Scientific Reports

https://doi.org/10.1038/s41598-026-39222-y

Article in Press

Adaptive multi mechanism integration in the crested porcupine optimizer for global optimization and engineering design problems

Received: 11 August 2025

Accepted: 3 February 2026

Published online: 16 February 2026

Cite this article as: Xie H., Mao J., Wan X. *et al.* Adaptive multi mechanism integration in the crested porcupine optimizer for global optimization and engineering design problems. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-39222-y

Hairong Xie, Jia Mao, Xijun Wan & Yifei Bai

Adaptive multi mechanism integration in the Crested Porcupine Optimizer for global optimization and engineering design problems

Hairong Xie[1] • Jia Mao[1*] • Xijun Wan[1] • Yifei Bai[1]

1 School of Transportation, Jilin University, Changchun 130022, Jilin, China

✉ *Corresponding author: Jia Mao

maojia@jlu.edu.cn

Abstract

The Crested Porcupine Optimizer (CPO), an emerging intelligent optimization algorithm, exhibits considerable potential for addressing complex engineering problems, yet its capabilities remain insufficiently investigated. Nevertheless, the original CPO is susceptible to premature convergence and suffers from insufficient population diversity. To effectively address these limitations, this paper proposes a multi-mechanism enhanced Crested Porcupine Optimizer (SDHCPO). Its core innovation lies in the integration of four key strategies: a Sobol-Opposition-Based Learning (Sobol-OBL) initialization strategy, which combines the Sobol sequence with opposition-based learning to generate an initial population that is more uniformly distributed in the high-dimensional search space; a cosine-annealing-based dynamic adjustment strategy that replaces the original random weights and substantially enhances convergence stability; the incorporation of the DE/rand/1 strategy in the first defense phase to disrupt positional dependence and prevent premature convergence; and a horizontal-vertical crossover strategy employed in the second defense phase to eliminate dimensional stagnation. Experimental results on two authoritative benchmark suites, CEC2017 and CEC2022, demonstrate that the proposed algorithm outperforms seven representative metaheuristic algorithms in terms of global exploration capability, local exploitation accuracy, and convergence robustness. Furthermore, empirical studies on five representative engineering design optimization problems show that SDHCPO consistently attains either the best-known solutions or highly competitive results reported in the literature, thereby further confirming its effectiveness and broad application potential for complex real-world engineering optimization tasks.

Keywords Metaheuristic algorithm · Crested Porcupine Optimizer · Sobol-OBL initialization · Differential Evolution strategy · Horizontal-Vertical crossover strategy · Cosine annealing dynamic adjustment strategy

1 Introduction

Global optimization problems are widespread in modern complex systems and engineering design, typically characterized by high nonlinearity, multimodality, multiple constraints, and high dimensionality. These characteristics pose significant computational challenges to traditional optimization methods [1]. Classical deterministic algorithms, such as gradient-based methods and dynamic programming, perform well in simple cases like convex optimization but often prove insufficient when addressing complex problems that are non-convex or non-differentiable. In contrast, intelligent optimization algorithms (IOAs), which do not rely on gradient information and employ population-based search mechanisms, have been widely applied to complex global optimization and engineering design problems [2]. Extensive studies have shown that such stochastic search algorithms exhibit superior problem-solving capabilities in practical engineering challenges such as mechanical structure design, parameter estimation, and signal processing [3]. In complex manufacturing environments, an efficient meta-heuristic with a critical-path-based neighborhood search has been developed for the dual-resource flexible job shop scheduling problem under production line reconfiguration, illustrating how tailored intelligent optimization can significantly enhance large-scale industrial scheduling performance[4]. However, as the scale and complexity of problems increase, the performance of intelligent optimization algorithms can degrade significantly, necessitating continuous development and improvement of new algorithms to meet these escalating challenges [5].

Over the past decades, researchers have proposed numerous mainstream intelligent optimization algorithms, encompassing various categories such as evolutionary algorithms (EAs) and swarm intelligence algorithms (SIAs). Typical representatives include Particle Swarm Optimization (PSO) [6], Genetic Algorithm (GA) [7], Differential Evolution (DE) [8], Grey Wolf Optimizer (GWO) [9], Whale Optimization Algorithm (WOA) [10], and Firefly Algorithm (FA) [11]. These algorithms have been successfully applied in various fields such as function optimization, scheduling, control, and machine learning [12]. However, each algorithm also has its inherent limitations: For example, Particle Swarm Optimization (PSO) is simple to implement and converges quickly, but it is prone to premature convergence and getting trapped in local optima [13]; GA possesses strong global search capability but involves complex encoding and exhibits uncertainty in the convergence process [14]; DE improves search efficiency through differential mutation, but it may suffer from premature convergence or stagnation when dealing with multimodal problems [15]; GWO simulates the social behavior of grey wolves and features a simple algorithmic structure, yet it tends to exhibit low convergence accuracy and local convergence issues in the fine-tuning phase [16]; WOA performs global exploration by mimicking the foraging behavior of whales, but in complex environments, it may suffer from poor optimization accuracy and can get trapped in local optima during early iterations [17]; FA conducts search through attractiveness-based movement among fireflies, but the standard FA lacks sufficient exploration capability, which may lead to premature convergence and reduced solution accuracy [18,19]. Overall, these mainstream intelligent algorithms often encounter bottlenecks such as premature convergence, entrapment in local optima, and slow convergence speed when tackling complex problems. To address these issues, researchers have introduced various enhancement strategies—such as memory mechanisms, mutation operations, and elitism—into individual algorithms to strengthen their global search capabilities [20,21]; On the other hand, hybrid intelligent algorithms have been developed by integrating multiple optimization strategies to balance global and local search capabilities. For example, a hybrid biogeography-based optimization algorithm that combines a hybrid migration operator with a feedback differential evolution mechanism has shown competitive performance on high-dimensional benchmark functions and several real-world engineering design problems [22]. According to the "No Free Lunch" theorem, no single algorithm can perform optimally across all optimization problems, which has further driven the continuous emergence of novel intelligent optimization algorithms [23]. In line with this viewpoint, a biogeography-based optimization variant that incorporates Lévy and Brownian movements together with a steepest-descent local search has been proposed, enabling efficient solution of large-scale global optimization and complex engineering design problems and exemplifying the problem-specific tailoring advocated by the theorem [24].

Against this background, collective behaviors in biological populations have provided a rich source of inspiration for algorithm design. A notable recent example is the Crested Porcupine Optimizer (CPO), a promising swarm intelligence algorithm proposed in 2024 and inspired by the distinctive defensive behavior of crested porcupines against predators. In real-world scenarios, crested porcupines employ four primary defensive strategies—visual intimidation, auditory deterrence, olfactory repulsion, and physical attack—to protect themselves when threatened. Building on this behavioral analogy, CPO divides the optimization process into exploration and exploitation phases and sequentially simulates the four defensive behaviors of crested porcupines during the iterative search. Initially, the algorithm generates a group of “porcupine” individuals as candidate solutions, which are randomly distributed across the global search space. During the exploration phase, CPO then employs “visual defense” and “auditory defense” mechanisms to guide individuals to probe in diverse directions, thereby promoting population diversity and dispersion. In the exploitation phase, the algorithm activates olfactory defense and physical attack strategies to guide individuals toward the neighborhood of the current best solution, thereby accelerating convergence. Notably, CPO incorporates a Cyclic Population Reduction (CPR) mechanism, in which the population size is periodically reduced as the iterations progress, mimicking the natural scenario in which not all porcupines engage in defense simultaneously. This mechanism enhances convergence efficiency while maintaining solution diversity.

Leveraging these bio-inspired strategies, CPO has demonstrated excellent performance on several benchmark tests and engineering applications. In the original study by Abdel-Basset et al., CPO outperformed methods such as GWO, WOA, and SSA by achieving superior optimal solutions across multiple engineering problems. Overall, the novel algorithmic framework of CPO, which effectively balances exploration and exploitation, provides a promising and powerful tool for engineering optimization [25].

Despite these advantages, subsequent studies have identified several notable limitations of CPO when tackling complex optimization problems. First, the original CPO is prone to becoming trapped in local optima when dealing with high-dimensional, multimodal, or irregular functions, causing the algorithm to stagnate near suboptimal solutions and hindering further exploration of superior candidates. In addition, its global search capability—particularly in the early search stage—is constrained by insufficient population diversity, making it difficult to thoroughly explore the solution space. Moreover, the convergence speed of CPO is suboptimal on certain benchmark functions, where slow or even oscillatory convergence has been observed, thereby reducing the overall efficiency of the algorithm [26]; At the same time, its exploitation accuracy is limited, as it lacks an effective local search mechanism in the fine-tuning phase, leading to final solutions that may still deviate from the true optimum [27]. Therefore, enhancing CPO’s global exploration capability and exploitation precision, while mitigating the risk of entrapment in local optima, has become a critical focus of subsequent research.

It is worth noting that the integration of multiple strategies has been proven effective in enhancing optimization algorithms. For example, to alleviate the premature convergence of the standard Firefly Algorithm, Villaruz et al. introduced an additional search mechanism inspired by bee colony scouting behavior, enabling fireflies trapped in local optima to perform guided random walks. This modification effectively helped the algorithm escape local optima and improved its convergence accuracy [28]. Similarly, Zhang et al. proposed a feedback biogeography-based optimization algorithm with steepest descent, in which a dynamic hybrid migration operator, a feedback differential evolution mechanism, and a steepest-descent local search are tightly integrated, achieving high solution accuracy and good scalability on a variety of large-scale benchmark tests and constrained engineering problems [29]. In the recently proposed Transient Search Optimization (TSO) algorithm, chaotic maps are introduced to replace certain random processes. Owing to the ergodicity and non-repetitiveness of chaotic sequences, this enhancement effectively improves the algorithm’s ability to avoid local optima and accelerates its convergence [30].

To this end, researchers have begun to explore the integration of various effective strategies into CPO to develop enhanced algorithmic variants [27,31]. By embedding diverse search mechanisms and parameter control schemes within the CPO framework, it is possible to substantially improve its global optimization performance while preserving its inherent advantages.

Building on the above insights, this paper proposes an adaptive multi-mechanism integrated Crested Porcupine Optimizer (SDHCPO), with the following main contributions:

- An improved CPO framework integrating four complementary mechanisms is developed. The Sobol-OBL initialization strategy combines low-discrepancy Sobol sequences with opposition-based learning to generate an initial population that is more uniformly distributed and provides broader coverage of the search space; a cosine-annealing-based dynamic adjustment strategy replaces random weights with a time-varying nonlinear decay factor, enabling a more controllable evolution of the search process between exploration and exploitation. On this basis, a DE/rand/1 mutation operator is introduced to intensify perturbations and reduce the risk of premature convergence, while a horizontal-vertical crossover mechanism alleviates inter-dimensional coupling and stagnation, thereby enhancing information recombination across dimensions. These four strategies operate synergistically within a unified framework, systematically improving the algorithm’s performance in terms of global search breadth, local exploitation accuracy, and convergence robustness.
- A multi-level evaluation framework is established that jointly captures external performance and internal search mechanisms. On the CEC2017 and CEC2022 benchmark suites, SDHCPO attains the best mean objective values on most test functions, and the Wilcoxon rank-sum test indicates that it achieves statistically significant superiority over the majority of competing algorithms. Subsequent dimensionality-extension experiments further show that the performance advantage of SDHCPO in terms of mean fitness becomes more pronounced as the problem dimension increases. Meanwhile, qualitative analyses from the perspectives of population distribution evolution and convergence trajectories characterize the internal search dynamics of SDHCPO, revealing its stage-wise dynamic adjustment of exploration intensity and exploitation precision at the algorithmic mechanism level.
- The applicability of SDHCPO to multi-constrained engineering optimization is validated through a series of engineering case studies. Several classical structural and mechanical design benchmark problems are selected to construct an engineering test suite spanning from low to high dimensionality. The experimental results show that SDHCPO consistently yields competitive, and in many cases best-known, design solutions with good stability across all test cases. In particular, for the 72-bar spatial truss problem, SDHCPO achieves significant weight reduction under multiple frequency constraints, indicating that the proposed framework can effectively tackle large-scale, highly constrained engineering optimization tasks and possesses strong potential for extension to more complex real-world engineering scenarios.

2 Standard CPO

The Crested Porcupine Optimizer (CPO) [25] is inspired by the defensive behavior of crested porcupines. When threatened, crested porcupines employ multiple defense mechanisms, including four principal strategies: visual, auditory, olfactory, and physical attack, which are activated in ascending order of aggressiveness to deter predators. In CPO, the visual and auditory defense mechanisms are used to model exploration behavior, whereas the olfactory and physical attack mechanisms correspond to exploitation behavior. The core of the algorithm lies in simulating how crested porcupines select among these defensive strategies according to the type and intensity of threats, thereby achieving a balance between exploration and exploitation throughout the optimization process.

2.1 Initialization

In CPO, as in other metaheuristic algorithms, the initialization phase employs Equation (1) to randomly generate solution vectors that are uniformly distributed within the upper and lower bounds of the search space:

$$\vec{X}_i = \vec{L} + \vec{r} \times (\vec{U} - \vec{L}), \qquad i = 1, 2, \ldots, N \tag{1}$$

where $\vec{X}_i$ denotes the ith candidate solution; $\vec{L}$ and $\vec{U}$ are the lower and upper bound vectors of the search space, respectively; and $\vec{r}$ is a random vector whose elements are uniformly distributed on [0,1]. By applying the above formula to all individuals in the population, the initialized population can be expressed as:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,j} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,j} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & x_{i,2} & \cdots & x_{i,j} & \cdots & x_{i,d} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,j} & \cdots & x_{N,d} \end{bmatrix} \tag{2}$$

After initialization, the fitness of each individual is evaluated, and the one with the best fitness value is selected as the current global best solution.

2.2 Search over the solution space

During the population search and position-update process, the procedure is divided into two main phases: global exploration and local exploitation. The selection between these two phases is determined by comparing two randomly generated values, t_8 and t_9:

2.2.1 Exploration

If $t_8 \geq t_9$, the global exploration phase is activated. In this phase, one of two defense strategies is selected by further comparing two additional random values, t_6 and t_7.

When $t_6 < t_7$, the first defense strategy is activated, which is mathematically formulated by Equation (3):

$$\overrightarrow{x_i^{t+1}} = \overrightarrow{x_i^t} + t_1 \times \left| 2 \times t_2 \times \overrightarrow{x_{CP}^t} - \overrightarrow{y_i^t} \right| \tag{3}$$

Where $\overrightarrow{x_{CP}^t}$ denotes the best solution at evaluation step, and $\overrightarrow{y_i^t}$ is a vector constructed between the current CP and a randomly selected CP from the population and represents the predator’s position in iteration. t_1 is a random variable following a normal distribution, and t_2 is a random variable uniformly distributed in [0,1]. The generation of $\overrightarrow{y_i^t}$ is defined by the mathematical expression in Equation (4):

$$\overrightarrow{y_i^t} = \frac{\overrightarrow{x_i^t} + \overrightarrow{x_r^t}}{2} \tag{4}$$

Where r is a random integer in the range [1,N]. When $t_6 < t_7$, the second defense strategy is executed, whose mathematical model is given by Equation (5):

$$\overrightarrow{x_i^{t+1}} = (1 - \overrightarrow{U_1}) \times \overrightarrow{x_i} + \overrightarrow{U_1} \times \left(\overrightarrow{y} + t_3 \times \left(\overrightarrow{x_{r1}^t} - \overrightarrow{x_{r2}^t} \right) \right) \tag{5}$$

Where U_1 is a binary vector whose elements take values of either 0 or 1. In this second defense strategy, the predator’s movement direction is determined based on two randomly selected individuals, $\overrightarrow{x_{r1}^t}$ and $\overrightarrow{x_{r2}^t}$.

2.2.2 Exploitation

When $t_8 \geq t_9$, local exploitation is performed, which is realized by adopting either the third or the fourth defense strategy; the specific strategy is selected by comparing two random values, t_{10} and T_f.

When $t_{10} < T_f$, the third defense strategy is activated, which is mathematically defined in Equation (6):

$$\overrightarrow{x_i^{t+1}} = (1 - \overrightarrow{U_1}) \times \overrightarrow{x_i} + \overrightarrow{U_1} \times \left(\overrightarrow{x_{r1}^t} + S_i^t \times \left(\overrightarrow{x_{r2}^t} - \overrightarrow{x_{r3}^t} \right) - t_3 \times \overrightarrow{d} \times g_t \times S_i^t \right) \tag{6}$$

Here, r_3 denotes a random integer range $[1, N]$; t_3 is a random number in the interval $[0,1]$; d is a direction control vector, g_t is the defense factor, and S_i^t is the scent dispersion factor. The corresponding calculation is defined as follows:

$$\vec{d} = \begin{cases} +1, & \text{if } \overrightarrow{rand} \le 0.5 \\ -1, & \text{else} \end{cases} \tag{7}$$

$$g_t = 2 \times rand \times \left(1 - \frac{t}{t_{\max}}\right)^{\frac{t}{t_{\max}}} \tag{8}$$

$$S_i^t = \exp\left(\frac{f(x_i^t)}{\sum_{k=1}^{N} f(x_k^t) + \epsilon}\right) \tag{9}$$

Where $f(x_i^t)$ denotes the fitness value of the ith individual at iteration t, and ϵ is a small constant introduced to avoid division by zero. When $t_{10} > T_f$, the fourth defense strategy is activated, and its mathematical model is given by Equation (10):

$$\vec{x}_i^{t+1} = \vec{x}_{CP}^t + \left(a(1 - t_4) + t_4\right) \times \left(\vec{d} \times \vec{x}_{CP}^t - \vec{x}_i^t\right) - K \tag{10}$$

$$K = t_5 \times d \times g_t \times \vec{F}_i^t \tag{11}$$

Where a denotes the velocity convergence factor; t_4 and t_5 are random numbers uniformly distributed in $[0,1]$; and $\vec{F}_i^t$ represents the inelastic collision force generated by the individual when attacking the predator, which is computed as follows:

$$\vec{F}_i^t = \vec{t}_6 \times m_i \times \left(\vec{v}_i^{t+1} - \vec{v}_i^t\right) \tag{12}$$

Where t_6 is a random vector whose elements lie in $[0,1]$; $\vec{v}_i^{t+1}$ denotes a randomly selected individual from the current population; and $\vec{v}_i^t$ refers to the current individual.

2.3 Cyclic Population Reduction

After each iteration, the population size is gradually reduced to accelerate convergence. Once it reaches a predefined minimum, the population size is then progressively increased to restore diversity. This process is repeated cyclically until the maximum number of iterations is reached. This dynamic population adjustment constitutes a distinctive feature of the CPO algorithm. The population size updating rule is given by:

$$N = N_{\min} + \left(N_{\max} - N_{\min}\right) \times \left(1 - \left(\frac{t \% \frac{T_{\max}}{T}}{\frac{T_{\max}}{T}}\right)\right) \tag{13}$$

Where **%** denotes the modulo operator.

3 Adaptive Multi-Mechanism Integrated CPO

To overcome the inherent limitations of standard CPO, particularly the uneven distribution of the initial population, its tendency toward premature convergence, and the oscillatory behavior of its convergence trajectories, this study proposes the SDHCPO algorithm. SDHCPO establishes a unified cooperative optimization framework that aims to dynamically regulate the balance between exploration and exploitation through the integration of four strategies. This framework follows a clear functional logic: in the exploration-dominated stage, the algorithm incorporates differential evolution and horizontal-vertical crossover strategies, which proactively inject diversity and break positional dependence, thereby expanding global search capability and preventing the population from collapsing prematurely into local optima. As the algorithm transitions to the exploitation-dominated stage, particularly in the fourth defense mechanism, a cosine-annealing mechanism is introduced. This strategy focuses on stabilizing the convergence trajectory by replacing random fluctuations with a deterministic decay schedule, thereby eliminating ineffective oscillatory jumps and enforcing precise convergence of the population toward the global optimum. Within this framework, a staged configuration that strengthens exploration in the early phase to expand the search boundary and emphasizes exploitation in the later phase to refine convergence accuracy, together with the complementary roles of each mechanism in diversity injection and trajectory stabilization, achieves a dynamically coordinated and finely balanced interaction between exploration and exploitation. Meanwhile, the Sobol-OBL initialization ensures from the outset a high-quality, uniformly distributed initial population in the search space. The theoretical foundations of each strategy and their synergistic effects are elaborated in the following sections.

3.1 Sobol-OBL Initialization

In swarm intelligence optimization algorithms, the diversity and uniformity of the initial population directly influence the global search capability and convergence speed. Although conventional random initialization is easy to implement, it often leads to uneven individual distribution and insufficient coverage in high-dimensional search spaces, making the algorithm prone to becoming trapped in local optima. Low-discrepancy sequences (LDS), owing to their uniform coverage properties, have therefore been widely adopted in the initialization stage. Among them, the Sobol sequence exhibits particularly good uniformity even in high-dimensional spaces. Nevertheless, Sobol-based initialization alone still fails to fully explore mutually opposite regions of the solution space. The Opposition-Based Learning (OBL) [32] mechanism enhances population diversity and enlarges the search range by generating symmetric opposite solutions of the current candidates within the search space. On this basis, a high-dimensional initialization strategy that combines the Sobol sequence with OBL (Sobol-OBL) is proposed. The Sobol sequence ensures globally uniform coverage, while OBL complements it by introducing solutions in symmetric regions. Together with a fitness-based ranking and selection mechanism, this strategy provides higher-quality initial solutions for subsequent iterations.

The Sobol sequence is a low-discrepancy quasi-random sequence that distributes points as uniformly as possible within a d dimensional hypercube. For a population of size N, the original value of the jth dimension of the ith Sobol sequence point is denoted by $\vec{S}_{q,i}$ and can be expressed as:

$$\vec{S}_{q,i} = \left[\sum_{k=1}^{M} \frac{b_k(i) \oplus a_{1,k}}{2^k}, \sum_{k=1}^{M} \frac{b_k(i) \oplus a_{2,k}}{2^k}, \ldots, \sum_{k=1}^{M} \frac{b_k(i) \oplus a_{d,k}}{2^k} \right] \tag{14}$$

In this expression, $b_k(i)$ denotes the kth bit in the binary representation of i, $a_{j,k}$ is the kth direction number for the jth dimension, and M is the bit length used in the binary representation. Subsequently, $\vec{S}_{q,i}$ is linearly mapped onto the actual search interval $[L,U]$ to obtain the Sobol-initialized individual:

$$\vec{X}_i^{\text{sobol}} = \vec{L} + \vec{S}_{q,i} \times (\vec{U} - \vec{L}), \quad i = 1, 2, \ldots, N \tag{15}$$

To illustrate the superiority of the Sobol sequence, 100 points are generated within the two-dimensional domain $[0,1]^2$, as shown in Figure 1. Visually, the points generated by the Sobol sequence exhibit markedly better uniformity and diversity compared with randomly generated points—an advantage that extends to high-dimensional spaces.

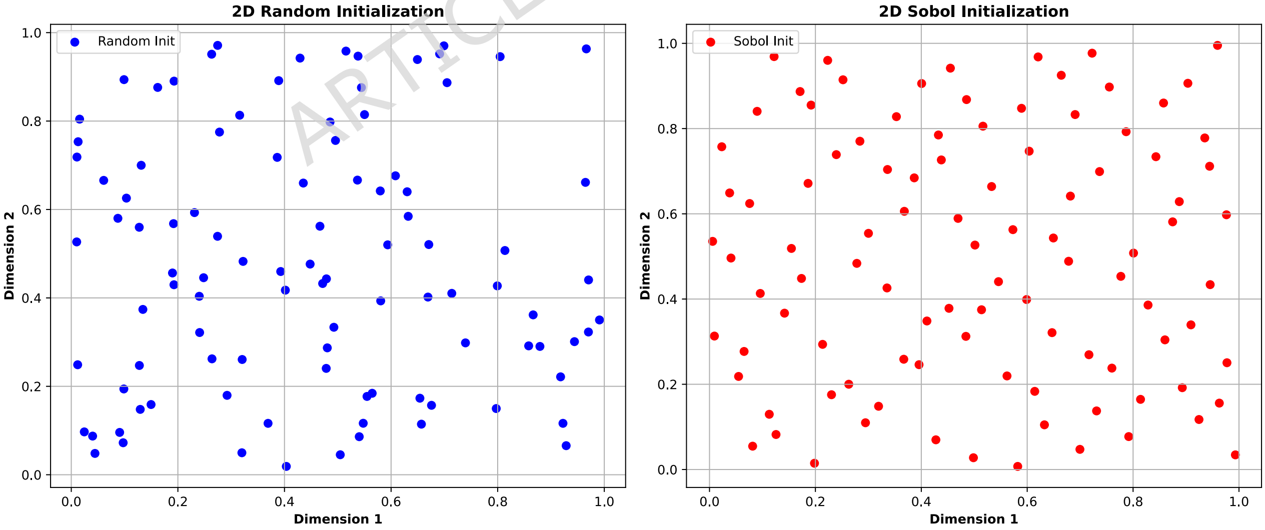


Fig. 1 Comparison of distribution between Sobol sequence and random initialization

Next, the OBL mechanism is introduced. For each initial individual $\vec{X}_i^{sobol}$, its opposite component in the jth dimension is defined as:

$$\vec{X}_i^{\text{op}} = \vec{L} + \vec{U} - \vec{X}_i^{\text{sobol}}, \quad i = 1, 2, \ldots, N \tag{16}$$

which maps the original value symmetrically with respect to the midpoint of the search space [$\vec{L},\vec{U}$]. Equation (16) generates the opposite solution $\vec{X}_i^{op}$ corresponding to $\vec{X}_i^{sobol}$; the two typically appear as a pair, enabling simultaneous exploration of opposite regions in the solution space and enhancing the global diversity of the population.

After Sobol initialization, the fitness values of the N Sobol individuals are first evaluated. The median fitness is then used as an automatically determined threshold to distinguish relatively good and poor solutions. For those individuals whose fitness is not better than the median, their opposite solutions are generated according to the OBL rule, and a replacement is performed only if the opposite solution yields a lower fitness value. In this way, opposition-based learning is selectively applied to the worse half of the initial Sobol population, improving solution quality while preserving the diversity of the better half. The resulting N individuals form the initialized population. The mathematical expression is:

$$\vec{X}_i = \begin{cases} \vec{X}_i^{sobol}, & f\left(\vec{X}_i^{sobol}\right) \le \mathrm{med}\left\{f(\vec{X}_k^{sobol})\right\}_{k=1}^{N}, \\ \arg\min_{\vec{Z} \in \{\vec{X}_i^{sobol}, \vec{X}_i^{op}\}} f(\vec{Z}), & otherwise, \end{cases} \qquad i = 1, \dots, N \tag{17}$$

where $\mathrm{med}\left\{f(\vec{X}_k^{sobol})\right\}_{k=1}^{N}$ denotes the median of the fitness values of all Sobol-initialized individuals, $\vec{X}_i^{op}$ is the opposite solution of $\vec{X}_i^{sobol}$ defined in (16), and $\vec{X}_i$ denotes the initialized individual obtained after the selective application of OBL.

3.2 Differential Evolution

Building on the high-quality initialization, the algorithm then enters the iterative search phase. In standard CPO, the first defense phase simulates the defensive response to predators; however, the unidirectional movement toward the current best solution can rapidly erode population diversity and induce premature convergence. To counterbalance this attraction without hindering convergence, the DE/rand/1 strategy [33] is specifically incorporated into this phase. In contrast to the original mechanism, which reinforces dependence on the current leader, the DE strategy introduces mutation and crossover operators to perturb candidate solutions. This perturbation mechanism breaks positional dependence, enabling the population to maintain sufficient diversity to explore previously unvisited regions, thereby continuously strengthening global search capability and preventing entrapment in local optima during the early stages of evolution. The steps of this strategy are as follows:

Step1: Select three distinct individuals $\vec{x}_{r1}^{t}$, $\vec{x}_{r2}^{t}$, and $\vec{x}_{r3}^{t}$ from the population($r1 \ne r2 \ne r3 \ne i$); then generate the mutant vector $\vec{v}_i^{t+1}$ using the DE/rand/1 mutation strategy as follows:

$$\vec{v}_i^{t+1} = \vec{x}_{r1}^{t} + CV \times (\vec{x}_{r2}^{t} - \vec{x}_{r3}^{t}) \tag{18}$$

Where CV is the mutation factor, which controls the magnitude of the perturbation.

Step2: Perform binomial crossover between the target individual $\vec{x}_i^t$ and the mutant vector $\vec{v}_i^{t+1}$ to generate the trial individual $\vec{u}_i^{t+1}$, as shown in Equation(18), where the crossover probability is governed by the parameter CR.

$$\vec{u}_{i,j}^{t+1} = \begin{cases} \vec{v}_{i,j}^{t+1}, & \text{if } rand(0,1) \le CR \text{ } or \text{ } j = j_{rand} \\ \vec{x}_{i,j}^{t}, & otherwise \end{cases} \tag{19}$$

Where j is the dimension index, $CR \in [0,1]$ is the crossover probability, and j_{rand} denotes a randomly selected dimension.

Step 3: Apply a greedy selection mechanism by comparing the fitness of the trial individual $\vec{u}_i^{t+1}$ with that of the target individual $\vec{x}_i^t$, as shown in Equation (20).

$$\vec{x}_i^{t+1} = \begin{cases} \vec{u}_i^{t+1}, & \text{if } f(\vec{x}_i^{t}) > f(\vec{u}_i^{t+1}) \\ \vec{x}_i^{t}, & otherwise \end{cases} \tag{20}$$

3.3 Horizontal-Vertical Crossover Strategy

Although the Differential Evolution strategy effectively disrupts positional dependence and mitigates early clustering, it treats individuals as holistic vectors and may therefore overlook stagnation in specific dimensions. Consequently, the second defense phase requires a dedicated mechanism to explicitly address dimensional stagnation.

In this phase, the original position update equation may cause certain dimensions of the solution vector to become inactive, thereby impeding progress toward the global optimum. To effectively eliminate this stagnation while preserving the enhanced global search capability achieved in the preceding stage, the Horizontal-Vertical Crossover strategy is introduced [34]. This strategy operates through a dual mechanism: horizontal crossover facilitates information exchange between distinct individuals to eliminate search blind spots, whereas vertical crossover explicitly targets stagnant dimensions, employing arithmetic crossover to enable them to escape local extrema. The horizontal crossover is performed using the following formula (21):

$$\vec{x}_i^{t+1} = h_1 \times \vec{x}_i^{t} + (1 - t_6) \times \vec{x}_r^{t} + c \times (\vec{x}_i^{t} - \vec{x}_r^{t}) \tag{21}$$

Where $\vec{x}_r^t$ denotes a randomly selected individual with an index different from i; $\eta_1 \in (0,1)$ is a random number, and c is a constant controlling the contribution of dimensional differences. The newly generated individual is then compared with the target individual, and the one with the smaller objective function value is retained.

Vertical crossover is an arithmetic crossover operator acting on two different dimensions of a single individual. In each crossover operation, only one dimension of the individual is updated while the others remain unchanged, thereby allowing stagnant dimensions to escape local optima without perturbing dimensions that may have already approached the optimum. Let two dimensions $j1$ and $j2$ be randomly selected. Then, the $j1$th dimension of the new individual is computed using the following formula (22):

$$\overrightarrow{x}_{i,j1}^{t+1} = \eta_2 \cdot \overrightarrow{x}_{i,j1}^{t} + (1-\tau_7) \cdot \overrightarrow{x}_{i,j2}^{t} \tag{22}$$

Where η_2 is a uniformly distributed random number in the interval (0,1).

3.4 Cosine Annealing Dynamic Adjustment Strategy

Following the extensive exploration and dimensional adjustments in the preceding phases, the algorithm must subsequently shift its focus toward stabilization and precise convergence. However, in the fourth defense strategy of the original CPO, the position update relies on random coefficients τ_4. The stochastic fluctuations of these terms can destabilize the update direction and lead to inconsistent convergence speed. In the improved CPO, a time-dependent nonlinear decay mechanism is introduced, whereby the original random terms are replaced with a deterministic weighting function. The core of this mechanism is a tunable cosine-annealing function:

$$W(t) = 0.5 \times \left(1 + \cos\left(\pi \times \left(\frac{t}{T_{\max}}\right)^{\gamma}\right)\right) \tag{23}$$

In this equation, the decay-rate control parameter $\gamma > 0$ governs the nonlinear attenuation behavior, while the weighting function $W(t)$ adapts over time according to the normalized iteration index $t/T_{\max}$. Its main advantage lies in the exponent term, which provides precise control over the shape of the decay curve: increasing γ produces a gentler decay in the early iterations, thereby preserving sufficient exploration in the initial phase; in the later iterations, the function transitions to a more rapid decay, strengthening fine-grained search around the vicinity of the optimum. As shown in Figure 2, larger values of γ lead to slower decay at the beginning and a steeper drop in the final phase. Benchmark experiments further confirm that $\gamma = 1.5$ yields the best overall performance in most cases. Consequently, at this stage, the high-quality individuals generated by the preceding strategies are efficiently exploited, and the focus of the search is gradually and deterministically shifted from maintaining diversity to fine-grained solution refinement. This unified decay trajectory prevents exploration and exploitation from being treated as disjoint processes and instead enables their continuous coordination over time. The position update formula for the fourth defense strategy is given by:

$$\vec{X}_i^{t+1} = \vec{X}_{CP}^{t} + \underbrace{0.5 \times \left(1+\cos\left(\pi \times \left(\frac{t}{T_{\max}}\right)^{\gamma}\right)\right)}_{W(t)} \times \left(\delta \times \vec{X}_{CP}^{t} - \vec{X}_i^{t}\right) - \tau_5 \times \delta \times \gamma_t \times \vec{F}_i^{t} \tag{24}$$

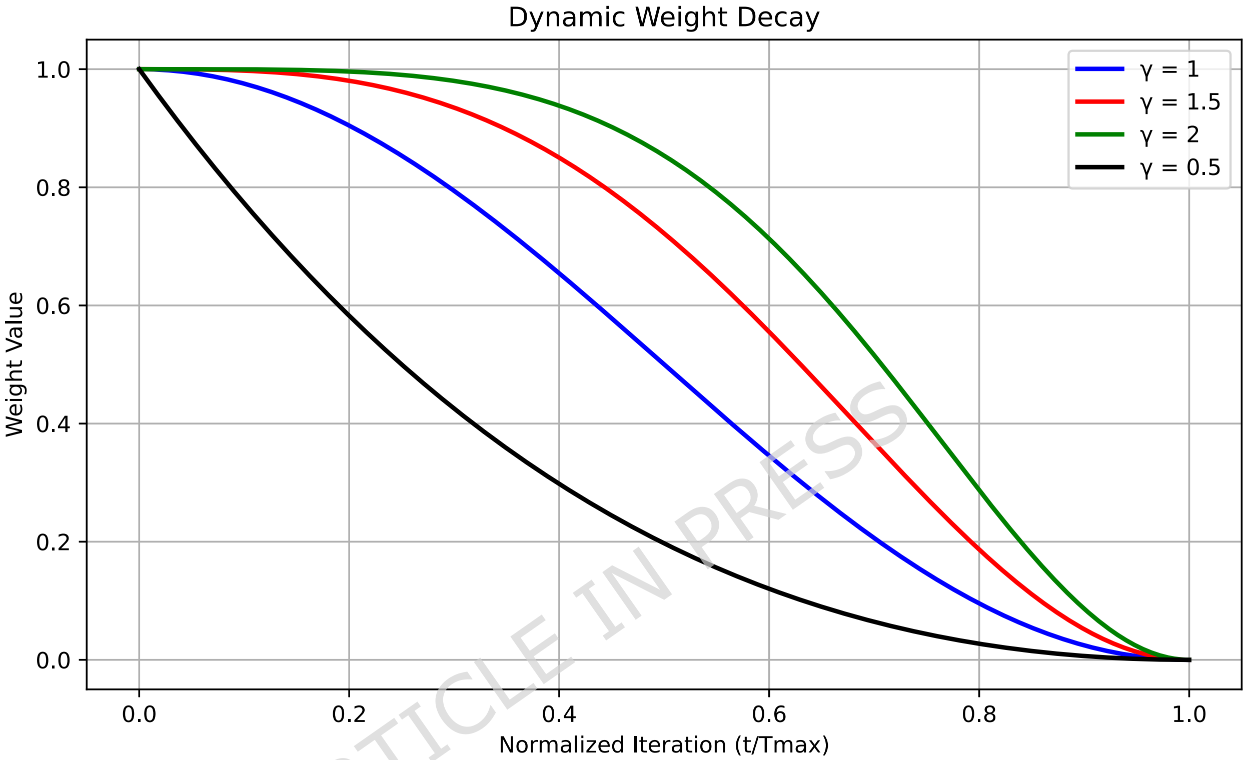


Fig. 2 Dynamic weight decay curves under different decay factors γ

3.5 Pseudocode and Flowchart of SDHCPO

This section presents the pseudocode of SDHCPO together with its corresponding flowchart. The overall workflow of the proposed algorithm is summarized in Figure 3.

Algorithm1 The pseudo-code of the SDHCPO algorithm.

```
Input: Parameters of CPO, such as $N_{\max}$, $T_{\max}$
Output: The global best solution
Set parameters $N_{\max}$, $N_{\min}$, T, $T_f$, $T_{\max}$, $g$.
Applying Sobol-OBL Initialization per Equation (17) to generate the position matrix..
While (t<$T_{\max}$) do
    Evaluate the fitness of the initial population and find the global best solution ( $\overrightarrow{X}_{CP}^{t}$ ).
    Equation (8) defines the update rule for $g_t$.
    Equation (23) defines the update rule for W(t).
    Update the population size according to Equation (13).
    For $i$=1 to N do
        Update parameters m, S, F, $d$.
        Generate a random number, $h_3$.
        If $h_3$ <0.5 // Exploration
            Generate a sequence of random numbers, $h_4$, $h_5$ and $h_6$.
            If $h_4$ <0.5 // First defense strategy
                Carry out Algorithm2.
                Adopting greedy selection strategy to optimize the population.
            Else // Second defense strategy
                If $h_5 < 0.5$ // Horizontal and Vertical Intersection
                    Carry out Algorithm3.
                Else
                    The CP position is adjusted via Equation (5)
                End if
            End if
        Else // Exploitation
            Generate a random number, $h_7$.
            If $h_7 < 0.5$ // Third defense strategy
                The CP position is adjusted via Equation (6)
            Else // Fourth defense strategy
                The CP position is adjusted via Equation (24)
            End if
        End if
        If $f\left(\overrightarrow{X}_{i}^{t+1}\right) > f\left(\overrightarrow{X}_{i}^{t}\right)$
            $\overrightarrow{X}_{i}^{t+1} = \overrightarrow{X}_{i}^{t}$
        End if
         $t = t + 1$
        End for
    End while
    Return $\overrightarrow{X}_{CP}^{t}$
    Output the global best solution
```

Algorithm2 DE/rand/1/bin

```
Selected three random donor vectors $\overrightarrow{x}_{r1}^{t}$ , $\overrightarrow{x}_{r2}^{t}$ , $\overrightarrow{x}_{r3}^{t}$ , where $r1$ ¹ $r2$ ¹ $r3$ ¹ $i$ .
Select random integer $j$ rand within $[1, d]$.
For $j=0$; $j<d$; $j=j+1$ do
    If $rand(0,1)$ £ $CR$ or $j=j_{rand}$ then
        $\overrightarrow{u}_{i,j}^{t+1}=\overrightarrow{v}_{i,j}^{t+1}$
    Else
        $\overrightarrow{u}_{i,j}^{t+1}=\overrightarrow{x}_{i}^{t}$
    End if
End for
```

Algorithm3 Horizontal and Vertical Intersection

```
If $h_6<0.5$ // Horizontal Intersection
    Randomly generate an individual from the population that is different from the current.
    The position of CP is updated according to Equation(21).
    Adopting greedy selection strategy to optimize the population.
Else // Vertical Intersection
    If d ³ 2
        Randomly select two different dimensions of the current solution vector.
        The position of CP is updated according to Equation(22).
        Adopting greedy selection strategy to optimize the population.
    End if
End if
```

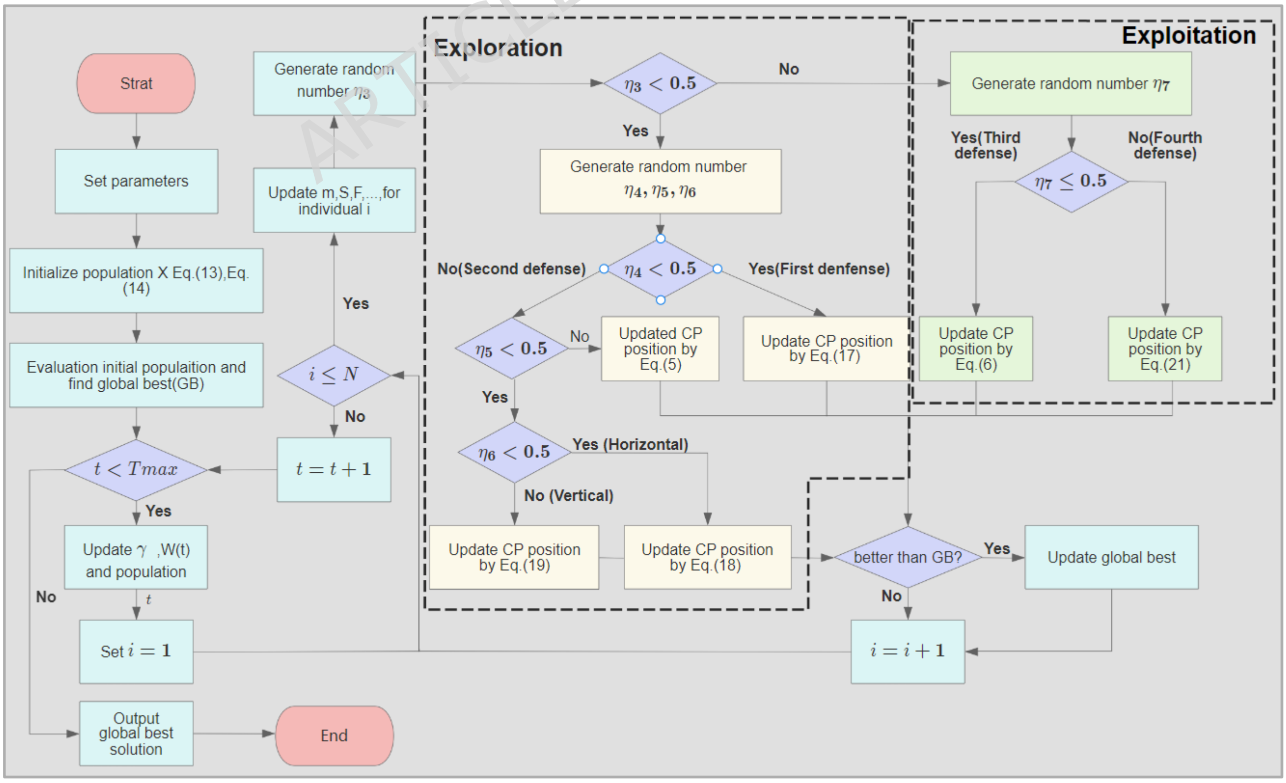

Exploration
Exploitation
Strat
Set parameters
Initialize population X Eq.(13),Eq.(14)
Evaluation initial populaition and find global best(GB)
$t < Tmax$
Yes
No
Update γ ,W(t) and population
t
Set $i = 1$
Output global best solution
End
Generate random number η_3
Update m,S,F,...,for individual i
Yes
$i \leq N$
No
$t = t + 1$
$\eta_3 < 0.5$
No
Yes
Generate random number η_4, η_5, η_6
No(Second defense)
$\eta_4 < 0.5$
Yes(First denfense)
$\eta_5 < 0.5$
No
Updated CP position by Eq.(5)
Update CP position by Eq.(17)
Yes
$\eta_6 < 0.5$
Yes (Horizontal)
No (Vertical)
Update CP position by Eq.(19)
Update CP position by Eq.(18)
Generate random number η_7
Yes(Third defense)
No(Fourth defense)
$\eta_7 \leq 0.5$
Update CP position by Eq.(6)
Update CP position by Eq.(21)
better than GB?
Yes
No
Update global best
$i = i + 1$

Fig. 3 Flowchart of the SDHCPO algorithm

3.6 Time Complexity Analysis

Time complexity is a key metric for evaluating the performance of optimization algorithms [35]. The time complexity of the SDHCPO algorithm is primarily determined by the population initialization and iterative update processes, with the main influencing factors being the maximum number of iterations $T_{\max}$, the problem dimensionality d, and initial population size N. In SDHCPO, population initialization combines the Sobol sequence with opposition-based learning, yielding a time complexity of $O(N \times d)$.The iterative update process employs a hybrid strategy, which includes Differential Evolution, Horizontal-Vertical Crossover, and cosine-annealing-based dynamic adjustment. This process has a time complexity of $O(T_{\max} \times N \times d)$, where N is dynamically adjusted through cyclic population reduction. In the best-case scenario: $O(SDHCPO) = O(N \times d) + O(T_{\max} \times N_{\min} \times d)$.

In the worst-case scenario: $O(SDHCPO) = O(N \times d) + O(T_{\max} \times N \times d)$.

4 Benchmark Test Results and Analysis

4.1 Experimental Setup and Overall Evaluation Framework

This section conducts a multi-level evaluation of the proposed SDHCPO algorithm, following a logical progression from overall performance, to internal mechanisms, and finally to high-dimensional extensions. First, SDHCPO is compared with several representative metaheuristic algorithms on the CEC2017 and CEC2022 benchmark test suites, in order to assess its overall competitiveness across different types of test functions; the CEC2017 suite is tested in 30 dimensions, whereas CEC2022 is evaluated in 20 dimensions. Second, a systematic ablation study and qualitative analysis on CEC2017 are performed to characterize the exploration-exploitation balance from an internal mechanism perspective and to clarify the roles and synergies of the four integrated strategies. Finally, a 50-dimensional extension experiment based on CEC2017 is carried out to examine the scalability and robustness of SDHCPO in higher-dimensional scenarios.

The performance of the improved algorithm is evaluated on two widely adopted benchmark test suites, CEC2017 [36] and CEC2022 [37]. CEC2017 comprises unimodal functions (UM: F1, F3; F2 has been officially removed), multimodal functions (MM: F4-F10), hybrid functions (H: F11-F20), and composition functions (C: F21-F30). CEC2022 includes unimodal functions (UM: F1), multimodal functions (MM: F2-F5), hybrid functions (H: F6-F8), and composition functions (C: F9-F12). To verify the effectiveness of SDHCPO, seven algorithms are selected as baselines: the original CPO [25], CFOA [38], PKO [39], CDO [40], MVO [41], HOA [42], and WOA [10], all implemented with the parameter settings recommended in their original publications; see Appendix A for details. Among them, CFOA, PKO, and CDO are recently proposed metaheuristic algorithms that represent the state of the art in this class of methods, whereas MVO, WOA, and HOA are classical swarm intelligence algorithms that have been widely used in the literature and frequently employed on CEC benchmarks and engineering optimization problems. By simultaneously including both recently proposed algorithms and classical representative methods, the competitiveness of SDHCPO can be assessed more convincingly. For each test function, all algorithms are independently run 30 times with a population size of 30 and a maximum of 500 iterations. All experiments are implemented in MATLAB R2022a on a workstation equipped with an AMD Ryzen 7 4800U 1.80 GHz processor and 16 GB of RAM.

Unless otherwise specified, statistical tests are conducted using the Wilcoxon rank-sum test [43] with SDHCPO as the reference method and a significance level of 0.05. In the “−/=/+” notation used throughout the paper, “−” indicates that SDHCPO is significantly inferior to the compared algorithm, “=” denotes no statistically significant difference, and “+” indicates that SDHCPO is significantly superior to the compared algorithm. The corresponding counts are used to summarize the overall advantage of SDHCPO relative to each competitor.

4.2 Results on the CEC2017 Benchmark Suite

In this subsection, the overall performance of SDHCPO is first compared with that of seven representative algorithms on the more challenging 30-dimensional CEC2017 test suite. Table 1 reports the mean, standard deviation, and Friedman ranking [44] of each algorithm on all CEC2017 test functions, with the smallest mean values highlighted in bold. It can be observed that SDHCPO achieves clearly superior mean objective values on most test functions and exhibits consistently stable performance across unimodal, multimodal, hybrid, and composition functions, indicating strong overall adaptability and cross-function robustness in the 30-dimensional benchmark setting.

Table 1. Performance Metrics of SDHCPO and Other Algorithms on CEC2017 (d=30)

Index	SDHCPO	CPO	CFOA	PKO	CDO	MVO	HOA	
Std	**8.826E+03**	5.932E+05	8.389E+09	5.573E+06	2.743E+08	6.298E+05	7.917E+09	1.9
Mean	**1.202E+04**	7.377E+05	4.636E+10	5.510E+06	5.287E+10	2.041E+06	3.911E+10	5.5
Std	9.399E+03	1.390E+04	7.551E+04	3.835E+04	**3.339E+03**	1.059E+04	8.119E+03	6.2
Mean	4.103E+04	6.263E+04	2.307E+05	1.703E+05	9.235E+04	**2.040E+04**	7.058E+04	2.4
Std	1.489E+01	2.335E+01	3.158E+03	2.315E+01	9.701E+01	**1.054E+01**	1.760E+03	4.6
Mean	5.058E+02	5.186E+02	1.070E+04	5.188E+02	5.570E+03	**4.995E+02**	7.802E+03	1.3
Std	2.196E+01	**1.598E+01**	4.181E+01	1.935E+01	1.631E+01	3.587E+01	3.028E+01	5.2
Mean	**5.922E+02**	6.944E+02	8.940E+02	5.928E+02	8.614E+02	6.175E+02	8.149E+02	8.5
Std	**4.659E-02**	7.423E-01	1.029E+01	3.351E+00	7.029E+00	1.479E+01	8.613E+00	9.8
Mean	**6.001E+02**	6.019E+02	6.832E+02	6.037E+02	6.752E+02	6.321E+02	6.657E+02	6.8
Std	2.053E+01	**1.674E+01**	1.402E+02	2.658E+01	1.776E+01	4.459E+01	6.224E+01	7.8
Mean	**8.585E+02**	9.398E+02	1.552E+03	8.752E+02	1.323E+03	8.830E+02	1.255E+03	1.2
Std	2.003E+01	**1.412E+01**	5.548E+01	2.229E+01	2.181E+01	3.477E+01	3.552E+01	4.0
Mean	**8.883E+02**	9.837E+02	1.135E+03	9.000E+02	1.110E+03	9.229E+02	1.060E+03	1.0
Std	**4.930E+01**	4.924E+02	3.457E+03	3.835E+02	6.634E+02	3.614E+03	1.806E+03	3.5
Mean	**9.336E+02**	1.315E+03	1.142E+04	1.261E+03	1.014E+04	6.879E+03	7.309E+03	1.2
Std	7.132E+02	4.159E+02	5.475E+02	5.968E+02	**3.150E+02**	6.698E+02	6.026E+02	9.1
Mean	5.648E+03	7.539E+03	9.206E+03	5.303E+03	9.009E+03	**4.912E+03**	7.619E+03	7.4
Std	**2.361E+01**	2.969E+01	4.992E+03	1.035E+02	1.254E+04	7.215E+01	2.319E+03	3.8
Mean	**1.226E+03**	1.274E+03	1.469E+04	1.418E+03	2.801E+04	1.362E+03	6.413E+03	1.0
Std	**3.686E+05**	9.060E+05	3.457E+09	2.508E+06	1.162E+08	9.680E+06	1.880E+09	3.1
Mean	**7.126E+05**	1.306E+06	8.340E+09	2.529E+06	9.876E+09	1.296E+07	6.585E+09	5.5
Std	**7.481E+03**	1.024E+04	3.656E+09	1.415E+05	1.325E+08	1.070E+05	1.695E+09	1.7
Mean	**1.394E+04**	2.466E+04	5.387E+09	7.805E+04	2.494E+09	1.403E+05	3.068E+09	1.5
Std	2.432E+03	**3.212E+02**	4.833E+06	7.881E+04	1.652E+05	4.673E+04	1.099E+06	3.4
Mean	3.180E+03	**1.944E+03**	5.595E+06	1.017E+05	2.843E+06	4.968E+04	1.832E+06	2.5
Std	**2.725E+03**	2.950E+03	2.909E+08	1.583E+04	7.055E+07	3.870E+04	1.292E+08	2.2
Mean	**4.219E+03**	5.125E+03	3.019E+08	2.201E+04	6.392E+08	7.310E+04	1.514E+08	1.2
Std	2.527E+02	**2.446E+02**	6.112E+02	2.614E+02	2.430E+03	3.046E+02	5.549E+02	6.5
Mean	**2.510E+03**	3.079E+03	4.649E+03	2.528E+03	9.488E+03	2.914E+03	4.649E+03	4.2
Std	1.279E+02	**1.047E+02**	9.196E+02	1.577E+02	1.783E+04	1.846E+02	3.690E+02	2.9

Mean	**1.879E+03**	2.050E+03	3.577E+03	2.121E+03	2.071E+04	2.268E+03	2.858E+03	2.7
Std	8.039E+04	**6.307E+04**	2.957E+07	1.199E+06	1.653E+06	6.035E+05	1.536E+07	1.4
Mean	**9.408E+04**	1.167E+05	3.295E+07	1.354E+06	1.056E+07	7.455E+05	1.352E+07	1.2
Std	5.836E+03	**4.334E+03**	4.390E+08	2.231E+04	6.524E+06	1.780E+06	3.770E+07	1.8
Mean	**5.566E+03**	6.267E+03	5.421E+08	2.477E+04	1.434E+08	2.157E+06	3.055E+07	1.9
Std	**1.255E+02**	1.616E+02	2.236E+02	1.723E+02	2.045E+02	1.782E+02	2.073E+02	2.1
Mean	**2.246E+03**	2.441E+03	3.074E+03	2.428E+03	3.011E+03	2.554E+03	2.767E+03	2.9
Std	2.093E+01	**1.421E+01**	4.746E+01	2.135E+01	1.918E+01	3.179E+01	2.772E+01	5.6
Mean	**2.390E+03**	2.481E+03	2.696E+03	2.397E+03	2.644E+03	2.423E+03	2.618E+03	2.6
Std	3.541E+00	**3.286E+00**	1.660E+03	2.127E+03	1.028E+03	1.649E+03	1.102E+03	1.4
Mean	**2.302E+03**	2.309E+03	9.030E+03	5.143E+03	1.023E+04	5.700E+03	8.134E+03	8.2
Std	2.201E+01	**1.540E+01**	1.287E+02	2.609E+01	8.481E+01	4.938E+01	1.314E+02	1.2
Mean	**2.742E+03**	2.849E+03	3.299E+03	2.754E+03	3.773E+03	2.789E+03	3.502E+03	3.1
Std	2.625E+01	**2.294E+01**	8.506E+01	2.239E+01	4.401E+01	2.934E+01	1.578E+02	9.6
Mean	**2.898E+03**	3.020E+03	3.450E+03	2.911E+03	3.846E+03	2.927E+03	3.824E+03	3.3

Table 1 Cont.

Fun	Index	SDHCPO	CPO	CFOA	PKO	CDO	MVO	HOA	WOA
F25	Std	**1.050E+01**	1.827E+01	7.489E+02	1.570E+01	3.044E+01	2.054E+01	2.122E+02	8.017E+01
	Mean	**2.895E+03**	2.910E+03	5.421E+03	2.916E+03	3.620E+03	2.906E+03	3.833E+03	3.222E+03
F26	Std	2.912E+02	8.708E+02	9.246E+02	**2.361E+02**	3.475E+02	6.543E+02	8.188E+02	8.084E+02
	Mean	4.721E+03	5.364E+03	1.037E+04	**4.600E+03**	9.049E+03	4.808E+03	9.515E+03	8.765E+03
F27	Std	**8.985E+00**	1.347E+01	2.247E+02	1.058E+01	4.526E+01	2.201E+01	2.061E+02	1.521E+02
	Mean	3.227E+03	3.277E+03	3.905E+03	**3.223E+03**	3.699E+03	3.235E+03	4.194E+03	3.488E+03
F28	Std	**2.115E+01**	2.221E+01	8.483E+02	2.752E+01	3.567E+01	3.863E+01	4.340E+02	3.401E+02
	Mean	**3.253E+03**	3.285E+03	6.866E+03	3.291E+03	5.056E+03	3.259E+03	5.830E+03	3.920E+03
F29	Std	**1.192E+02**	1.578E+02	1.091E+03	1.462E+02	3.756E+02	2.169E+02	7.567E+02	6.425E+02
	Mean	**3.662E+03**	3.997E+03	6.717E+03	3.840E+03	6.574E+03	4.031E+03	6.288E+03	5.513E+03
F30	Std	**1.369E+04**	7.121E+04	4.820E+08	1.676E+05	8.595E+08	3.913E+06	2.704E+08	1.111E+08
	Mean	**2.454E+04**	1.160E+05	6.293E+08	1.087E+05	3.126E+09	4.939E+06	4.026E+08	9.166E+07
Std rank		**1.93**	2.10	7.34	3.31	4.59	4.38	6.07	6.28
Mean rank		**1.24**	2.97	7.45	2.79	6.93	3.10	5.93	5.55
Overall rank		**1**	2	8	3	5	4	7	6

As shown by the numerical results in Table 1, SDHCPO achieves noticeable improvements in both the mean and standard deviation of the objective values, with the reduction in the mean being particularly pronounced. For several functions, such as F1, F9, F12, F18, and F30, the mean objective values of SDHCPO are lower than those of the original CPO and other metaheuristic algorithms by one or even multiple orders of magnitude. For the multimodal functions F4-F10, algorithms such as the original CPO, CFOA, and CDO tend to become trapped in local minima within the complex multimodal landscape as the dimensionality increases, leading to generally higher mean objective values. In contrast, SDHCPO is able to escape local optima while maintaining adequate exploration intensity, thereby keeping the final mean values at a markedly lower level. This advantage is further amplified on the hybrid functions F11-F20 and the composition functions F21-F30. It can be observed that the mean values of most competing algorithms deteriorate significantly on these two categories of functions, with some objective values even reaching the order of 10^8-10^9, whereas the mean values of SDHCPO remain far below this range. These results indicate that, compared with other competing algorithms, SDHCPO exhibits substantially stronger global search capability and higher convergence accuracy.

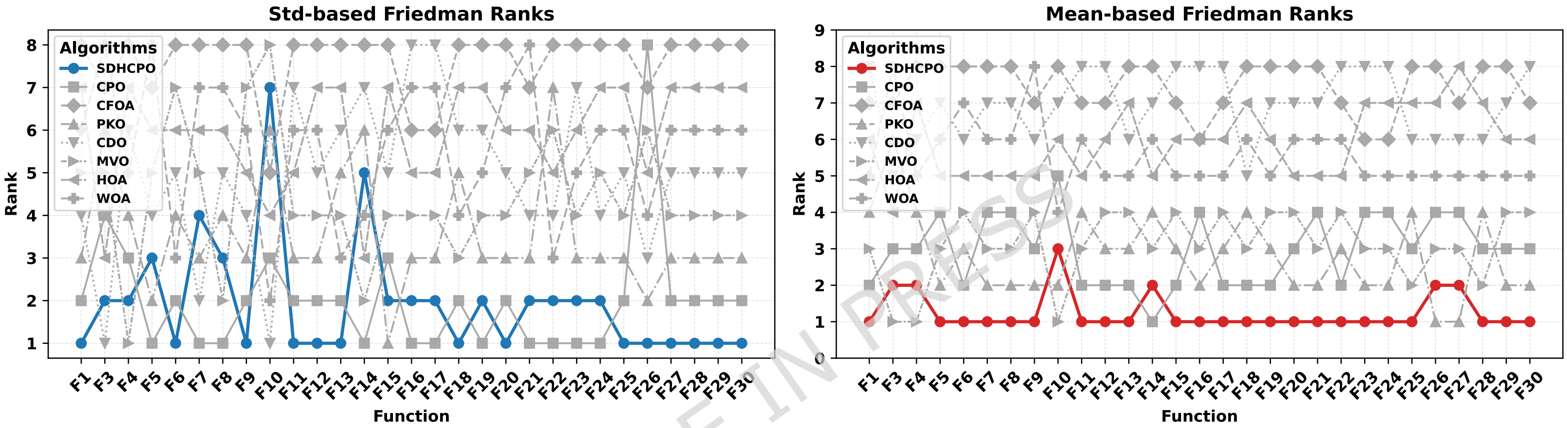


Fig. 4 Friedman Rank Evolution of SDHCPO vs. Competitors on CEC2017 Test Suite (d=30)

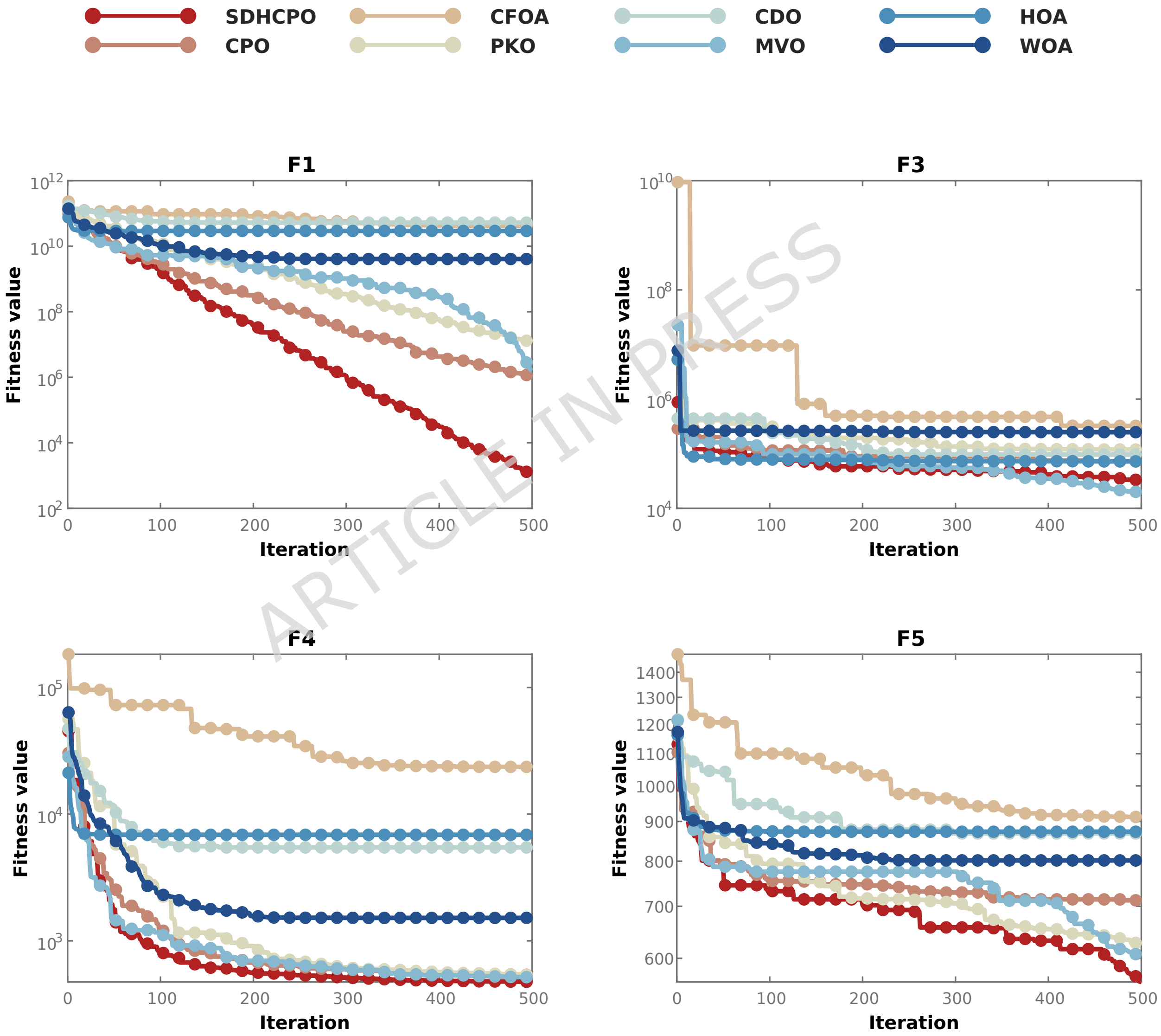
Figure 4 presents the per-function ranking distribution of the eight algorithms on the 30 test functions of CEC2017. It can be observed that SDHCPO consistently ranks among the top methods on almost all test functions, with only minor fluctuations on a few cases, which further corroborates, from a ranking perspective, the high accuracy and robustness indicated by the preceding statistical results.

Table 2. Wilcoxon Rank-Sum p-Values: SDHCPO vs. Algorithms on CEC2017 (d=30)

Fun	CPO-p	CFOA-p	PKO-p	CDO-p	MVO-p	HOA-p	WOA-p
1	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11
3	4.118E-06	3.020E-11	3.020E-11	3.020E-11	1.558E-08	1.777E-10	3.020E-11
4	4.982E-04	3.020E-11	1.041E-04	3.020E-11	2.116E-01	3.020E-11	3.020E-11
5	3.020E-11	3.020E-11	1.087E-01	3.020E-11	3.339E-03	3.020E-11	3.020E-11
6	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11

7	3.020E-11	3.020E-11	1.767E-03	3.020E-11	2.062E-01	3.020E-11	3.020E-11
8	3.020E-11	3.020E-11	8.650E-01	3.020E-11	5.971E-05	3.020E-11	3.020E-11
9	6.696E-11	3.020E-11	5.072E-10	3.020E-11	3.020E-11	3.020E-11	3.020E-11
10	3.020E-11	3.020E-11	7.959E-03	3.020E-11	9.031E-04	1.329E-10	3.159E-10
11	5.533E-08	3.020E-11	6.121E-10	3.020E-11	2.390E-08	3.020E-11	3.020E-11
12	4.060E-02	3.020E-11	4.183E-09	3.020E-11	9.919E-11	3.020E-11	3.020E-11
13	2.681E-04	3.020E-11	2.879E-06	3.020E-11	3.020E-11	3.020E-11	3.020E-11
14	5.106E-01	3.020E-11	3.338E-11	3.020E-11	6.066E-11	3.020E-11	3.020E-11
15	2.254E-04	3.020E-11	1.613E-10	3.020E-11	3.020E-11	3.020E-11	3.020E-11
16	1.011E-08	3.020E-11	7.394E-01	3.020E-11	1.383E-02	3.020E-11	3.020E-11
17	1.493E-04	3.020E-11	1.529E-05	3.020E-11	3.368E-05	3.020E-11	3.020E-11
18	8.650E-01	3.020E-11	1.174E-09	3.020E-11	1.473E-07	3.690E-11	6.066E-11
19	2.062E-01	3.020E-11	4.118E-06	3.020E-11	3.020E-11	3.020E-11	3.020E-11
20	2.783E-07	3.020E-11	2.433E-05	3.020E-11	7.599E-07	3.020E-11	3.020E-11
21	3.020E-11	3.020E-11	1.537E-01	3.020E-11	7.221E-06	3.020E-11	3.020E-11
22	5.573E-10	3.690E-11	1.464E-10	3.020E-11	2.154E-10	4.504E-11	4.077E-11
23	4.077E-11	3.020E-11	8.771E-02	3.020E-11	3.183E-03	3.020E-11	3.020E-11
24	3.020E-11	3.020E-11	9.941E-01	3.020E-11	3.034E-03	3.020E-11	3.020E-11
25	7.697E-04	3.020E-11	2.226E-01	3.020E-11	6.204E-01	3.020E-11	3.020E-11
26	6.356E-05	3.020E-11	9.069E-03	3.020E-11	6.736E-06	3.020E-11	3.020E-11
27	5.494E-11	3.020E-11	5.828E-03	3.020E-11	5.555E-02	3.020E-11	3.020E-11
28	1.784E-04	3.020E-11	1.385E-06	3.020E-11	5.692E-01	3.020E-11	3.020E-11
29	1.329E-10	3.020E-11	1.019E-05	3.020E-11	1.174E-09	3.020E-11	3.020E-11
30	4.311E-08	3.020E-11	1.032E-02	3.020E-11	3.020E-11	3.020E-11	3.020E-11
-/=/+	0/3/26	0/0/29	1/7/21	0/0/29	2/5/22	0/0/29	0/0/29

Table 2 reports the p-values of SDHCPO against the seven comparison algorithms on each test function, together with the overall “−/=/+” counts. It can be seen that the vast majority of p-values are far below 0.05, indicating that SDHCPO exhibits statistically significant superiority over the competing algorithms on almost all functions. It is worth emphasizing that the Wilcoxon-based significance results are highly consistent with the aforementioned “order-of-magnitude gap” in the mean values: for functions where the mean objective value of SDHCPO is much lower than that of the competitors, the corresponding p-values typically shrink to the order of 10^{-8} or even 10^{-11}, implying that the performance gap cannot be attributed to random fluctuations. By contrast, the few cases with slightly larger p-values or an “=“ outcome mostly correspond to functions on which all algorithms have already approached the theoretical optimum. In such situations, the standard deviation and p-values are no longer the primary focus of comparison and do not alter the conclusion that SDHCPO holds a clear overall advantage.



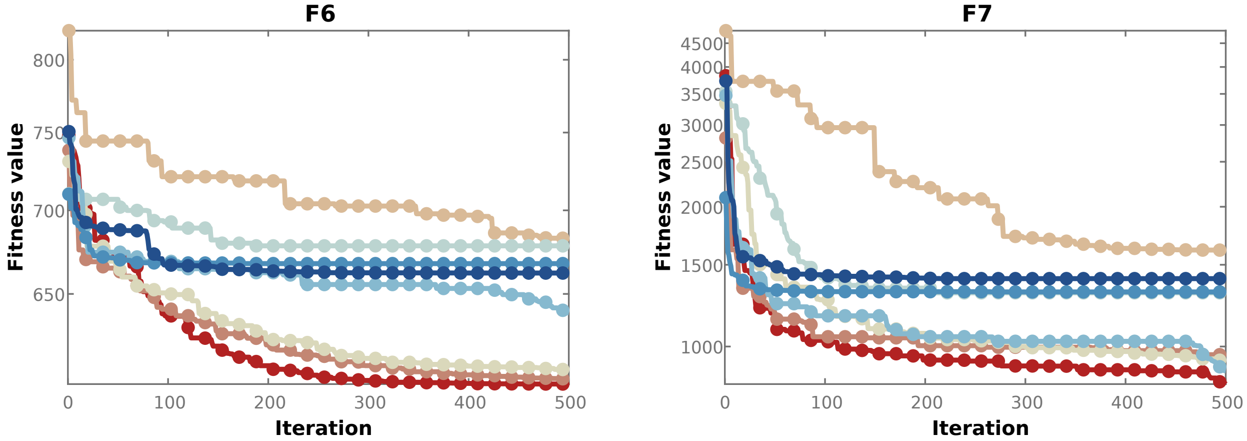
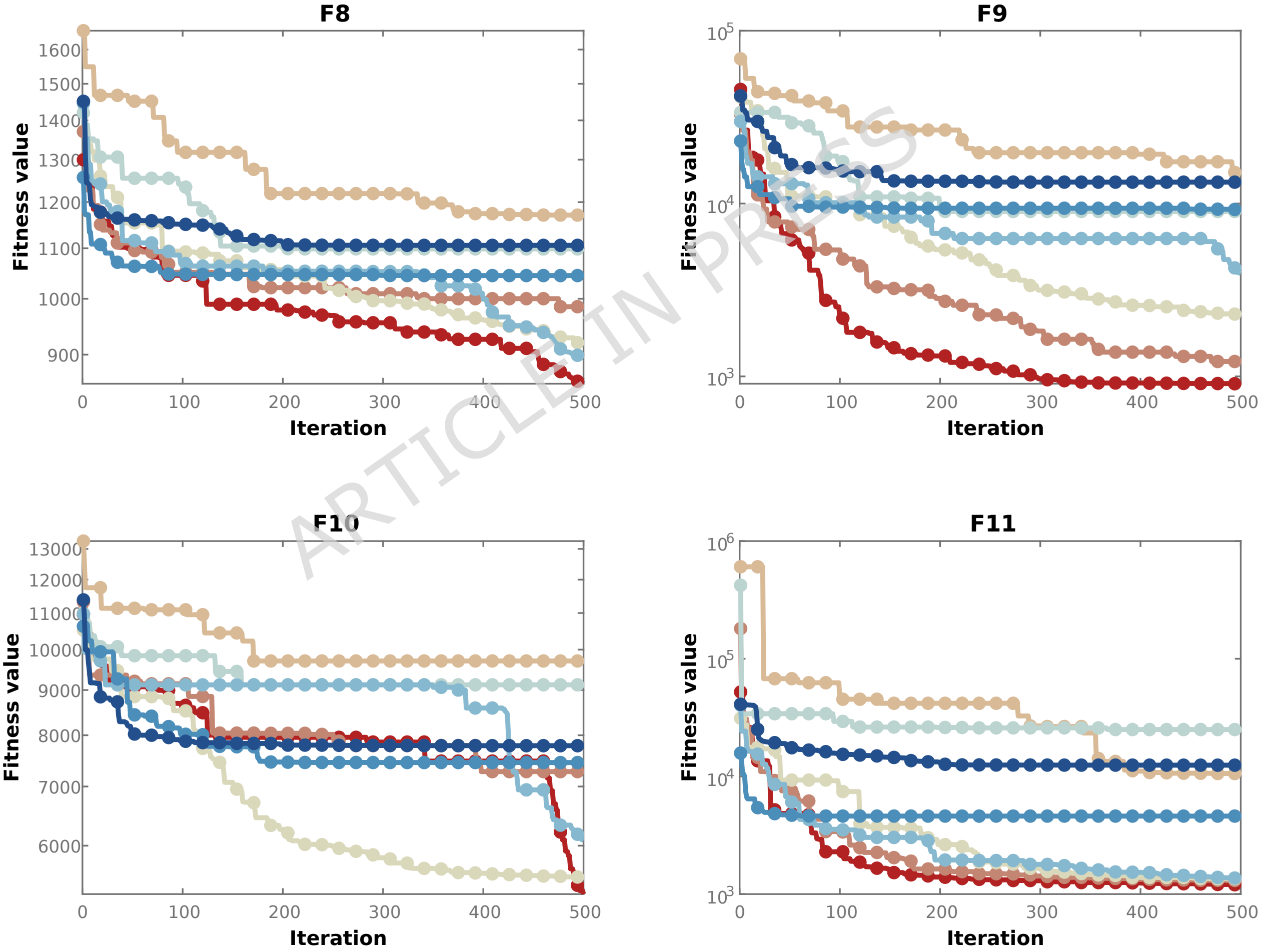


Fig. 5 SDHCPO vs. algorithms: CEC2017 convergence curves (F1, F3-F7)



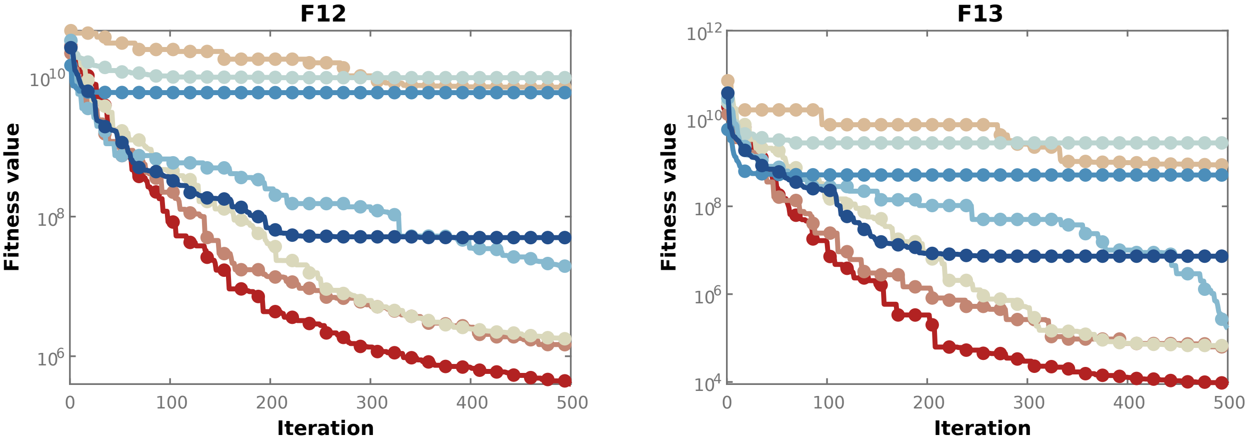
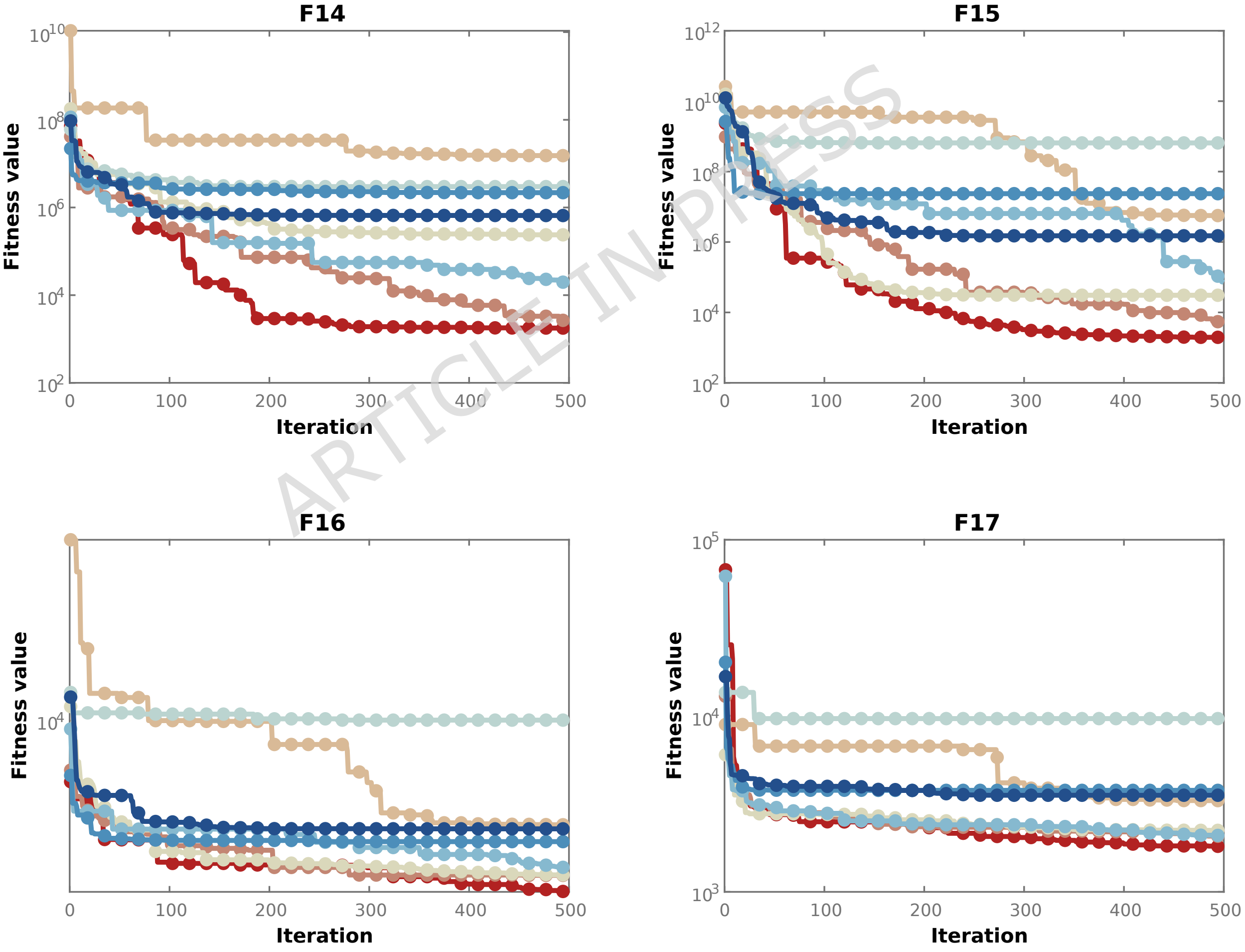


Fig. 6 SDHCPO vs. algorithms: CEC2017 convergence curves (F8-F13)



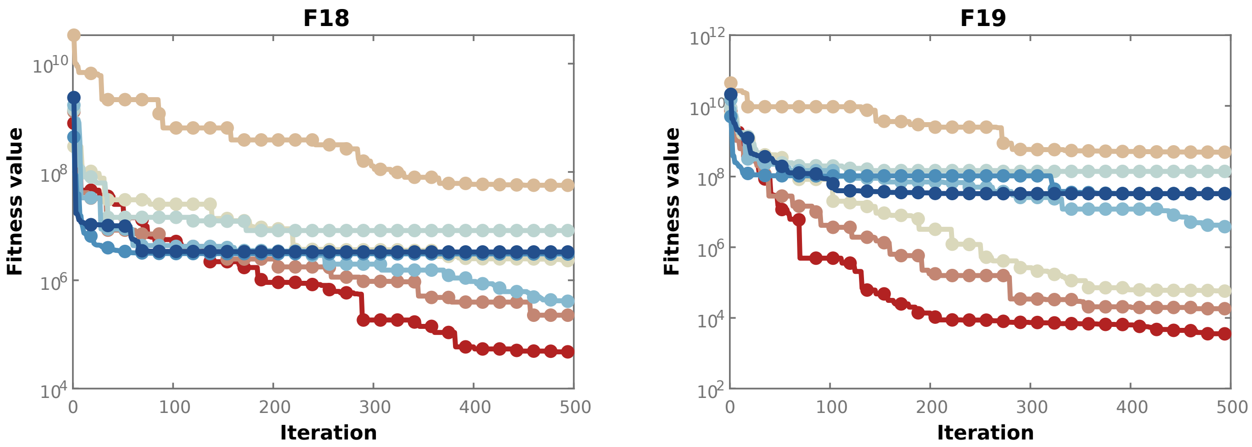
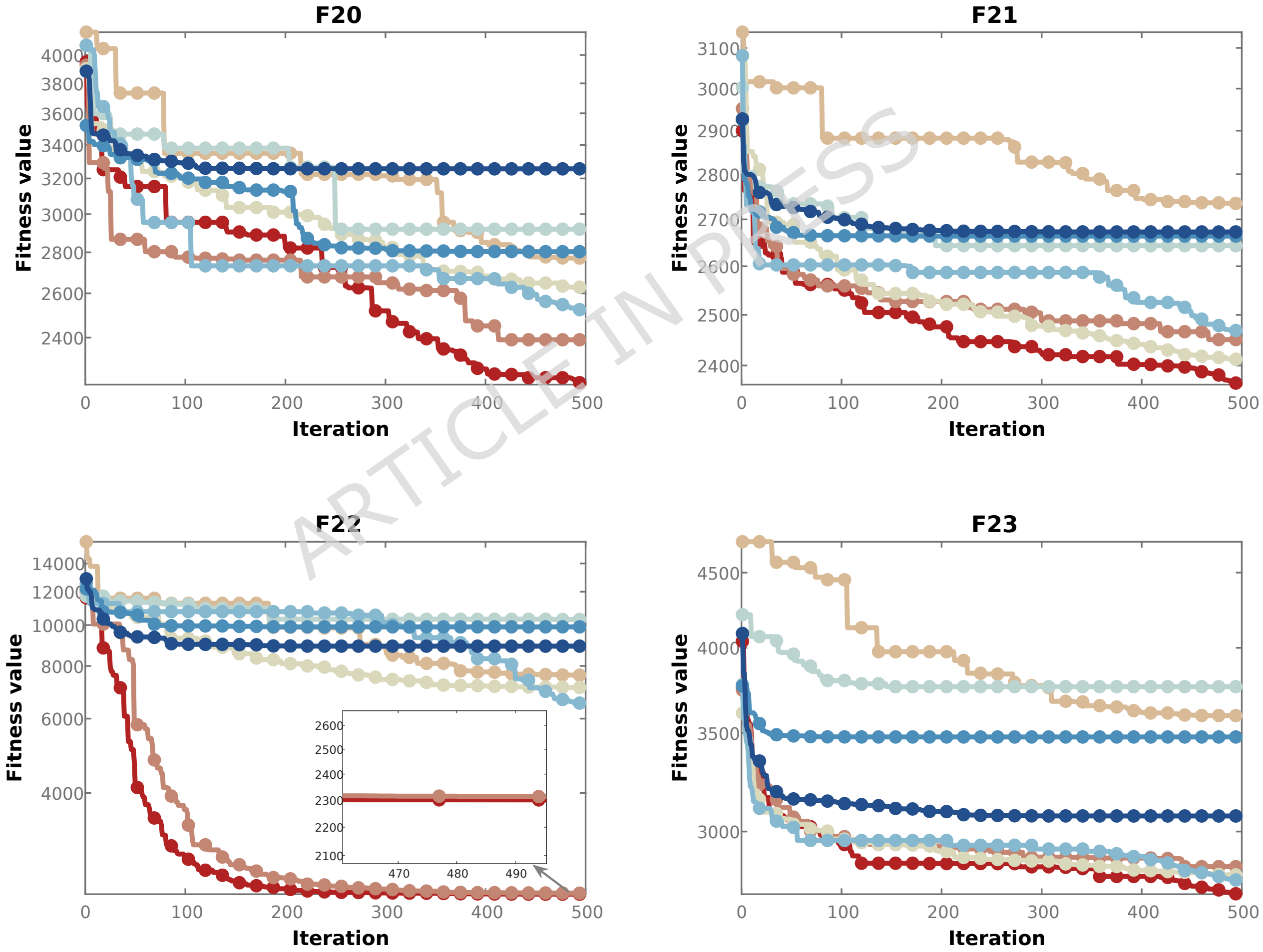


Fig. 7 SDHCPO vs. algorithms: CEC2017 convergence curves (F14-F19)



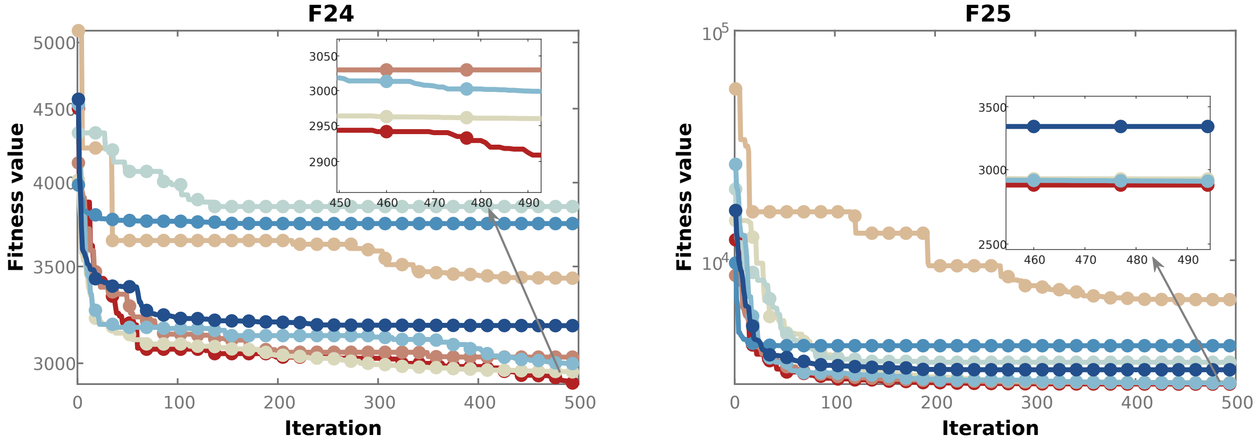
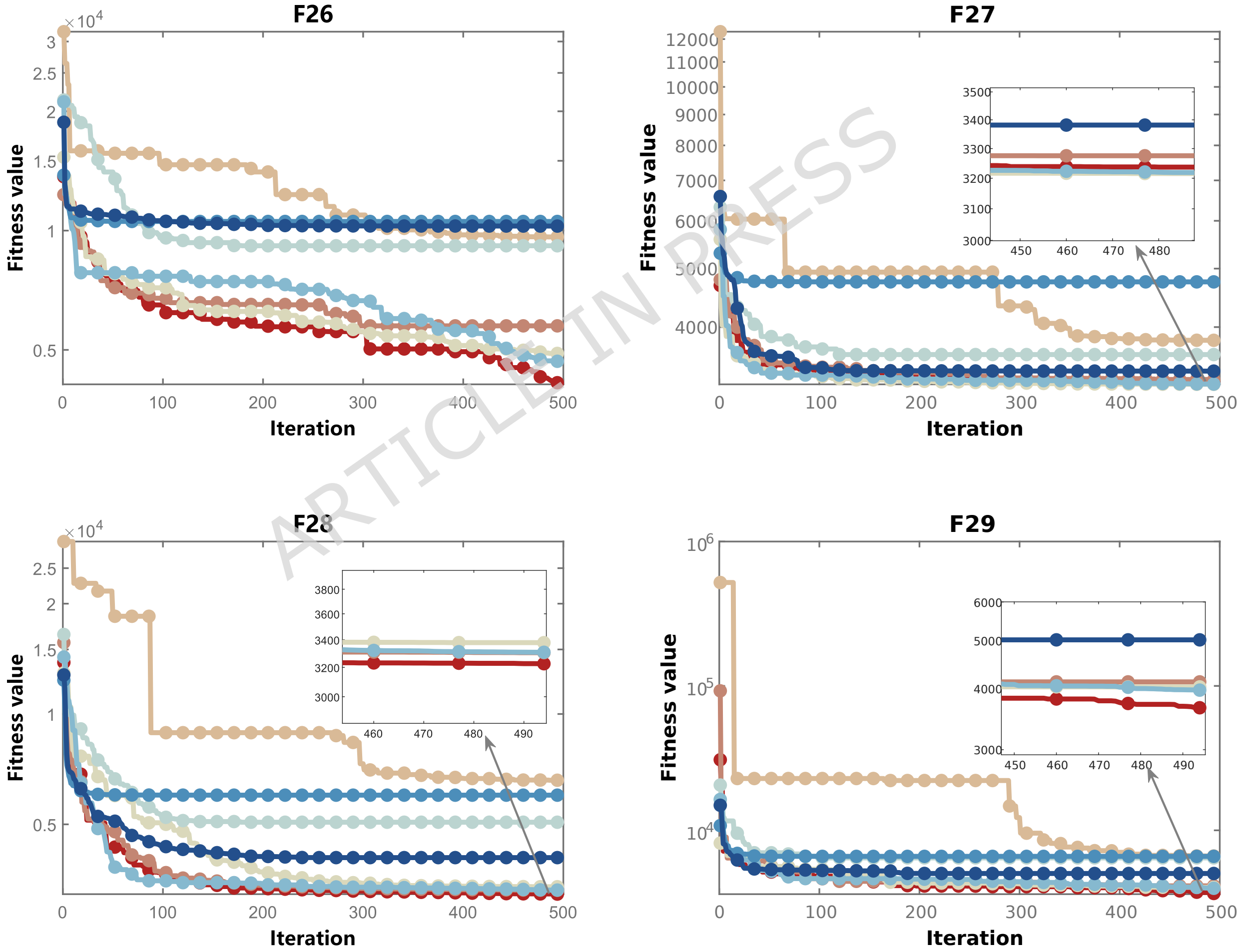


Fig. 8 SDHCPO vs. algorithms: CEC2017 convergence curves (F20-F25)



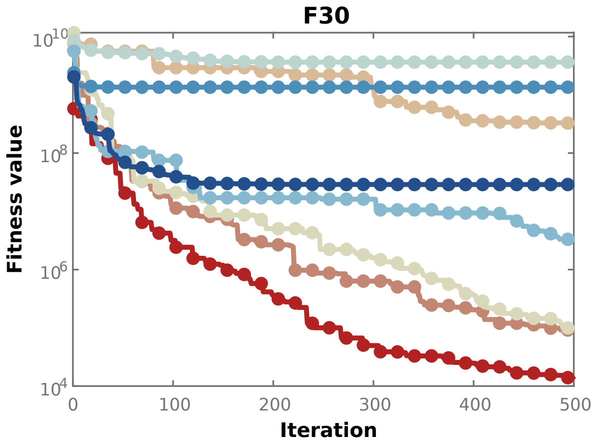


Fig. 9 SDHCPO vs. algorithms: CEC2017 convergence curves (F26-F30)

The convergence behavior on CEC2017 is illustrated in Figures 5-9. By analyzing the convergence curves for the F1-F30 test functions, SDHCPO exhibits a pronounced overall advantage in both convergence speed and convergence accuracy. In most cases, its convergence curves remain at the lowest level among all algorithms, and, in particular, the fitness values of SDHCPO show the steepest decline during the early iterations. This indicates that SDHCPO possesses highly efficient global search capability and can rapidly approach the vicinity of the optimal solution. As the iterations proceed, SDHCPO continues to improve the solutions and effectively avoids being trapped in local optima, ultimately converging to fitness values that are clearly superior to those of the competing algorithms, for example on functions F1, F9, F12, F18, and F30. These results provide strong evidence of its superiority in obtaining high-precision solutions and the stability of its search process.

4.3 Results on the CEC2022 Benchmark Suite

To further evaluate the generalization capability of SDHCPO across different function classes, this subsection compares SDHCPO with the same set of competing algorithms on the 20-dimensional CEC2022 test suite. Table 3 reports the mean and standard deviation of the best objective values obtained on the 12 test functions. It can be observed that SDHCPO achieves the smallest mean fitness on 10 out of the 12 functions and ranks second on the remaining two, while also attaining the best average Friedman rank. These results indicate that, on the new benchmark suite, SDHCPO maintains an overall performance advantage consistent with that observed on CEC2017.

Table 3. Performance Metrics of SDHCPO and Other Algorithms on CEC2022

Index	**SDHCPO**	**CPO**	**CFOA**	**PKO**	**CDO**	**MVO**	**HOA**	
Std	2.515E+03	3.810E+03	3.407E+04	1.250E+04	2.452E+03	**1.205E+01**	6.987E+03	9.0
Mean	7.059E+03	1.297E+04	8.376E+04	3.490E+04	2.971E+04	**3.180E+02**	3.175E+04	3.5
Std	1.110E+01	1.211E+01	4.769E+02	**1.109E+01**	4.478E+01	1.582E+01	3.153E+02	6.4

Mean	4.560E+02	4.620E+02	1.914E+03	4.561E+02	2.087E+03	**4.488E+02**	1.386E+03	6.1
Std	**6.167E-03**	2.149E-01	1.358E+01	7.955E-01	5.362E+00	1.349E+01	8.986E+00	1.3
Mean	**6.000E+02**	6.003E+02	6.734E+02	6.008E+02	6.663E+02	6.204E+02	6.539E+02	6.6
Std	1.022E+01	1.488E+01	2.027E+01	1.552E+01	**9.692E+00**	2.634E+01	1.673E+01	3.8
Mean	**8.335E+02**	9.032E+02	9.749E+02	8.400E+02	9.555E+02	8.753E+02	9.254E+02	9.4
Std	**9.416E-01**	2.371E+01	1.637E+03	3.821E+01	2.482E+02	1.490E+03	5.482E+02	1.7
Mean	**9.005E+02**	9.121E+02	4.260E+03	9.304E+02	3.502E+03	2.353E+03	2.688E+03	4.3
Std	**1.461E+03**	1.825E+04	5.336E+08	9.770E+04	9.034E+08	6.507E+03	5.120E+08	4.1
Mean	**3.655E+03**	2.208E+04	7.031E+08	6.943E+04	5.670E+09	1.150E+04	6.341E+08	4.1
Std	**7.363E+00**	1.024E+01	7.736E+01	2.218E+01	3.442E+01	8.206E+01	6.253E+01	7.6
Mean	**2.039E+03**	2.064E+03	2.231E+03	2.071E+03	2.353E+03	2.143E+03	2.203E+03	2.2
Std	**1.337E+00**	2.163E+00	2.928E+02	4.422E+00	1.166E+01	9.548E+01	9.250E+01	1.0
Mean	**2.227E+03**	2.232E+03	2.515E+03	2.231E+03	2.256E+03	2.316E+03	2.338E+03	2.3
Std	**1.315E-01**	3.537E-01	2.091E+02	3.671E-01	1.046E+02	4.737E-01	1.803E+02	5.2
Mean	**2.481E+03**	2.482E+03	2.932E+03	2.481E+03	3.469E+03	2.482E+03	3.021E+03	2.6
Std	**3.451E+01**	8.076E+01	1.747E+03	1.013E+03	9.556E+02	7.934E+02	1.384E+03	1.3
Mean	**2.510E+03**	2.540E+03	5.297E+03	3.759E+03	6.068E+03	3.880E+03	4.938E+03	4.7
Std	**1.018E-01**	3.727E+01	1.733E+04	1.483E+01	3.424E+01	1.177E+01	1.083E+03	3.7
Mean	**2.900E+03**	2.915E+03	4.567E+04	2.920E+03	8.544E+03	2.982E+03	7.203E+03	3.9
Std	**5.716E+00**	1.147E+01	1.259E+02	6.412E+00	5.178E+01	3.332E+01	1.984E+02	1.2
Mean	**2.944E+03**	2.987E+03	3.310E+03	2.949E+03	3.592E+03	2.974E+03	3.716E+03	3.1
ank	**1.333**	2.750	7.417	3.417	4.250	4.500	6.000	
rank	**1.167**	2.833	7.250	2.917	6.833	3.250	5.917	
rank	**1**	2	8	3	7	4	6	

At the level of individual functions, the improvements achieved by SDHCPO on multimodal, hybrid, and composition functions are particularly pronounced. For the multimodal functions F2-F5, SDHCPO ranks second on F2—slightly inferior to MVO yet still competitive—while achieving the smallest mean objective values among all algorithms on F3-F5. In contrast, the mean values of CFOA, CDO, HOA, and WOA are often several times, or even several orders of magnitude, higher. This indicates that SDHCPO possesses stronger global optimization capability and greater resistance to premature convergence when dealing with complex landscapes featuring multiple local minima. For the hybrid functions F6-F8, the performance gains are even more striking. In particular, on F6, SDHCPO reduces the mean objective value to the order of 10^3, whereas MVO remains above 10^4, and CPO together with the other algorithms perform several orders of magnitude worse. For the composition functions F9-F12, SDHCPO performs on par with PKO on F9, while on F10-F12 it is markedly superior to all competing algorithms, indicating that it can also effectively avoid entrapment in local optima in complex composite landscapes. SDHCPO likewise exhibits a clear advantage in convergence stability on the vast majority of functions. Except for F1 and F4, SDHCPO attains the smallest standard deviation among all algorithms on the remaining 10 functions, with particularly small fluctuation magnitudes on F3, F5, F6, and F9-F12. This demonstrates highly reproducible convergence behavior across repeated runs.

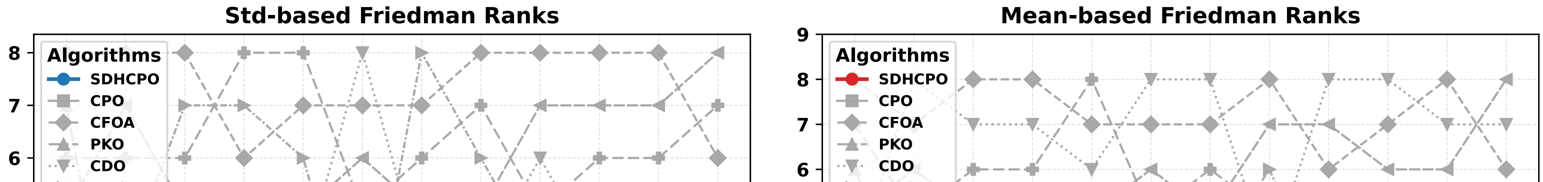


Fig. 10 Friedman Rank Evolution of SDHCPO vs. Competitors on CEC2022 Test Suite

Figure 10 presents the per-function ranking distribution of the eight algorithms on the 12 test functions of the 20-dimensional CEC2022 suite. It can be observed that SDHCPO ranks among the top methods on almost all functions, reflecting stable and consistently strong ranking performance across different function types. These results indicate that, on the CEC2022 test set, SDHCPO achieves a clearly superior overall solving capability and robustness.

Table 4. Wilcoxon Rank-Sum p-Values: SDHCPO vs. Algorithms on CEC2022

Fun	**CPO-p**	**CFOA-p**	**PKO-p**	**CDO-p**	**MVO-p**	**HOA-p**	**WOA-p**
1	2.195E-08	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11
2	1.058E-03	3.020E-11	3.478E-01	3.020E-11	9.031E-04	3.020E-11	3.020E-11
3	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11
4	3.690E-11	3.020E-11	1.120E-01	3.020E-11	6.518E-09	3.020E-11	3.020E-11
5	1.174E-09	3.020E-11	3.497E-09	3.020E-11	3.338E-11	3.020E-11	3.020E-11
6	6.722E-10	3.020E-11	8.891E-10	3.020E-11	1.174E-09	3.020E-11	3.020E-11
7	2.610E-10	3.020E-11	1.698E-08	3.020E-11	1.464E-10	3.020E-11	3.020E-11
8	5.072E-10	3.020E-11	3.592E-05	3.020E-11	1.070E-09	3.020E-11	3.020E-11
9	6.066E-11	3.020E-11	3.147E-02	3.020E-11	2.872E-10	3.020E-11	3.020E-11

Table 4.Cont.

Fun	**CPO-p**	**CFOA-p**	**PKO-p**	**CDO-p**	**MVO-p**	**HOA-p**	**WOA-p**
10	6.669E-03	7.389E-11	2.879E-06	3.020E-11	2.015E-08	6.696E-11	1.777E-10
11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11
12	1.094E-	3.020E-	5.264E-	3.020E-	4.459E-	3.020E-	3.020E-

	10	11	04	11	04	11	11
-/=/+	0/0/12	0/0/12	0/2/10	0/0/12	1/0/11	0/0/12	0/0/12

These findings are further corroborated by the Wilcoxon rank-sum test. As shown in Table 4, the p-values of SDHCPO against most competing algorithms are substantially below 0.05, and the corresponding "−/=/+" statistics are dominated by "+", indicating that, on the CEC2022 test set, SDHCPO is overall statistically superior to the vast majority of comparison methods. Overall, the CEC2022 experimental results are highly consistent with the conclusions drawn from CEC2017: SDHCPO maintains a leading Friedman ranking and statistically significant advantages across different dimensions and benchmark suites, with its strengths particularly pronounced on structurally complex multimodal, hybrid, and composition functions. This indicates that the four enhancement strategies proposed in this paper do not merely overfit a specific benchmark set, but can effectively improve the accuracy, stability, and cross-problem robustness of the algorithm over a broader family of problems, thereby providing further support for the subsequent ablation studies and high-dimensional extension analysis.

4.4 Melting Experiment Results

The comparisons with multiple metaheuristic algorithms show that SDHCPO already achieves substantially better overall performance than the original CPO and other representative methods on both benchmark suites. However, numerical results alone are insufficient to disentangle the individual contributions of the four enhancement strategies or to determine whether the observed performance gains primarily stem from a particular mechanism or from their synergistic interaction. To this end, two groups of ablation experiments are designed on the 30-dimensional CEC2017 test set, centered on the four key strategies. In the first group, the original CPO is used as the baseline, and only one strategy is activated at a time, yielding the variants SCPO, DCPO, HCPO, and CCPO, which are used to characterize the independent effect of each individual strategy. In the second group, the fully integrated SDHCPO serves as the baseline, and one strategy is deactivated at a time to construct the variants w/o C, w/o H, w/o D, and w/o S, thereby examining the necessity of each strategy from the opposite perspective. The configurations of all variants are encoded using a binary switch scheme, where 1 indicates that the corresponding strategy is enabled and 0 indicates that it is disabled, providing an intuitive representation of the differences among variants along the four strategy dimensions, as summarized in Table 5.

Table 5. Variants of CPO constructed from four strategies.

Algorithm	**S: Sobol-OBL Initialization**	**D: Differential Evolution**	**H: Horizontal-Vertical Crossover**	**C: Cosine Annealing Dynamic Adjustment**

CPO	0	0	0	0
SCPO	1	0	0	0
DCPO	0	1	0	0
HCPO	0	0	1	0
CCPO	0	0	0	1
SDHCPO(all strategies enabled)	1	1	1	1

The ablation results for individual strategies are reported in Table 6. Overall, SCPO, DCPO, HCPO, and CCPO achieve varying degrees of improvement over CPO in terms of mean and/or standard deviation on most test functions, indicating that each of the four strategies can yield substantial benefits when applied in isolation. On unimodal functions, the effect of the differential evolution mutation is particularly pronounced, significantly enhancing the search efficiency and solution stability for single-peak problems. For functions such as F4 and F6, DCPO achieves lower mean and standard deviation than CPO, indicating that DE mutation can effectively strengthen global search and suppress premature convergence. HCPO substantially reduces the final mean on multimodal or complex functions such as F5, F8, F10, and F16, yielding higher-precision solutions than CPO. CCPO exhibits more stable performance on medium- and high-index functions and provides the smallest or near-smallest mean among the single-strategy variants on many hybrid and composition functions, with clear improvements over CPO on F21-F29.

Table 6. Performance Metrics of CPO and Single-Strategy Variants on CEC2017

Fun	Index	CPO	SCPO	DCPO	HCPO	CCPO	SDHCPO
F1	Std	5.932E+05	4.280E+05	1.108E+04	4.840E+06	4.536E+05	**8.826E+03**
	Mean	7.377E+05	6.020E+05	**8.484E+03**	5.829E+06	6.930E+05	1.202E+04
F3	Std	1.390E+04	1.304E+04	1.004E+04	1.612E+04	**7.969E+03**	9.399E+03
	Mean	6.263E+04	6.191E+04	6.388E+04	5.062E+04	5.862E+04	**4.103E+04**
F4	Std	2.335E+01	2.150E+01	1.883E+01	1.653E+01	1.693E+01	**1.489E+01**
	Mean	5.186E+02	5.158E+02	5.064E+02	5.297E+02	5.246E+02	**5.058E+02**
F5	Std	1.598E+01	1.529E+01	**1.390E+01**	2.900E+01	2.512E+01	2.196E+01
	Mean	6.944E+02	6.672E+02	6.841E+02	6.286E+02	6.454E+02	**5.922E+02**
F6	Std	7.423E-01	6.706E-01	6.953E-02	2.251E+00	5.900E-01	**4.659E-02**
	Mean	6.019E+02	6.018E+02	**6.001E+02**	6.048E+02	6.018E+02	**6.001E+02**
F7	Std	1.674E+01	1.663E+01	**1.507E+01**	4.640E+01	2.685E+01	2.053E+01
	Mean	9.398E+0	9.234E+0	9.297E+0	9.249E+0	8.902E+0	**8.585E+**

		2	2	2	2	2	**02**
F8	Std	**1.412E+01**	1.478E+01	1.739E+01	2.950E+01	2.060E+01	2.003E+01
	Mean	9.837E+02	9.821E+02	9.770E+02	9.187E+02	9.231E+02	**8.883E+02**
F9	Std	4.924E+02	2.308E+02	7.038E+01	7.523E+02	1.335E+02	**4.930E+01**
	Mean	1.315E+03	1.197E+03	9.431E+02	2.189E+03	1.099E+03	**9.336E+02**
F10	Std	4.159E+02	3.552E+02	**3.409E+02**	4.057E+02	5.891E+02	7.132E+02
	Mean	7.539E+03	7.462E+03	7.789E+03	6.406E+03	6.040E+03	**5.648E+03**
F11	Std	2.969E+01	2.729E+01	**1.953E+01**	6.698E+01	2.678E+01	2.361E+01
	Mean	1.274E+03	1.270E+03	1.256E+03	1.261E+03	1.263E+03	**1.226E+03**
F12	Std	9.060E+05	7.954E+05	7.489E+05	1.391E+06	6.439E+05	**3.686E+05**
	Mean	1.306E+06	9.918E+05	8.830E+05	2.136E+06	1.262E+06	**7.126E+05**
F13	Std	1.024E+04	8.785E+03	8.673E+03	7.815E+03	8.433E+03	**7.481E+03**
	Mean	2.466E+04	1.781E+04	1.487E+04	**8.952E+03**	1.930E+04	1.394E+04
F14	Std	**3.212E+02**	2.924E+03	1.497E+03	1.283E+04	1.659E+03	2.432E+03
	Mean	**1.944E+03**	2.685E+03	2.730E+03	8.658E+03	2.575E+03	3.180E+03
F15	Std	2.950E+03	2.723E+03	2.040E+03	5.550E+03	**1.705E+03**	2.725E+03
	Mean	5.125E+03	4.924E+03	4.387E+03	5.309E+03	4.411E+03	**4.219E+03**
F16	Std	2.446E+02	**1.741E+02**	1.759E+02	2.495E+02	2.528E+02	2.527E+02
	Mean	3.079E+03	3.020E+03	3.090E+03	2.643E+03	2.642E+03	**2.510E+03**
F17	Std	**1.047E+02**	1.466E+02	1.294E+02	1.366E+02	1.229E+02	1.279E+02
	Mean	2.050E+03	2.042E+03	2.044E+03	1.952E+03	1.966E+03	**1.879E+03**
F18	Std	**6.307E+04**	1.164E+05	2.453E+05	2.956E+05	6.733E+04	8.039E+04
	Mean	1.167E+05	1.057E+05	1.316E+05	2.458E+05	1.110E+05	**9.408E+04**

Table 6. Cont.

Fun	Index	CPO	SCPO	DCPO	HCPO	CCPO	SDHCPO
F19	Std	4.334E+03	**3.619E+03**	6.374E+03	4.587E+03	4.259E+03	5.836E+03
	Mean	6.267E+03	5.969E+03	7.292E+03	6.730E+03	6.679E+03	**5.566E+03**
F20	Std	1.616E+02	**9.667E+01**	1.462E+02	1.307E+02	1.265E+02	1.255E+02

	Mean	2.441E+03	2.415E+03	2.475E+03	2.312E+03	2.320E+03	**2.246E+03**
F21	Std	**1.421E+01**	1.770E+01	1.495E+01	2.751E+01	2.029E+01	2.093E+01
	Mean	2.481E+03	2.441E+03	2.471E+03	2.427E+03	2.424E+03	**2.390E+03**
F22	Std	3.286E+00	3.001E+00	**2.155E+00**	9.620E+02	4.703E+00	3.541E+00
	Mean	2.309E+03	2.305E+03	2.303E+03	2.494E+03	2.303E+03	**2.302E+03**
F23	Std	**1.540E+01**	1.834E+01	2.310E+01	3.812E+01	2.003E+01	2.201E+01
	Mean	2.849E+03	2.845E+03	2.836E+03	2.782E+03	2.809E+03	**2.742E+03**
F24	Std	2.294E+01	1.893E+01	**1.487E+01**	3.081E+01	2.609E+01	2.625E+01
	Mean	3.020E+03	2.992E+03	3.009E+03	2.935E+03	2.965E+03	**2.898E+03**
F25	Std	1.827E+01	1.681E+01	1.370E+01	2.518E+01	1.294E+01	**1.050E+01**
	Mean	2.910E+03	2.915E+03	2.901E+03	2.928E+03	2.907E+03	**2.895E+03**
F26	Std	8.708E+02	8.430E+02	8.268E+02	1.117E+03	8.183E+02	**2.912E+02**
	Mean	5.364E+03	5.196E+03	5.256E+03	**4.407E+03**	4.807E+03	4.721E+03
F27	Std	1.347E+01	1.275E+01	1.235E+01	1.462E+01	1.291E+01	**8.985E+00**
	Mean	3.277E+03	3.261E+03	3.244E+03	3.255E+03	3.265E+03	**3.227E+03**
F28	Std	2.221E+01	2.191E+01	2.305E+01	3.179E+01	2.256E+01	**2.115E+01**
	Mean	3.285E+03	3.262E+03	3.256E+03	3.309E+03	3.283E+03	**3.253E+03**
F29	Std	1.578E+02	1.462E+02	1.314E+02	1.873E+02	1.718E+02	**1.192E+02**
	Mean	3.997E+03	3.991E+03	3.987E+03	3.776E+03	3.801E+03	**3.662E+03**
F30	Std	7.121E+04	6.867E+04	5.487E+04	6.338E+04	7.631E+04	**1.369E+04**
	Mean	1.160E+05	1.001E+05	6.268E+04	6.711E+04	1.289E+05	**2.454E+04**

The Friedman statistics in Table 7 show that, in terms of the mean objective value, SDHCPO attains an average rank of 1.241, which is substantially lower than that of all single-strategy variants, followed in order by CCPO, DCPO, HCPO, and SCPO. For the standard deviation, SDHCPO likewise achieves the lowest average rank, only slightly higher than a few single-strategy configurations that behave more conservatively on individual functions. These results indicate that the four strategies do not simply stack within a single framework, but rather exhibit complementary effects at different stages and on different function types, enabling SDHCPO to further reduce the objective values while maintaining stability. The Wilcoxon test results against SDHCPO further show that, although some single-strategy variants can approach SDHCPO on a small number of functions, SDHCPO still significantly outperforms them on roughly twenty functions, indicating that no single strategy can reproduce the overall advantage of the fully integrated framework.

Table 7. Friedman/Wilcoxon statistics of CPO and single-strategy variants on CEC2017

Algorithm	Avg Rank (Std)	Avg Rank (Mean)	Overall Rank	-/=/+ (Mean)
CPO	3.724	5.103	5	0/3/26
SCPO	3.138	3.793	4	0/4/25
DCPO	2.724	3.655	2	1/8/20
HCPO	5.276	3.828	6	2/4/23
CCPO	3.448	3.276	3	0/4/25
SDHCPO	**2.690**	**1.241**	**1**	**-/-/-**

After analyzing the configurations with single strategies enabled, a complementary set of experiments is designed to further assess the necessity of each enhancement module from the opposite perspective. Using SDHCPO as the baseline, four variants are constructed by selectively disabling one strategy at a time. Specifically, by turning off the Sobol-based initialization, Differential Evolution mutation, Horizontal-Vertical Crossover, and cosine-annealing-based dynamic adjustment, the variants w/o S, w/o D, w/o H, and w/o C are obtained, respectively. The corresponding 0-1 configurations are summarized in Table 8.

Table 8. Strategy configurations of SDHCPO and disabled-strategy variants.

Algorithm	S: Sobol-OBL Initialization	D: Differential Evolution	H: Horizontal-Vertical Crossover	C: Cosine Annealing Dynamic Adjustment
SDHCPO(all strategies enabled)	1	1	1	1
w/o C	1	1	1	0
w/o H	1	1	0	1
w/o D	1	0	1	1
w/o S	0	1	1	1
CPO	0	0	0	0

The ablation results for the configurations with individual strategies disabled are reported in Table 9. Overall, removing any single strategy from SDHCPO leads to varying degrees of performance degradation on a considerable number of benchmark functions, with the effect being particularly pronounced on high-dimensional multimodal, hybrid, and composition functions. Compared with the fully integrated SDHCPO, all four variants exhibit generally higher mean fitness values and slightly larger standard deviations, indicating that none of the modules is a redundant add-on; instead, they jointly underpin the algorithm’s global search capability and convergence stability in complex search spaces.

Table 9. Performance Metrics of SDHCPO and Single-Strategy-Disabled Variants on CEC2017

Fun	Index	SDHCPO	w/o C	w/o H	w/o D	w/o S	CPO
F1	Std	8.826E+03	6.443E+04	**6.808E+03**	3.424E+06	1.081E+04	5.932E+05
	Mean	1.202E+04	7.225E+04	**7.593E+03**	5.360E+06	1.287E+04	7.377E+05
F3	Std	**9.399E+03**	1.075E+04	1.124E+04	1.057E+04	1.021E+04	1.390E+04
	Mean	**4.103E+04**	4.643E+04	5.938E+04	4.554E+04	4.177E+04	6.263E+04
F4	Std	1.489E+01	**1.310E+01**	1.695E+01	1.942E+01	2.015E+01	2.335E+01
	Mean	**5.058E+02**	5.129E+02	5.078E+02	5.331E+02	5.113E+02	5.186E+02
F5	Std	2.196E+01	2.757E+01	2.637E+01	2.152E+01	2.336E+01	**1.598E+01**
	Mean	**5.922E+02**	6.205E+02	6.114E+02	6.097E+02	5.995E+02	6.944E+02
F6	Std	4.659E-02	7.320E-02	**4.175E-02**	1.205E+00	5.381E-02	7.423E-01
	Mean	**6.001E+02**	6.002E+02	6.002E+02	6.034E+02	6.004E+02	6.019E+02
F7	Std	2.053E+01	3.255E+01	2.319E+01	3.213E+01	2.306E+01	**1.674E+01**
	Mean	**8.585E+02**	8.927E+02	8.719E+02	8.859E+02	8.604E+02	9.398E+02
F8	Std	2.003E+01	2.999E+01	1.808E+01	1.858E+01	2.062E+01	**1.412E+01**
	Mean	**8.883E+02**	9.043E+02	9.061E+02	8.992E+02	8.908E+02	9.837E+02
F9	Std	4.930E+01	3.101E+02	**4.734E+01**	4.401E+02	5.753E+01	4.924E+02
	Mean	**9.336E+02**	1.170E+03	9.360E+02	1.538E+03	9.379E+02	1.315E+03
F10	Std	7.132E+02	7.440E+02	7.691E+02	4.974E+02	6.607E+02	**4.159E+02**
	Mean	**5.648E+03**	6.174E+03	5.956E+03	5.797E+03	5.688E+03	7.539E+03
F11	Std	**2.361E+01**	2.448E+01	2.617E+01	6.054E+01	3.058E+01	2.969E+01
	Mean	1.226E+03	**1.211E+03**	1.232E+03	1.259E+03	1.231E+03	1.274E+03
F12	Std	**3.686E+05**	8.891E+05	4.309E+05	9.400E+05	4.373E+05	9.060E+05
	Mean	7.126E+05	1.046E+06	**6.764E+05**	1.899E+06	7.501E+05	1.306E+06

Table 9. Cont.

Fun	Index	SDHCPO	w/o C	w/o H	w/o D	w/o S	CPO
F13	Std	7.481E+03	1.002E+04	1.264E+04	**7.075E+03**	8.215E+03	1.024E+04
	Mean	1.394E+04	**1.010E+04**	1.761E+04	1.044E+04	1.891E+04	2.466E+04

F14	Std	2.432E+03	1.319E+04	9.620E+02	1.941E+04	2.714E+03	**3.212E+02**
	Mean	3.180E+03	8.973E+03	2.132E+03	1.084E+04	2.557E+03	**1.944E+03**
F15	Std	2.725E+03	2.827E+03	2.942E+03	2.542E+03	**2.300E+03**	2.950E+03
	Mean	**4.219E+03**	6.106E+03	4.868E+03	4.290E+03	4.499E+03	5.125E+03
F16	Std	2.527E+02	3.367E+02	2.655E+02	**2.247E+02**	2.508E+02	2.446E+02
	Mean	**2.510E+03**	2.628E+03	2.735E+03	2.518E+03	2.515E+03	3.079E+03
F17	Std	1.279E+02	1.314E+02	1.116E+02	**9.676E+01**	1.155E+02	1.047E+02
	Mean	**1.879E+03**	1.965E+03	1.930E+03	1.933E+03	1.907E+03	2.050E+03
F18	Std	8.039E+04	1.311E+05	1.571E+05	1.848E+05	9.263E+04	**6.307E+04**
	Mean	**9.408E+04**	1.853E+05	1.873E+05	1.679E+05	1.293E+05	1.167E+05
F19	Std	5.836E+03	7.452E+03	**3.149E+03**	6.898E+03	4.003E+03	4.334E+03
	Mean	**5.566E+03**	7.958E+03	5.617E+03	8.341E+03	5.572E+03	6.267E+03
F20	Std	1.255E+02	1.230E+02	1.228E+02	**9.908E+01**	1.128E+02	1.616E+02
	Mean	**2.246E+03**	2.274E+03	2.304E+03	2.285E+03	2.284E+03	2.441E+03
F21	Std	2.093E+01	3.754E+01	2.166E+01	2.298E+01	2.278E+01	**1.421E+01**
	Mean	**2.390E+03**	2.393E+03	2.418E+03	2.404E+03	2.391E+03	2.481E+03
F22	Std	3.541E+00	1.423E+03	7.698E+02	4.716E+00	8.314E+02	**3.286E+00**
	Mean	**2.302E+03**	2.676E+03	2.442E+03	2.321E+03	2.453E+03	2.309E+03
F23	Std	2.201E+01	2.790E+01	2.272E+01	2.708E+01	2.525E+01	**1.540E+01**
	Mean	2.742E+03	**2.736E+03**	2.779E+03	2.764E+03	2.760E+03	2.849E+03
F24	Std	2.625E+01	2.796E+01	2.969E+01	**2.191E+01**	2.836E+01	2.294E+01
	Mean	**2.898E+03**	2.908E+03	2.944E+03	2.933E+03	2.910E+03	3.020E+03
F25	Std	**1.050E+01**	1.769E+01	1.153E+01	2.124E+01	1.197E+01	1.827E+01
	Mean	**2.895E+03**	2.913E+03	2.897E+03	2.925E+03	2.897E+03	2.910E+03
F26	Std	2.912E+02	7.101E+02	**2.366E+02**	7.406E+02	5.233E+02	8.708E+02
	Mean	4.721E+03	**4.021E+03**	4.811E+03	4.510E+03	4.454E+03	5.364E+03
F27	Std	8.985E+00	1.167E+01	8.361E+00	1.717E+01	**6.648E+00**	1.347E+01
	Mean	**3.227E+03**	3.229E+03	3.231E+03	3.255E+03	3.227E+03	3.277E+03

F28	Std	2.115E+01	2.213E+01	**2.081E+01**	2.228E+01	2.180E+01	2.221E+01
	Mean	**3.253E+03**	3.264E+03	3.258E+03	3.295E+03	3.255E+03	3.285E+03
F29	Std	**1.192E+02**	1.428E+02	1.428E+02	1.643E+02	1.476E+02	1.578E+02
	Mean	**3.662E+03**	3.679E+03	3.773E+03	3.693E+03	3.669E+03	3.997E+03
F30	Std	**1.369E+04**	1.461E+04	4.242E+04	3.690E+04	1.844E+04	7.121E+04
	Mean	2.454E+04	**2.266E+04**	7.642E+04	4.690E+04	2.825E+04	1.160E+05

These conclusions are systematically corroborated by the Friedman rankings and the Wilcoxon “−/=/+” statistics. As shown in Table 10, SDHCPO attains by far the lowest average Friedman rank in terms of the mean objective value, followed in order by w/o S, w/o C, w/o H, w/o D, and CPO, indicating that disabling any single strategy leads to a certain degree of overall performance degradation, with the deterioration caused by removing Differential Evolution or Horizontal-Vertical Crossover being particularly pronounced;For the standard deviation, SDHCPO likewise achieves the lowest average rank, whereas all other variants and the original CPO exhibit substantially higher ranks, indicating that the complete SDHCPO configuration offers superior convergence accuracy and stability. The Wilcoxon tests with SDHCPO as the baseline lead to the same conclusion: relative to w/o C, w/o H, w/o D, and w/o S, SDHCPO is marked with “+” on most functions, while “−” outcomes are very rare and mainly concentrated on relatively simple functions. From a statistical perspective, this further confirms that removing any single module degrades the overall performance of SDHCPO.

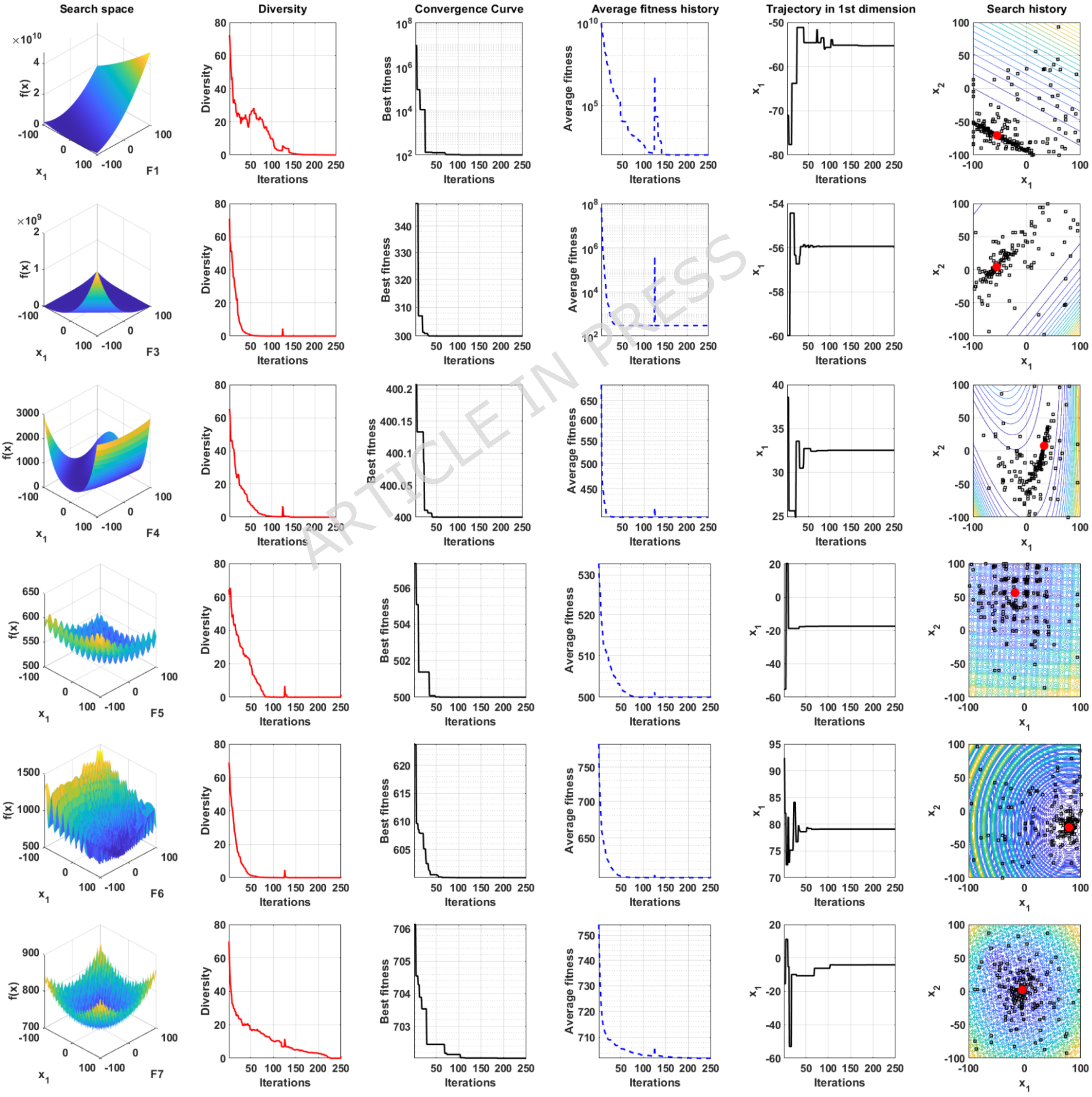
Table 10. Friedman/Wilcoxon statistics of SDHCPO and disabled-strategy variants on CEC2017

Algorithm	**Avg Rank (Std)**	**Avg Rank (Mean)**	**Overall Rank**	**-/=/+ (Mean)**
SDHCPO	**2.448**	**1.448**	**1**	**-/-/-**
w/o C	4.379	3.586	4	3/8/18
w/o H	3.172	3.724	3	1/12/16
w/o D	4.069	4.310	5	0/10/19
w/o S	3.310	2.655	2	0/17/12
CPO	3.586	5.172	6	0/3/24

Taken together with the single-strategy ablation results, these findings indicate that all four strategies are individually effective while also exhibiting strong synergy within the fully integrated framework. There is neither a single strategy that alone accounts for all performance gains nor any redundant module that fails to contribute to the overall behavior. Instead, both structurally and statistically, the four components jointly underpin the superior performance of SDHCPO.

4.5 Qualitative Analysis of Search Dynamics

To further elucidate the internal mechanisms of SDHCPO, this section selects representative unimodal and multimodal functions from CEC2017 and conducts a series of visualization analyses of its iterative behavior, including population diversity evolution, convergence curves, mean best fitness, 1-Dimensional Search Path, and search-history distributions, in order to investigate the exploration-exploitation balance achieved through the synergy of the four integrated strategies [25]. The resulting search dynamics visualizations are provided in Figure 11.



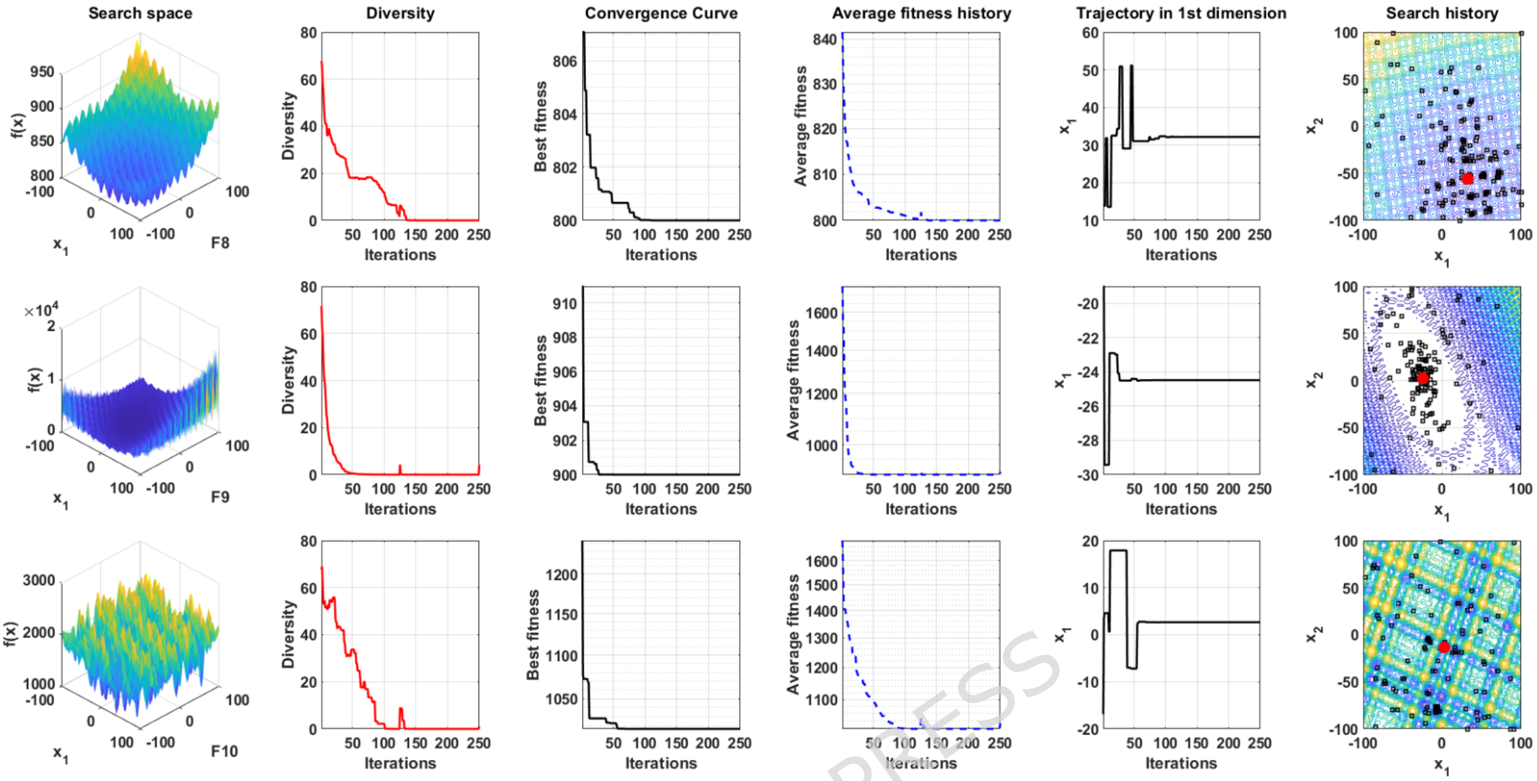


Fig. 11 Search Dynamics of SDHCPO

First, considering the population diversity curves in the second column, it can be observed that, for unimodal functions, diversity remains at a relatively high level in the initial stage and then decreases steadily with the number of iterations, eventually stabilizing. This indicates that the algorithm performs sufficiently extensive global exploration at the outset and can rapidly achieve a transition from broad exploration to fine-grained exploitation. For multimodal functions, by contrast, the decline in diversity is noticeably slower, and the curves maintain a relatively wide fluctuation range in the early and middle stages, suggesting that the algorithm conducts thorough global exploration and effectively avoids premature convergence to local optima.

The column of convergence curves provides a more direct illustration of the dynamic switching mechanism between exploration and exploitation in SDHCPO. For unimodal functions, the curves generally exhibit a fast-then-slow descending pattern: SDHCPO rapidly locks onto a region near the global optimum, after which the curves descend more gradually with almost no large oscillations, indicating stable and fine-grained convergence. For multimodal functions, the convergence curves display a more typical piecewise monotonic behavior: over several iteration intervals, the best fitness remains nearly unchanged, followed by a pronounced drop and the formation of a new plateau. This pattern clearly demonstrates the ability of SDHCPO to escape local optima.

The mean fitness history curves further show that SDHCPO exhibits small jumps in the middle iterations. This indicates that the algorithm does not simply pursue monotonically accelerated convergence; instead, once local exploitation within a region has progressed to a certain extent, SDHCPO deliberately expands the search radius and re-enhances population diversity. In doing so, it temporarily sacrifices mean fitness to gain more thorough global exploration capability, thereby achieving a more effective overall balance between exploration and exploitation.

The 1-Dimensional Search Path in the fifth column show that, for unimodal functions, the trajectories undergo large positional changes over a wide domain in the early iterations, after which the search interval gradually shrinks and eventually stabilizes in the vicinity of the global optimum. For multimodal functions, the trajectories maintain large-range transitions over a considerable number of iterations, frequently crossing the boundaries of different attraction basins, which reflects a pronounced cross-basin global exploration capability. As the algorithm enters the late iterations, the activity range of the trajectories contracts markedly and becomes essentially confined to the neighborhood of a single attraction basin, indicating that the search focus has gradually shifted from exploration to exploitation in this stage.

The search-history scatter plots in the sixth column further illustrate the behavior of SDHCPO from a spatial distribution perspective. For unimodal functions, the individual trajectories are widely and diffusely distributed within the objective region in the early iterations, covering most of the feasible domain. As the iterations progress, the scatter points gradually contract toward the region containing the global optimum and form a high-density cluster in its neighborhood, thereby achieving a smooth transition from global exploration to local exploitation. For multimodal functions, the scatter points are widely distributed across multiple competing regions in the early iterations, indicating that the population is concurrently evaluating different peak-valley structures. As the algorithm enters the middle and late stages, the scatter cloud gradually recedes from suboptimal regions and concentrates in a few dominant areas, eventually forming a more compact cluster near the global optimum. This process provides a direct visualization of SDHCPO’s ability to coordinate global exploration and local exploitation in complex energy landscapes.

4.6 High-Dimensional Testing Results

Building on the preceding 30-dimensional experiments, this section increases the problem dimensionality to 50 and performs extended tests on the same set of comparison algorithms. Overall, as the dimensionality increases, the optimization difficulty rises for all methods; however, the advantage of SDHCPO is not only preserved but becomes even more pronounced in high-dimensional settings. The performance metrics of all algorithms on CEC2017 for the 50-dimensional case are summarized in Table 11.

Table 11. Performance Metrics of SDHCPO and Other Algorithms on CEC2017 (d=50)

Index	SDHCPO	CPO	CFOA	PKO	CDO	MVO	HOA	
Std	8.423E+06	9.516E+07	1.210E+10	1.626E+08	1.123E+09	**4.466E+06**	8.301E+09	4.
Mean	**1.452E+07**	1.883E+08	1.029E+11	2.095E+08	8.056E+10	1.786E+07	8.790E+10	2.
Std	2.381E+04	2.172E+04	1.146E+05	1.019E+05	1.647E+04	3.369E+04	**1.034E+04**	1.
Mean	**1.379E+05**	1.801E+05	4.532E+05	4.136E+05	1.967E+05	1.477E+05	1.597E+05	3.
Std	4.842E+01	4.893E+01	5.989E+03	7.442E+01	3.629E+02	**3.788E+01**	4.847E+03	1.
Mean	6.489E+02	7.148E+02	3.093E+04	6.980E+02	2.278E+04	**6.006E+02**	2.341E+04	5.
Std	2.694E+01	**2.491E+01**	4.186E+01	5.231E+01	2.694E+01	7.471E+01	3.898E+01	6.
Mean	**7.470E+02**	9.260E+02	1.218E+03	7.772E+02	1.117E+03	7.959E+02	1.074E+03	1.
Std	**6.105E-01**	2.221E+00	6.886E+00	8.526E+00	4.823E+00	1.207E+01	5.984E+00	1.
Mean	**6.017E+02**	6.105E+02	6.997E+02	6.130E+02	6.947E+02	6.541E+02	6.823E+02	6.
Std	3.529E+01	5.244E+01	1.712E+02	6.258E+01	**3.059E+01**	8.165E+01	8.336E+01	1.
Mean	**1.093E+03**	1.251E+03	2.410E+03	1.144E+03	1.829E+03	1.197E+03	1.823E+03	1.
Std	3.704E+01	**2.671E+01**	6.292E+01	4.516E+01	2.807E+01	7.025E+01	3.672E+01	7.
Mean	1.063E+03	1.217E+03	1.504E+03	**1.062E+03**	1.507E+03	1.086E+03	1.414E+03	1.
Std	**1.226E+03**	2.584E+03	8.774E+03	2.603E+03	3.258E+03	1.118E+04	4.739E+03	7.
Mean	**2.819E+03**	8.375E+03	4.396E+04	5.094E+03	3.412E+04	2.343E+04	2.669E+04	3.

Table 11. Cont.

Fun	Index	SDHCPO	CPO	CFOA	PKO	CDO	MVO	HOA	WOA
F10	Std	8.849E+02	**4.544E+02**	9.001E+02	9.453E+02	5.159E+02	1.121E+03	8.736E+02	8.636E+02
	Mean	1.059E+04	1.352E+04	1.542E+04	9.607E+03	1.566E+04	**8.072E+03**	1.334E+04	1.334E+04
F11	Std	**1.139E+02**	2.146E+02	1.543E+04	2.642E+03	8.049E+02	1.235E+02	2.762E+03	1.919E+03
	Mean	**1.597E+03**	1.832E+03	4.278E+04	4.844E+03	1.978E+04	1.696E+03	1.905E+04	8.431E+03
F12	Std	**4.890E+06**	1.001E+07	1.176E+10	2.900E+07	4.510E+08	5.301E+07	1.236E+10	1.391E+09
	Mean	**8.926E+06**	2.054E+07	5.457E+10	4.265E+07	5.635E+10	9.778E+07	5.254E+10	4.568E+09
F13	Std	**1.998E+03**	3.790E+04	8.506E+09	5.186E+05	1.245E+09	1.424E+05	9.338E+09	3.097E+08
	Mean	**4.484E+03**	2.390E+04	2.602E+10	4.070E+05	4.737E+10	3.559E+05	2.799E+10	5.282E+08
F14	Std	**9.004E+04**	2.209E+05	3.120E+07	7.754E+05	5.741E+06	2.561E+05	3.333E+07	4.813E+06
	Mean	**1.157E+05**	1.563E+05	3.883E+07	1.135E+06	7.361E+07	3.653E+05	5.112E+07	6.809E+06
F15	Std	**3.599E+03**	1.262E+04	2.249E+09	8.172E+04	2.847E+08	8.831E+04	2.257E+09	8.688E+07
	Mean	**6.693E+03**	1.668E+04	4.196E+09	8.441E+04	1.574E+10	1.393E+05	4.171E+09	9.271E+07
F16	Std	**3.322E+02**	4.369E+02	1.123E+03	4.785E+02	4.735E+02	3.787E+02	1.133E+03	1.129E+03
	Mean	**3.257E+03**	4.525E+03	7.660E+03	3.716E+03	9.680E+03	3.507E+03	6.970E+03	6.611E+03
F17	Std	2.420E+02	**2.198E+02**	1.130E+04	3.103E+02	2.655E+02	3.938E+02	9.255E+02	6.577E+02
	Mean	**2.958E+03**	3.491E+03	1.301E+04	3.201E+03	8.642E+03	3.327E+03	4.891E+03	4.709E+03

F18	Std	**1.323E+06**	1.459E+06	8.887E+07	2.706E+06	2.990E+06	2.061E+06	4.725E+07	2.942E+07
	Mean	**1.440E+06**	2.310E+06	1.438E+08	4.710E+06	2.071E+08	2.860E+06	8.806E+07	5.222E+07
F19	Std	**6.109E+03**	7.865E+03	1.031E+09	5.689E+04	2.856E+07	4.511E+06	7.379E+08	1.792E+07
	Mean	**1.670E+04**	1.950E+04	2.071E+09	4.778E+04	1.916E+09	6.663E+06	1.409E+09	2.070E+07
F20	Std	2.487E+02	**1.886E+02**	3.782E+02	3.914E+02	2.784E+02	3.154E+02	2.355E+02	3.515E+02
	Mean	**2.994E+03**	3.678E+03	4.467E+03	3.195E+03	4.209E+03	3.260E+03	3.713E+03	3.947E+03
F21	Std	3.296E+01	2.188E+01	8.515E+01	5.547E+01	**2.085E+01**	6.122E+01	6.630E+01	1.330E+02
	Mean	**2.538E+03**	2.702E+03	3.123E+03	2.540E+03	3.000E+03	2.571E+03	2.976E+03	3.101E+03
F22	Std	2.137E+03	5.557E+03	1.009E+03	1.698E+03	**4.579E+02**	1.112E+03	8.627E+02	1.183E+03
	Mean	1.188E+04	1.230E+04	1.711E+04	1.060E+04	1.684E+04	**9.863E+03**	1.504E+04	1.452E+04
F23	Std	5.159E+01	**3.305E+01**	1.740E+02	5.327E+01	2.852E+02	6.224E+01	2.738E+02	1.755E+02
	Mean	3.021E+03	3.183E+03	4.125E+03	**3.006E+03**	4.958E+03	3.025E+03	4.502E+03	3.866E+03
F24	Std	5.598E+01	**2.693E+01**	2.041E+02	4.656E+01	1.009E+02	7.014E+01	2.162E+02	1.560E+02
	Mean	3.167E+03	3.345E+03	4.298E+03	**3.134E+03**	4.616E+03	3.148E+03	5.001E+03	3.960E+03
F25	Std	3.996E+01	5.946E+01	3.073E+03	7.149E+01	1.998E+02	**3.349E+01**	1.295E+03	7.021E+02
	Mean	3.163E+03	3.243E+03	1.564E+04	3.236E+03	7.443E+03	**3.072E+03**	1.155E+04	5.438E+03
F26	Std	5.468E+02	2.030E+03	1.667E+03	3.934E+02	**2.527E+02**	6.064E+02	7.439E+02	1.840E+03
	Mean	6.814E+03	7.859E+03	1.810E+04	**6.348E+03**	1.642E+04	6.583E+03	1.588E+04	1.493E+04
F27	Std	**6.635E+01**	8.286E+01	5.809E+02	7.180E+01	9.625E+02	8.342E+01	7.707E+02	6.086E+02
	Mean	3.500E+03	3.726E+03	5.651E+03	3.497E+03	8.767E+03	**3.472E+03**	7.038E+03	4.933E+03
F28	Std	6.871E+01	8.010E+01	1.535E+03	2.124E+02	1.365E+02	**3.367E+01**	8.940E+02	6.619E+02
	Mean	3.538E+03	3.747E+03	1.184E+04	3.792E+03	9.376E+03	**3.332E+03**	1.046E+04	6.072E+03
F29	Std	2.927E+02	**2.722E+02**	3.791E+04	4.613E+02	2.216E+03	4.528E+02	1.021E+04	1.929E+03
	Mean	**4.397E+03**	5.173E+03	3.793E+04	5.166E+03	2.458E+04	5.075E+03	2.252E+04	9.495E+03
F30	Std	**6.020E+05**	6.063E+06	1.967E+09	1.135E+07	2.768E+08	2.969E+07	1.462E+09	1.170E+08
	Mean	**2.381E+06**	1.190E+07	4.604E+09	1.008E+07	5.471E+09	9.066E+07	3.055E+09	3.154E+08
Std rank		**2.14**	2.45	6.86	4.28	4.14	4.14	6.00	5.97
Mean rank		**1.52**	3.41	7.41	2.72	7.03	2.55	5.97	5.34
Overall rank		**1**	2	8	4	5	3	7	6

High-dimensional experimental results show that, on the 50-dimensional CEC2017 benchmark, SDHCPO still attains the best or near-best mean objective values on the vast majority of functions, and its average Friedman rank remains markedly superior to those of the other algorithms, with an overall ordering consistent with the 30-dimensional case. In contrast, although the original CPO can still produce reasonably good best-so-far values on several functions, its mean performance deteriorates significantly at 50 dimensions, a phenomenon that is particularly pronounced on the hybrid and composition functions. In other words, while the mean ranking of CPO is still acceptable at 30 dimensions, the performance gap in terms of average behavior between SDHCPO and CPO becomes further amplified in the 50-dimensional setting.

In terms of standard deviation, the fluctuation level of all algorithms increases as the dimensionality rises. However, the growth in SDHCPO's standard deviation is relatively moderate, and its overall variability remains at a low-to-medium level without exhibiting any pronounced numerical instability. On some functions, the improvement in standard deviation is less marked than that in the mean, but at least comparable stability to the 30-dimensional case is maintained. This indicates that, in high-dimensional spaces, SDHCPO can significantly reduce the objective value without incurring a substantial loss of robustness.

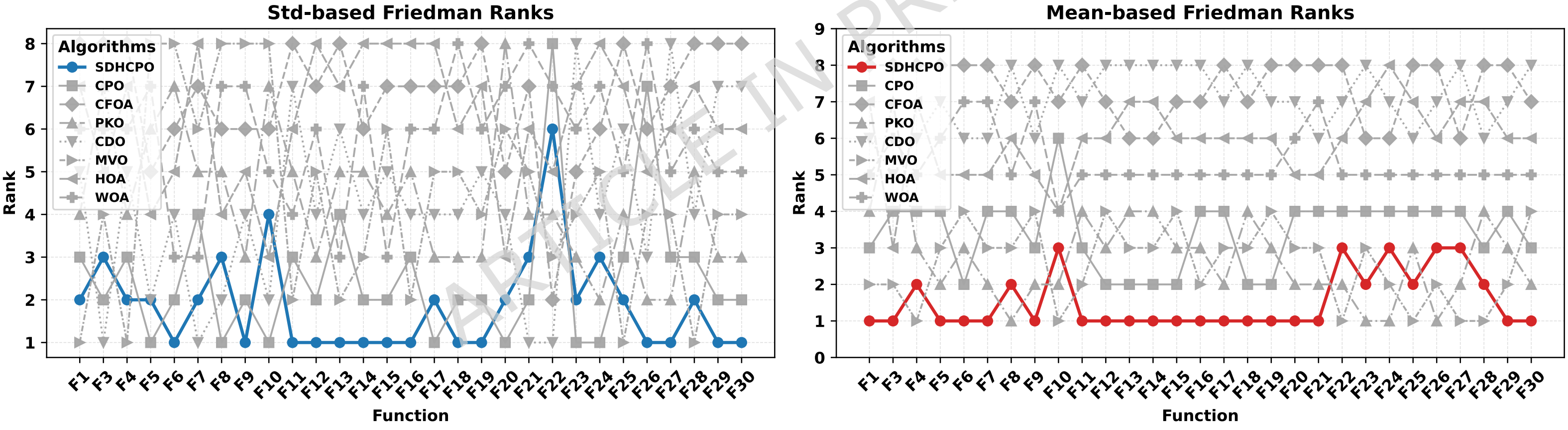


Fig. 12 Friedman Rank Evolution of SDHCPO vs. Competitors on CEC2017 Test Suite (d=50)

Figure 12 depicts the per-function ranking distribution of all algorithms on the 50-dimensional CEC2017 test set. It can be observed that SDHCPO consistently ranks among the top methods on most functions.

Table 12. Wilcoxon Rank-Sum p-Values: SDHCPO vs. Algorithms on CEC2017 (d=50)

Fun	CPO-p	CFOA-p	PKO-p	CDO-p	MVO-p	HOA-p	WOA-p
1	3.020E-11	3.020E-11	1.777E-10	3.020E-11	1.784E-04	3.020E-11	3.020E-11
3	1.167E-05	3.020E-11	3.020E-11	3.338E-11	8.236E-02	7.697E-04	3.020E-11

4	9.063E-08	3.020E-11	2.006E-04	3.020E-11	2.510E-02	3.020E-11	3.020E-11
5	3.020E-11	3.020E-11	3.265E-02	3.020E-11	1.765E-02	3.020E-11	3.020E-11
6	3.020E-11	3.020E-11	6.696E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11
7	3.020E-11	3.020E-11	4.444E-07	3.020E-11	1.407E-04	3.020E-11	3.020E-11
8	3.020E-11	3.020E-11	2.416E-02	3.020E-11	4.637E-03	3.020E-11	3.020E-11
9	2.034E-09	3.020E-11	2.499E-03	3.020E-11	3.020E-11	3.020E-11	3.020E-11
10	4.504E-11	3.338E-11	9.514E-06	3.020E-11	4.975E-11	1.287E-09	3.497E-09
11	7.043E-07	3.020E-11	4.975E-11	3.020E-11	3.778E-02	3.020E-11	3.020E-11
12	4.686E-08	3.020E-11	2.670E-09	3.020E-11	3.020E-11	3.020E-11	3.020E-11
13	5.494E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11	3.020E-11
14	2.282E-01	3.020E-11	3.020E-11	3.020E-11	6.765E-05	3.020E-11	3.020E-11
15	2.891E-03	3.020E-11	2.133E-05	3.020E-11	3.020E-11	3.020E-11	3.020E-11
16	4.504E-11	3.020E-11	3.147E-02	3.020E-11	2.891E-03	3.020E-11	3.020E-11

Table 12. Cont.

Fun	CPO-p	CFOA-p	PKO-p	CDO-p	MVO-p	HOA-p	WOA-p
17	7.773E-09	3.020E-11	3.339E-03	3.020E-11	6.528E-08	4.077E-11	3.020E-11
18	1.629E-02	3.020E-11	6.010E-08	3.020E-11	1.767E-03	3.020E-11	3.020E-11
19	1.076E-02	3.020E-11	4.637E-03	3.020E-11	3.020E-11	3.020E-11	3.020E-11
20	7.389E-11	3.020E-11	4.459E-04	3.020E-11	2.813E-02	1.359E-07	3.820E-10
21	3.020E-11	3.020E-11	8.073E-01	3.020E-11	3.778E-02	3.020E-11	3.020E-11
22	8.564E-04	3.338E-11	4.676E-02	3.020E-11	6.283E-06	8.993E-11	1.613E-10
23	3.020E-11	3.020E-11	6.414E-01	3.020E-11	1.958E-01	3.020E-11	3.020E-11
24	3.020E-11	3.020E-11	6.567E-02	3.020E-11	9.049E-02	3.020E-11	3.020E-11
25	5.859E-06	3.020E-11	1.058E-03	3.020E-11	4.200E-10	3.020E-11	3.020E-11
26	1.108E-06	3.020E-11	1.761E-01	3.020E-11	1.624E-01	3.020E-11	3.020E-11
27	4.077E-11	3.020E-11	1.501E-02	3.020E-11	2.519E-01	3.020E-11	3.020E-11
28	3.338E-11	3.020E-11	1.857E-09	3.020E-11	3.020E-11	3.020E-11	3.020E-11
29	1.957E-10	3.020E-11	3.352E-08	3.020E-11	3.497E-09	3.020E-11	3.020E-11

30	3.020E-11	3.020E-11	3.520E-07	3.020E-11	3.020E-11	3.020E-11	3.020E-11
-/=/+	0/0/29	0/0/29	1/4/25	0/0/29	5/4/20	0/0/29	0/0/29

Table 12 reports the Wilcoxon rank-sum test results for SDHCPO against each comparison algorithm on the high-dimensional CEC2017 set. It can be seen that, relative to CPO, CFOA, CDO, HOA, and WOA, SDHCPO is significantly superior on all 29 functions (0/0/29). Against PKO, the result is 1/4/25, and against MVO it is 5/4/20, indicating that in the high-dimensional setting SDHCPO still enjoys statistically significant advantages on the vast majority of functions, and is only comparable to or slightly inferior on a small subset of cases when compared with these two stronger algorithms.

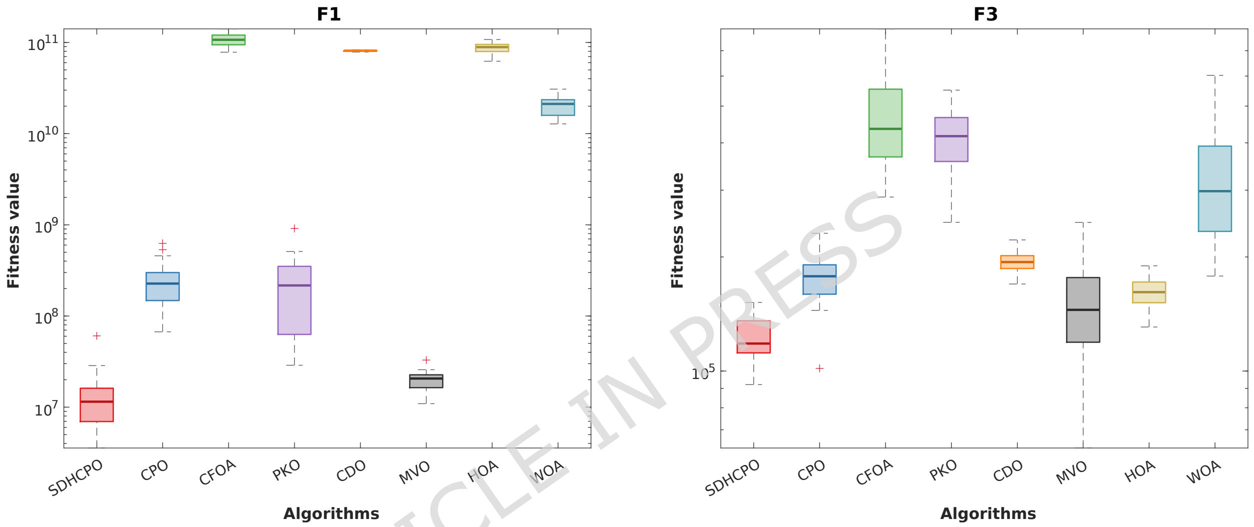
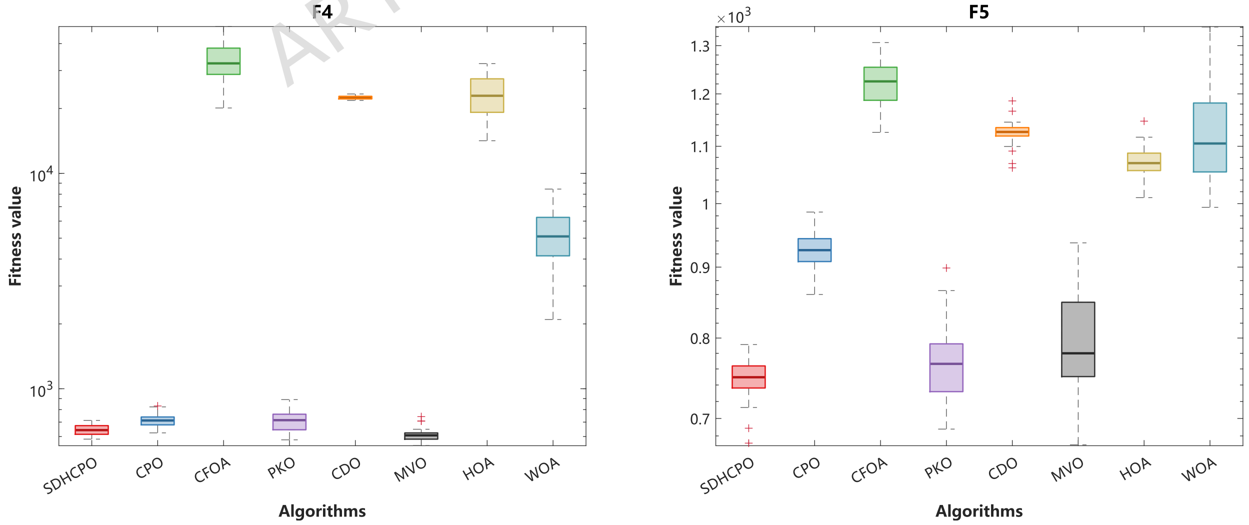


Fig. 13 CEC2017 boxplots: SDHCPO vs. algorithms (d=50, F1, F3)



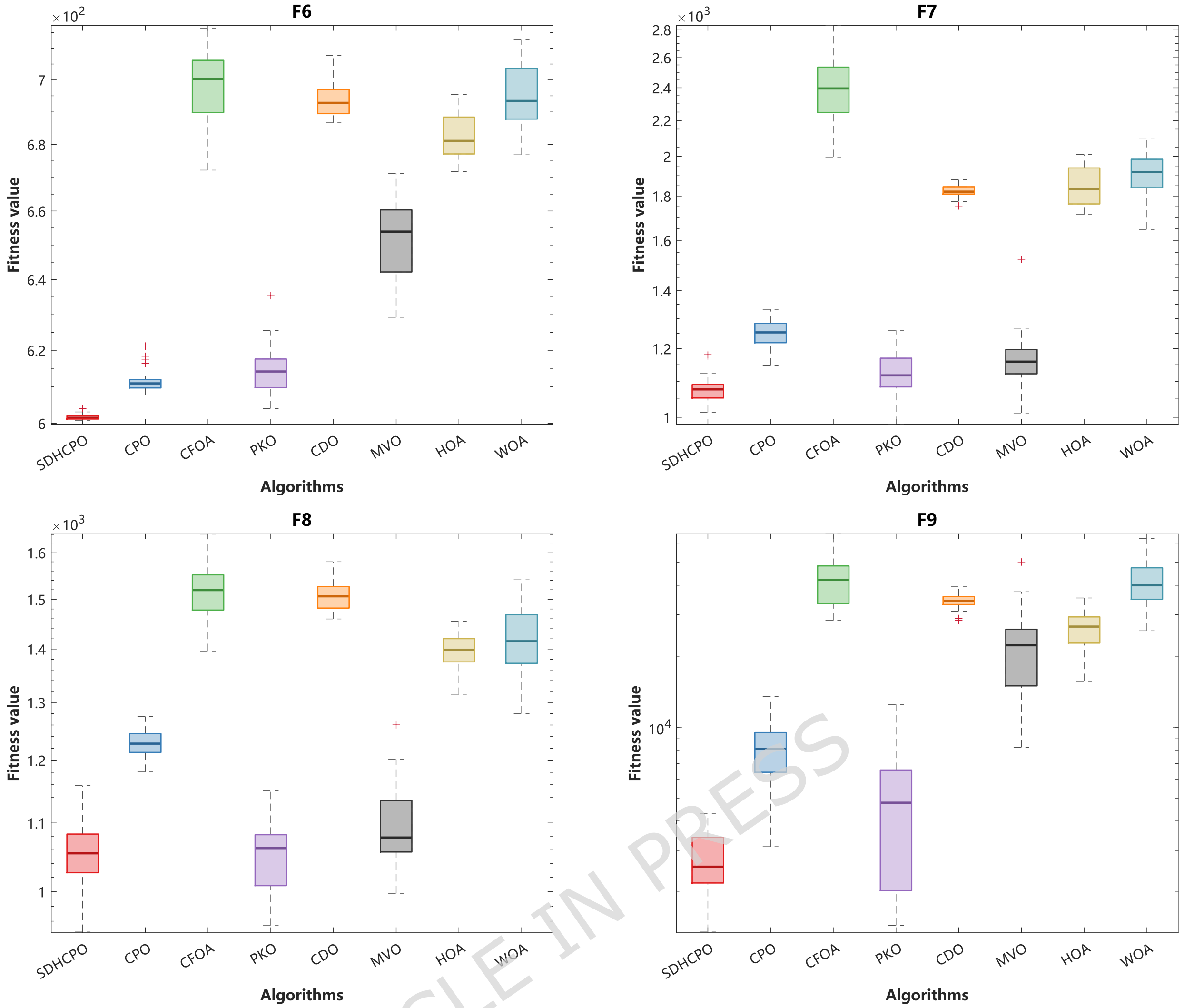
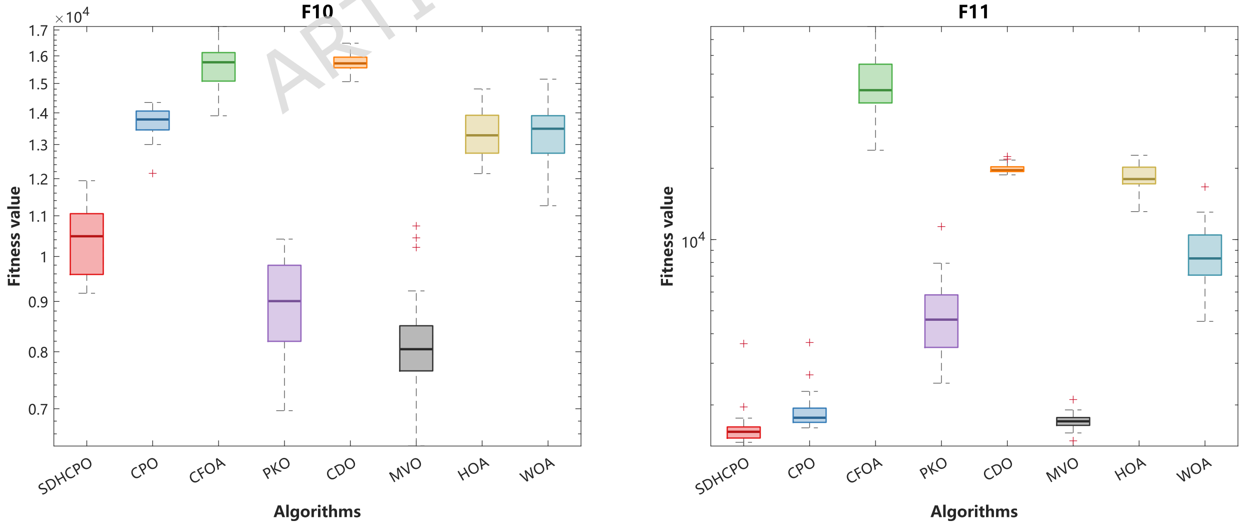


Fig. 14 CEC2017 boxplots: SDHCPO vs. algorithms (d=50, F4-F9)



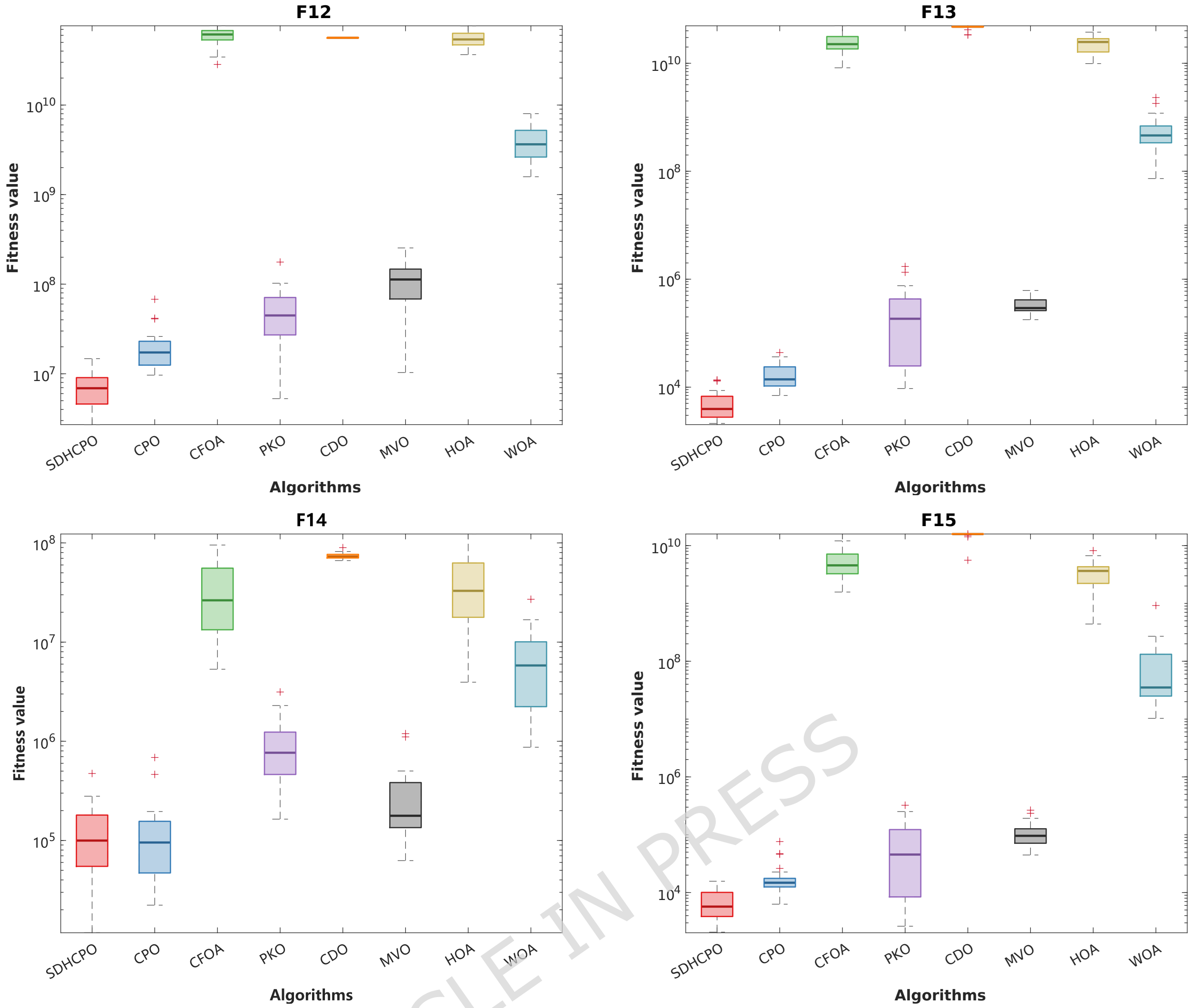
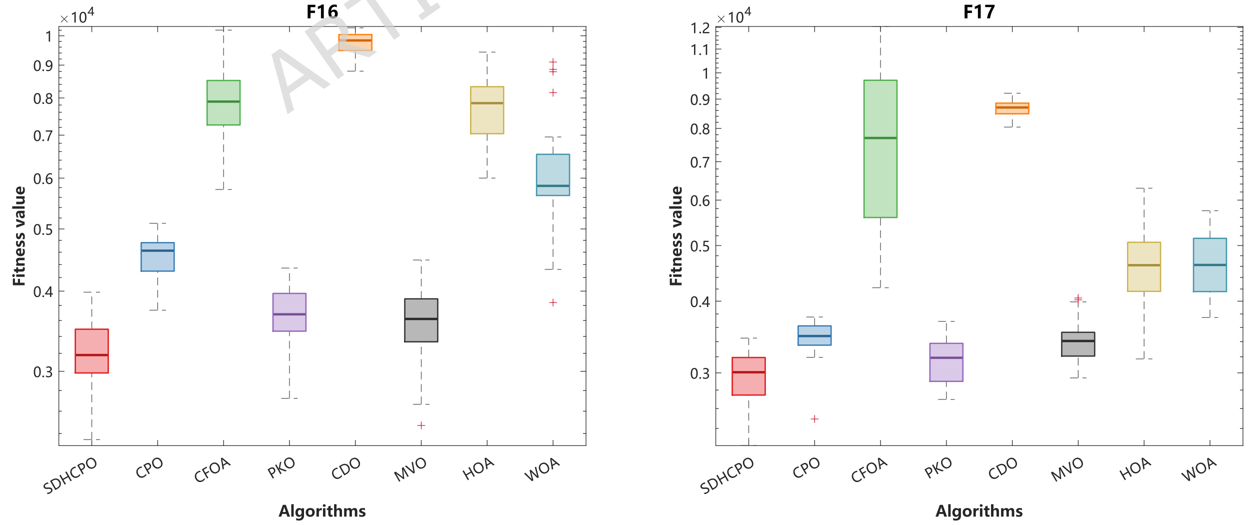


Fig. 15 CEC2017 boxplots: SDHCPO vs. algorithms (d=50, F10-F15)



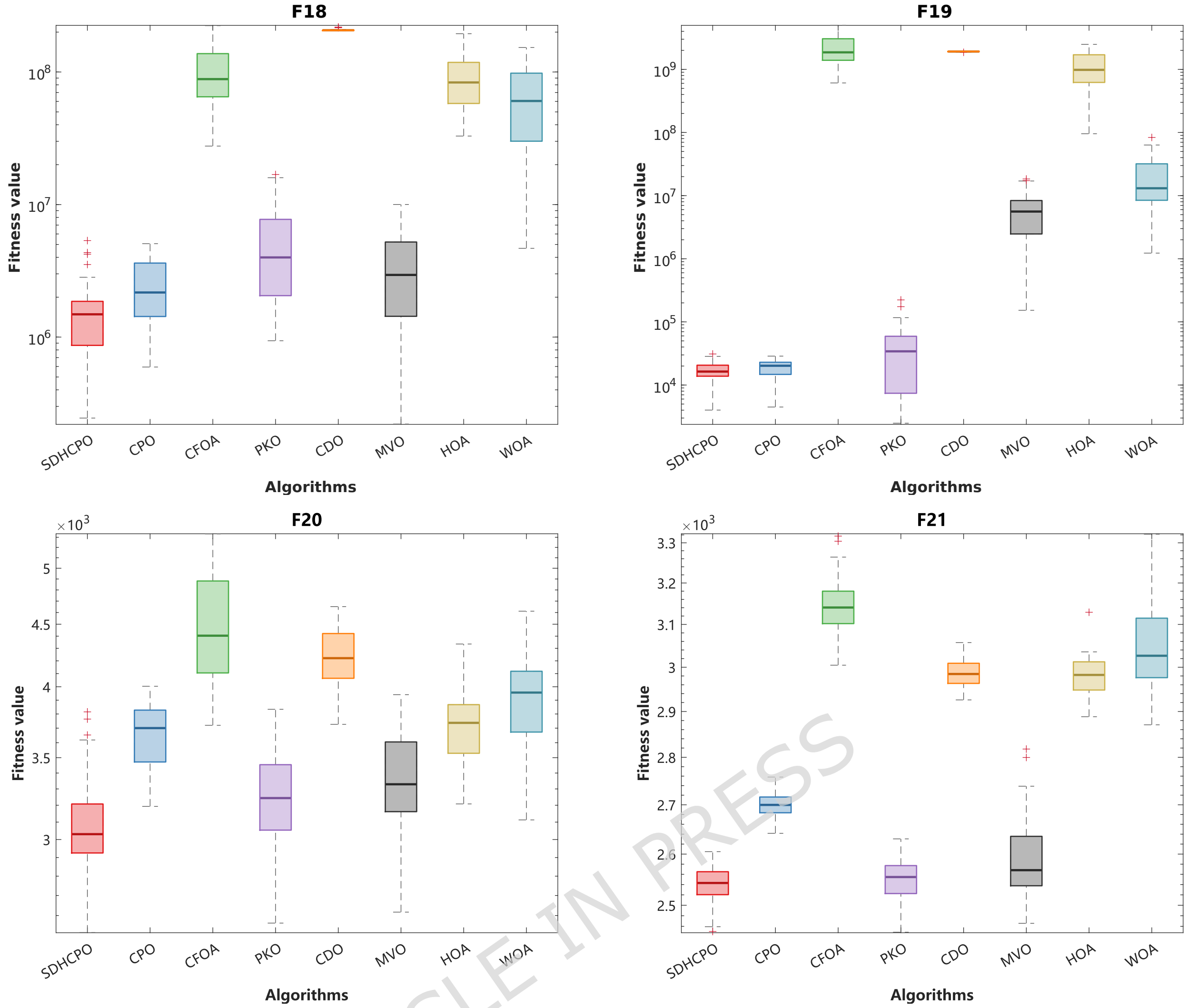
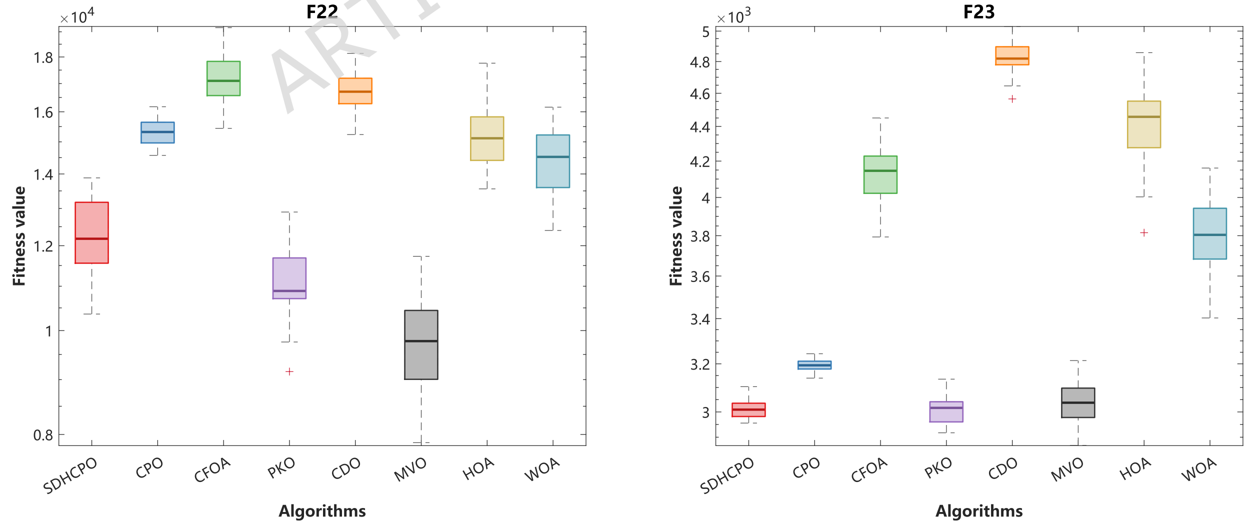


Fig. 16 CEC2017 boxplots: SDHCPO vs. algorithms (d=50, F16-F21)



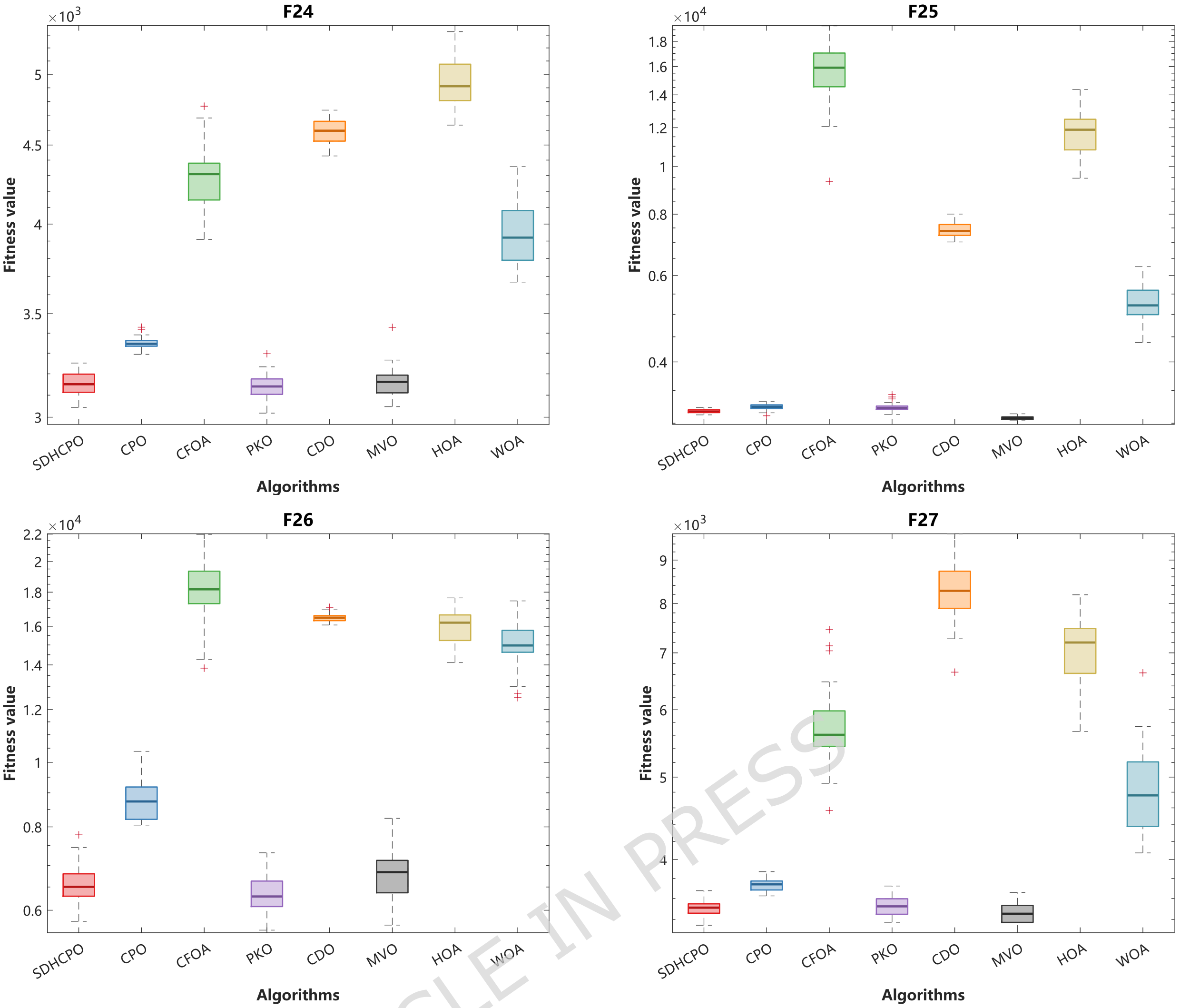
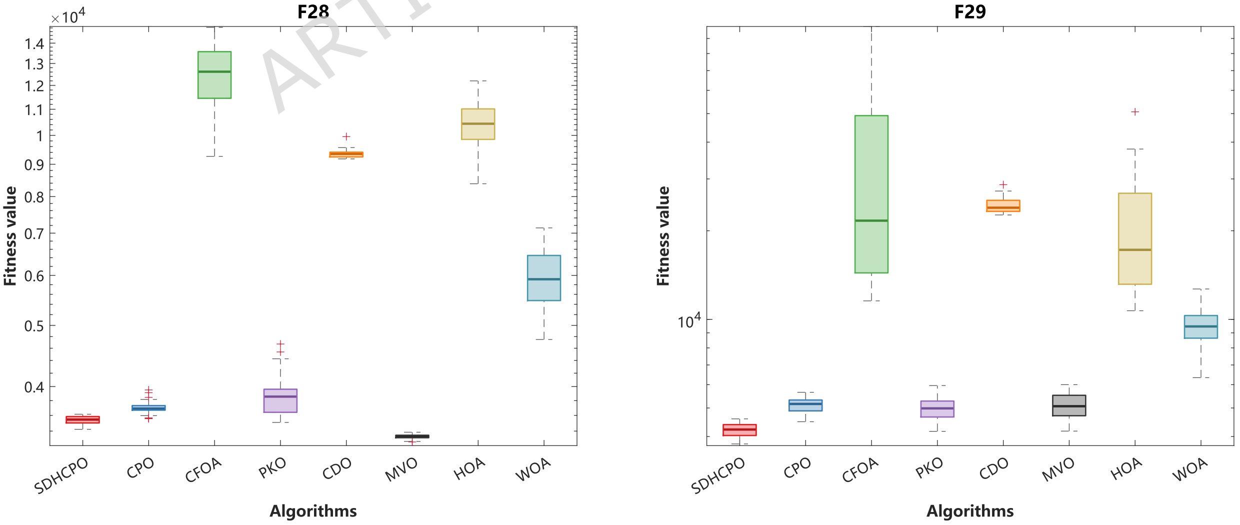


Fig. 17 CEC2017 boxplots: SDHCPO vs. algorithms (d=50, F22-F27)



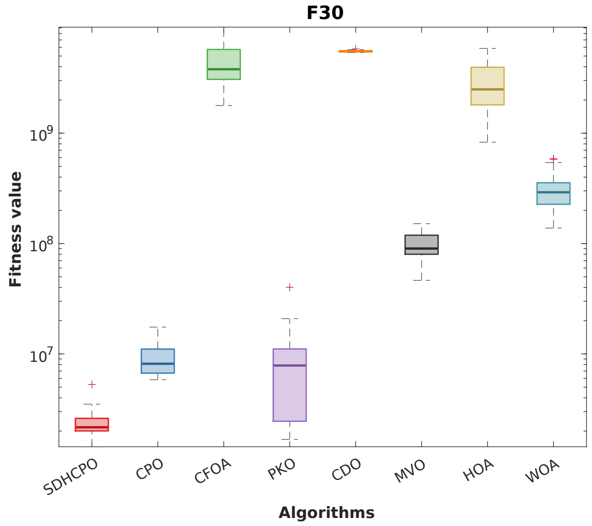


Fig. 18 CEC2017 boxplots: SDHCPO vs. algorithms (d=50, F28-F30)

The boxplots in Figures 13-18 further demonstrate that SDHCPO exhibits marked advantages in solution accuracy, stability, and robustness. For almost all test functions, the boxes corresponding to SDHCPO lie in the lowest fitness-value region, indicating not only the best median performance but also that at least three quarters of its runs outperform even the best results of the competing algorithms. At the same time, the boxes of SDHCPO are generally short, with relatively narrow whiskers, especially in comparison with algorithms such as PKO, CFOA, and WOA. This reflects the outstanding stability and consistency of SDHCPO across multiple independent runs, with very low performance variability and little susceptibility to random perturbations. Only on a few functions, such as F4, F14, F25, and F28, do the boxplots of CPO or MVO appear numerically comparable to those of SDHCPO. Overall, however, SDHCPO delivers the best comprehensive performance and robustness on the 50-dimensional CEC2017 test set.

Taken together, the 30 and 50-dimensional results show that, compared with the original CPO and other competing algorithms, SDHCPO exhibits a more pronounced average performance advantage in 50 dimensions, particularly on high-dimensional hybrid and composition functions, thereby demonstrating strong scalability with respect to dimensionality. Although the overall standard deviation increases, the fluctuation level of SDHCPO remains competitive relative to the other algorithms, and no high-dimensional degradation or divergence is observed. This indicates that the four strategies designed in this study continue to function synergistically in the 50-dimensional setting and effectively alleviate the search difficulties caused by the curse of dimensionality.

5 Application Validation of SDHCPO in Engineering Optimization Problems

To assess the effectiveness and robustness of SDHCPO in real-world engineering optimization, this section considers five representative engineering design problems: welded beam design, tension/compression spring design, pressure vessel design, three-bar truss design, and 72-bar spatial truss design. These problems share the common characteristic of requiring the optimization of multiple design parameters under various constraints, with objectives such as minimizing structural weight or reducing material cost. To ensure the fairness and comparability of the experimental results, all tests are conducted under consistent settings: a population size of 30, a maximum of 500 iterations, and 30 independent runs for each problem.

5.1 Welded Beam Design Problem

The welded beam design problem is a constrained optimization task aimed at minimizing the weight of the beam [45]. The design variables comprise four geometric parameters: weld thickness h, beam length l, beam thickness t, and weld width b. The objective function evaluates the beam weight based on these geometric relationships. At the same time, the problem is subject to seven constraints, including limits on shear stress $t(x)$, bending stress $s(x)$, and deflection $d(x)$, as well as geometric and stress-equilibrium constraints, to ensure the structural safety and feasibility of the design. The problem configuration is depicted in Figure 19.

As shown in Table 13, SDHCPO achieves superior performance on the welded beam design problem, as indicated by the bolded optimal value, mean, and standard deviation. The corresponding optimal solution is given by: weld thickness h=0.2057, weld length l=3.4704, beam width t=9.0366, and beam height b=0.2507.

Consider variable $X = [x_1, x_2, x_3, x_4] = [h, l, t, b]$

Minimize $f(x) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$

Subject to

$$
\begin{gathered}
g_1(X) = t(x) - t_{max} \pounds 0 \\
g_2(X) = s(x) - s_{max} \pounds 0 \\
g_3(X) = d(x) - d_{max} \pounds 0 \\
g_4(X) = x_1 - x_4 \pounds 0 \\
g_5(X) = P - P_c(x) \pounds 0 \\
g_6(X) = 0.125 - x_1 \pounds 0 \\
g_7(X) = 1.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \pounds 0
\end{gathered}
$$

Variable range

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

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

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \qquad \tau'' = \frac{MR}{J}$$

$$M = P\left(L + \frac{x_2}{2}\right)$$

Where

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

$$J = 2\left(\sqrt{2}x_1x_2\left(\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right)\right)$$

$$P_c(x) = \frac{4.013E\sqrt{\frac{x_2^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$\sigma(x) = \frac{6PL}{x_4x_2^2}, \qquad \delta(x) = \frac{4PL^3}{Ex_3^3x_4}$$

$$P = 6000\ \text{lb}, L = 14\ \text{in}, \delta_{max} = 0.25\ \text{in}, E = 30 \times 10^6\ \text{psi}$$
$$G = 12 \times 10^6\ \text{psi}, \tau_{max} = 13600\ \text{psi}, \sigma_{max} = 30000\ \text{psi}$$

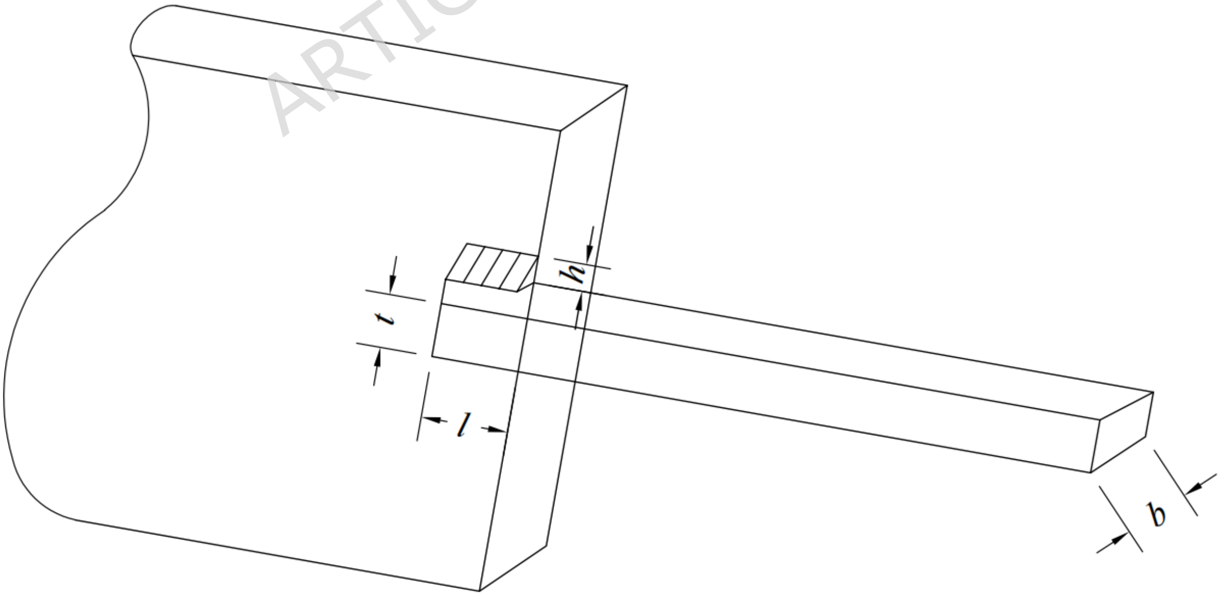


Fig.19 Schematic diagram of cantilever beam

Table 13. Results of All Algorithms on the Welded Beam Design Problem

Algorithm	Optimal Values for Variables				Optimal Weight	Average Value	Standard Deviation
	h	l	t	b			

SDHCPO	0.2057358	3.4703569	9.0366243	0.2057296	**1.7248478**	**1.7248503**	**0.0000046**
CPO	0.2057280	3.4705097	9.0366974	0.2057325	1.7248870	1.7253144	0.0005242
CFOA	0.1990293	3.9577508	8.7223931	0.2241047	1.8619768	2.7454742	0.8181638
PKO	0.2057301	3.4704767	9.0366592	0.2057300	1.7248600	1.7251203	0.0003491
CDO	0.2042892	3.5006486	9.2297572	0.2084563	1.7813186	1.8457670	0.0321628

Table 13. Cont.

Algorithm	Optimal Values for Variables				Optimal Weight	Average Value	Standard Deviation
	h	l	t	b			
SDHCPO	0.2057358	3.4703569	9.0366243	0.2057296	**1.7248478**	**1.7248503**	**0.0000046**
MVO	0.2048328	3.4985344	9.0374170	0.2059497	1.7290605	1.7539214	0.0341230
HOA	0.2209976	3.8431975	7.5597302	0.3086653	2.2104538	3.1764949	0.4571604
WOA	0.1577181	4.6001361	9.6190287	0.2029958	1.8737189	2.6543622	0.5600200

5.2 Pressure Vessel Design Problem

The pressure vessel design problem seeks to minimize the structural material weight [46]. It involves four design variables: shell thickness T_s , head thickness T_h, inner radius R, and cylindrical shell length L, collectively denoted as$[y_1, y_2, y_3, y_4]$. The design is constrained by four inequalities: g_1 and g_2 impose minimum thickness requirements on the shell and head to prevent structural failure caused by insufficient material; g_3 limits the stress level; and g_4 specifies an upper bound on the cylindrical shell length. The design variables are bounded within engineeringly feasible ranges, with y_1, y_2 Î $[0,99]$, and y_3, y_4 Î $[10,200]$. The corresponding structural schematic is illustrated in Figure 20.

SDHCPO attains the minimum weight of 5734.913157 for the pressure vessel design problem, with the corresponding design variables reported in Table 14. This value is the best among all compared algorithms, and the standard deviation over 30 runs is markedly lower than that of the other methods.

Consider variable $Y = [y_1, y_2, y_3, y_4] = [T_s, T_h, R, L]$

Minimize $f(y) = 0.6224 y_1 y_3 y_4 + 1.7781 y_2 y_3^2 + 3.1661 y_1^2 y_4 + 19.84 y_1^2 y_3$

Subject to

$$g_1(Y) = -y_1 + 0.0193 y_2 \leq 0$$
$$g_2(Y) = -y_2 + 0.00954 y_3 \leq 0$$
$$g_3(Y) = -\pi y_2 y_3^2 y_4 - \frac{4}{3} \pi y_3^3 + 1296000 \leq 0$$
$$g_4(Y) = y_4 - 240 \leq 0$$

Variable range

$$0 \leq y_1, y_2 \leq 99$$
$$10 \leq y_3, y_4 \leq 200$$

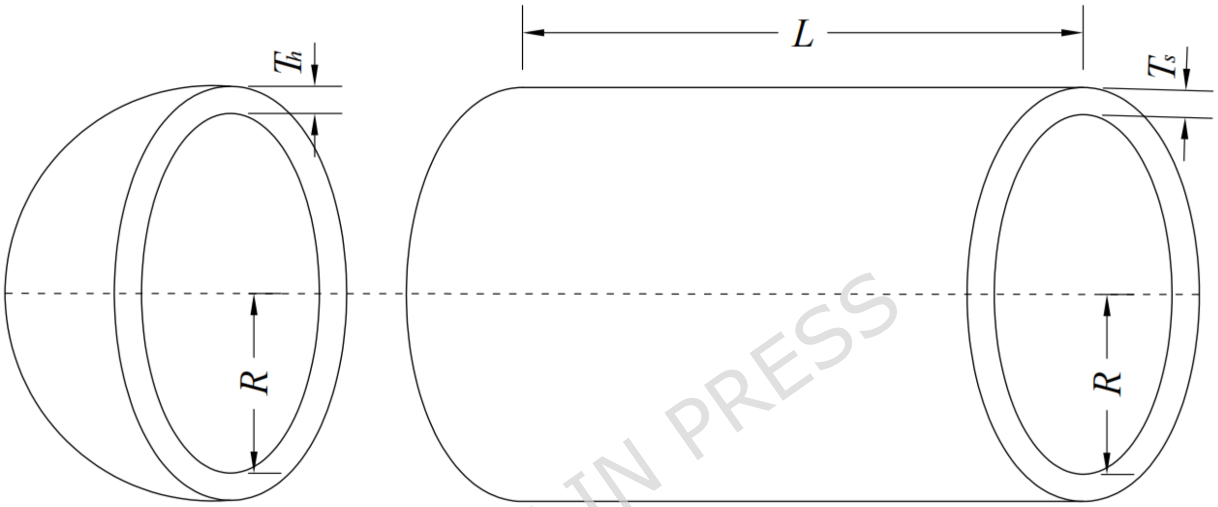


Fig.20 Schematic diagram of pressure vessel

Table 14. Results of All Algorithms on the Pressure Vessel Design Problem

Algorithm	Optimal Values for Variables				Optimal Weight	Average Value	Standard Deviation
	Ts	Th	R	L			
SDHCPO	0.7424336	0.3701961	40.3196187	200	**5734.9131570**	**5734.9132223**	**0.0001859**
CPO	0.7424216	0.3700774	40.3196324	199.9998095	5734.9151170	5735.2739790	1.6779232
CFOA	0.7678084	0.3867046	41.5987812	186.5353308	5865.9261129	40789.5180883	50282.0649233
PKO	0.7424356	0.3701942	40.3196187	200	5734.9131634	5750.2190817	47.0773456
CDO	0.7216561	0.3554438	40.4484786	200	5856.0091856	6012.4315885	97.0115631
MVO	0.8407224	0.4058078	44.6692450	147.3116683	5926.2125862	6591.0809350	427.1301470
HOA	1.0105148	0.4865822	53.6884007	74.2060890	6458.6563412	7689.6378519	704.9414773
WOA	0.7362395	0.4161056	40.6854150	194.9697617	5871.7764090	13133.0184431	16126.1868194

5.3 Tension/Compression Spring Design Problem

The tension/compression spring design problem [47] involves three variables: z_1 represents the wire diameter d, z_2 the mean coil diameter D, and z_3 the number of active coils N. The objective is to minimize the spring weight, subject to four constraints. The schematic of this problem is shown in Figure 21.

The comparison results reported in Table 15 show that SDHCPO identifies the best set of design parameters, achieving a minimum objective value of 0.0135672 and a mean value of 0.0135777, ranking first among all algorithms. This confirms the effectiveness of SDHCPO in solving the spring design problem.

Consider variable $Z = [z_1, z_2, z_3] = [d, D, N]$

Minimize $f(z) = z_1^2 z_2 z_3 + 2 z_1^2 z_2$

Subject to

$$g_1(Z) = 1 - \frac{z_2^2 z_3}{71785 z_1^2} \le 0$$

$$g_2(Z) = \frac{4 z_2^2 - z_1 z_2}{12566(z_1^2 z_2 - z_1^3)} + \frac{1}{5108 z_1^2} - 1 \le 0$$

$$g_3(Z) = 1 - \frac{140.45 z_1}{z_2^2 z_3} \le 0$$

$$g_4(Z) = \frac{z_1 + z_2}{1.5} - 1 \le 0$$

Variable range $0.05 \le z_1 \le 2,\quad 0.25 \le z_2 \le 1.3 \quad 2 \le z_3 \le 15$

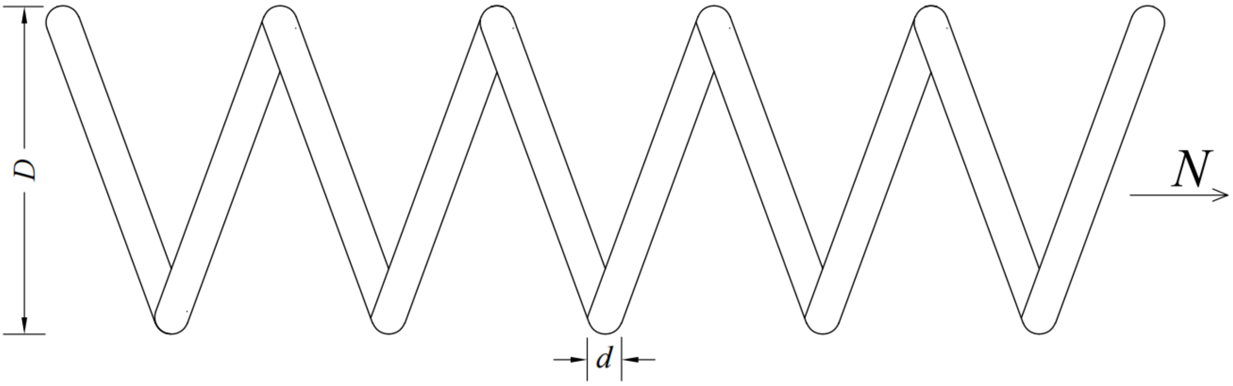


Fig.21 Schematic diagram of tension/compression spring

Table 15. Results of All Algorithms on the Tension/Compression Spring Design Problem

Algorithm	Optimal Values for Variables			Optimal Weight	Average Value	Standard Deviation
	d	D	N			

SDHCPO	0.0529873	0.3974161	10.1590673	**0.0135672**	**0.0135777**	**0.0000170**
CPO	0.0529578	0.3966766	10.1964045	0.0135684	0.0135824	0.0000220
CFOA	0.0527075	0.3904548	10.5235112	0.0135844	0.0229571	0.0135615
PKO	0.0529984	0.3976885	10.1458126	0.0135674	0.0136130	0.0000559
CDO	0.0000559	0.3280588	14.8412948	0.0138123	0.0139953	0.0003072
MVO	0.0500000	0.3267802	14.9233195	0.0138255	0.0178244	0.0015443
HOA	0.0525083	0.3844025	10.8709435	0.0136412	0.0150239	0.0013185
WOA	0.0533331	0.4060281	9.7491859	0.0135693	0.0143547	0.0011676

5.4 Three-Bar Truss Design Problem

The three-bar truss design optimization problem aims to minimize the structural material volume, thereby indirectly reducing the weight [48]. It involves two design variables: A_1 and A_3, the cross-sectional area of the two diagonal members, and A_2, the cross-sectional area of the vertical member. The objective function is determined by the member lengths and cross-sectional areas, with a base length parameter l=100cm. The design must satisfy three stress-related constraints. A schematic of the three-bar truss is provided in Figure 22.

As shown in Table 16, SDHCPO, CPO, and PKO all perform well on this problem, each obtaining the same optimal solution with a standard deviation of 0.

Consider variable $X = [x_1, x_2] = [A_1, A_2]$

Minimize $f(x) = l \times \left(2\sqrt{2}x_1 + x_2\right)$

Subject to

$$g_1(X)\frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0$$

$$g_2(X) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0$$

$$g_3(X) = \frac{1}{x_1 + \sqrt{2}x_2}P - \sigma \le 0$$

Variable range $0 \le x_1, x_2 \le 1$

Where $l = 100\text{cm}, P = 2\text{KN/cm}^2, s = 2\text{KN/cm}^2$

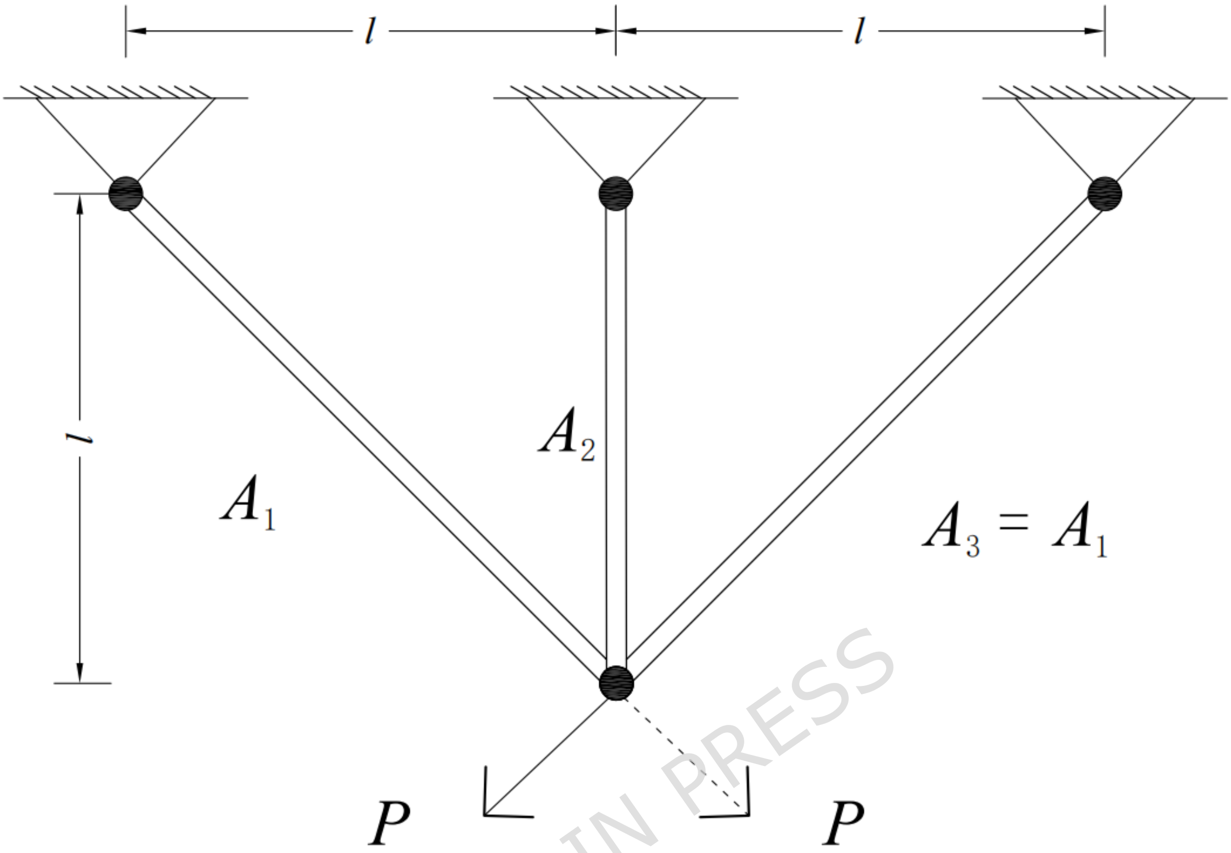


Fig.22 Schematic diagram of three-bar truss

Table 16. Results of All Algorithms on the Three-Bar Truss Design Problem

Algorithm	Optimal Values for Variables		Optimal Weight	Average Value	Standard Deviation
	A1	A2			
SDHCPO	0.7884152	0.4081138	**263.8523464**	**263.8523464**	**0**
CPO	0.7884152	0.4081138	**263.8523464**	**263.8523464**	**0**
CFOA	0.7882057	0.4087165	263.8523850	264.7752980	1.4950354
PKO	0.7884152	0.4081138	**263.8523464**	**263.8523464**	**0**
CDO	0.7885754	0.4077093	263.8524967	263.9357268	0.0864609
MVO	0.7884444	0.4080303	263.8523471	263.8524352	0.0001067
HOA	0.7884593	0.4079891	263.8523479	263.8981730	0.1027500
WOA	0.7887717	0.4071060	263.852439	265.160821	3.2927772

	8	6

5.5 72-Bar Spatial Truss Optimization Problem

Distinct from the foregoing classical benchmarks that involve only a small number of design variables, this subsection introduces a 72-bar spatial truss optimization problem with 16 design variables and multiple frequency constraints, in order to further assess the potential and practical applicability of the algorithm for high-dimensional, complex, and constrained real-world engineering problems [49].

This structure consists of 20 nodes and 72 members. To preserve structural symmetry, the 72 members are partitioned into 16 design groups according to their geometric locations, and all members within the same group share an identical cross-sectional area, resulting in 16 continuous design variables. The detailed grouping scheme is provided in Table 17.

Table 17. Member grouping scheme for the 72-bar spatial truss.

Group number (k)	Member type	Member indices (e)
	Level 1	
1	Columns	1, 2, 3, 4
2	Face diagonals	5, 6, 7, 8, 9, 10, 11, 12
3	Horizontals	13, 14, 15, 16
4	Internal diagonals	17, 18
	Level 2	
5	Columns	19, 20, 21, 22
6	Face diagonals	23, 24, 25, 26, 27, 28, 29, 30
7	Horizontals	31, 32, 33, 34
8	Internal diagonals	35, 36
	Level 3	
9	Columns	37, 38, 39, 40
10	Face diagonals	41, 42, 43, 44, 45, 46, 47, 48
11	Horizontals	49, 50, 51, 52
12	Internal diagonals	53, 54
	Level 4	
13	Columns	55, 56, 57, 58
14	Face diagonals	59, 60, 61, 62, 63, 64, 65, 66
15	Horizontals	67, 68, 69, 70
16	Internal diagonals	71, 72

The truss is made of aluminum alloy with elastic modulus $E » 6.9´10^{10} N/m^2$ and mass density $r » 2770 kg/m^3$. A nonstructural lumped mass of 2270 kg is attached to each of the top-layer nodes 1-4. The design variables are the 16 grouped cross-sectional areas x_k, which form the design vector $\mathbf{x} = (x_1, x_2, ¼, x_{16})^•$ .The kth variable x_k controls the cross-sectional areas of all members belonging to the kth group. The cross-sectional area of the eth member is denoted by $A_e(\mathbf{x})$,determined by the corresponding group variable x_k, and its length is denoted by L_e.

The optimization objective is to minimize the self-weight of the structure subject to multiple frequency constraints, with the objective function determined jointly by the material density ρ, the cross-sectional areas $A_e(\mathbf{x})$, and the lengths L_e of all members. The structural self-weight, together with the lumped masses at the top nodes, forms the global mass matrix $M(\mathbf{x})$, which, combined with the global stiffness matrix $K(\mathbf{x})$ determined by the design variables, defines a generalized eigenvalue problem. By solving this eigenvalue problem under fixed boundary conditions at the four bottom nodes, the eigenvalues $\lambda_r(\mathbf{x})$ and mode shapes ϕ_r are obtained, from which the first three natural frequencies $f_r(\mathbf{x})$ are computed.

In this problem, constraints are imposed on the first three natural frequencies: the first and second frequencies must not be lower than 4Hz, and the third must not be lower than 6Hz. Under these constraints, the design vector $\mathbf{x}$ is adjusted to obtain a 72-bar spatial truss configuration with minimum mass.

Consider variable $\mathbf{x}=(x_1,x_2,\ldots,x_{16})^{\bullet}$

Minimize $\min_{\mathbf{x}} f(\mathbf{x})=\rho\sum_{e=1}^{72}A_e(\mathbf{x})L_e$

Subject to

$$K(\mathbf{x})\phi_r=\lambda_r(\mathbf{x})M(\mathbf{x})\phi_r,\quad r=1,2,3$$

$$f_r(\mathbf{x})=\frac{\sqrt{\lambda_r(\mathbf{x})}}{2\pi},\quad r=1,2,3$$

$$f_1(\mathbf{x})\geq 4\text{Hz},\quad f_2(\mathbf{x})\geq 4\text{Hz},\quad f_3(\mathbf{x})\geq 6\text{Hz}$$

Variable range

$$A_{\min}\leq x_k\leq A_{\max},\quad k=1,\ldots,16$$

$$A_{\min}=6.45\times10^{-5}\text{m}^2\approx0.645\text{cm}^2$$

$$A_{\max}=5.0\times10^{-3}\text{m}^2\approx50\text{cm}^2$$

Table 18. Results of All Algorithms on the 72-Bar Spatial Truss Design Problem

Algorithm	Optimal Values for Variables			Optimal Weight	Average Value	Standard Deviation
	f_1	f_2	f_3			
SDHCPO	4.0000	4.0001	5.1900	116.6002	**116.7921**	**0.1158**
CPO	3.9997	4.0091	5.2730	117.7778	118.8341	0.5758
CFOA	4.7858	5.7236	7.8279	490.0387	658.3237	111.3120
PKO	3.9999	4.0006	5.2066	116.5697	116.8338	0.1850
CDO	4.0020	4.0238	5.2280	119.7953	122.6835	1.9200
MVO	4.0012	4.0141	5.2258	117.9871	131.2249	10.7874
HOA	4.0260	4.0294	5.3867	130.4777	145.4744	8.2826
WOA	4.0000	4.0632	5.3168	163.4465	261.8906	62.5870

According to the statistical results in Table 18, SDHCPO exhibits superior overall performance on this high-dimensional problem, achieving an average structural weight of 116.7921 and a standard deviation of 0.1158, both of which are the best among all compared algorithms. This outcome provides strong evidence that, as the search space expands sharply with increasing dimensionality, SDHCPO attains higher solution accuracy and robustness than CPO, PKO, and other competitors, effectively escaping local optima and stably converging to high-quality solutions.

Conclusion

This paper proposes a multi-mechanism integrated Crested Porcupine Optimizer (SDHCPO) that enhances the original CPO by incorporating four innovative strategies, among which the Sobol-OBL initialization and cosine-annealing-based dynamic adjustment are core components introduced for the first time. The Sobol-OBL initialization combines the low-discrepancy Sobol sequence with opposition-based learning to produce an initial population that is uniformly distributed across the solution space, effectively alleviating population clustering and reducing unexplored regions caused by purely random initialization, thereby laying a solid foundation for global search. The cosine-annealing-based dynamic adjustment strategy replaces random weights with a time-dependent nonlinear decay factor, substantially improving the stability of position updates in the fourth defense phase and enhancing the consistency of convergence. On this basis, the integration of Differential Evolution and the Horizontal-Vertical Crossover strategy further breaks positional dependence and eliminates dimensional stagnation, thereby jointly strengthening the algorithm's exploration capability and exploitation accuracy.

In the numerical experiments, SDHCPO is first compared with seven representative metaheuristic algorithms on the CEC2017 and CEC2022 benchmark functions. The results show that SDHCPO attains markedly lower mean fitness values on most test functions and achieves the best overall Friedman ranking. Moreover, the Wilcoxon rank-sum tests with SDHCPO as the reference method indicate that, for the vast majority of functions, the significance comparisons consistently support the statistical superiority of SDHCPO over the competing algorithms. Further ablation studies show that each of the four strategies yields varying degrees of performance improvement when activated individually, while their fully integrated configuration produces a pronounced synergistic effect, with particularly substantial reductions in the mean objective values on high-dimensional multimodal, hybrid, and composition functions. Together with qualitative analyses of population diversity evolution, fitness history, and one-dimensional search trajectories, the results show that SDHCPO maintains strong global exploration capability in the early iterations and then achieves a smooth transition to fine-grained local exploitation in the middle and late stages. This provides a mechanistic explanation for its convergence behavior across different function classes and clarifies the sources of its performance advantages. High-dimensional extension experiments further demonstrate that, on the 50-dimensional CEC2017 tests, SDHCPO achieves order-of-magnitude reductions in mean fitness compared with the original CPO and, even when the improvement in standard deviation is relatively modest, still maintains stable global optimization capability under this more demanding dimensional setting.

In terms of engineering applications, SDHCPO is validated on several classical engineering case studies, including traditional low-dimensional problems such as welded beam design and pressure vessel design, as well as a high-dimensional frequency-constrained optimization problem for a 72-bar spatial truss. The numerical results show that SDHCPO generally attains better objective values and smaller variability across these engineering examples. In particular, for large-scale, highly constrained structural optimization problems such as the 72-bar truss, SDHCPO still maintains high-quality convergence and strong solution robustness.

Despite the strong competitiveness of SDHCPO on both benchmark tests and engineering applications, several aspects remain open for improvement. For example, the coordination between exploration and exploitation on certain high-dimensional composition functions still has room for optimization, and the parameter sensitivity and computational overhead of some integrated strategies on specific problem types merit further investigation. Future work will focus on complex urban traffic signal timing as a key application scenario to further extend and validate SDHCPO. In multi-intersection coordinated control problems, a unified optimization model can be formulated by integrating multiple performance indicators—such as average delay, queue length, and emission levels—under traffic safety and signal control constraints, with SDHCPO serving as the core solver to systematically assess its convergence efficiency and robustness in large-scale road networks. Building on this foundation, future research will target multi-objective traffic control problems and integrate SDHCPO into decomposed or hierarchical control frameworks, thereby strengthening its optimization capability with respect to multiple dimensions such as traffic efficiency, equity, and environmental benefits. In this way, SDHCPO is expected to evolve into an engineering-feasible intelligent optimization tool for high-dimensional, strongly constrained, and multi-criteria-coupled urban traffic systems.

Data availability

This study is a benchmark based algorithm study and does not rely on any proprietary or externally collected dataset. All data generated or analyzed during this study are included in this published article. The implementation code and analysis scripts used to reproduce the reported tables, figures, and statistical analyses are available from the corresponding author upon reasonable request.

Conflict of interest

The authors declare no competing interests, whether financial or non-financial.

Funding

This research received no external funding.

References

1. Floudas, C.A., Akrotirianakis, I.G., Caratzoulas, S., Meyer, C.A., Kallrath, J.: Global optimization in the 21st century: Advances and challenges. Computers & Chemical Engineering (2005). https://doi.org/10.1016/j.compchemeng.2005.02.006
2. Mandal, P.K.: A review of classical methods and Nature-Inspired Algorithms (NIAs) for optimization problems. Results in Control and Optimization (2023). https://doi.org/10.1016/j.rico.2023.100315
3. Abualigah, L., Elaziz, M.A., Khasawneh, A.M., Alshinwan, M., Ibrahim, R.A., Al-Qaness, M.A., Gandomi, A.H.: Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: a comprehensive survey, applications, comparative analysis, and results. Neural Computing and Applications (2022). https://doi.org/10.1007/s00521-021-06747-4
4. Zhang, Z., Li, X., Gao, L., Liu, Q., Huang, J.: Tackling dual-resource flexible job shop scheduling problem in the production line reconfiguration scenario: An efficient meta-heuristic with critical path-based neighborhood search. Advanced Engineering Informatics (2025). https://doi.org/10.1016/j.aei.2025.103282
5. Chopard, B., Tomassini, M.: Performance and limitations of metaheuristics. In: An introduction to metaheuristics for optimization, pp. 191-203. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-93073-2_11
6. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, Vol.4, pp.1942-1948. IEEE (1995). https://doi.org/10.1109/ICNN.1995.488968
7. Holland, J.H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.

MIT Press (1992)
8. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization (1997). https://doi.org/10.1023/A:1008202821328
9. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in Engineering Software (2014). https://doi.org/10.1016/j.advengsoft.2013.12.007
10. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Advances in Engineering Software (2016). https://doi.org/10.1016/j.advengsoft.2016.01.008
11. Yang, X.S.: Firefly algorithms for multimodal optimization. In: International Symposium on Stochastic Algorithms, pp.169-178. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04944-6_14
12. Rajwar, K., Deep, K., Das, S.: An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. Artificial Intelligence Review (2023). https://doi.org/10.1007/s10462-023-10470-y
13. Qiao, J., Wang, G., Yang, Z., Luo, X., Chen, J., Li, K., Liu, P.: A hybrid particle swarm optimization algorithm for solving engineering problem. Scientific Reports (2024). https://doi.org/10.1038/s41598-024-59034-2
14. Vie, A., Kleinnijenhuis, A.M., Farmer, D.J.: Qualities, challenges and future of genetic algorithms: a literature review. arXiv Preprint arXiv:2011.05277 (2020). https://doi.org/10.48550/arXiv.2011.05277
15. Qin, Y., Deng, L., Li, C., Zhang, L.: CIR-DE: A chaotic individual regeneration mechanism for solving the stagnation problem in differential evolution. Swarm and Evolutionary Computation (2024). https://doi.org/10.1016/j.swevo.2024.101718
16. Chen, S., Zheng, J.: A hybrid grey wolf optimizer for engineering design problems. Journal of Combinatorial Optimization (2024). https://doi.org/10.1007/s10878-024-01189-9
17. Deng, L., Liu, S.: Deficiencies of the whale optimization algorithm and its validation method. Expert Systems with Applications (2024). https://doi.org/10.1016/j.eswa.2023.121544
18. Fister, I., Fister Jr, I., Yang, X.S., Brest, J.: A comprehensive review of firefly algorithms. Swarm and Evolutionary Computation (2013). https://doi.org/10.1016/j.swevo.2013.06.001
19. Li, Y., Zhao, Y., Shang, Y., Liu, J.: An improved firefly algorithm with dynamic self-adaptive adjustment. PLOS ONE (2021). https://doi.org/10.1371/journal.pone.0255951
20. Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, Vol.3, pp.1875-1882. IEEE (1999). https://doi.org/10.1109/CEC.1999.785502
21. Liang, Y., Leung, K.S.: Genetic algorithm with adaptive elitist-population strategies for multimodal function optimization. Applied Soft Computing

(2011). https://doi.org/10.1016/j.asoc.2010.06.017

22. Zhang, Z., Gao, Y., Liu, Y., Zuo, W.: A hybrid biogeography-based optimization algorithm to solve high-dimensional optimization problems and real-world engineering problems. Applied Soft Computing (2023). https://doi.org/10.1016/j.asoc.2023.110514
23. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation (2002). https://doi.org/10.1109/4235.585893
24. Zhang, Z., Gao, Y.: Solving large-scale global optimization problems and engineering design problems using a novel biogeography-based optimization with Lévy and Brownian movements. International Journal of Machine Learning and Cybernetics (2023). https://doi.org/10.1007/s13042-022-01642-3
25. Abdel-Basset, M., Mohamed, R., Abouhawwash, M.: Crested Porcupine Optimizer: A new nature-inspired metaheuristic. Knowledge-Based Systems (2024). https://doi.org/10.1016/j.knosys.2023.111257
26. Liu, S., Jin, Z., Lin, H., Lu, H.: An improve crested porcupine algorithm for UAV delivery path planning in challenging environments. Scientific Reports (2024). https://doi.org/10.1038/s41598-024-71485-1
27. Liu, H., Zhou, R., Zhong, X., Yao, Y., Shan, W., Yuan, J., Wang, Z.: Multi-strategy enhanced crested porcupine optimizer: CAPCPO. Mathematics (2024). https://doi.org/10.3390/math12193080
28. Villaruz, J.A., Gerardo, B.D., Gamao, A.O., Medina, R.P.: Scouting Firefly Algorithm and its Performance on Global Optimization Problems. International Journal of Advanced Computer Science and Applications (2023). https://doi.org/10.14569/IJACSA.2023.0140350
29. Zhang, Z., Gao, Y., Guo, E.: A supercomputing method for large-scale optimization: a feedback biogeography-based optimization with steepest descent method. Journal of Supercomputing (2023). https://doi.org/10.1007/s11227-022-04644-8
30. Altay, O., Altay, E.V.: A novel chaotic transient search optimization algorithm for global optimization, real-world engineering problems and feature selection. PeerJ Computer Science (2023). https://doi.org/10.7717/peerj-cs.1526
31. Li, J., Bai, J., Wang, J.: High-Precision Trajectory-Tracking Control of Quadrotor UAVs Based on an Improved Crested Porcupine Optimiser Algorithm and Preset Performance Self-Disturbance Control. Drones (2025). https://doi.org/10.3390/drones9060420
32. Tizhoosh, H.R.: Opposition-based learning: a new scheme for machine intelligence. In: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vol.1, pp.695-701. IEEE (2005). https://doi.org/10.1109/CIMCA.2005.1631345
33. Das, S., Suganthan, P.N.: Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation (2010).

https://doi.org/10.1109/TEVC.2010.2059031

34. Meng, A.B., Chen, Y.C., Yin, H., Chen, S.Z.: Crisscross optimization algorithm and its application. Knowledge-Based Systems (2014). https://doi.org/10.1016/j.knosys.2014.05.004
35. Doerr, C.: Complexity theory for discrete black-box optimization heuristics. In: Theory of Evolutionary Computation: Recent Developments in Discrete Optimization, pp.133-212. Springer International Publishing, Cham (2019)
36. Wu, G., Mallipeddi, R., Suganthan, P.N.: 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)
37. Luo, W., Lin, X., Li, C., Yang, S., Shi, Y.: Benchmark functions for CEC 2022 competition on seeking multiple optima in dynamic environments. arXiv Preprint arXiv:2201.00523 (2022). https://doi.org/10.48550/arXiv.2201.00523
38. Jia, H., Wen, Q., Wang, Y., Mirjalili, S.: Catch fish optimization algorithm: a new human behavior algorithm for solving clustering problems. Cluster Computing (2024). https://doi.org/10.1007/s10586-024-04618-w
39. Bouaouda, A., Hashim, F.A., Sayouti, Y., Hussien, A.G.: Pied kingfisher optimizer: a new bio-inspired algorithm for solving numerical optimization and industrial engineering problems. Neural Computing and Applications (2024). https://doi.org/10.1007/s00521-024-09879-5
40. Shehadeh, H.A.: Chernobyl disaster optimizer (CDO): a novel meta-heuristic method for global optimization. Neural Computing and Applications (2023). https://doi.org/10.1007/s00521-023-08261-1
41. Mirjalili, S., Mirjalili, S.M., Hatamlou, A.: Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications (2016). https://doi.org/10.1007/s00521-015-1870-7
42. Oladejo, S.O., Ekwe, S.O., Mirjalili, S.: The Hiking Optimization Algorithm: A novel human-based metaheuristic approach. Knowledge-Based Systems (2024). https://doi.org/10.1016/j.knosys.2024.111880
43. Wilcoxon, F.: Individual comparisons by ranking methods. In: Breakthroughs in statistics: Methodology and distribution, pp. 196-202. Springer, New York (1992). https://doi.org/10.1007/978-1-4612-4380-9_16
44. Friedman, M.: The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. Journal of the American Statistical Association (1937). https://doi.org/10.1080/01621459.1937.10503522
45. He, Q., Wang, L.: An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering Applications of Artificial Intelligence (2007). https://doi.org/10.1016/j.engappai.2006.03.003
46. Mohammed, H., Rashid, T.: A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design. Neural Computing and Applications (2020). https://doi.org/10.1007/s00521-020-

04823-9

47. Akdağ, O.: A modified tunicate swarm algorithm for engineering optimization problems. Arabian Journal for Science and Engineering (2023). https://doi.org/10.1007/s13369-023-07803-y
48. Qiu, Y., Yang, X., Chen, S.: An improved gray wolf optimization algorithm solving to functional optimization and engineering design problems. Scientific Reports (2024). https://doi.org/10.1038/s41598-024-64526-2
49. Mashinchi Joubari, M., Pashaei, M.H., Fathi, A.: Sizing optimization of truss structures under frequency constraints with artificial bee colony algorithm. Journal of Mathematics and Computer Science (2014). https://doi.org/10.22436/jmcs.09.02.01