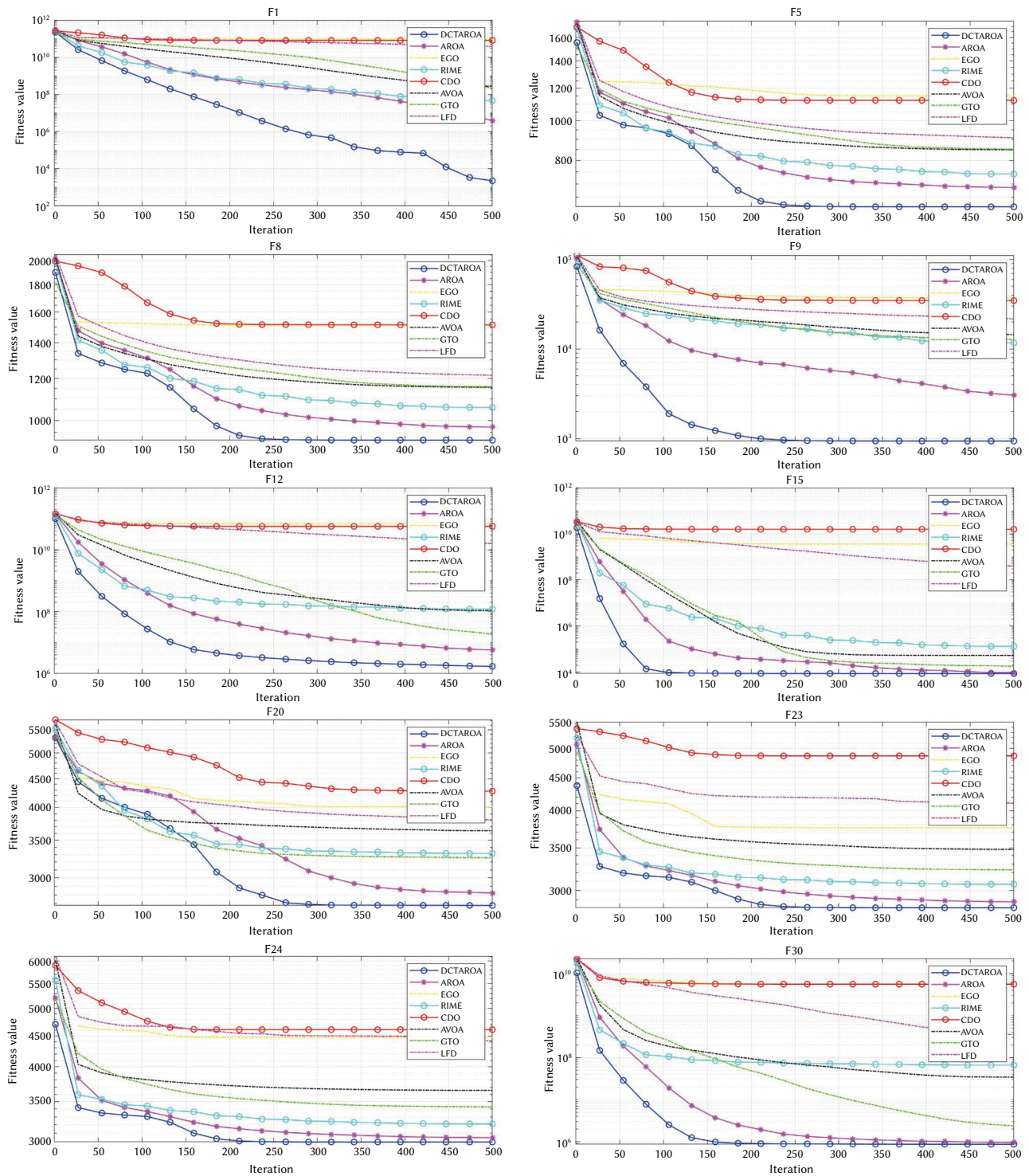


Fig. 6. Average convergence curves of eight single optimization algorithms on representative CEC2017 functions (50-dimensional) (Appendix).

daily streamflow prediction,” *Stochastic Environmental Research and Risk Assessment*, vol. 34, pp. 1755–1773, 2020, doi: <https://doi.org/10.1007/s00477-020-01874-1>

- [23] D. H. Wolpert, W. G. Macready, “No free lunch theorems for optimization,” *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997, doi: <https://doi.org/10.1109/4235.585893>
- [24] L. Yin, Y. Deng, C. Wang, Y. Chen, W. Liang, K. Li, “A three-dimensional safe escape path dynamic planning method based on multi-modal fire information sensing and deep reinforcement learning,” *The Journal of Supercomputing*, vol. 81, no. 17, p. 1574, 2025.
- [25] B. S. Yildiz, N. Pholdee, S. Bureerat, A. R. Yildiz, S. M. Sait, “Robust

design of a robot gripper mechanism using new hybrid grasshopper optimization algorithm,” *Expert Systems*, vol. 38, no. 3, p. e12666, 2021, doi: <https://doi.org/10.1111/exsy.12666>

- [26] A. A. Ewees, M. Abd Elaziz, E. H. Houssein, “Improved grasshopper optimization algorithm using opposition-based learning,” *Expert Systems with Applications*, vol. 112, pp. 156–172, 2018, doi: <https://doi.org/10.1016/j.eswa.2018.06.023>
- [27] Y. Li, X. Yu, J. Liu, “An opposition-based butterfly optimization algorithm with adaptive elite mutation in solving complex high-dimensional optimization problems,” *Mathematics and Computers in Simulation*, vol. 204, pp. 498–528, 2023, doi: <https://doi.org/10.1016/j.matcom.2022.08.020>

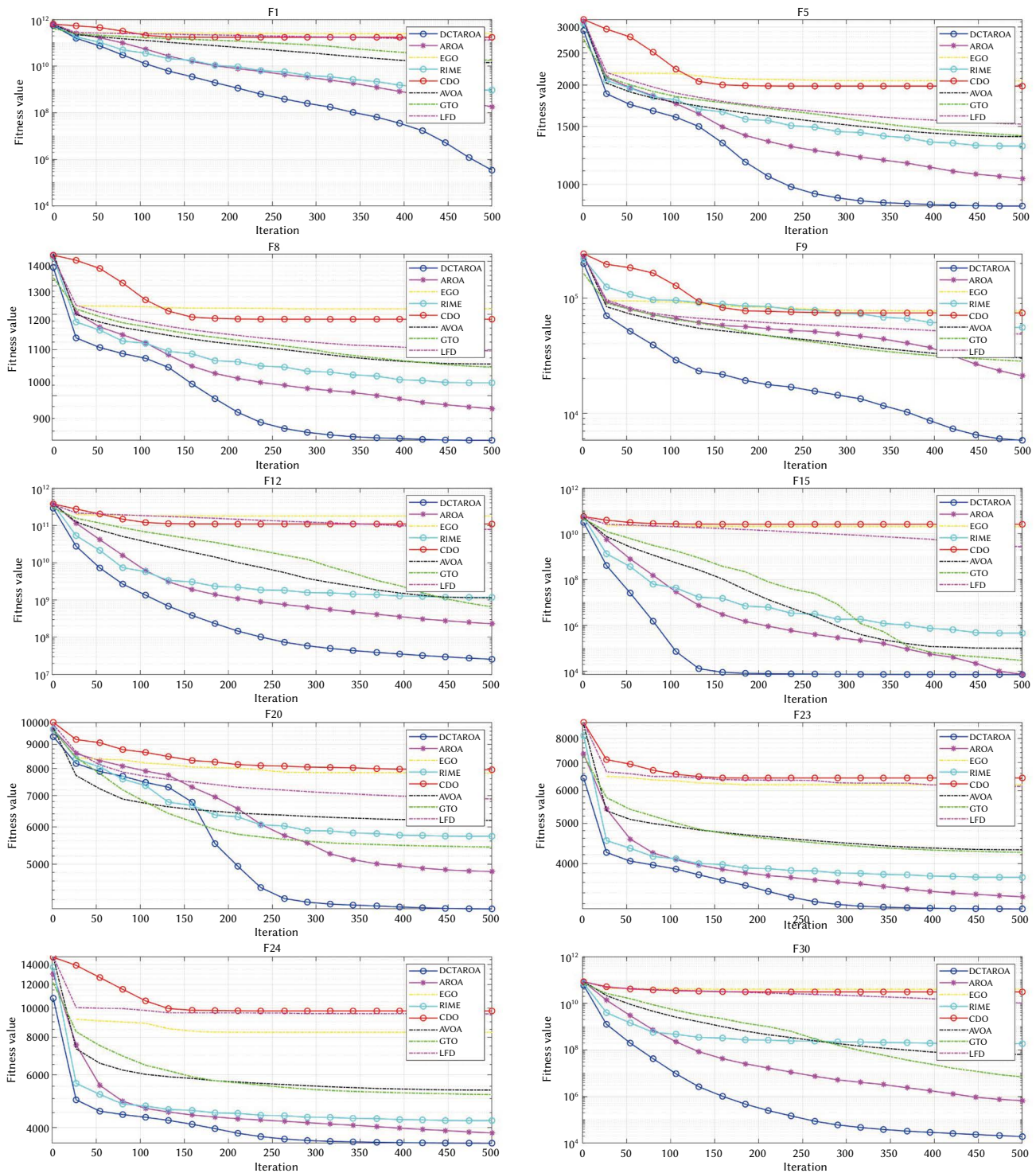


Fig. 7. Average convergence curves of eight single optimization algorithms on representative CEC2017 functions (100-dimensional) (Appendix).

- [28] S. Arora, P. Anand, "Chaotic grasshopper optimization algorithm for global optimization," *Neural Computing and Applications*, vol. 31, pp. 4385–4405, 2019, doi: <https://doi.org/10.1007/s00521-018-3343-2>
- [29] J. Wu, H. Wang, N. Li, P. Yao, Y. Huang, Z. Su, Y. Yu, "Distributed trajectory optimization for multiple solar-powered UAVs target tracking in urban environment by Adaptive Grasshopper Optimization Algorithm," *Aerospace Science and Technology*, vol. 70, pp. 497–510, 2017, doi: <https://doi.org/10.1016/j.ast.2017.08.037>
- [30] A. K. Shukla, "Detection of anomaly intrusion utilizing self-adaptive grasshopper optimization algorithm," *Neural Computing and Applications*, vol. 33, no. 13, pp. 7541–7561, 2021, doi: <https://doi.org/10.1007/s00521-020-05500-7>
- [31] J. Luo, H. Chen, Y. Xu, H. Huang, X. Zhao, others, "An improved grasshopper optimization algorithm with application to financial stress prediction," *Applied Mathematical Modelling*, vol. 64, pp. 654–668, 2018, doi: <https://doi.org/10.1016/j.apm.2018.07.044>
- [32] H. T. Kahraman, S. Aras, E. Gedikli, "Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms," *Knowledge-Based Systems*, vol. 190, p. 105169, 2020, doi: <https://doi.org/10.1016/j.knsys.2019.105169>
- [33] H. T. Kahraman, H. Bakir, S. Duman, M. Kati, S. Aras, U. Guvenc, "Dynamic FDB selection method and its application: modeling and optimizing of directional overcurrent relays coordination," *Applied Intelligence*, pp. 1–36, 2022, doi: <https://doi.org/10.1007/s10489-021-02629-3>

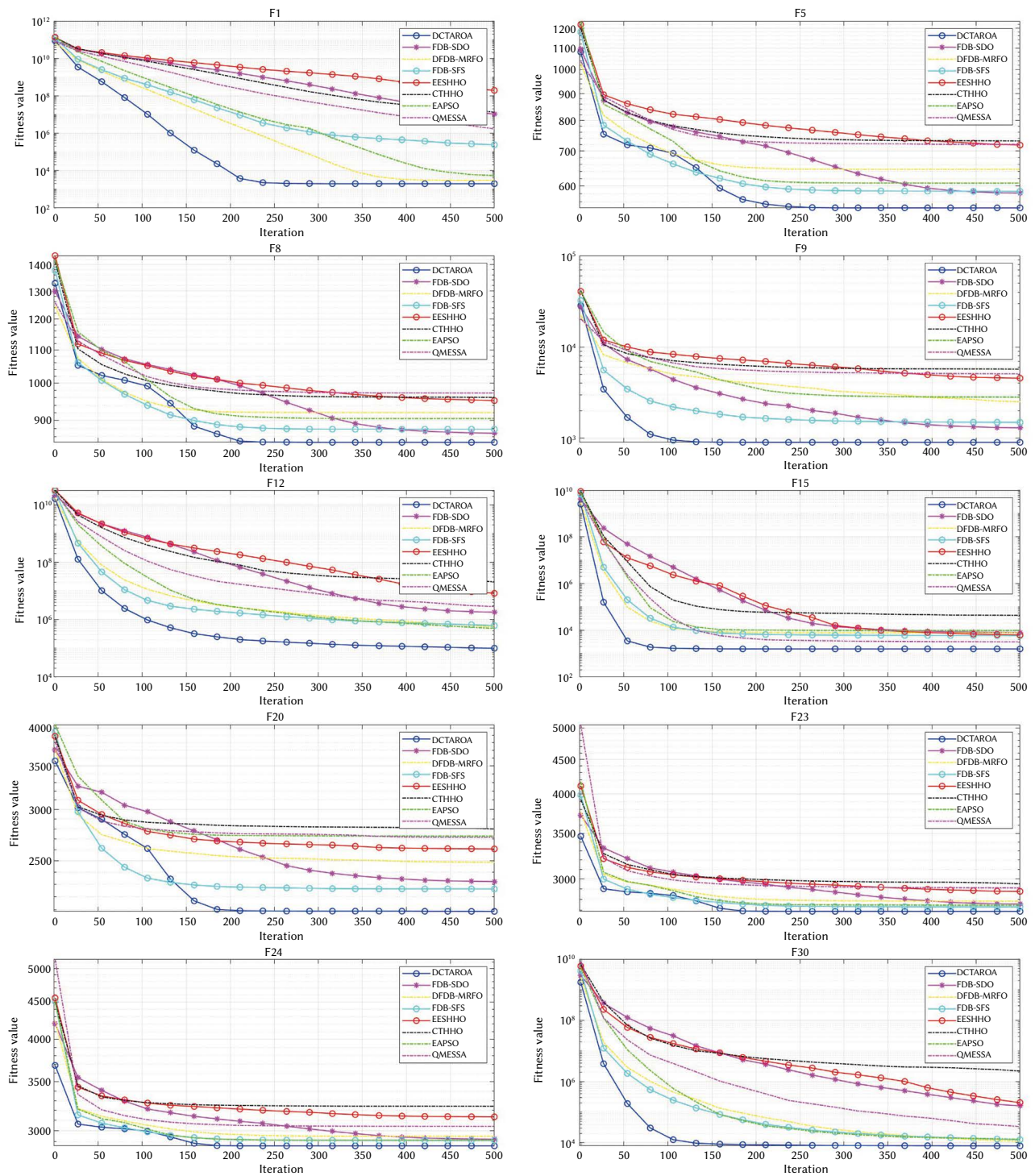


Fig. 8. Average convergence curves of eight advanced optimization algorithms on representative CEC2017 functions (30-dimensional) (Appendix).

- [34] 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, p. 105501, 2023, doi: <https://doi.org/10.1016/j.engappai.2022.105501>
- [35] B. Ozkaya, H. T. Kahraman, S. Duman, U. Guvenc, "Fitness-Distance-Constraint (FDC) based guide selection method for constrained optimization problems," *Applied Soft Computing*, vol. 144, p. 110479, 2023, doi: <https://doi.org/10.1016/j.asoc.2023.110479>
- [36] Y. Sonmez, S. Duman, H. T. Kahraman, M. Kati, S. Aras, U. Guvenc, "Fitness-distance balance based artificial ecosystem optimisation to solve

- transient stability constrained optimal power flow problem," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 36, no. 5, pp. 745–784, 2024, doi: <https://doi.org/10.1080/0952813X.2022.2104388>
- [37] H. Bakir, "Dynamic fitness-distance balance-based artificial rabbits optimization algorithm to solve optimal power flow problem," *Expert Systems with Applications*, vol. 240, p. 122460, 2024, doi: <https://doi.org/10.1016/j.eswa.2023.122460>
- [38] C. Li, J. Li, H. Chen, A. A. Heidari, "Memetic Harris Hawks Optimization: Developments and perspectives on project scheduling and QoS-aware web service composition," *Expert Systems with Applications*, vol. 171, p. 114529, 2021, doi: <https://doi.org/10.1016/j.eswa.2020.114529>

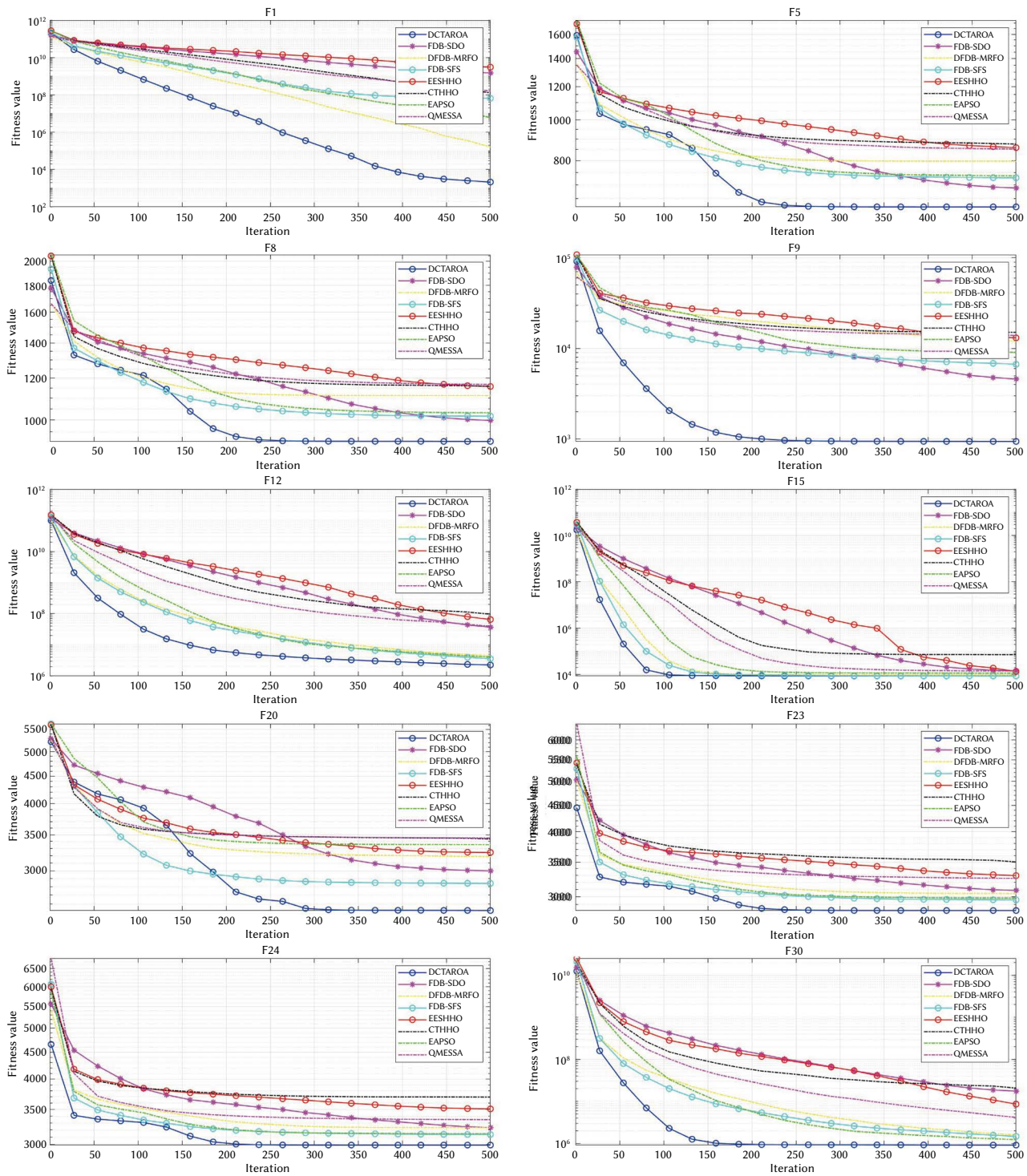


Fig. 9. Average convergence curves of eight advanced optimization algorithms on representative CEC2017 functions (50-dimensional) (Appendix).

- [39] R. Wu, H. Huang, J. Wei, C. Ma, Y. Zhu, Y. Chen, Q. Fan, "An improved sparrow search algorithm based on quantum computations and multi-strategy enhancement," *Expert Systems with Applications*, vol. 215, p. 119421, 2023, doi: <https://doi.org/10.1016/j.eswa.2022.119421>
- [40] K. Cymerys, M. Oszust, "Attraction-Repulsion Optimization Algorithm for Global Optimization Problems," *Swarm and Evolutionary Computation*, vol. 84, p. 101459, 2024, doi: <https://doi.org/10.1016/j.swevo.2023.101459>
- [41] Z. Su, H. Wang, "A novel robust hybrid gravitational search algorithm for reusable launch vehicle approach and landing trajectory optimization," *Neurocomputing*, vol. 162, pp. 116–127, 2015, doi: <https://doi.org/10.1016/j.neucom.2015.03.063>
- [42] D. Mustafi, G. Sahoo, "A hybrid approach using genetic algorithm and the differential evolution heuristic for enhanced initialization of the k-means algorithm with applications in text clustering," *Soft Computing*, vol. 23, pp. 6361–6378, 2019, doi: <https://doi.org/10.1007/s00500-018-3289-4>
- [43] H. Nenavath, R. K. Jatoth, "Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking," *Applied Soft Computing*, vol. 62, pp. 1019–1043, 2018, doi: <https://doi.org/10.1016/j.asoc.2017.09.039>
- [44] S. S. Jadon, R. Tiwari, H. Sharma, J. C. Bansal, "Hybrid artificial bee colony algorithm with differential evolution," *Applied Soft Computing*, vol. 58, pp. 11–24, 2017, doi: <https://doi.org/10.1016/j.asoc.2017.04.018>

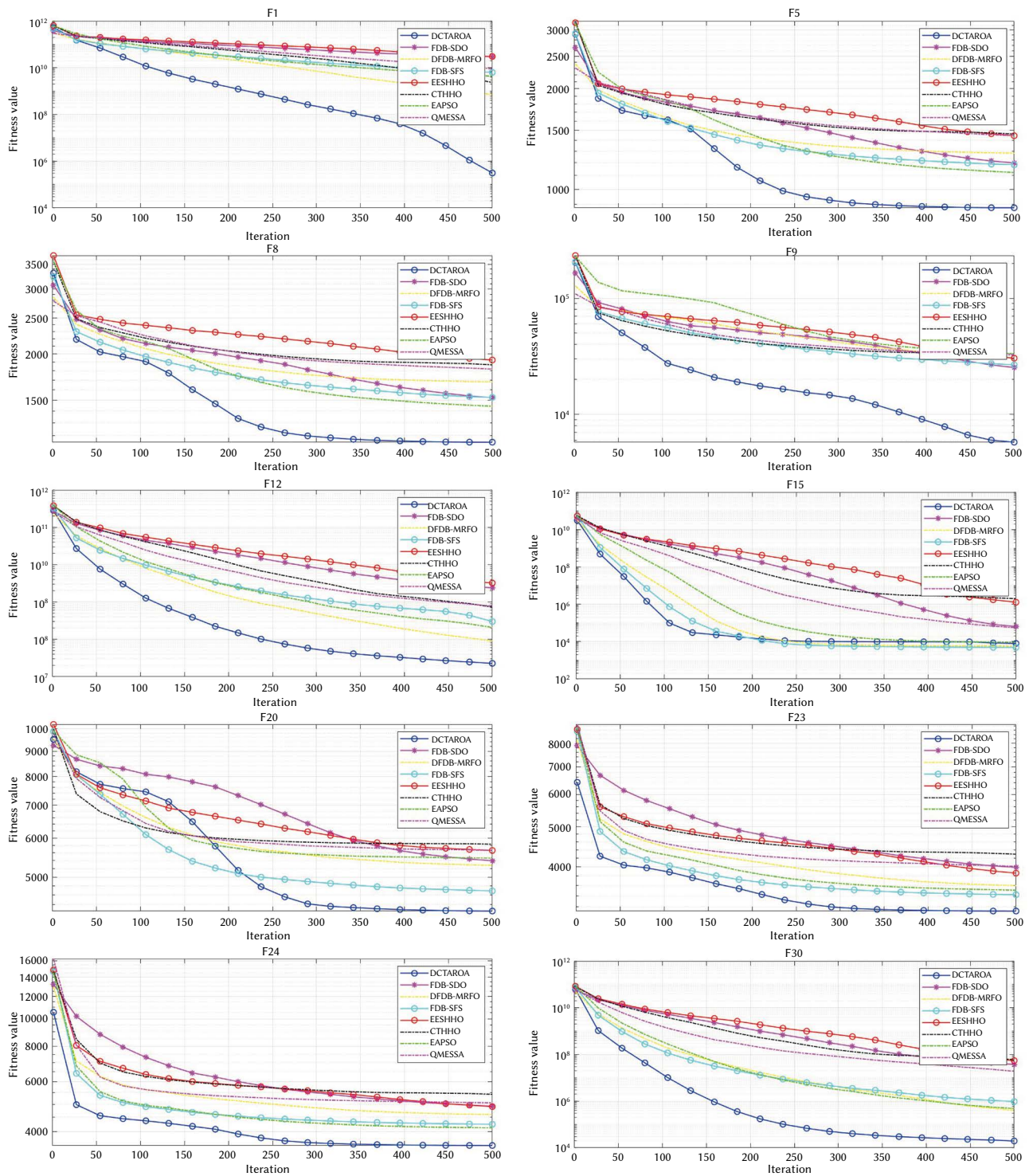


Fig. 10. Average convergence curves of eight advanced optimization algorithms on representative CEC2017 functions (100-dimensional).

- [45] A. Layeb, "Tangent search algorithm for solving optimization problems," *Neural Computing and Applications*, vol. 34, no. 11, pp. 8853–8884, 2022, doi: <https://doi.org/10.1007/s00521-022-06908-z>
- [46] L. Deng, S. Liu, "A multi-strategy improved slime mould algorithm for global optimization and engineering design problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 404, p. 115764, 2023, doi: <https://doi.org/10.1016/j.cma.2022.115764>
- [47] K. Wang, G. K. Rohde, J. Xiao, "Edge feature enhancement approach using hilbert transform of Cauchy distribution and its applications," *IET Image Processing*, vol. 15, no. 12, pp. 2891–2909, 2021, doi: <https://doi.org/10.1049/ipr2.12275>
- [48] G. Sun, Y. Lan, R. Zhao, "Differential evolution with Gaussian mutation and dynamic parameter adjustment," *Soft Computing*, vol. 23, pp. 1615–1642, 2019, doi: <https://doi.org/10.1007/s00500-017-2885-z>
- [49] G. Wu, R. Mallipeddi, P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization," *National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report*, 2017, [Online]. Available: <https://www.researchgate.net/publication/317228117>
- [50] F. Kutlu Onay, S. B. Aydemir, "Chaotic hunger games search optimization algorithm for global optimization and engineering problems,"

Mathematics and Computers in Simulation, vol. 192, pp. 514–536, 2022, doi: <https://doi.org/10.1016/j.matcom.2021.09.014>

- [51] A. Mohammadzadeh, S. Mirjalili, “Eel and grouper optimizer: a nature-inspired optimization algorithm,” *Cluster Computing*, pp. 1–42, 2024, doi: <https://doi.org/10.1007/s10586-024-04545-w>
- [52] H. Su, D. Zhao, A. A. Heidari, L. Liu, X. Zhang, M. Mafarja, H. Chen, “RIME: A physics-based optimization,” *Neurocomputing*, vol. 532, pp. 183–214, 2023, doi: <https://doi.org/10.1016/j.neucom.2023.02.010>
- [53] H. A. Shehadeh, “Chernobyl disaster optimizer (CDO): A novel metaheuristic method for global optimization,” *Neural Computing and Applications*, vol. 35, no. 15, pp. 10733–10749, 2023, doi: <https://doi.org/10.1007/s00521-023-08261-1>
- [54] B. Abdollahzadeh, F. S. Gharehchopogh, S. Mirjalili, “African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems,” *Computers and Industrial Engineering*, vol. 158, p. 107408, 2021, doi: <https://doi.org/10.1016/j.cie.2021.107408>
- [55] B. Abdollahzadeh, F. Soleimani Gharehchopogh, S. Mirjalili, “Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems,” *International Journal of Intelligent Systems*, vol. 36, no. 10, pp. 5887–5958, 2021, doi: <https://doi.org/10.1002/int.22535>
- [56] E. H. Houssein, M. R. Saad, F. A. Hashim, H. Shaban, M. Hassaballah, “Levy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems,” *Engineering Applications of Artificial Intelligence*, vol. 94, p. 103731, 2020, doi: <https://doi.org/10.1016/j.engappai.2020.103731>
- [57] M. Kati, H. T. Kahraman, “Improving supply-demand-based optimization algorithm with FDB method: a comprehensive research on engineering design problems,” *Journal of Engineering Sciences*, vol. 8, no. 5, pp. 156–172, 2020, doi: <https://doi.org/10.21923/jesd.829508>
- [58] S. Aras, E. Gedikli, H. T. Kahraman, “A novel stochastic fractal search algorithm with fitness-distance balance for global numerical optimization,” *Swarm and Evolutionary Computation*, vol. 61, p. 100821, 2021, doi: <https://doi.org/10.1016/j.swevo.2020.100821>
- [59] M. Wang, J.-S. Wang, X.-D. Li, M. Zhang, 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, 2022, doi: <https://doi.org/10.1007/s10489-021-03080-0>
- [60] Y. Zhang, “Elite archives-driven particle swarm optimization for large scale numerical optimization and its engineering applications,” *Swarm and Evolutionary Computation*, vol. 76, p. 101212, 2023, doi: <https://doi.org/10.1016/j.swevo.2022.101212>
- [61] F. Wilcoxon, “Individual Comparisons by Ranking Methods,” *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945, doi: <https://doi.org/10.2307/3001968>
- [62] K. Li, H. Huang, S. Fu, C. Ma, Q. Fan, Y. Zhu, “A multi-strategy enhanced northern goshawk optimization algorithm for global optimization and engineering design problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 415, p. 116199, 2023, doi: <https://doi.org/10.1016/j.cma.2023.116199>
- [63] M. H. Nadimi-Shahraki, H. Zamani, “DMDE: Diversity-maintained multi-trial vector differential evolution algorithm for non-decomposition large-scale global optimization,” *Expert Systems with Applications*, vol. 198, p. 116895, 2022, doi: <https://doi.org/10.1016/j.eswa.2022.116895>
- [64] N. Wu, W. Zhao, Y. Chen, J. Xiao, J. Wang, W. Liang, K.-C. Li, N. Sukhija, “HFSL: heterogeneity split federated learning based on client computing capabilities,” *The Journal of Supercomputing*, vol. 81, no. 1, pp. 1–28, 2025, doi: <https://doi.org/10.1007/s11227-024-06632-6>
- [65] J. Zhang, X. Li, K. Gu, W. Liang, K. Li, “Secure aggregation in heterogeneous federated learning for digital ecosystems,” *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 1995–2003, 2023, doi: <https://doi.org/10.1109/TCE.2023.3330501>
- [66] F. Feng, Q. Zhou, Z. Shen, X. Yang, L. Han, J. Wang, “The application of a novel neural network in the detection of phishing websites,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–15, 2024, doi: <https://doi.org/10.1007/s12652-018-0786-3>
- [67] Q. Zhou, F. Feng, Z. Shen, R. Zhou, M.-Y. Hsieh, K.-C. Li, “A novel approach for mobile malware classification and detection in Android systems,” *Multimedia Tools and Applications*, vol. 78, pp. 3529–3552, 2019, doi: <https://doi.org/10.1007/s11042-018-6498-z>
- [68] F. Feng, X. Liu, B. Yong, R. Zhou, Q. Zhou, “Anomaly detection in ad-

hoc networks based on deep learning model: A plug and play device,” *Ad Hoc Networks*, vol. 84, pp. 82–89, 2019, doi: <https://doi.org/10.1016/j.adhoc.2018.09.014>



Fang Feng

learning, and medical diagnosis.



Mingjiang Cai

Mingjiang Cai received his Master's degree in the College of Public Management, Guizhou University of Finance and Economics, China. He is working in the Nanming District Commerce Bureau in Guiyang, China. His research interests are machine learning and data mining.



Hung-Wei Li

Hung-Wei Li received a B.S. degree in computer science and information engineering from Providence University in 2025. His current interests include edge computing, edge intelligence, and machine learning.



Kuan-Ching Li

Kuan-Ching Li is currently a Life Distinguished Professor at Providence University, where he is also the Director of the High-Performance Computing and Networking Center, established by collaborations with industry. Besides publishing articles in renowned journals and conferences, he is a co-author/co-editor of more than 50 books published by well-known publishers, holds several patents, and serves as an associate and guest editor for various scientific journals. His research interests include Big Data, parallel and distributed computing, and emerging technologies. Dr. Li is a Fellow of the IET and a Senior Member of the IEEE.