Cluster Computing
https://doi.org/10.1007/s10586-024-04587-0

DEA^{2}H^2: differential evolution architecture based adaptive hyper-heuristic algorithm for continuous optimization

Rui Zhong[1] · Jun Yu[2]

Received: 14 March 2024 / Revised: 26 April 2024 / Accepted: 17 May 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract
This paper proposes a novel differential evolution (DE) architecture based hyper-heuristic algorithm (DEA^{2}H^2) for solving continuous optimization tasks. A representative hyper-heuristic algorithm consists of two main components: low-level and high-level components. In the low-level component, DEA^{2}H^2 leverages ten DE-derived search operators as low-level heuristics (LLHs). In the high-level component, we incorporate a success-history-based mechanism inspired by the success-history-based parameter adaptation in success-history adaptive DE (SHADE). Specifically, if a parent individual successfully evolves an offspring individual using a specific search operator, that corresponding operator is preserved for subsequent iterations. On the contrary, if the evolution is unsuccessful, the search operator is replaced by random initialization. To validate the effectiveness of DEA^{2}H^2, we conduct comprehensive numerical experiments on both CEC2020 and CEC2022 benchmark functions, as well as eight engineering problems. We compare the performance of DEA^{2}H^2 against fifteen well-known metaheuristic algorithms (MA). Additionally, ablation experiments are performed to investigate the effectiveness of the success-history-based high-level component independently. The experimental results and statistical analyses affirm the superiority and robustness of DEA^{2}H^2 across diverse optimization tasks, highlighting its potential as an effective tool for continuous optimization problems. The source code of this research can be downloaded from https://github.com/RuiZhong961230/DEA2H2.

Keywords Differential evolution (DE) · Hyper-heuristic (HH) · Success history knowledge · Continuous optimization

1 Introduction

The pursuit of optimal solutions to complex problems is a fundamental challenge across diverse domains, ranging from engineering and logistics to finance and artificial intelligence [1–4]. In this quest, traditional optimization techniques frequently encounter limitations due to the non-convex, high-dimensional, non-linear, and multi-modal nature of real-world problems [5–7]. The emergence of metaheuristic algorithms (MAs) has revolutionized the landscape of optimization by providing versatile and efficient approaches capable of navigating complex solution spaces.

With the rapid development of the MA community, hundreds and thousands of novel MA approaches are proposed each year. While this phenomenon significantly contributes to the prosperity of the MA community, scholars and researchers have raised concerns about the proliferation of MA approaches. Sörensen Kenneth [8] expressed reservations about drawing inspiration from various sources, such as insects, mammals, and natural phenomena, to launch new MA approaches. He argued that such inspiration might be too far-fetched and could potentially lead the MA community away from scientific rigor. Claus et al. [9] identified a concerning trend in the MA community, noting the proposal of numerous metaphor-based metaheuristics inspired by a wide array of natural, artificial, social, and sometimes supernatural

✉ Jun Yu
yujun@ie.niigata-u.ac.jp

Rui Zhong
rui.zhong.u5@elms.hokudai.ac.jp

[1] Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan

[2] Institute of Science and Technology, Niigata University, Niigata, Japan

Published online: 08 June 2024

phenomena. They raised the issue that these proposals often lack a clear motivation beyond the authors' desire to publish papers, potentially posing a detriment to the MA community. Villalón et al. [10] conducted a critical analysis of specific algorithms, including the grey wolf optimizer (GWO), firefly algorithm (FA), and bat algorithm (BA), within the context of particle swarm optimization (PSO). They concluded that these approaches were not novel but rather a reiteration of ideas initially introduced for particle swarm optimization. These ideas were later reintroduced with new natural metaphors. These concerns have also extended to other algorithms, such as harmony search (HS) [11] and the intelligent water drops algorithm (IWDA) [12], both of which have faced criticism for a perceived lack of originality.

In light of the concerns raised within the MA community, it is crucial to promote the healthy and sustainable advancement of the MA field. The emergence of the hyper-heuristic (HH) framework offers a promising avenue for advancing the MA community, providing an alternative approach that avoids the potential pitfalls associated with overstated metaphors [13]. In traditional MAs, operators such as crossover, mutation, and selection are typically predefined and applied in a fixed sequence. For example, in a conventional genetic algorithm (GA), crossover, mutation, and selection are activated iteratively according to a predefined sequence. Similarly, differential evolution (DE) iterates through mutation, crossover, and selection sequentially, while PSO comprises two fixed search operators: location update and velocity update. However, these expert-designed optimization sequences are also optimizable from the HH perspective [14]. The HH framework focuses on designing algorithms that generate or select low-level heuristics (LLHs) dynamically, rather than prescribing fixed sequences of heuristics. This high-level, automatic, and intelligent approach allows the HH framework to learn the experience from the past optimization and determine the subsequent selections during the optimization process [15].

In the past decade, the HH framework has captivated widespread attention, catalyzing the creation of numerous applications aimed at tackling complex real-world challenges. These applications traverse diverse domains, highlighting the remarkable adaptability of the HH framework. Within the realm of HH research, one of the most prominent areas of focus is around automatic configuration for optimization sequences-a concept synonymous with the self-adaptation mechanism in the high-level component. For example, Zhao et al. [16] proposed a cooperative multi-stage hyper-heuristic (CMS-HH) algorithm tailored for combinatorial optimization problems. The CMS-HH employs a genetic algorithm to perturb the initial solution, thereby enhancing solution diversity. During the search phase, an online learning mechanism based on multi-armed bandits and relay hybridization technology is employed to improve solution quality. Additionally, a multi-point search strategy collaborates with a single-point search operator when the solution state remains unchanged over continuous time. Zhang et al. [17] presented a deep reinforcement learning-based hyper-heuristic algorithm for combinatorial optimization problems with uncertainties. The algorithm features a data-driven heuristic selection module constructed using deep reinforcement learning techniques. This approach was validated in two combinatorial optimization scenarios: a real-world container terminal truck routing problem with uncertain service times and the widely studied online 2D strip packing problem. Furthermore, Qin et al. [18] devised a reinforcement learning-based hyper-heuristic algorithm for addressing a practical heterogeneous vehicle routing problem. This problem involves routing a predefined fleet with varying vehicle capacities to serve a sequence of customers while minimizing the maximum routing time of vehicles. The algorithm employs several meta-heuristics with diverse characteristics as LLHs, with policy-based reinforcement learning employed as the high-level selection strategy. While the combinatorial optimization domain has been a focal point for many works, there has been a relatively limited emphasis on the continuous optimization domain [19, 20]. Therefore, the motivation of this research is to propose an adaptive hyper-heuristic algorithm named DEA^2H^2, based on the DE architecture specifically for continuous optimization. In DEA^2H^2, the LLHs module incorporates representative search operators derived from DE. Inspired by the parameter adaptation strategies in success-history adaptive DE (SHADE) [21], the success history knowledge is fully utilized to dynamically adjust the optimization sequence, serving as the high-level component in the proposed DEA^2H^2 algorithm. The main contribution of this paper is as follows:

- We propose a DE-architecture-based adaptive hyper-heuristic algorithm named DEA^2H^2 for continuous optimization.
- DEA^2H^2 inherits the well-established search operators from DE as LLHs.
- Motivated by SHADE, we design a success-history-based mechanism to act as the high-level component of DEA^2H^2 to construct the optimization sequence automatically and intelligently.
- Comprehensive performance investigation on CEC2020 and CEC2022 benchmark functions and eight engineering problems including comparison experiments and ablation experiments are conducted to evaluate the performance of DEA^2H^2.

The remainder of this paper is organized as follows: Sect. 2 provides an overview of related works, including preliminaries on the HH framework, the fundamental components of DE, and the success history adaptation strategy in SHADE. Section 3 presents the details of our proposed DEA^{2}H^2. The numerical experiments and statistical analyses are discussed in Sect. 4. The performance of DEA^{2}H^2 is then analyzed in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Related works

2.1 Hyper-heuristic (HH) framework

Hyper-heuristic (HH) algorithm is an optimization technique that operates at a higher level, focusing on the selection or generation of LLHs to solve computational problems, rather than directly solving the problems themselves. In essence, the HH algorithm aims to find or generate sequences of heuristics that demonstrate superior robustness and scalability across a range of problems. Figure 1 illustrates a representative architecture of the HH framework.

A complete HH algorithm is comprised of two main components: the high-level component and the low-level component. The high-level component encompasses overarching strategies or mechanisms that guide the selection or generation of LLHs. It provides a methodology for decision-making and adaptation throughout the optimization process. On the other hand, the low-level component comprises the actual LLHs that are applied to problem instances to construct solutions. These LLHs serve as the fundamental building blocks used by the HH algorithm to explore and exploit the problem space.

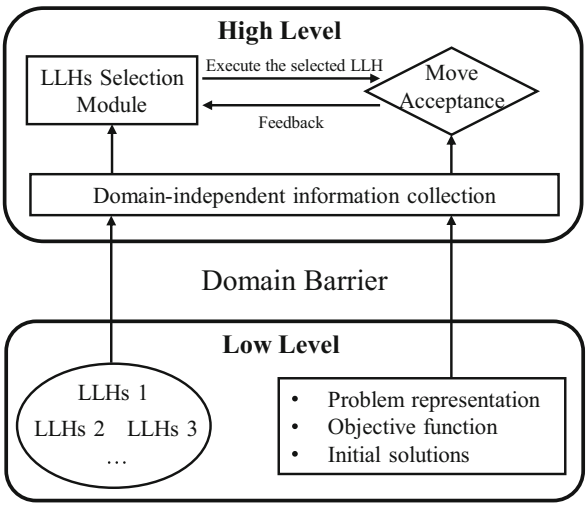


Fig. 1 A representative architecture of the HH framework [22]

2.2 Differential evolution (DE)

In this section, we present an introduction to the conventional DE algorithm, with a focus on its key components: population initialization, mutation, crossover, and selection. It's important to note that this explanation is in the context of minimization problems.

Population initialization: The initial and crucial step of DE is population initialization, as formulated in Eq. (1).

$$P = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ x_{31} & x_{32} & \cdots & x_{3M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NM} \end{bmatrix}$$
$$x_{ij} = r \cdot (UB_j - LB_j) + LB_j \tag{1}$$

where P denotes the population, X_i represents the i^{th} individual, and x_{ij} signifies the value of i^{th} individual in the j^{th} dimension. UB_j and LB_j correspond to the upper and lower bound of the j^{th} dimension, respectively, while r is a random value following a uniform distribution within the range (0,1).

Mutation operator: The name of DE is renowned for its mutation operator. The simplest *DE/cur/*1 scheme generates new candidate solutions by perturbing the current individual with a subtracting vector, as formulated in Eq. (2).

$$V_i = X_i + F \cdot (X_{r_1} - X_{r_2}) \tag{2}$$

where F is a scaling factor, r_1 and r_2 are randomly selected and mutually different integers from [1, N].

Crossover operator: The crossover operator combines information from the mutated individual with the corresponding parent individual and determines the components of the mutated individual that will be carried over to the subsequent iteration. The binomial crossover, involving the mutated individual V_i and parent individual X_i, is formulated in Eq. (3).

$$U_{i,j} = \begin{cases} V_{i,j}, & if\ r < Cr\ or\ j = jrand \\ X_{i,j}, & otherwise \end{cases} \tag{3}$$

where Cr denotes the crossover rate in (0, 1) and *jrand* is a random integer in [1, N].

Selection: DE adopts a one-to-one greedy selection strategy to determine the surviving individual, which is formulated in Eq. (4).

$$X_i = \begin{cases} U_i, & if\ f(U_i) < f(X_i) \\ X_i, & otherwise \end{cases} \tag{4}$$

2.3 Success history adaptation strategy in SHADE

Renowned for its efficacy in global optimization, DE operates by iteratively evolving a population of candidate solutions through the manipulation of solution vectors. However, the performance of DE is highly contingent upon the appropriate tuning of its control parameters (i.e., scaling parameter F and crossover rate Cr). This challenge often requires expert knowledge or extensive experimentation.

As one of the most well-established variants of DE, SHADE stands as a pioneering advancement within the realm of evolutionary computation, specifically in the domain of the parameter adaptation of DE. SHADE initializes the matrix of crossover rate $M_{Cr,i}$ and scaling factor $M_{F,i}$ ($i \in \{1, 2, ..., H\}$) with 0.5, where H is the length of the matrix. In a certain iteration, the specific index idx is randomly sampled from $\{1, 2, ..., H\}$. Then, the scaling factor F_j and crossover rate Cr_j for the j^{th} individual are sampled using Eq. (5).

$$\begin{aligned} F_j &= randn(M_{F,idx}, 0.1) \\ Cr_j &= randc(M_{Cr,idx}, 0.1) \end{aligned} \tag{5}$$

where $randn(\mu, \sigma^2)$ and $randc(\mu, \sigma^2)$ are values randomly generated from Gaussian and Cauchy distributions with expectation μ and variance σ^2. When F_j is larger than 1, it is truncated to 1, and if F_j is smaller than 0, it will resample through Eq. (5). Similarly, when Cr_j is out of [0, 1], Eq. (5) is re-activated to generate a new Cr_j.

After the evaluation, the scaling factor F_j and crossover rate Cr_j from successful evolution (i.e. the fitness of the offspring individual is better than the parent individual) are recorded into S_F and S_{Cr}. At the end, $M_{Cr,idx}$ and $M_{F,idx}$ are updated using Eq. (6).

$$\begin{aligned} M_{Cr,idx} &= \begin{cases} mean_{WA}(S_{Cr}), & if\ S_{Cr} \neq \emptyset \\ M_{Cr,idx}, & otherwise \end{cases} \\ M_{F,idx} &= \begin{cases} mean_{WL}(S_F), & if\ S_F \neq \emptyset \\ M_{F,idx}, & otherwise \end{cases} \end{aligned} \tag{6}$$

Here, $mean_{WA}(S_{Cr})$ and $mean_{WL}(S_F)$ can be calculated by Eq. (7).

$$\begin{aligned} mean_{WA}(S_{Cr}) &= \sum_{k=1}^{|S_{Cr}|} w_k \cdot S_{Cr,k} \\ mean_{WL}(S_F) &= \frac{\sum_{k=1}^{|S_F|} w_k \cdot S_{F,k}^2}{\sum_{k=1}^{|S_F|} w_k \cdot S_{F,k}} \\ w_k &= \frac{\Delta f_k}{\sum_{i=1}^{|S_{Cr}|} \Delta f_i} \end{aligned} \tag{7}$$

where $\Delta f_i = |f(X_i) - f(O_i)|$ represents the absolute difference between the parent individual X_i and the offspring individual O_i. The success-history-based adaptation scheme hypothesizes that the individual-specific scaling factor F_i and crossover rate Cr_i of the i^{th} individual with successful evolution are potentially beneficial for other individuals to construct superior offspring solutions. Therefore, this adaptive response to problem characteristics may effectively navigate complex solution spaces and converge toward high-quality solutions across a diverse range of optimization problems.

3 Our proposal: DEA^{2}H^2

This section begins by illustrating the flowchart of our proposed DEA^{2}H^2 in Fig. 2. Following the architecture shared by most HH approaches, DEA^{2}H^2 initializes parameters and the population. Additionally, it constructs a unique success-history memory to store the determined LLHs. During the internal iteration, DEA^{2}H^2 samples the corresponding LLH for the i^{th} individual based on the success-history memory of LLHs. It then utilizes this LLH to generate an offspring individual. If the constructed offspring has a better fitness value, it replaces the parent individual and survives; otherwise, this offspring individual is discarded, and the success-history memory of LLHs is updated. These procedures continue until the computational budget is exhausted.

In the subsequent sections, we will delve into the detailed design of LLHs and the high-level component of DEA^{2}H^2.

3.1 Low-level heuristics

In the HH framework, low-level heuristics (LLHs) play a fundamental role in the optimization process. LLHs, also known as search operators, are elementary problem-solving procedures that manipulate solutions within a search space. Unlike domain-specific heuristics tailored to particular optimization problems, LLHs are designed to be generic and applicable across various problem domains. In this paper, three categories of easy-implemented and efficient search operators are employed as LLHs: local search operator, differential mutation, and crossover.

Local search with uniform distribution: Local search is a fundamental optimization technique that focuses on refining a given solution by slight perturbation to move to neighboring solutions that are locally better in terms of the objective function. Equation (8) presents the basic local search operator in the continuous search domain.

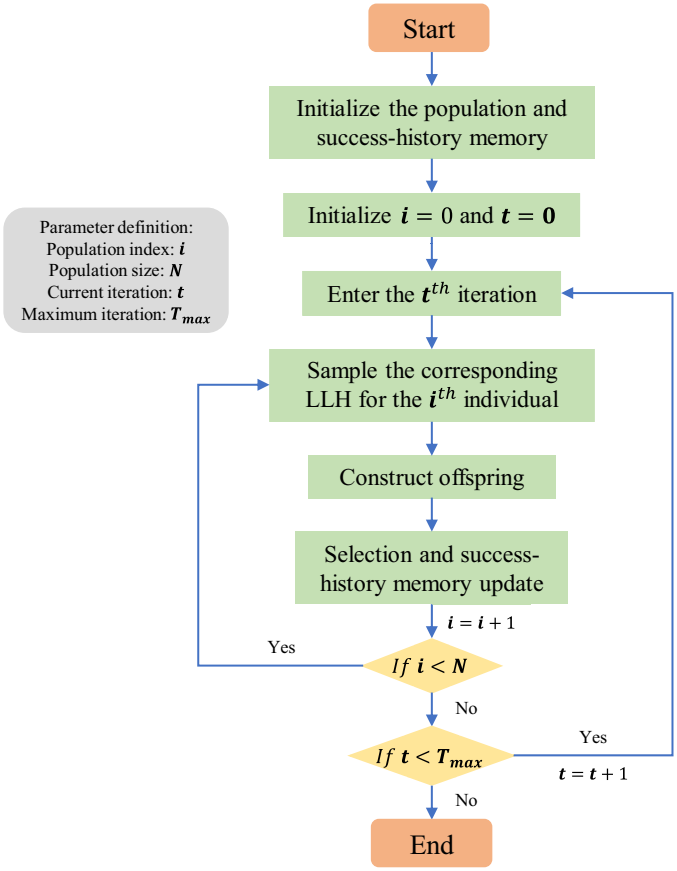


Fig. 2 The flowchart of DEA^{2}H^2

$$X_{i,j}^{new} = X_{i,j} + \delta_j \delta_j \sim U(-R, R) \tag{8}$$

where δ represents the perturbation vector. R is the parameter of the search radius.

Local search with Gaussian distribution: This operator utilizes the Gaussian distribution random value generator to construct the perturbation vector δ.

$$X_{i,j}^{new} = X_{i,j} + \delta_j \delta_j \sim R \cdot N(0, 1) \tag{9}$$

$N(0, 1)$ represents the random value following the standard Gaussian distribution.

Lévy flight: Lévy flight describes a type of random movement characterized by occasional long jumps or flights interspersed with shorter, more localized steps. Equation (10) formulates the Lévy flight with the mathematical model.

$$X_{i,j}^{new} = X_{i,j} + \delta_j$$
$$\delta_j \sim \frac{u}{|v|^{\frac{1}{\beta}}} \tag{10}$$

where β is the Lévy distribution index bounded as $0 < \beta \leq 2$, while u and v are such that

$$u \sim N(0, \sigma^2), u \sim N(0, 1) \tag{11}$$

and the standard deviation σ is defined by

$$\sigma = \left\{ \frac{\Gamma(1+\beta) \sin\left(\frac{\pi\beta}{2}\right)}{\beta \Gamma\left(\frac{1+\beta}{2}\right) 2^{\frac{\beta-1}{2}}} \right\}^{\frac{1}{\beta}} \tag{12}$$

the gamma function Γ for an integer z can be formulated as

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt \tag{13}$$

Differential mutation: The differential mutation operation is responsible for generating new candidate solutions by perturbing existing solutions in the population. It is one of the primary mechanisms through which DE explores the search space and evolves toward better solutions. Therefore, DEA^{2}H^2 inherits the advantages of DE and employs three common schemes, as expressed in Eq. (14).

$$DE/rand/1 : X_i^{new} = X_{r_1} + r \cdot (X_{r_2} - X_{r_3})$$
$$DE/best/1 : X_i^{new} = X_{best} + r \cdot (X_{r_1} - X_{r_2})$$
$$DE/cur2best/1 : X_i^{new} = X_i + r \cdot (X_{best} - X_i) + r \cdot (X_{r_1} - X_{r_2}) \tag{14}$$

To increase the randomness in differential mutation, the scaling factor is used in place of the random value r, which follows the uniform distribution in $U(0, 1)$.

Crossover operator: The crossover operator is an important component that facilitates the exchange of information between candidate solutions in the population. Working in tandem with the differential mutation operator, the crossover operator plays a crucial role in generating new trial solutions and driving the search process toward better solutions. The basic idea behind the crossover operator in DE is to probabilistically select components from either the mutant vector or the target vector to form the trial solution. This is typically done using a crossover probability parameter, which determines the likelihood of selecting components from the mutant vector or the target vector. In this study, the binomial crossover and exponential crossover are employed, which are explained as follows:

The mathematical model of binomial crossover can be found in Eq. (3). However, the crossover operator, along with the local search and differential mutation, are utilized independently in the context of the HH framework. Therefore, the mutated vector V_i in Eq. (3) is replaced by the current best individual X_{best} and random individual X_r, developing the binomial crossover with the current best and binomial crossover with the random individual.

The exponential crossover focuses on inheriting sequential genes in the offspring individual. A visual demonstration is presented in Fig. 3.

For a pair of mutated vector V_i and parent individual X_i, the exponential crossover first locates a random position denoted as the "start" in Fig. 3. Then, the subsequent genes will inherit the value from the mutated vector V_i if the independently random value *rand*() is smaller than the crossover rate; otherwise, the process of the exponential crossover is terminated, and the values of the remaining genes are duplicated from the parent individual X_i. Similarly, the exponential crossover with the current best and the exponential crossover with the random individual are included in DEA^{2}H^2.

3.2 High-level component

The original SHADE algorithm hypothesizes that the scaling factor F and crossover rate Cr obtained from individuals with successful evolution carry navigation significance. As the search progresses, the mean values of Cr (μ_{Cr}) and F (μ_F) gradually approach optimal values for the given problem. Inspired by the success-history knowledge utilization in SHADE, we hypothesize that the LLHs obtained from individuals with successful evolution are suitable for optimizing the specific problem, and this success-history information is employed in DEA^{2}H^2 to act as the high-level component. A demonstration of the high-level component mechanism is visualized in Fig. 4.

Fig. 4a presents a summary of the ten LLHs involved in the DEA^{2}H^2 algorithm. Initially, the success-history memory of the LLHs matrix (M_{LLHs}) is randomly sampled from the LLHs pool. Figure 4 (b) illustrates a successful evolution for the 2^{nd} individual. Initially, DEA^{2}H^2 selects the corresponding LLH from the success-history memory, which, in this case, is the local search with Lévy flight. Then, a better offspring individual is constructed, leading to the maintenance of the success-history memory. Conversely, Fig. 4c depicts a failed evolution. In such a scenario, the corresponding LLH is re-initialized with a random sample.

In summary, the pseudocode of DEA^{2}H^2 is presented in Algorithm 1.

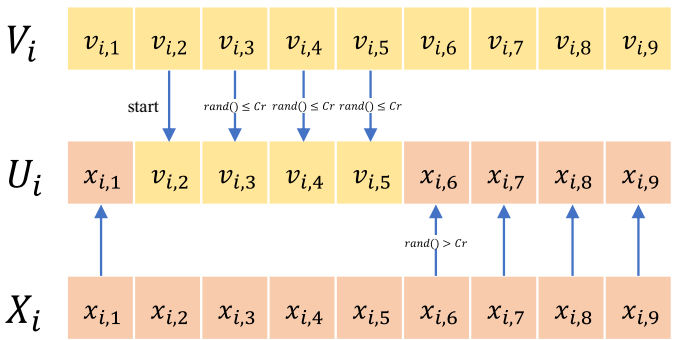


Fig. 3 A demonstration of exponential crossover [23]

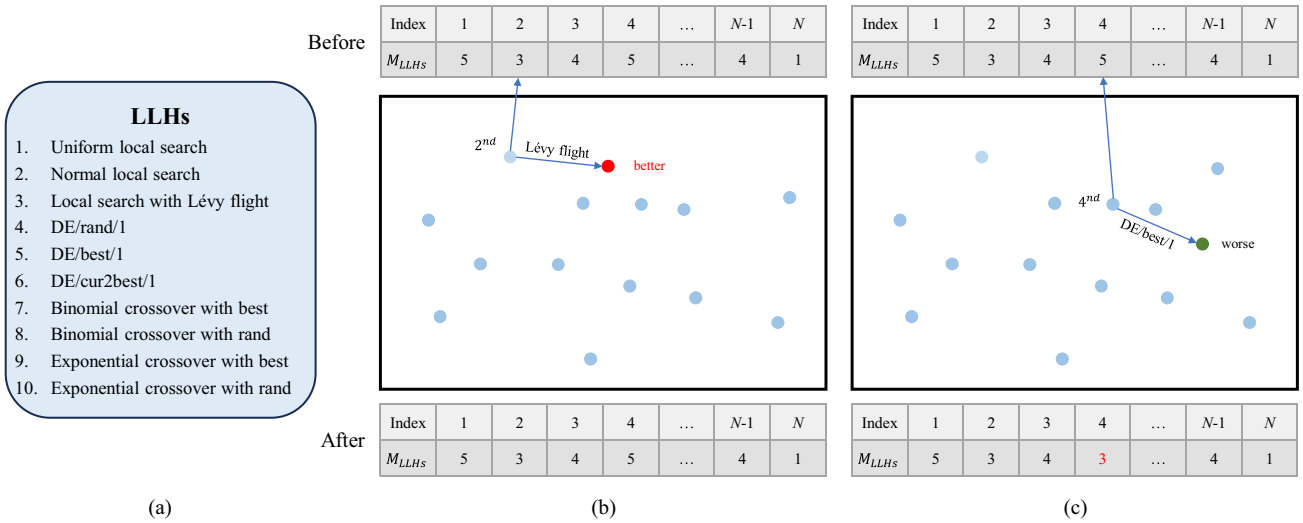


Fig. 4 A demonstration of the high-level component mechanism

Algorithm 1 DEA^{2}H^2

Require: Population size: N, Dimension: D, Maximum iteration: T_{max},
Ensure: Optimum: X_{best}

Initialize the population, parameters, and the LLHs pool
$X_{best} \leftarrow \textbf{best}(P)$
$t \leftarrow 0$
while $t < T$ **do**
  for $i = 0$ *to* N **do**
    Sample the i^{th} LLH from success-history memory
    Construct the offspring with the corresponding LLH
    if Offspring has a better fitness value **then**
      Replace the parent individual
    else
      Update success-history memory with re-initialization
    end if
  end for
  $X_{best} \leftarrow \textbf{best}(P)$
  $t \leftarrow t + 1$
end while
return X_{best}

4 Numerical experiments

This section provides a comprehensive overview of the detailed experiment settings and corresponding results. Section 4.1 focuses on delineating the specific details of the experiment settings, including experimental environments, benchmark functions, and competitor algorithms along with their parameters. Section 4.2 presents the experimental results and provides thorough statistical analyses.

4.1 Experimental settings

4.1.1 Experimental environments and implementation

We conducted numerical experiments using Python 3.11 programming language. The experiments were executed on a Lenovo Legion R9000P laptop running Windows 11. The system is equipped with an AMD Ryzen 7 5800 H processor with Radeon Graphics clocked at 3.20 GHz and 16GB of RAM.

4.1.2 Benchmark functions

We adopted three well-established benchmarks to evaluate our proposed DEA^{2}H^2 comprehensively, which are summarized as follows:

- CEC2020 benchmark functions are listed in Table 1 and provided by OpFuNu library [24].
- CEC2022 benchmark functions are listed in Table 2 and provided by OpFuNu library [24].
- Eight engineering problems are described as follows and made available using the ENOPPY library [25]. We only provide mathematical models and demonstrations. More detailed information can be found in [26].

Cantilever beam design (CBD): The mathematical model of CBD is formulated in Eq. (15), while the visualization is provided in Fig. 5.

$$\begin{aligned} &\textbf{min}\ f(X) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5) \\ &\textbf{s.t.}\ g(X) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0 \\ &\textbf{where}\ 0.01 \le x_i \le 100,\ i \in \{1,2,3,4,5\} \end{aligned} \tag{15}$$

Corrugated bulkhead design (CBHD): The mathematical model of the CBHD is presented in Eq. (16).

$$\begin{aligned} &\textbf{min}\ f(X) = \frac{5.885 x_4 (x_1 + x_3)}{x_1 + \sqrt{|x_3^2 - x_2^2|}} \\ &\textbf{s.t.}\ g_1(X) = -x_4 x_2 \left(0.4 x_1 + \frac{x_3}{6}\right) \\ &\qquad + 8.94\left(x_1 + \sqrt{|x_3^2 - x_2^2|}\right) \le 0 \\ &\qquad g_2(X) = -x_4 x_2^2 \left(0.2 x_1 + \frac{x_3}{12}\right) + 2.2\left(8.94\left(x_1 + \sqrt{|x_3^2 - x_2^2|}\right)\right)^{4/3} \le 0 \\ &\qquad g_3(X) = -x_4 + 0.0156 x_1 + 0.15 \le 0 \\ &\qquad g_4(X) = -x_4 + 0.0156 x_3 + 0.15 \le 0 \\ &\qquad g_5(X) = -x_4 + 1.05 \le 0 \\ &\qquad g_6(X) = -x_3 + x_2 \le 0 \\ &\textbf{where}\ 0 \le x_1, x_2, x_3 \le 100 \\ &\qquad 0 \le x_4 \le 5 \end{aligned} \tag{16}$$

Compression/tension spring problem (CSP): Eq. (17) describes the mathematical model of CSP, and the visualization is in Fig. 6.

Table 1 Summary of the CEC2020 benchmark functions: Uni.=Unimodal function, Multi.=Multimodal function, Hybrid.=Hybrid function, Comp.=Composition function

No	Func	Feature	Optimum
f_1	Shifted and Rotated Bent Cigar Function	Uni	100
f_2	Shifted and Rotated Schwefel's function	Multi.	1100
f_3	Shifted and Rotated Lunacek bi-Rastrigin function		700
f_4	Expanded Rosenbrock's plus Griewangk's function		1900
f_5	Hybrid function 1 (N = 3)	Hybrid.	1700
f_6	Hybrid function 2 (N = 4)		1600
f_7	Hybrid function 3 (N = 5)		2100
f_8	Composition function 1 (N = 3)	Comp.	2200
f_9	Composition function 2 (N = 4)		2400
f_{10}	Composition function 3 (N = 5)		2500
Search range: $[-100, 100]^D$			

Table 2 Summary of the CEC2022 benchmark functions: Uni.=Unimodal function, Basic.=Basic function, Hybrid.=Hybrid function, Comp.=Composition function

Func	Description	Feature	Optimum
f_1	Shifted and full Rotated Zakharov	Uni	300
f_2	Shifted and full Rotated Rosenbrock	Basic.	400
f_3	Shifted and full Rotated Expanded Schaffer f_6		600
f_4	Shifted and full Rotated Non-Continuous Rastrigin		800
f_5	Shifted and full Rotated Levy		900
f_6	Hybrid function 1 (N = 3)	Hybrid.	1800
f_7	Hybrid function 2 (N = 6)		2000
f_8	Hybrid function 3 (N = 5)		2200
f_9	Composition function 1 (N = 5)	Comp.	2300
f_{10}	Composition function 2 (N = 4)		2400
f_{11}	Composition function 3 (N = 5)		2600
f_{12}	Composition function 3 (N = 6)		2700
Search range: $[-100, 100]^D$			

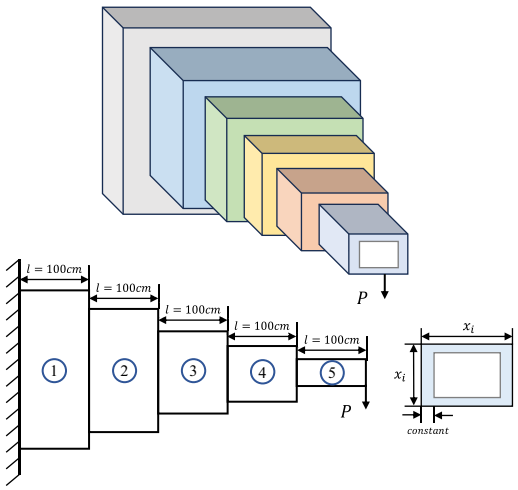


Fig. 5 The demonstration of the cantilever beam design problem

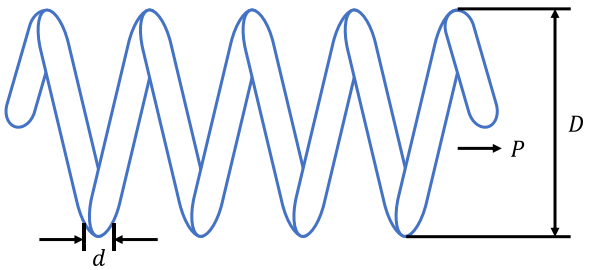


Fig. 6 The demonstration of the compression/tension spring design problem

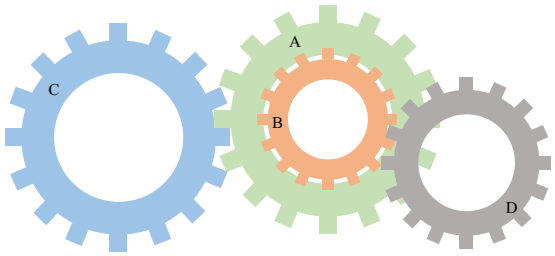


Fig. 7 The demonstration of the gear train design problem

$$
\begin{aligned}
&\text{minimize}\\
&f(X) = (x_3 + 2)x_2x_1^2\\
&\text{subject to}\\
&g_1(X) = 1 - \frac{x_2^3x_3}{71785x_1^4} \le 0\\
&g_2(X) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0\\
&g_3(X) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0\\
&g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \le 0\\
&\text{where}\\
&0.05 \le x_1 \le 2\\
&0.25 \le x_2 \le 1.3\\
&2 \le x_3 \le 15
\end{aligned}
\tag{17}
$$

Gear train design (GTD): Eq. (18) presents the mathematical model while Fig. 7 visualizes this problem.

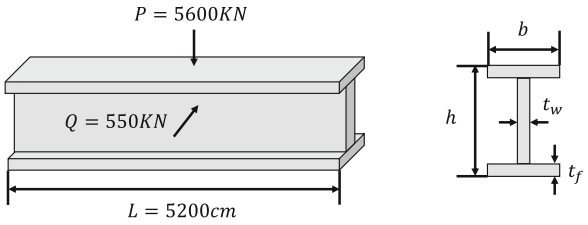


Fig. 8 The demonstration of the I-beam design problem

$$\min f(X) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4}\right)^2 \tag{18}$$

$$\text{where} x_1, x_2, x_3, x_4 \in \{12, 13, 14, ..., 60\}$$

I-beam design (IBD): The mathematical model of IBD is expressed using Eq. (19), and Fig. 8 visualizes the demonstration.

$$\begin{aligned}
\min f(X) =& \frac{5000}{x_3(x_2 - 2x_4)^3/12 + (x_1 x_4^3/6) + 2x_1 x_4 (x_2 - x_4/2)^2} \\
\text{s.t. } g_1(X) =& 2x_1 x_3 + x_3(x_2 - 2x_4) - 300 \le 0 \\
g_2(X) =& \frac{180000 x_2}{x_3(x_2 - 2x_4)^3 + 2x_1 x_3 (4x_4^2 + 3x_2(x_2 - 2x_4))} \\
&+ \frac{15000 x_1}{(x_2 - 2x_4)x_3^2 + 2x_3 x_1^3} - 56 \le 0 \\
\text{where } & 10 \le x_1 \le 50 \\
& 10 \le x_2 \le 80 \\
& 0.9 \le x_3, x_4 \le 5
\end{aligned} \tag{19}$$

Pressure vessel problem (PVP): The mathematical model and the visualized demonstration of PVP are demonstrated in Eq. (20) and Fig. 9, respectively.

$$\begin{aligned}
\min f(X) =& 0.6224 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3 \\
\text{s.t. } g_1(X) =& -x_1 + 0.0193 x_3 \le 0 \\
g_2(X) =& -x_2 + 0.00954 x_3 \le 0 \\
g_3(X) =& -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \le 0 \\
g_4(X) =& x_4 - 240 \le 0 \\
\text{where } & 0 \le x_1, x_2 \le 99 \\
& 10 \le x_3, x_4 \le 200
\end{aligned} \tag{20}$$

Speed reducer design (SRD): The mathematical model of the SRD is formulated in Eq. (21).

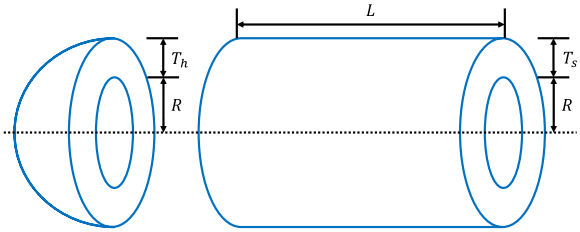


Fig. 9 The demonstration of the pressure vessel problem

$$\begin{aligned}
\min f(X) =& 0.7854 x_1 x_2^2 (3.3333 x_3^2 + 14.9334 x_3 - 43.0924) \\
&- 1.508 x_1 (x_6^2 + x_7^2) + \\
&7.4777 (x_6^3 + x_7^3) + 0.7854 (x_4 x_6^2 + x_5 x_7^2) \\
\text{s.t. } g_1(X) =& \frac{27}{x_1 x_2^2 x_3} - 1 \le 0 \\
g_2(X) =& \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0 \\
g_3(X) =& \frac{1.93 x_4^3}{x_2 x_6^4 x_3} - 1 \le 0 \\
g_4(X) =& \frac{1.93 x_5^3}{x_2 x_7^4 x_3} - 1 \le 0 \\
g_5(X) =& \frac{\sqrt{(745 x_4 / x_2 x_3)^2 + 16900000}}{110 x_6^3} - 1 \le 0 \\
g_6(X) =& \frac{\sqrt{(745 x_5 / x_2 x_3)^2 + 157500000}}{85 x_7^3} - 1 \le 0 \\
g_7(X) =& \frac{x_2 x_3}{40} - 1 \le 0 \\
g_8(X) =& \frac{5 x_2}{x_1} - 1 \le 0 \\
g_9(X) =& \frac{x_1}{12 x_2} - 1 \le 0 \\
g_{10}(X) =& \frac{1.5 x_6 + 1.9}{x_4} - 1 \le 0 \\
g_{11}(X) =& \frac{1.1 x_7 + 1.9}{x_5} - 1 \le 0 \\
\text{where } & 2.6 \le x_1 \le 2.6 \\
& 0.7 \le x_2 \le 0.8 \\
& 17 \le x_3 \le 28 \\
& 7.3 \le x_4 x_5 \le 8.3 \\
& 2.9 \le x_6 \le 3.9 \\
& 5 \le x_7 \le 5.5
\end{aligned} \tag{21}$$

Tubular column design (TCD): The mathematical model and demonstration of TCD are presented in Eq. (22) and Fig. 10, respectively.

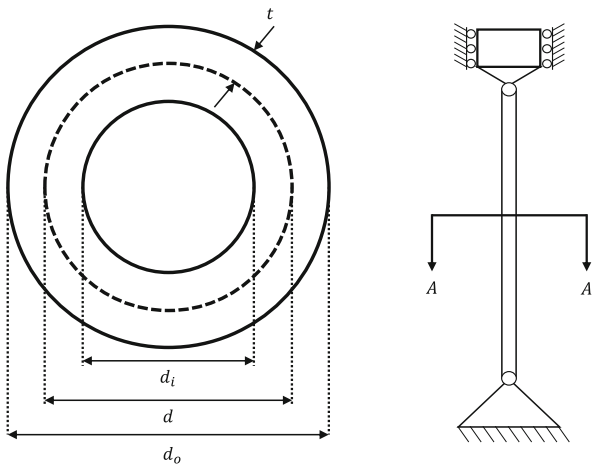


Fig. 10 The demonstration of the tubular column design problem

$$\begin{aligned}
&\textbf{min}\ f(X) = 9.8x_1x_2 + 2x_1 \\
&\textbf{s.t.}\ g_1(X) = \frac{P}{\pi x_1 x_2 \sigma_y} - 1 \le 0 \\
&\qquad g_2(X) = \frac{8PL^2}{\pi^3 E x_1 x_2 (x_1^2 + x_2^2)} - 1 \le 0 \\
&\textbf{where}\ 2 \le x_1 \le 14 \\
&\qquad 0.2 \le x_2 \le 0.8
\end{aligned} \tag{22}$$

4.1.3 Competitor algorithms and parameters

Fifteen well-known and state-of-the-art MA approaches are employed as competitor algorithms in this study to evaluate the performance of DEA^{2}H^2. They are listed as follows:

- Classic MAs: genetic algorithm (GA) [27], PSO [28], DE [29], and covariance matrix adaptation evolution strategy (CMA-ES) [30].
- DE-based MAs: JADE [31], SHADE [21], and SHADE with linear population size reduction (L-SHADE) [32].
- Highly-cited MAs: cuckoo search (CS) [33], grey wolf optimizer (GWO) [34], sine cosine algorithm (SCA) [35], and whale optimization algorithm (WOA) [36].
- Latest MAs: golden jackal optimization (GJO) [37], Fick's law algorithm (FLA) [38], weIghted meaN oF vectOrs (INFO) [39], and rime-ice optimizer (RIME) [40].

The population size for all algorithms is set to 100 individuals. To ensure a comprehensive evaluation, the maximum number of fitness evaluations (FEs) varies depending on the benchmark functions and engineering problems. For CEC2020 and CEC2022 benchmark functions, the maximum FEs are fixed at 1000 $\cdot D$ (D denotes the dimension size), while for the eight engineering problems, it is set to 10,000 FEs. To account for the stochastic nature of MAs, each algorithm is executed with 30 independent trial runs. This helps mitigate the impact of randomness and provides a more robust assessment of their performance. Detailed parameter settings for the competitor algorithms are presented in Table 3, aligning with the recommendations provided in the respective literature. These parameters are carefully chosen to ensure fair comparisons and reproducibility of results across different algorithms and problem instances.

Additionally, the primary MA approaches and proposed DEA^{2}H^2 cannot deal with constrained optimization problems directly. Thus, we equip them with the static penalty function, as defined by Eq. (23).

$$F(R_i) = f(R_i) + w \cdot \sum_{i=1}^{m} (\max(0, g_i(R_i))) \tag{23}$$

$F(\cdot)$ is the fitness function, $f(\cdot)$ is the objective function, and $g_i(\cdot)$ is the constraint function. w is a constant set to $10e7$ by default.

4.2 Experimental results and statistical analyses

This section presents the experimental results and statistical analyses on CEC2020, CEC2022, and engineering optimization problems, respectively. Additionally, the ablation experiments are conducted to investigate the effectiveness of the success-history-based high-level component. For any pair of compared algorithms, the Mann–Whitney U test is employed to determine the significance. If the significance exists, we apply the Holm multiple comparison test [41] to correct the p-value obtained from the Mann–Whitney U test. Marks $+$, $\approx$, and $-$ denote that our proposed DEA^{2}H^2 is significantly better, has no significance, or is significantly worse than the specific competitor algorithm. The average rank is computed to demonstrate the scalability of algorithms in diverse benchmark functions, and the fitness value of the best-performing algorithm is denoted in bold.

4.2.1 Performance on CEC2020 benchmark functions

Tables 4, 5, 6, and 7 summarize the experimental and statistical analyses on 10-D, 30-D, 50-D, and 100-D CEC2020 benchmark functions. Figures 11 and 12 visualize the corresponding convergence curves.

4.2.2 Performance on CEC2022 benchmark functions

Tables 8 and 9 summarize the experimental results and statistical analyses on 10-D and 20-D CEC2022 benchmark

Table 3 The parameter setting of competitor algorithms

MAs	Parameters	Value
GA	Crossover probability *pc*	0.95
	Mutation probability *pm*	0.025
	Selection	tournament
PSO	Inertia factor *w*	1
	Acceleration coefficients c_1 and c_2	2.05
	Max. and min. speed	2, -2
DE	Mutation scheme	DE/cur-to-rand/1
	Scaling factor *F*	0.8
	Crossover rate *Cr*	0.9
CMA-ES	parameter-free	
JADE	Mutation scheme	DE/cur-to-pbest/1
	Mean of scaling factor *F*	0.5
	Mean of crossover rate *Cr*	0.5
SHADE	Mean of scaling factor *F*	0.5
	Mean of crossover rate *Cr*	0.5
L-SHADE	Mean of scaling factor *F*	0.5
	Mean of crossover rate *Cr*	0.5
CS	Probability p_a	0.25
GWO	parameter-free	
SCA	Constant *A*	2
WOA	Constant *b*	1
GJO	parameter-free	
FLA	Factors C_1, C_2, C_3, C_4, and C_5	0.5, 2, 0.1, 0.2, and 5
	Factor *D*	0.01
INFO	parameter-free	
RIME	Parameter *w*	5
DEA^2H^2	Search radius *R*	1

functions. Convergence curves are demonstrated in Fig. 13.

4.2.3 Performance on engineering problems

Table 10 summarizes the experimental results and statistical analyses on engineering optimization problems. Given that stability is a crucial performance indicator in real-world scenarios, the best and the worst optimum found in 30 independent trial runs are also provided. Figure 14 presents the corresponding convergence curves.

In summary, the average ranks of optimizers in various benchmarks are visualized in Fig. 15.

4.2.4 Ablation experiments on CEC2020 benchmark functions

To assess the effectiveness of our proposed success-history-based high-level component, we replaced the high-level component of DEA^2H^2 with learning-free random selection [42] and created DEA^2H^2-rand. In DEA2H2-rand, each LLH has an equal probability of being chosen and activated. The experimental results and statistical analyses are presented in Table 11.

5 Discussion

This section discusses the performance of DEA^2H^2. Section 5.1 analyzes the computational complexity of DEA^2H^2 theoretically, Sect. 5.2 presents the advantages of DEA^2H^2 in both LLHs and high-level component design. Finally, Sects. 5.3, 5.4, and 5.5 analyze the performance of DEA^2H^2 on CEC2020, CEC2022, and engineering problems.

5.1 Computational complexity analysis

Considering the population size of *N*, the dimension size of *D*, and the maximum iteration of *T*, the computational complexity of population initialization in DEA^2H^2 is

Table 4 Experimental results and statistical analyses on 10-D CEC2020 benchmark functions

Func.		GA	PSO	DE	CMA-ES	JADE	SHADE	L-SHADE
f_1	Mean	2.99e+06 +	4.26e+08 +	7.01e+08 +	2.54e+07 +	3.46e+07 +	1.39e+07 +	1.65e+07 +
	SD	2.34e+06	4.50e+08	2.15e+08	9.87e+06	2.46e+07	4.67e+06	7.03e+06
f_2	Mean	2.79e+08 +	9.12e+09 +	6.30e+10 +	2.40e+09 +	8.99e+08 +	5.80e+08 +	5.93e+08 +
	SD	1.97e+08	8.66e+09	1.85e+10	8.04e+08	3.70e+08	2.16e+08	2.03e+08
f_3	Mean	9.81e+07 +	1.42e+09 +	3.45e+10 +	1.22e+09 +	4.08e+08 +	3.77e+08 +	4.12e+08 +
	SD	6.84e+07	3.17e+09	9.56e+09	4.22e+08	1.77e+08	1.30e+08	1.42e+08
f_4	Mean	1.90e+03 +	1.91e+03 +	1.93e+03 +	1.91e+03 +	1.90e+03 +	1.90e+03 +	1.90e+03 +
	SD	1.26e+00	4.02e+00	1.98e+01	8.96e−01	7.16e−01	6.54e−01	4.95e−01
f_5	Mean	1.10e+04 +	1.56e+05 +	1.27e+04 +	**2.10e+03** ≈	8.62e+03 +	6.96e+03 +	6.57e+03 +
	SD	1.03e+04	1.99e+05	2.27e+03	1.08e+02	2.44e+03	1.81e+03	2.06e+03
f_6	Mean	8.90e+03 +	6.83e+03 +	2.85e+03 +	1.95e+03 +	2.62e+03 +	2.32e+03 +	2.52e+03 +
	SD	7.69e+03	4.70e+03	3.66e+02	1.44e+02	3.73e+02	2.85e+02	3.75e+02
f_7	Mean	1.31e+04 +	2.42e+05 +	6.24e+04 +	6.44e+03 +	1.06e+04 +	1.18e+04 +	1.18e+04 +
	SD	8.53e+03	3.72e+05	2.87e+04	1.58e+03	3.45e+03	5.22e+03	4.66e+03
f_8	Mean	2.31e+03 +	2.32e+03 +	2.32e+03 +	2.31e+03 +	2.31e+03 +	2.31e+03 +	2.31e+03 +
	SD	1.75e+00	3.20e+00	1.47e+00	1.14e+00	1.13e+00	1.11e+00	1.10e+00
f_9	Mean	2.67e+03 +	3.63e+03 +	3.88e+03 +	2.89e+03 +	2.96e+03 +	2.88e+03 +	2.91e+03 +
	SD	3.27e+01	6.06e+02	1.32e+02	3.94e+01	7.84e+01	9.88e+01	1.08e+02
f_{10}	Mean	3.00e+03 ≈	3.09e+03 +	3.10e+03 +	3.00e+03 ≈	3.02e+03 +	2.99e+03 ≈	3.00e+03 ≈
	SD	2.20e+01	4.56e+01	3.45e+01	5.59e+00	1.60e+01	4.93e+00	7.10e+00
+/≈/−		9/1/0	10/0/0	10/0/0	8/2/0	10/0/0	9/1/0	9/1/0
Ave. rank		5.5	13.2	13.2	6.9	7.9	5.6	6.4

Func.		CS	GWO	SCA	WOA	GJO	FLA	INFO	RIME	DEA^2H^2
f_1	Mean	6.33e+09 +	7.97e+06 +	1.42e+08 +	3.41e+07 +	3.42e+08 +	8.10e+06 +	8.76e+06 +	4.39e+05 +	**1.84e+03**
	SD	1.91e+09	3.50e+07	5.08e+07	6.24e+07	2.63e+08	4.15e+06	4.02e+07	3.27e+05	2.38e+03
f_2	Mean	5.75e+11 +	2.80e+07 +	9.19e+09 +	4.58e+09 +	4.16e+10 +	7.32e+08 +	6.99e+08 +	4.09e+07 +	**1.40e+04**
	SD	1.76e+11	3.80e+07	3.77e+09	1.03e+10	2.64e+10	3.12e+08	1.83e+09	3.59e+07	4.07e+04
f_3	Mean	2.05e+11 +	1.47e+08 +	2.66e+09 +	6.27e+08 +	9.58e+09 +	2.78e+08 +	1.14e+08 +	1.51e+07 +	**1.36e+05**
	SD	4.97e+10	1.64e+08	1.14e+09	9.75e+08	1.33e+10	1.40e+08	1.07e+08	1.16e+07	1.94e+05
f_4	Mean	7.25e+03 +	1.90e+03 +	1.91e+03 +	1.94e+03 +	1.91e+03 +	1.90e+03 +	1.91e+03 +	1.90e+03 +	**1.90e+03**
	SD	5.15e+03	7.16e−01	8.49e−01	4.84e+01	1.10e+01	6.71e−01	1.21e+01	7.21e−01	7.72e−01
f_6	Mean	1.17e+06 +	2.89e+04 +	2.50e+04 +	9.44e+03 +	1.55e+05 +	1.77e+04 +	2.60e+04 +	1.20e+04 +	2.84e+03
	SD	8.17e+05	1.93e+04	9.73e+03	6.74e+03	9.74e+04	9.02e+03	1.35e+04	7.30e+03	3.72e+03
f_6	Mean	4.10e+04 +	3.70e+03 +	3.22e+03 +	1.27e+04 +	5.09e+03 +	4.40e+03 +	1.02e+04 +	3.45e+03 +	**1.72e+03**
	SD	3.04e+04	2.71e+03	6.94e+02	8.96e+03	2.36e+03	3.02e+03	6.84e+03	3.09e+03	1.40e+02
f_7	Mean	1.43e+06 +	2.74e+04 +	3.13e+04 +	2.27e+04 +	8.77e+04 +	2.32e+04 +	2.70e+04 +	1.96e+04 +	**2.77e+03**
	SD	9.78e+05	1.49e+04	2.30e+04	1.54e+04	6.48e+04	1.30e+04	1.57e+04	1.17e+04	1.03e+03
f_8	Mean	2.35e+03 +	2.31e+03 ≈	2.31e+03 +	2.32e+03 +	2.31e+03 +	2.31e+03 +	2.31e+03 +	**2.31e+03** ≈	2.31e+03
	SD	8.72e+00	2.89e+00	1.06e+00	7.21e+00	2.20e+00	1.91e+00	6.31e+00	1.75e+00	2.14e+00
f_9	Mean	6.05e+03 +	2.75e+03 +	3.07e+03 +	2.79e+03 +	3.18e+03 +	2.73e+03 +	2.69e+03 +	2.63e+03 +	**2.59e+03**
	SD	7.50e+02	1.23e+02	9.28e+01	1.62e+02	4.24e+02	4.99e+01	8.38e+01	4.20e+01	3.89e+01
f_{10}	Mean	3.36e+03 +	**2.99e+03** ≈	3.03e+03 +	3.09e+03 +	3.01e+03 +	2.99e+03 ≈	3.05e+03 +	2.99e+03 ≈	3.00e+03
	SD	1.11e+02	1.16e+01	1.29e+01	6.31e+01	1.75e+01	1.79e+01	4.82e+01	1.20e+01	3.48e+01
+/∼/−		10/0/0	8/2/0	10/0/0	10/0/0	10/0/0	9/1/0	10/0/0	8/2/0	–
Ave. rank		16.0	5.6	11.3	11.0	11.8	6.6	9.3	3.7	**2.0**

Table 5 Experimental results and statistical analyses on 30-D CEC2020 benchmark functions

Func.		GA	PSO	DE	CMA-ES	JADE	SHADE	L-SHADE	CS	GWO	SCA	WOA	GJO	FLA	INFO	RIME	DEA^{2}H^2
f_1	Mean	1.98e+09 +	3.48e+09 +	3.80e+10 +	3.15e+09 +	3.78e+06 +	1.02e+06 +	1.58e+06 +	5.97e+10 +	4.70e+08 +	2.68e+09 +	7.04e+09 +	1.13e+10 +	1.08e+08 +	1.99e+09 +	4.51e+06 +	**3.77e+03**
	SD	3.02e+08	1.50e+09	4.49e+09	5.27e+08	1.31e+06	4.40e+05	7.33e+05	1.05e+10	3.96e+08	4.76e+08	4.02e+09	2.54e+09	2.26e+07	1.82e+09	1.78e+06	4.78e+03
f_2	Mean	2.09e+11 +	6.70e+11 +	3.71e+12 +	2.97e+11 +	3.96e+08 +	2.29e+08 +	2.35e+08 +	6.68e+12 +	5.13e+10 +	3.44e+11 +	7.43e+11 +	1.22e+12 +	1.13e+10 +	2.84e+11 +	4.98e+08 +	**7.01e+05**
	SD	3.85e+10	4.44e+11	4.17e+11	4.73e+10	1.37e+08	1.59e+08	1.67e+08	9.74e+11	6.17e+10	7.60e+10	4.11e+11	3.63e+11	2.95e+09	2.06e+11	3.34e+08	6.86e+05
f_3	Mean	6.98e+10 +	1.85e+11 +	1.27e+12 +	1.01e+11 +	2.73e+08 +	1.19e+08 +	1.27e+08 +	2.01e+12 +	1.77e+10 +	1.13e+11 +	2.37e+11 +	4.36e+11 +	3.91e+09 +	1.08e+11 +	1.49e+08 +	**2.79e+05**
	SD	1.20e+10	1.11e+11	1.41e+11	1.86e+10	1.13e+08	6.21e+07	6.70e+07	3.79e+11	1.45e+10	2.13e+10	1.29e+11	1.07e+11	9.54e+08	8.08e+10	7.09e+07	3.21e+05
f_4	Mean	1.94e+03 +	8.40e+03 +	1.05e+05 +	2.10e+03 +	1.92e+03 ≈	1.92e+03 −	1.92e+03 −	8.89e+05 +	1.92e+03 ≈	2.16e+03 +	7.07e+03 +	6.04e+03 +	1.92e+03 ≈	2.23e+03 +	**1.91e+03** −	1.92e+03
	SD	6.69e+00	1.24e+04	4.79e+04	1.12e+02	1.70e+00	1.25e+00	1.42e+00	3.88e+05	4.67e+00	1.73e+02	6.07e+03	4.89e+03	2.33e+00	2.87e+02	2.23e+00	7.85e+00
f_5	Mean	8.49e+05 +	2.24e+07 +	1.43e+07 +	5.50e+05 +	1.92e+05 +	3.82e+05 +	2.53e+05 +	1.01e+08 +	5.93e+05 +	1.73e+06 +	2.21e+06 +	3.41e+06 +	3.78e+05 +	6.95e+05 +	5.99e+05 +	**6.80e+04**
	SD	2.60e+05	2.11e+07	4.77e+06	1.61e+05	4.91e+04	9.35e+04	5.42e+04	4.11e+07	3.50e+05	8.43e+05	4.34e+06	1.44e+06	1.72e+05	5.42e+05	3.29e+05	2.60e+04
f_6	Mean	4.54e+04 +	1.42e+06 +	1.91e+05 +	6.08e+03 +	1.27e+04 +	1.44e+04 +	1.24e+04 +	1.89e+08 +	3.68e+04 +	3.32e+04 +	5.67e+04 +	2.48e+06 +	3.99e+04 +	2.77e+04 +	3.40e+04 +	**3.11e+03**
	SD	4.78e+04	4.68e+06	3.00e+05	1.19e+03	4.05e+03	3.85e+03	3.05e+03	9.23e+07	1.02e+04	1.07e+04	7.30e+04	2.72e+06	1.82e+04	9.24e+03	1.54e+04	1.80e+03
f_7	Mean	1.51e+06 +	5.48e+07 +	4.16e+07 +	5.82e+05 +	4.32e+05 +	1.01e+06 +	5.37e+05 +	5.44e+08 +	9.62e+05 +	3.04e+06 +	1.59e+06 +	2.20e+07 +	1.14e+06 +	9.50e+05 +	9.52e+05 +	**3.47e+04**
	SD	7.89e+05	5.89e+07	1.17e+07	1.32e+05	1.05e+05	3.44e+05	1.68e+05	1.85e+08	5.15e+05	1.49e+06	1.19e+06	1.58e+07	9.90e+05	7.12e+05	5.81e+05	2.30e+04
f_8	Mean	2.40e+03 ≈	2.73e+03 +	2.65e+03 +	2.45e+03 +	2.38e+03 −	**2.37e+03** −	2.37e+03 −	4.15e+03 +	2.38e+03 −	2.48e+03 +	3.33e+03 +	2.49e+03 +	2.38e+03 −	2.87e+03 +	2.38e+03 −	2.43e+03
	SD	7.78e+00	1.87e+02	2.92e+01	9.18e+00	2.04e+00	1.58e+00	1.54e+00	4.56e+02	9.73e+00	1.07e+01	5.42e+02	2.80e+01	3.75e+00	3.62e+02	8.45e+00	1.86e+01
f_9	Mean	6.17e+03 +	1.04e+04 +	1.32e+04 +	6.31e+03 +	2.71e+03 +	**2.62e+03** −	2.64e+03 ≈	3.52e+04 +	3.26e+03 +	7.04e+03 +	1.17e+04 +	1.14e+04 +	3.39e+03 +	7.72e+03 +	2.75e+03 +	2.64e+03
	SD	2.82e+02	3.82e+03	4.68e+02	3.33e+02	2.38e+01	3.49e+00	1.04e+01	4.00e+03	5.49e+02	3.49e+02	5.86e+03	1.37e+03	8.65e+01	2.51e+03	3.93e+01	1.47e+02
f_{10}	Mean	3.03e+03 +	3.63e+03 +	5.04e+03 +	3.10e+03 +	2.96e+03 +	**2.92e+03** −	2.92e+03 −	7.89e+03 +	2.97e+03 +	3.19e+03 +	3.30e+03 +	3.39e+03 +	2.93e+03 ≈	3.10e+03 +	2.93e+03 ≈	2.93e+03
	SD	1.89e+01	5.37e+02	2.99e+02	3.05e+01	1.01e+01	5.55e−01	8.63e−01	1.49e+03	3.41e+01	5.36e+01	1.55e+02	1.78e+02	9.90e+00	6.66e+01	5.01e+00	1.58e+01
+/≈/−		9/1/0	10/0/0	10/0/0	10/0/0	8/1/1	6/0/4	6/1/3	10/0/0	8/1/1	10/0/0	10/0/0	10/0/0	7/2/1	10/0/0	7/1/2	–
Ave. rank		8.6	13.3	14.3	7.9	4.0	3.0	**2.6**	16.0	6.6	10.3	12.8	13.2	6.2	9.3	5.1	2.8

Table 6 Experimental results and statistical analyses on 50-D CEC2020 benchmark functions

Func.		GA	PSO	DE	CMA-ES	JADE	SHADE	L-SHADE	CS	GWO	SCA	WOA	GJO	FLA	INFO	RIME	DEA^{2}H^2
f_1	Mean	1.69e+10 +	1.98e+10 +	1.02e+11 +	1.02e+10 +	1.46e+08 +	3.66e+05 +	1.84e+06 +	1.20e+11 +	2.25e+09 +	9.74e+09 +	2.05e+10 +	3.51e+10 +	3.06e+08 +	1.02e+10 +	1.50e+07 +	**1.37e+04**
	SD	1.78e+09	4.88e+09	9.01e+09	2.46e+09	5.29e+07	1.24e+05	8.38e+05	1.88e+10	1.26e+09	1.37e+09	8.09e+09	5.92e+09	4.04e+07	5.99e+09	5.37e+06	2.49e+04
f_2	Mean	1.83e+12 +	2.48e+12 +	1.19e+13 +	1.34e+12 +	1.68e+10 +	5.24e+07 +	2.44e+08 +	1.40e+13 +	2.79e+11 +	1.24e+12 +	2.36e+12 +	4.07e+12 +	3.67e+10 +	1.19e+12 +	2.06e+09 +	**2.25e+06**
	SD	1.75e+11	6.33e+11	1.00e+12	3.45e+11	4.60e+09	2.11e+07	8.36e+07	1.73e+12	1.47e+11	1.57e+11	1.07e+12	4.51e+11	5.70e+09	4.98e+11	1.13e+09	4.95e+06
f_3	Mean	6.12e+11 +	7.09e+11 +	4.32e+12 +	4.35e+11 +	3.24e+09 +	2.60e+07 +	8.68e+07 +	4.51e+12 +	1.09e+11 +	3.58e+11 +	6.33e+11 +	1.24e+12 +	1.19e+10 +	3.36e+11 +	5.21e+08 +	**1.46e+06**
	SD	5.46e+10	2.02e+11	3.09e+11	1.29e+11	1.43e+09	1.39e+07	5.42e+07	5.90e+11	5.54e+10	5.67e+10	2.14e+11	1.54e+11	1.28e+09	1.63e+11	1.71e+08	5.61e+06
f_4	Mean	5.34e+03 +	7.47e+04 +	1.49e+06 +	7.54e+03 +	1.96e+03 ≈	1.94e+03 −	1.93e+03 −	4.15e+06 +	1.96e+03 ≈	4.73e+03 +	3.38e+04 +	4.48e+04 +	1.94e+03 −	4.89e+03 +	**1.93e+03** −	1.96e+03
	SD	1.22e+03	1.38e+05	6.47e+05	3.16e+03	7.61e+00	2.44e+00	2.30e+00	1.89e+06	4.35e+01	1.70e+03	4.68e+04	3.12e+04	3.55e+00	2.81e+03	5.59e+00	1.67e+01
f_5	Mean	1.42e+07 +	7.79e+07 +	5.45e+07 +	2.12e+06 +	4.16e+06 +	6.52e+06 +	4.86e+06 +	2.57e+08 +	4.21e+06 +	7.42e+06 +	1.03e+07 +	2.96e+07 +	3.43e+06 +	4.16e+06 +	4.40e+06 +	**1.35e+05**
	SD	3.99e+06	5.73e+07	1.71e+07	3.95e+05	1.22e+06	1.84e+06	1.36e+06	8.85e+07	3.04e+06	2.88e+06	6.72e+06	1.64e+07	1.55e+06	2.13e+06	1.88e+06	6.51e+04
f_6	Mean	8.73e+04 +	6.53e+06 +	9.27e+07 +	3.48e+04 +	1.91e+04 +	3.51e+04 +	2.33e+04 +	4.15e+09 +	5.63e+05 +	8.94e+05 +	6.00e+05 +	1.48e+08 +	6.63e+04 +	6.91e+05 +	6.29e+04 +	**3.58e+03**
	SD	5.37e+04	9.11e+06	2.88e+07	9.39e+03	5.99e+03	1.35e+04	6.56e+03	1.63e+09	8.44e+05	4.06e+05	1.07e+06	1.88e+08	9.02e+04	9.72e+05	2.80e+04	3.42e+03
f_7	Mean	9.36e+07 +	7.20e+08 +	7.55e+08 +	1.21e+07 +	4.40e+06 +	1.08e+07 +	7.36e+06 +	7.18e+09 +	5.08e+06 +	3.44e+07 +	2.77e+07 +	2.77e+08 +	6.98e+06 +	6.84e+06 +	5.64e+06 +	**1.14e+05**
	SD	2.22e+07	6.68e+08	2.11e+08	3.09e+06	1.19e+06	3.32e+06	2.25e+06	2.49e+09	2.91e+06	1.12e+07	2.63e+07	1.21e+08	4.06e+06	3.93e+06	3.07e+06	4.09e+04
f_8	Mean	2.68e+03 −	4.14e+03 +	3.19e+03 +	2.66e+03 ≈	2.45e+03 −	**2.43e+03** −	2.44e+03 −	1.01e+04 +	2.45e+03 −	2.82e+03 +	1.02e+04 +	3.01e+03 +	2.44e+03 −	6.64e+03 +	2.47e+03 −	2.71e+03
	SD	2.06e+01	5.91e+02	8.93e+01	4.33e+01	1.01e+01	4.65e+00	7.45e+00	1.60e+03	2.03e+01	5.88e+01	2.37e+03	1.45e+02	1.07e+01	1.96e+03	1.70e+01	3.62e+02
f_9	Mean	1.54e+04 +	2.27e+04 +	2.70e+04 +	1.14e+04 +	3.37e+03 +	**2.66e+03** −	2.69e+03 −	7.52e+04 +	6.20e+03 +	1.42e+04 +	2.87e+04 +	2.61e+04 +	4.19e+03 +	2.05e+04 +	2.93e+03 +	2.85e+03
	SD	5.47e+02	3.62e+03	2.01e+03	1.40e+03	6.40e+02	1.02e+02	2.07e+01	7.80e+03	1.22e+03	1.01e+03	9.78e+03	4.09e+03	2.68e+02	1.03e+04	8.04e+01	5.27e+02
f_{10}	Mean	5.06e+03 +	8.75e+03 +	1.23e+04 +	4.60e+03 +	3.62e+03 +	**3.32e+03** ≈	3.34e+03 ≈	2.60e+04 +	3.83e+03 +	5.59e+03 +	5.59e+03 +	6.90e+03 +	3.47e+03 +	4.47e+03 +	3.46e+03 +	3.33e+03
	SD	2.19e+02	2.61e+03	1.64e+03	3.07e+02	4.44e+01	3.16e+01	2.90e+01	4.38e+03	2.10e+02	2.60e+02	1.00e+03	7.37e+02	1.82e+02	5.84e+02	1.58e+02	6.48e+01
+/≈/−		9/0/1	10/0/0	10/0/0	9/1/0	8/1/1	6/1/3	6/1/3	10/0/0	8/1/1	10/0/0	10/0/0	10/0/0	8/0/2	10/0/0	8/0/2	–
Ave. rank		10.3	13.3	14.4	7.9	4.4	**2.8**	3.2	15.9	6.5	9.8	12.2	13.3	5.2	8.9	4.4	3.0

Table 7 Experimental results and statistical analyses on 100-D CEC2020 benchmark functions

Func.		GA	PSO	DE	CMA-ES	JADE	SHADE	L-SHADE	CS	GWO	SCA	WOA	GJO	FLA	INFO	RIME	DEA^{2}H^2
f_1	Mean	1.09e+11 +	9.83e+10 +	3.32e+11 +	8.71e+10 +	2.72e+07 +	**6.57e+03** −	5.40e+04 −	2.88e+11 +	1.51e+10 +	4.37e+10 +	7.65e+10 +	1.50e+11 +	1.27e+09 +	4.86e+10 +	5.91e+07 +	6.05e+06
	SD	5.57e+09	1.02e+10	2.02e+10	1.04e+10	7.13e+06	3.78e+03	3.79e+04	3.83e+10	5.55e+09	4.30e+09	1.60e+10	1.12e+10	1.66e+08	1.24e+10	1.76e+07	6.88e+06
f_2	Mean	1.10e+13 +	1.10e+13 +	3.23e+13 +	9.08e+12 +	3.68e+09 +	**4.23e+05** −	6.27e+06 −	3.09e+13 +	1.80e+12 +	4.84e+12 +	8.33e+12 +	1.54e+13 +	1.25e+11 +	4.76e+12 +	5.56e+09 +	4.25e+08
	SD	6.40e+11	1.20e+12	2.51e+12	1.70e+12	1.07e+09	2.46e+05	2.80e+06	3.26e+12	7.29e+11	4.70e+11	2.18e+12	1.10e+12	1.33e+10	1.57e+12	1.34e+09	5.08e+08
f_3	Mean	3.85e+12 +	3.63e+12 +	1.11e+13 +	3.06e+12 +	1.52e+09 +	**1.41e+05** −	3.15e+06 −	1.11e+13 +	5.49e+11 +	1.65e+12 +	2.78e+12 +	5.03e+12 +	4.69e+10 +	1.76e+12 +	2.18e+09 +	1.24e+08
	SD	2.13e+11	4.53e+11	1.02e+12	3.83e+11	3.91e+08	8.37e+04	1.95e+06	1.53e+12	1.49e+11	1.83e+11	5.27e+11	3.05e+11	5.66e+09	4.40e+11	6.55e+08	1.54e+08
f_4	Mean	3.56e+05 +	9.32e+06 +	2.10e+07 +	7.91e+04 +	2.06e+03 ≈	**1.98e+03** −	1.98e+03 −	1.08e+07 +	3.40e+03 +	4.32e+04 +	8.65e+04 +	4.03e+05 +	2.00e+03 −	2.30e+04 +	2.00e+03 −	2.07e+03
	SD	1.03e+05	4.21e+06	6.64e+06	3.66e+04	1.10e+01	4.62e+00	5.68e+00	4.55e+06	2.26e+03	1.26e+04	4.81e+04	9.99e+04	6.94e+00	1.58e+04	1.49e+01	5.56e+01
f_5	Mean	4.11e+08 +	1.18e+09 +	7.70e+08 +	2.74e+07 +	4.80e+07 +	4.88e+07 +	4.51e+07 +	1.12e+09 +	2.93e+07 +	9.28e+07 +	5.49e+07 +	2.37e+08 +	3.00e+07 +	2.93e+07 +	2.44e+07 +	**1.05e+06**
	SD	5.19e+07	4.45e+08	1.04e+08	7.32e+06	7.41e+06	1.08e+07	9.12e+06	2.99e+08	1.21e+07	1.76e+07	2.30e+07	4.04e+07	7.15e+06	1.07e+07	7.76e+06	4.87e+05
f_6	Mean	8.49e+07 +	2.77e+10 +	3.43e+09 +	2.82e+05 +	6.06e+04 +	8.62e+03 +	9.30e+03 +	2.55e+10 +	3.57e+06 +	1.12e+07 +	3.09e+07 +	2.32e+09 +	8.51e+05 +	6.24e+07 +	1.98e+05 +	**7.45e+03**
	SD	1.99e+07	2.00e+10	1.24e+09	8.66e+04	4.49e+04	8.56e+03	4.73e+03	9.27e+09	8.36e+06	5.59e+06	6.63e+07	1.51e+09	1.34e+06	1.51e+08	8.13e+04	6.10e+03
f_7	Mean	1.68e+09 +	2.11e+10 +	5.21e+09 +	3.66e+07 +	2.25e+07 +	2.08e+07 +	2.30e+07 +	2.29e+10 +	5.30e+07 +	3.79e+08 +	1.72e+08 +	2.19e+09 +	2.62e+07 +	4.95e+07 +	1.26e+07 +	**7.97e+05**
	SD	2.86e+08	1.32e+10	1.48e+09	1.16e+07	5.41e+06	7.61e+06	6.57e+06	4.43e+09	3.53e+07	1.02e+08	1.31e+08	5.70e+08	1.25e+07	3.79e+07	4.12e+06	3.43e+05
f_8	Mean	3.48e+03 ≈	2.54e+04 +	4.51e+03 +	3.53e+03 ≈	2.42e+03 −	**2.42e+03** −	2.42e+03 −	2.32e+04 +	2.57e+03 −	3.60e+03 +	2.40e+04 +	6.52e+03 +	2.49e+03 −	1.46e+04 +	2.51e+03 −	3.46e+03
	SD	7.42e+01	8.22e+03	3.77e+02	2.91e+02	5.03e+00	4.28e+00	6.34e+00	2.45e+03	2.95e+01	1.35e+02	3.59e+03	5.23e+02	1.16e+01	2.26e+03	1.50e+01	7.13e+02
f_9	Mean	7.26e+04 +	2.28e+05 +	1.16e+05 +	6.53e+04 +	3.87e+03 +	**2.60e+03** −	2.62e+03 −	2.23e+05 +	2.60e+04 +	7.70e+04 +	1.13e+05 +	1.17e+05 +	9.39e+03 +	8.74e+04 +	3.89e+03 +	3.50e+03
	SD	3.03e+03	4.26e+04	1.27e+04	7.45e+03	2.99e+02	4.84e−01	3.95e+00	2.33e+04	5.46e+03	9.33e+03	2.28e+04	8.83e+03	1.22e+03	2.54e+04	1.25e+02	1.50e+03
f_{10}	Mean	1.14e+04 +	4.39e+04 +	4.19e+04 +	8.51e+03 +	3.95e+03 +	**3.49e+03** ≈	3.59e+03 +	4.20e+04 +	4.54e+03 +	7.79e+03 +	7.30e+03 +	1.26e+04 +	3.63e+03 +	6.23e+03 +	3.63e+03 +	3.50e+03
	SD	5.76e+02	8.89e+03	7.87e+03	1.14e+03	2.99e+01	4.26e+01	5.78e+01	8.80e+03	3.35e+02	4.23e+02	7.90e+02	9.14e+02	8.16e+01	1.18e+03	8.07e+01	5.66e+01
+/≈/−		9/1/0	10/0/0	10/0/0	9/1/0	8/1/1	3/1/6	4/0/6	10/0/0	9/0/1	10/0/0	10/0/0	10/0/0	8/0/2	10/0/0	8/0/2	–
Ave. rank		11.7	14.5	14.3	8.7	4.6	**2.1**	2.8	15.1	7.0	9.5	10.7	13.2	5.6	8.9	4.1	3.2

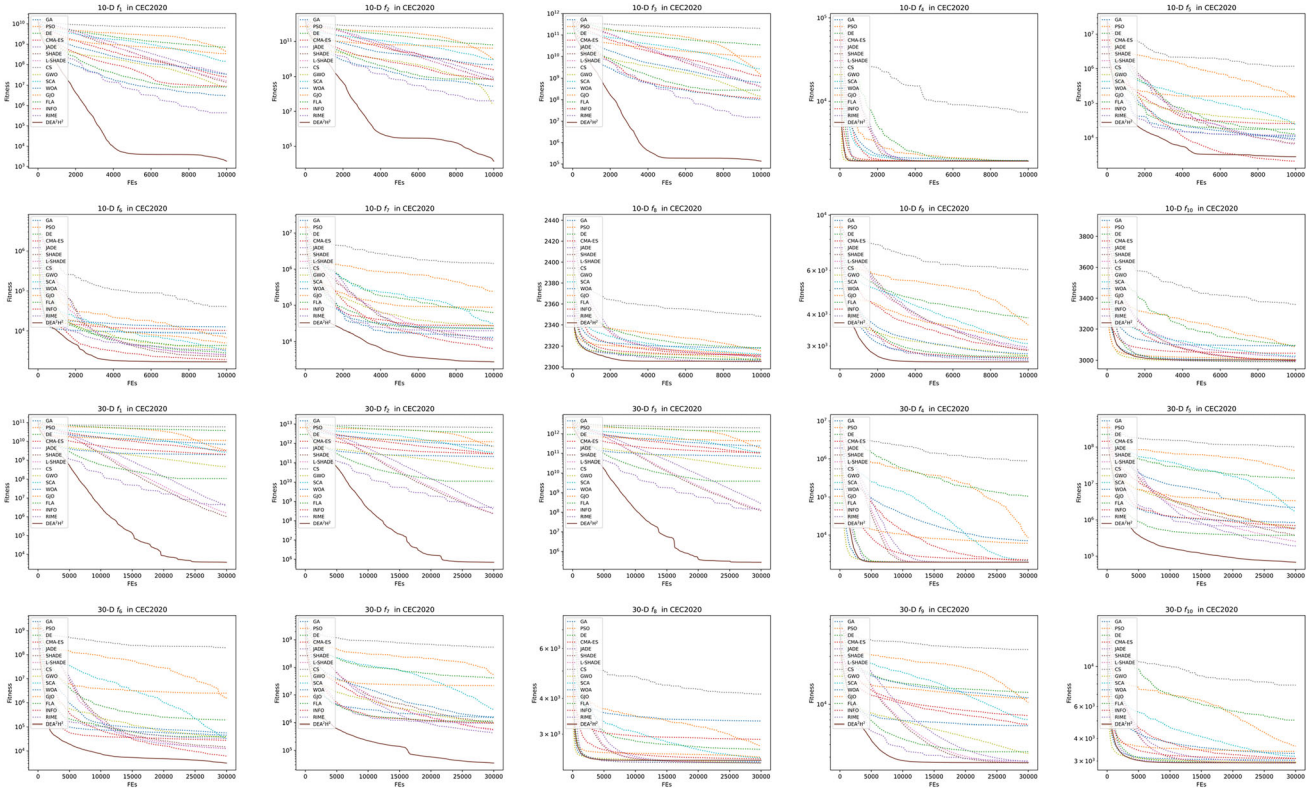


Fig. 11 Convergence curves on 10-D and 30-D CEC2020 benchmark functions

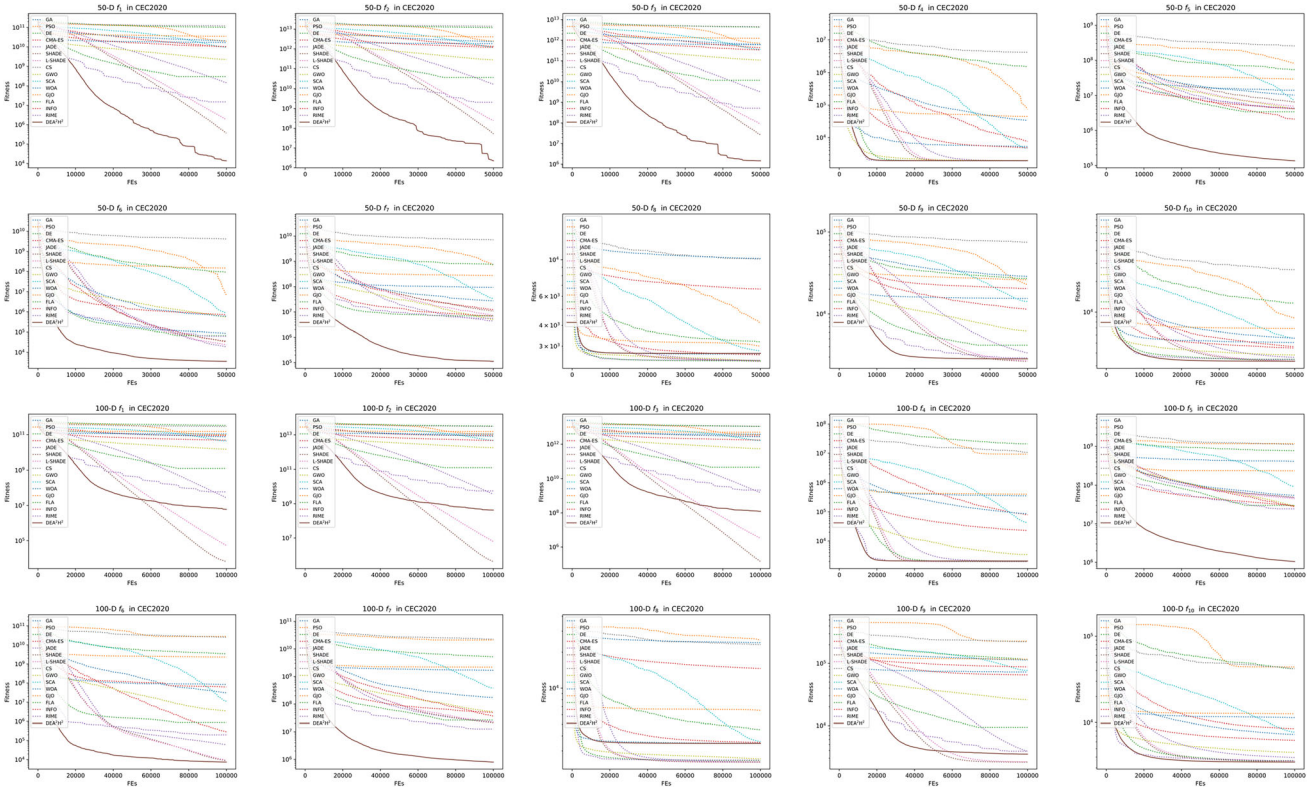


Fig. 12 Convergence curves on 50-D and 100-D CEC2020 benchmark functions

Table 8 Experimental results and statistical analyses on 10-D CEC2022 benchmark functions

Func.		GA	PSO	DE	CMA-ES	JADE	SHADE	L-SHADE	CS	GWO	SCA	WOA	GJO	FLA	INFO	RIME	DEA^{2}H^2
f_1	Mean	3.20e+03 +	1.91e+03 +	6.94e+03 +	4.38e+02 +	1.22e+03 +	1.20e+03 +	1.12e+03 +	9.59e+03 +	3.32e+02 +	6.38e+02 +	3.39e+03 +	6.47e+02 +	3.14e+02 +	3.62e+02 +	3.01e+02 +	**3.00e+02**
	SD	1.20e+03	1.44e+03	1.75e+03	5.24e+01	3.73e+02	3.30e+02	2.87e+02	2.90e+03	2.51e+01	1.47e+02	2.10e+03	6.25e+02	6.00e+00	1.66e+02	1.30e+00	6.97e−10
f_2	Mean	4.69e+02 +	4.97e+02 +	4.58e+02 +	4.15e+02 +	4.19e+02 +	4.20e+02 +	4.19e+02 +	9.20e+02 +	4.18e+02 +	4.39e+02 +	4.86e+02 +	4.47e+02 +	4.15e+02 +	4.27e+02 +	4.13e+02 +	**4.07e+02**
	SD	1.48e+01	3.69e+01	1.23e+01	2.62e+00	8.42e+00	7.56e+00	6.89e+00	2.01e+02	2.03e+01	1.78e+01	8.80e+01	1.88e+01	2.03e+01	2.78e+01	1.90e+01	9.56e+00
f_3	Mean	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	**6.00e+02**
	SD	7.97e−03	5.02e−03	1.23e−02	6.89e−04	2.11e−04	6.88e−05	8.44e−05	9.33e−02	1.70e−03	2.52e−03	7.70e−03	1.17e−02	2.08e−04	4.89e−04	2.01e−05	1.96e−13
f_4	Mean	8.01e+02 +	8.01e+02 +	8.01e+02 +	8.01e+02 +	8.00e+02 +	8.00e+02 +	8.00e+02 ≈	8.01e+02 +	8.00e+02 +	8.00e+02 +	8.00e+02 +	8.00e+02 +	8.00e+02 ≈	8.00e+02 ≈	**8.00e+02** ≈	8.00e+02
	SD	1.53e−01	2.00e−01	2.03e−01	1.76e−01	7.32e−02	7.45e−02	8.41e−02	2.17e−01	3.64e−01	1.31e−01	3.28e−01	1.53e−01	1.10e−01	3.02e−01	1.34e−01	1.66e−01
f_5	Mean	9.01e+02 ≈	9.01e+02 ≈	9.02e+02 +	9.00e+02 −	9.00e+02 ≈	9.00e+02 ≈	9.00e+02 ≈	9.04e+02 +	**9.00e+02** −	9.00e+02 −	9.03e+02 +	9.00e+02 ≈	9.00e+02 −	9.01e+02 ≈	9.00e+02 −	9.01e+02
	SD	1.26e−01	6.85e−01	5.63e−01	7.71e−02	2.01e−01	1.42e−01	1.01e−01	6.85e−01	1.45e−01	6.40e−02	1.59e+00	2.77e−01	6.11e−01	1.08e+00	4.08e−01	7.75e−01
f_6	Mean	1.26e+05 +	1.26e+06 +	1.05e+05 +	**4.12e+03** ≈	2.71e+04 +	3.85e+04 +	3.34e+04 +	6.54e+07 +	3.85e+04 +	2.44e+05 +	2.68e+04 +	6.16e+04 +	6.09e+04 +	3.48e+04 +	2.88e+04 +	7.13e+03
	SD	2.02e+05	2.15e+06	3.78e+04	6.53e+02	1.12e+04	2.49e+04	1.73e+04	3.85e+07	1.93e+04	1.51e+05	1.06e+04	3.85e+04	3.32e+04	2.05e+04	1.55e+04	8.93e+03
f_7	Mean	2.05e+03 +	2.13e+03 +	2.07e+03 +	2.08e+03 +	2.03e+03 +	2.04e+03 +	2.03e+03 +	2.41e+03 +	2.04e+03 +	2.07e+03 +	2.14e+03 +	2.12e+03 +	2.04e+03 +	2.10e+03 +	2.03e+03 +	**2.03e+03**
	SD	8.55e+00	6.05e+01	1.17e+01	1.62e+01	4.14e+00	3.71e+00	3.04e+00	1.24e+02	2.67e+01	1.59e+01	9.68e+01	9.19e+01	1.83e+01	5.24e+01	1.80e+01	8.26e+00
f_8	Mean	2.40e+03 +	3.74e+03 +	2.25e+03 +	2.23e+03 +	2.24e+03 +	2.24e+03 +	2.24e+03 +	9.42e+05 +	2.50e+03 +	2.31e+03 +	3.04e+03 +	3.70e+03 +	2.28e+03 +	2.42e+03 +	2.23e+03 ≈	**2.22e+03**
	SD	2.49e+02	4.23e+03	7.30e+00	3.23e+00	6.57e+00	7.83e+00	5.73e+00	2.38e+06	5.43e+02	4.60e+01	7.42e+02	6.21e+02	1.06e+02	2.14e+02	4.93e+00	1.06e+00
f_9	Mean	2.60e+03 +	2.66e+03 +	2.67e+03 +	2.58e+03 ≈	2.36e+03 −	2.37e+03 −	2.36e+03 −	2.97e+03 +	2.63e+03 +	**2.35e+03** −	2.76e+03 +	2.56e+03 ≈	2.55e+03 ≈	2.61e+03 +	2.54e+03 ≈	2.56e+03
	SD	8.59e+01	1.63e+02	4.50e+01	1.31e+02	2.76e+01	4.35e+01	3.33e+01	1.14e+02	1.28e+02	3.40e+01	1.22e+02	1.93e+02	1.73e+02	1.50e+02	1.80e+02	1.59e+02
f_{10}	Mean	2.62e+03 ≈	2.62e+03 +	2.62e+03 ≈	2.61e+03 ≈	2.61e+03 ≈	2.61e+03 ≈	2.61e+03 ≈	2.68e+03 +	2.63e+03 + ≈	**2.61e+03** ≈	2.67e+03 +	2.66e+03 +	2.63e+03 ≈	2.67e+03 +	2.62e+03 ≈	2.63e+03
	SD	3.25e+01	9.08e+00	4.23e+00	8.06e−01	3.34e+00	2.39e+00	2.95e+00	2.71e+01	5.51e+01	1.09e+00	1.19e+02	7.10e+01	1.01e+02	1.33e+02	4.48e+01	5.72e+01
f_{11}	Mean	2.63e+03 −	2.62e+03 −	2.66e+03 ≈	2.65e+03 ≈	2.61e+03 −	2.60e+03 −	2.61e+03 −	2.83e+03 +	**2.60e+03** −	2.61e+03 −	2.65e+03 ≈	2.61e+03 −	2.63e+03 −	2.69e+03 +	2.63e+03 −	2.66e+03
	SD	4.71e+00	1.36e+01	1.51e+02	1.75e+02	2.11e+00	1.39e+00	1.68e+00	1.08e+02	1.93e+00	2.70e+00	1.59e+02	1.38e+01	1.57e+02	2.62e+02	1.57e+02	2.18e+02
f_{12}	Mean	2.88e+03 ≈	2.91e+03 +	2.87e+03 ≈	2.87e+03 ≈	2.87e+03 ≈	2.87e+03 ≈	2.87e+03 ≈	3.02e+03 +	**2.87e+03** ≈	2.89e+03 +	2.92e+03 +	2.87e+03 ≈	2.87e+03 ≈	2.88e+03 ≈	2.87e+03 ≈	2.87e+03
	SD	2.01e+00	1.98e+01	7.31e−01	5.12e−01	2.04e+01	5.86e−01	7.64e−01	5.10e+01	9.19e−01	5.04e+00	6.12e+01	7.43e+00	1.11e+00	2.28e+01	1.17e+00	2.04e+00
+/≈/−		8/3/1	10/1/1	9/3/0	6/5/1	7/3/2	7/3/2	6/4/2	12/0/0	8/2/2	8/1/3	11/1/0	8/3/1	6/4/2	9/3/0	5/5/2	–
Ave. rank		10.7	12.2	12.0	6.8	5.5	6.1	5.1	16.0	6.7	7.6	12.5	10.0	6.2	9.9	**3.5**	4.8

Table 9 Experimental results and statistical analyses on 20-D CEC2022 benchmark functions

Func.		GA	PSO	DE	CMA-ES	JADE	SHADE	L-SHADE	CS	GWO	SCA	WOA	GJO	FLA	INFO	RIME	DEA^{2}H^2
f_1	Mean	2.27e+04 +	1.30e+04 +	3.43e+04 +	2.59e+03 +	3.27e+03 +	2.47e+03 +	2.46e+03 +	3.63e+04 +	5.54e+02 +	4.62e+03 +	6.21e+03 +	6.64e+03 +	3.55e+02 +	2.22e+03 +	3.04e+02 +	**3.00e+02**
	SD	3.65e+03	5.87e+03	4.84e+03	4.39e+02	9.02e+02	5.34e+02	5.39e+02	6.15e+03	1.78e+02	1.13e+03	3.69e+03	2.06e+03	1.58e+01	1.41e+03	2.00e+00	1.48e−06
f_2	Mean	8.73e+02 +	7.21e+02 +	1.05e+03 +	5.17e+02 +	4.95e+02 +	4.56e+02 +	4.62e+02 +	2.72e+03 +	4.77e+02 +	5.58e+02 +	6.17e+02 +	5.82e+02 +	4.66e+02 +	5.16e+02 +	4.58e+02 +	**4.45e+02**
	SD	8.41e+01	1.63e+02	1.36e+02	1.52e+01	1.27e+01	5.73e+00	9.78e+00	5.53e+02	2.28e+01	2.53e+01	8.45e+01	4.95e+01	3.67e+01	5.52e+01	1.31e+01	2.20e+01
f_3	Mean	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.01e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	**6.00e+02**
	SD	4.88e−02	1.08e−01	6.40e−02	5.57e−03	6.32e−05	1.07e−05	2.73e−05	1.97e−01	4.78e−03	1.22e−02	7.58e−02	7.14e−02	4.17e−04	1.29e−02	2.82e−05	6.22e−13
f_4	Mean	8.04e+02 +	8.04e+02 +	8.04e+02 +	8.04e+02 +	8.01e+02 ≈	8.01e+02 ≈	8.01e+02 ≈	8.04e+02 +	8.02e+02 +	8.02e+02 +	8.01e+02 ≈	8.02e+02 +	8.01e+02 ≈	8.01e+02 ≈	**8.01e+02** ≈	8.01e+02
	SD	2.74e−01	4.07e−01	3.27e−01	4.27e−01	1.88e−01	1.95e−01	1.58e−01	4.51e−01	1.21e+00	2.88e−01	6.31e−01	6.45e−01	4.68e−01	4.00e−01	3.46e−01	3.75e−01
f_5	Mean	9.05e+02 +	9.05e+02 ≈	9.15e+02 +	9.03e+02 ≈	9.02e+02 ≈	9.02e+02 ≈	9.02e+02 ≈	9.16e+02 +	**9.01e+02** −	9.01e+02 −	9.08e+02 +	9.02e+02 ≈	9.03e+02 ≈	9.03e+02 ≈	9.02e+02 ≈	9.03e+02
	SD	7.76e−01	3.28e+00	2.46e+00	5.98e−01	9.65e−01	6.60e−01	8.21e−01	2.55e+00	6.39e−01	3.19e−01	3.53e+00	9.23e−01	2.21e+00	1.31e+00	1.69e+00	1.69e+00
f_6	Mean	3.33e+08 +	2.63e+08 +	2.19e+08 +	4.95e+06 +	1.89e+06 +	7.66e+06 +	3.04e+06 +	2.21e+09 +	9.52e+04 +	1.19e+07 +	8.64e+04 +	2.76e+07 +	1.07e+06 +	3.88e+05 +	7.78e+04 +	**4.58e+04**
	SD	1.11e+08	3.64e+08	7.11e+07	1.89e+06	7.69e+05	3.28e+06	1.28e+06	8.81e+08	9.05e+04	6.50e+06	4.20e+04	4.13e+07	5.49e+05	1.73e+06	2.87e+04	1.80e+04
f_7	Mean	2.76e+03 +	2.94e+03 +	2.82e+03 +	2.31e+03 +	2.11e+03 +	2.15e+03 +	2.12e+03 +	4.74e+03 +	2.08e+03 +	2.24e+03 +	3.28e+03 +	2.26e+03 +	2.09e+03 +	2.71e+03 +	2.06e+03 ≈	**2.05e+03**
	SD	1.65e+02	5.55e+02	2.30e+02	6.08e+01	1.81e+01	2.57e+01	1.97e+01	5.39e+02	3.67e+01	9.08e+01	6.35e+02	2.31e+02	3.81e+01	5.41e+02	1.64e+01	1.97e+01
f_8	Mean	1.69e+06 +	4.23e+07 +	1.11e+05 +	2.42e+03 +	2.63e+03 +	2.92e+03 +	2.65e+03 +	8.72e+09 +	4.75e+03 +	4.08e+03 +	5.79e+03 +	7.70e+03 +	4.25e+03 +	3.78e+03 +	4.10e+03 +	**2.25e+03**
	SD	4.98e+06	1.10e+08	2.67e+05	7.47e+01	1.77e+02	2.44e+02	1.92e+02	1.29e+10	1.24e+03	1.03e+03	2.45e+03	3.29e+03	1.42e+03	6.71e+02	1.10e+03	3.91e+01
f_9	Mean	3.04e+03 +	3.21e+03 +	2.78e+03 +	2.65e+03 +	2.69e+03 +	2.67e+03 +	2.67e+03 +	4.62e+03 +	2.66e+03 +	2.83e+03 +	2.79e+03 +	2.84e+03 +	2.65e+03 +	2.69e+03 +	2.65e+03 +	**2.64e+03**
	SD	5.12e+01	2.21e+02	5.01e+01	6.49e+00	4.44e+01	8.02e+00	7.54e+00	4.51e+02	2.08e+01	3.97e+01	1.54e+02	9.51e+01	6.60e+00	5.13e+01	6.78e+00	1.08e−02
f_{10}	Mean	2.94e+03 ≈	2.83e+03 ≈	3.59e+03 ≈	3.15e+03 ≈	2.81e+03 ≈	2.82e+03 ≈	2.87e+03 ≈	3.73e+03 +	3.56e+03 ≈	**2.79e+03** ≈	4.56e+03 +	3.64e+03 ≈	4.03e+03 ≈	4.50e+03 +	3.22e+03 ≈	3.49e+03
	SD	1.31e+02	3.48e+01	1.36e+03	8.56e+02	2.13e+01	2.61e+01	2.45e+02	6.64e+02	1.14e+03	3.56e+01	1.21e+03	1.31e+03	9.79e+02	1.21e+03	7.96e+02	1.08e+03
f_{11}	Mean	2.77e+03 +	2.80e+03 +	2.73e+03 +	2.62e+03 +	2.61e+03 +	**2.60e+03** −	2.60e+03 −	6.18e+03 +	2.61e+03 +	2.63e+03 +	2.97e+03 +	2.65e+03 +	2.63e+03 +	2.75e+03 +	2.60e+03 +	2.60e+03
	SD	4.29e+01	3.40e+02	2.87e+01	2.19e+00	1.84e+00	2.87e−01	8.58e−01	1.15e+03	1.00e+01	4.52e+00	5.82e+02	3.25e+01	1.48e+02	3.83e+02	5.89e+00	6.25e+00
f_{12}	Mean	3.10e+03 +	3.24e+03 +	2.96e+03 ≈	2.95e+03 −	2.97e+03 ≈	**2.95e+03** −	2.95e+03 −	3.54e+03 +	2.96e+03 ≈	3.13e+03 +	3.18e+03 +	3.03e+03 +	2.97e+03 ≈	3.08e+03 +	2.96e+03 ≈	2.98e+03
	SD	2.05e+01	9.55e+01	5.36e+00	6.52e+00	5.75e+00	3.39e+00	2.86e+00	1.15e+02	2.03e+01	2.75e+01	1.45e+02	4.12e+01	1.81e+01	9.80e+01	1.59e+01	2.52e+01
+/≈/−		11/1/0	10/2/0	10/2/0	9/2/1	8/4/0	7/3/2	7/3/2	12/0/0	9/2/1	10/1/1	11/1/0	10/2/0	8/4/0	10/2/0	7/5/0	–
Ave. rank		12.9	12.7	12.4	7.7	6.0	4.8	4.9	15.7	5.9	8.5	11.6	10.8	6.4	8.4	3.7	**3.3**

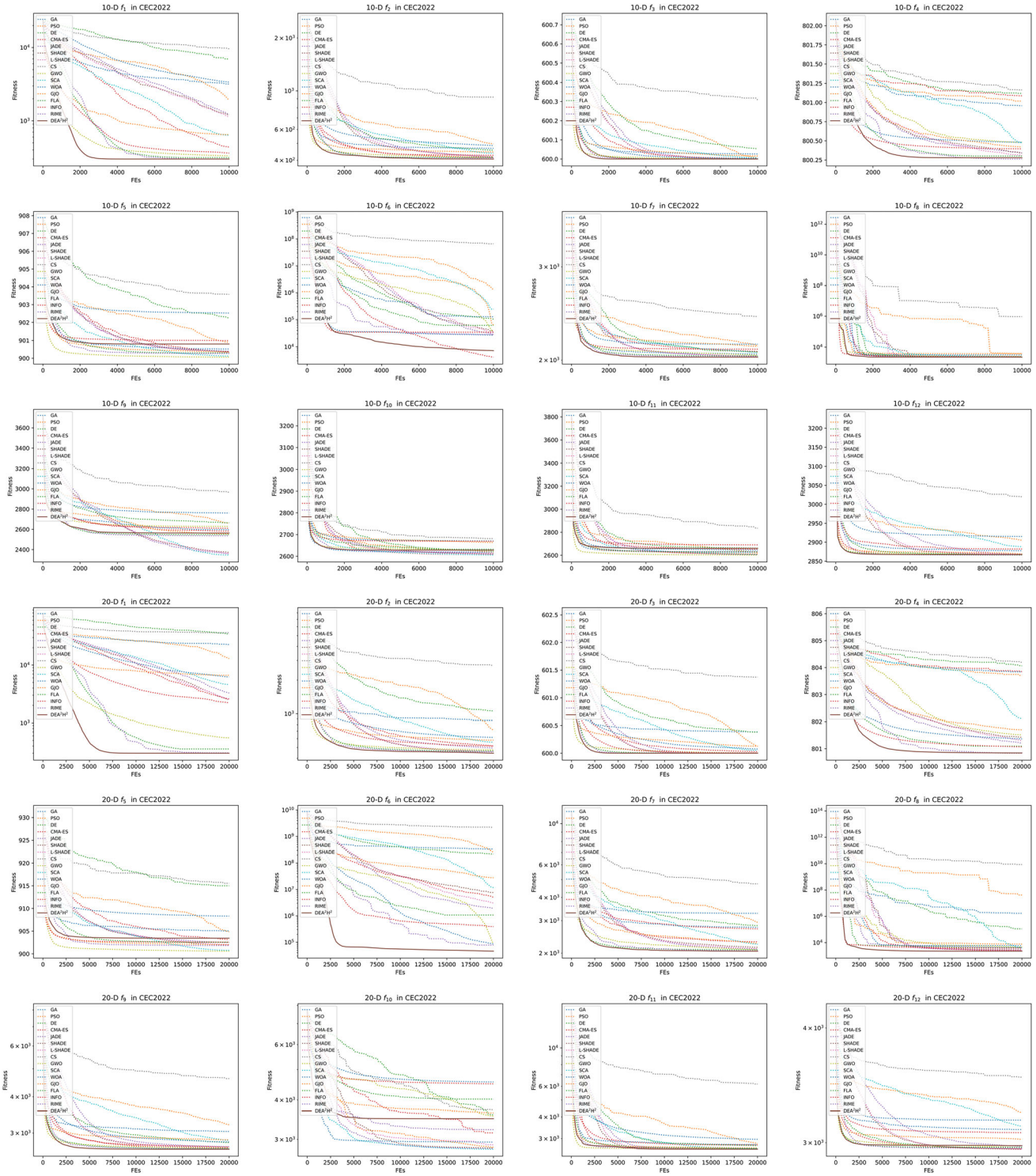


Fig. 13 Convergence curves on 10-D and 20-D CEC2022 benchmark functions

$O(N \cdot D)$. For each individual, the computational complexity of any LLH operation is $O(D)$, and the update process of the success-history memory is $O(1)$. Therefore, the overall computational complexity of DEA^{2}H^2 can be approximately expressed as $O(T \cdot N \cdot D)$.

This analysis indicates that the computational requirements of DEA^{2}H^2 grow linearly with the population size, dimension size, and maximum iteration. Understanding the computational complexity provides insights into the algorithm's efficiency and scalability.

Table 10 Experimental results and statistical analyses on engineering problems

Func.		GA	PSO	DE	CMA-ES	JADE	SHADE	L-SHADE	CS	GWO	SCA	WOA	GJO	FLA	INFO	RIME	DEA^{2}H^2
CBD	Mean	1.520e+00 +	2.204e+00 +	2.015e+00 +	1.559e+00 +	1.365e+00 +	1.354e+00 +	1.354e+00 +	3.683e+00 +	1.340e+00 +	1.353e+00 +	4.361e+00 +	1.341e+00 +	1.671e+00 +	1.345e+00 +	1.401e+00 +	**1.340e+00**
	SD	1.299e-01	2.871e-01	2.139e-01	8.311e-02	1.452e-02	6.559e-03	6.797e-03	7.126e-01	1.593e-04	5.027e-03	1.323e+00	5.532e-04	2.717e-01	4.382e-03	4.339e-02	2.619e-04
	Best	1.369e+00	1.682e+00	1.550e+00	1.397e+00	1.349e+00	1.344e+00	1.343e+00	2.098e+00	1.340e+00	1.343e+00	2.101e+00	1.340e+00	1.408e+00	1.340e+00	1.348e+00	1.340e+00
	Worst	1.826e+00	3.121e+00	2.585e+00	1.734e+00	1.419e+00	1.371e+00	1.366e+00	4.890e+00	1.341e+00	1.365e+00	7.591e+00	1.343e+00	2.732e+00	1.357e+00	1.514e+00	1.341e+00
CBHD	Mean	9.222e+00 +	7.439e+00 +	6.949e+00 +	6.874e+00 +	7.079e+00 +	6.914e+00 +	6.945e+00 +	8.206e+00 +	6.862e+00 +	7.003e+00 +	7.322e+00 +	7.251e+00 +	6.980e+00 +	6.898e+00 +	6.868e+00 +	**6.851e+00**
	SD	1.082e+00	2.430e-01	3.671e-02	9.187e-03	1.654e-01	3.306e-02	8.368e-02	4.613e-01	8.570e-03	4.093e-02	4.652e-01	4.655e-01	2.003e-01	1.063e-01	1.467e-02	2.213e-02
	Best	6.970e+00	6.964e+00	6.890e+00	6.852e+00	6.862e+00	6.856e+00	6.863e+00	7.374e+00	6.848e+00	6.923e+00	6.872e+00	6.858e+00	6.879e+00	6.843e+00	6.846e+00	6.843e+00
	Worst	1.086e+01	7.919e+00	7.036e+00	6.901e+00	7.581e+00	6.980e+00	7.251e+00	9.498e+00	6.882e+00	7.079e+00	8.991e+00	7.966e+00	7.974e+00	7.304e+00	6.894e+00	6.950e+00
CSP	Mean	7.364e-03 +	8.812e-03 +	6.076e-03 +	6.076e-03 +	6.078e-03 +	6.076e-03 +	6.076e-03 +	1.113e-02 +	6.077e-03 +	6.408e-03 +	6.722e-03 +	6.078e-03 +	6.084e-03 +	6.265e-03 +	6.086e-03 +	**6.076e-03**
	SD	1.026e-03	1.381e-03	3.646e-15	3.007e-11	2.612e-06	1.979e-07	2.187e-07	2.049e-03	4.694e-07	1.158e-04	1.288e-03	1.347e-06	1.784e-05	6.784e-04	2.474e-05	0.000e+00
	Best	6.198e-03	6.803e-03	6.076e-03	6.076e-03	6.076e-03	6.076e-03	6.076e-03	6.430e-03	6.076e-03	6.183e-03	6.076e-03	6.076e-03	6.076e-03	6.076e-03	6.076e-03	6.076e-03
	Worst	1.020e-02	1.363e-02	6.076e-03	6.076e-03	6.089e-03	6.077e-03	6.077e-03	1.579e-02	6.078e-03	6.702e-03	1.031e-02	6.081e-03	6.178e-03	9.734e-03	6.210e-03	6.076e-03
GTD	Mean	2.347e-09 +	6.432e-09 +	1.838e-10 +	4.365e-11 +	5.814e-12 +	5.194e-12 +	4.131e-12 +	3.467e-07 +	2.553e-11 +	1.150e-10 +	**0.000e+00** ≈	4.097e-11 +	3.410e-10 +	1.655e-13 +	1.651e-11 +	**0.000e+00**
	SD	3.998e-09	1.158e-08	2.820e-10	7.241e-11	1.745e-11	9.050e-12	7.898e-12	8.675e-07	5.624e-11	2.715e-10	0.000e+00	8.123e-11	8.086e-10	3.508e-13	3.216e-11	0.000e+00
	Best	3.804e-13	6.535e-12	8.179e-13	2.502e-16	5.041e-16	1.528e-17	9.016e-18	1.461e-11	2.607e-15	1.083e-14	0.000e+00	3.129e-14	1.440e-15	5.904e-17	8.160e-17	0.000e+00
	Worst	1.545e-08	6.275e-08	9.423e-10	3.635e-10	9.708e-11	4.161e-11	3.503e-11	4.665e-06	2.480e-10	1.480e-09	0.000e+00	2.796e-10	4.155e-09	1.858e-12	1.379e-10	0.000e+00
IBD	Mean	1.771e-04 +	1.868e-04 +	1.746e-04 +	1.746e-04 +	1.746e-04 +	1.746e-04 +	1.746e-04 +	1.961e-04 +	1.746e-04 +	1.814e-04 +	1.990e-04 +	1.746e-04 +	1.785e-04 +	1.746e-04 +	1.746e-04 +	**1.746e-04**
	SD	3.638e-06	4.466e-06	2.661e-11	1.680e-14	2.097e-09	1.301e-10	1.941e-10	1.603e-05	4.757e-10	1.852e-06	5.129e-05	1.485e-09	2.095e-05	6.014e-12	4.345e-09	1.355e-19
	Best	1.749e-04	1.798e-04	1.746e-04	1.746e-04	1.746e-04	1.746e-04	1.746e-04	1.761e-04	1.746e-04	1.777e-04	1.746e-04	1.746e-04	1.746e-04	1.746e-04	1.746e-04	1.746e-04
	Worst	1.932e-04	1.964e-04	1.746e-04	1.746e-04	1.746e-04	1.746e-04	1.746e-04	2.283e-04	1.746e-04	1.856e-04	3.733e-04	1.746e-04	2.913e-04	1.746e-04	1.746e-04	1.746e-04
PVP	Mean	9.244e+03 +	1.647e+04 +	6.577e+03 ≈	8.336e+03 +	2.999e+03 −	3.227e+03 −	**2.968e+03** −	3.251e+04 +	7.616e+03 +	8.619e+03 +	3.555e+04 +	7.662e+03 +	9.452e+03 +	7.105e+03 +	7.720e+03 +	7.064e+03
	SD	2.137e+03	3.094e+03	2.373e+03	1.893e+00	6.178e+01	1.051e+03	5.466e+01	1.706e+04	1.846e+03	1.511e+03	6.295e+04	1.690e+03	4.292e+03	2.326e+03	2.185e+03	2.301e+03
	Best	3.793e+03	1.177e+04	2.899e+03	8.334e+03	2.904e+03	2.912e+03	2.898e+03	1.265e+04	2.897e+03	3.767e+03	2.904e+03	2.899e+03	2.941e+03	2.893e+03	2.900e+03	2.892e+03
	Worst	1.362e+04	2.303e+04	8.335e+03	8.341e+03	3.141e+03	8.475e+03	3.119e+03	9.530e+04	8.351e+03	1.005e+04	2.669e+05	8.404e+03	1.501e+04	9.091e+03	1.006e+04	8.335e+03
SRD	Mean	2.995e+03 +	3.156e+03 +	2.991e+03 +	2.990e+03 +	2.995e+03 +	2.988e+03 +	2.989e+03 +	6.226e+03 +	3.011e+03 +	3.269e+03 +	3.785e+03 +	3.033e+03 +	8.809e+03 +	2.998e+03 +	3.003e+03 +	**2.987e+03**
	SD	4.492e+00	5.313e+01	1.774e+00	9.769e-01	2.773e+00	5.682e-01	7.889e-01	1.327e+04	4.713e+00	1.115e+02	8.245e+02	1.246e+01	2.587e+04	1.878e+01	8.867e+00	1.847e-01
	Best	2.988e+03	3.030e+03	2.988e+03	2.988e+03	2.990e+03	2.987e+03	2.988e+03	3.128e+03	3.003e+03	3.057e+03	2.994e+03	3.008e+03	3.014e+03	2.987e+03	2.989e+03	2.987e+03
	Worst	3.006e+03	3.257e+03	2.997e+03	2.992e+03	3.003e+03	2.989e+03	2.991e+03	7.764e+04	3.025e+03	3.525e+03	5.828e+03	3.060e+03	1.480e+05	3.088e+03	3.026e+03	2.988e+03
TCD	Mean	3.071e+01 +	3.035e+01 +	3.015e+01 ≈	**3.015e+01** −	3.016e+01 +	3.016e+01 +	3.016e+01 +	3.072e+01 +	3.016e+01 +	3.020e+01 +	3.107e+01 +	3.018e+01 +	3.083e+01 +	3.015e+01 +	3.043e+01 +	3.015e+01
	SD	7.337e-01	9.032e-02	8.503e-05	3.461e-05	6.910e-03	5.405e-03	4.300e-03	3.375e-01	5.315e-03	3.227e-02	1.262e+00	2.029e-02	7.091e-01	2.571e-03	3.068e-01	3.541e-03
	Best	3.017e+01	3.019e+01	3.015e+01	3.015e+01	3.015e+01	3.015e+01	3.015e+01	3.021e+01	3.015e+01	3.016e+01	3.015e+01	3.016e+01	3.023e+01	3.015e+01	3.017e+01	3.015e+01
	Worst	3.355e+01	3.059e+01	3.015e+01	3.015e+01	3.018e+01	3.018e+01	3.017e+01	3.175e+01	3.017e+01	3.027e+01	3.580e+01	3.024e+01	3.343e+01	3.016e+01	3.161e+01	3.017e+01
+/≈/−		8/0/0	8/0/0	6/2/0	7/0/1	7/0/1	7/0/1	7/0/1	8/0/0	8/0/0	8/0/0	7/1/0	8/0/0	8/0/0	8/0/0	8/0/0	–
Ave. rank		12.1	13.6	6.1	5.6	7.2	4.7	4.6	15.1	6.1	10.6	13.1	8.3	12.3	5.6	8.6	**1.8**

5.2 Advantages of DEA^{2}H^2

This section analyzes the advantages of DEA^{2}H^2 from two aspects: the LLHs design and the high-level component design.

Advantages of the LLHs design: This study introduces ten efficient and easy-implemented LLHs for constructing a versatile and effective LLHs pool capable of addressing various optimization tasks. LLHs play a crucial role in guiding the optimization process towards promising regions of the search space, and therefore, the design of the LLHs pool significantly influences the algorithm's ability to find high-quality solutions. Each search operator provides unique advantages that contribute to the overall efficacy of the optimization process. For instance, local search with a uniform distribution excels in exploitative search, systematically refining solutions in the vicinity of promising regions. On the other hand, Lévy flight, known for its ability to generate long-range jumps, facilitates the escape from local optima, preventing premature convergence. Additionally, the DE/rand/1 operator is recognized for its superior explorative behavior, allowing the algorithm to efficiently explore the search space and discover diverse solution alternatives. Incorporating these high-quality LLHs into the DEA^{2}H^2 framework enhances its robustness and adaptability across diverse optimization challenges. The diverse nature of the LLHs pool reduces the algorithm's sensitivity to specific problem features or search space structures, mitigating the risk of premature convergence or stagnation in suboptimal solutions. This enhanced robustness not only improves the algorithm's reliability but also enables it to maintain competitive performance across a wide range of optimization scenarios.

Advantages of the high-level component design: Motivated by the success-history parameter adaptation strategy employed in SHADE, we introduce a novel enhancement to DEA^{2}H^2. Our enhancement involves the incorporation of a success-history memory mechanism as a high-level component within DEA^{2}H^2. This mechanism aims to improve the algorithm's adaptability and performance by leveraging past evolutionary successes to guide the optimization process. Specifically, the success-history memory tracks the evolutionary progress by recording the performance of parent individuals in generating offspring with improved fitness values. When a parent individual successfully produces an offspring with a superior fitness value, this evolutionary event is considered a success. In such cases, the corresponding Local Low-Level Heuristic (LLH) responsible for generating the successful offspring is retained for future use. Conversely, if the offspring fails to exhibit improved fitness, the LLH is replaced with a randomly selected alternative.

The concept of success-history memory has been previously explored in the context of parameter adaptation strategies, particularly in the SHADE algorithm proposed by Tanabe and Fukunaga [21]. In SHADE, the success-history memory mechanism is utilized to dynamically adjust the scaling factor (F) and crossover rate (Cr) based on past evolutionary successes. This adaptive mechanism enables SHADE to drive the evolution of these parameters towards problem-specific optima, thereby enhancing the algorithm's performance across diverse optimization tasks. Similarly, we hypothesize that integrating the success-history memory mechanism into DEA^{2}H^2 can effectively guide the evolution of the optimization sequence towards promising regions of the search space. By leveraging past successes to inform the selection of LLHs, our proposed approach enhances the algorithm's ability to adapt to the problem's characteristics and navigate complex optimization landscapes.

5.3 Performance analysis on CEC2020 benchmark functions

The CEC2020 benchmark suite, encompassing ten well-established functions with diverse characteristics and dimensions, serves as a challenging platform for a comprehensive evaluation of optimization algorithms. Across this suite, DEA^{2}H^2 consistently outperforms competitor algorithms in various scenarios, demonstrating the outstanding scalability and robustness of DEA^{2}H^2.

DEA^{2}H^2 demonstrates remarkable exploitation ability across both unimodal function f_1 and multimodal functions f_2 to f_4. This remarkable performance is attributed to the inclusion of efficient local search operators within the LLHs pool. These operators enable DEA^{2}H^2 to effectively refine solutions in the vicinity of promising regions, thereby facilitating the exploitation of local optima. Moreover, the intelligent success-history-based high-level component plays a pivotal role in guiding decision-making processes, constructing the optimization sequence intelligently and automatically, further enhancing the algorithm's ability to exploit promising regions of the search space.

In the case of hybrid functions from f_5 to f_8, characterized by complex fitness landscape characteristics, DEA^{2}H^2 significantly outperforms competitor algorithms across all instances. These functions pose challenges in terms of exploration and escaping from local optima due to their intricate nature. From the experimental results and statistical analyses, DEA^{2}H^2 achieves a delicate balance between exploitation and exploration, which is crucial for addressing optimization challenges effectively. This balance is primarily attributed to the integration of versatile search operators into its LLHs and the effective design of

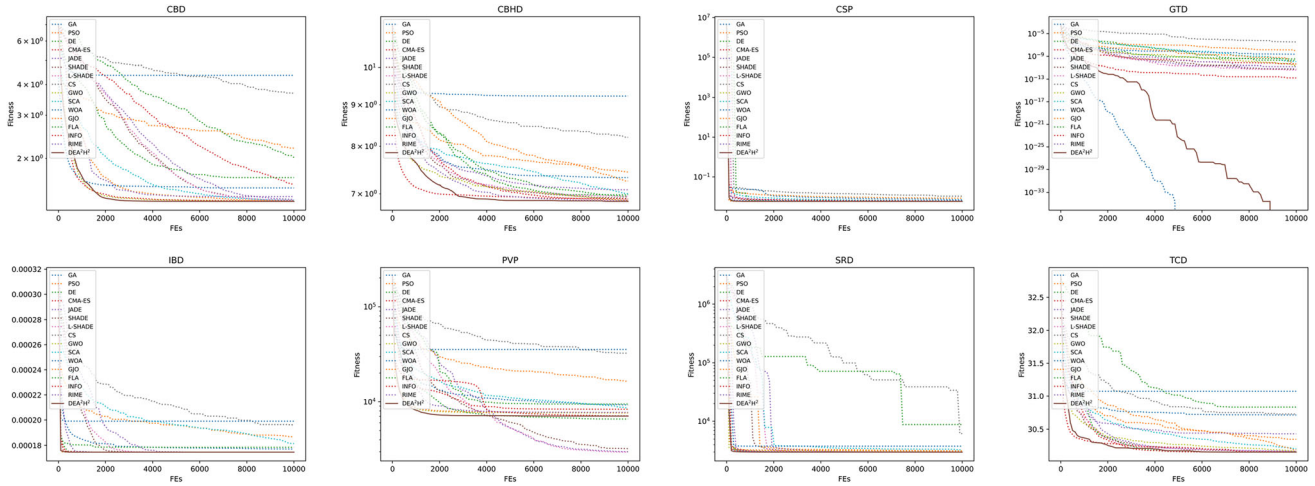


Fig. 14 Convergence curves on engineering problems

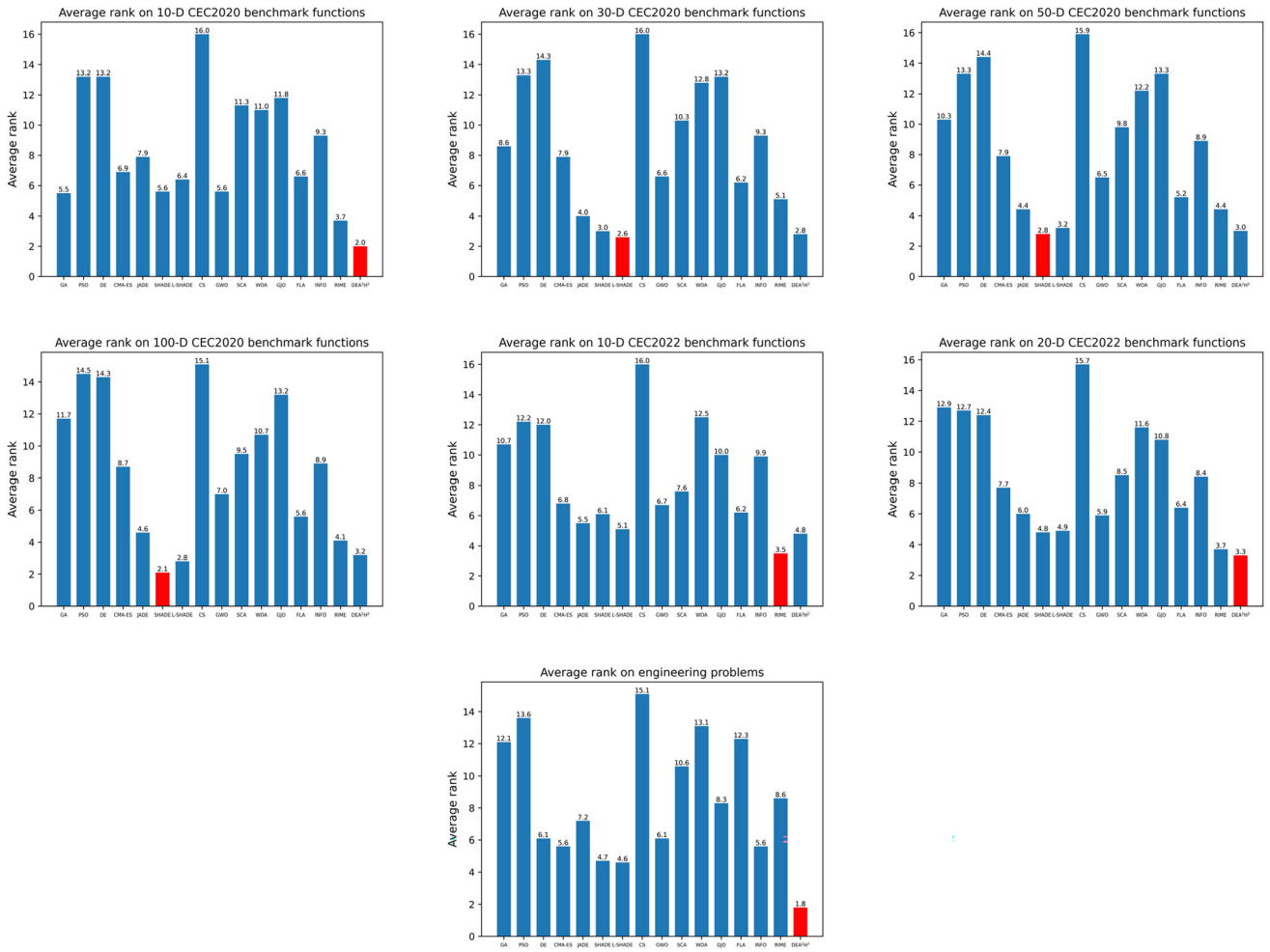


Fig. 15 the average ranks of optimizers in various benchmarks

the success-history-based high-level component. By leveraging the strengths of these components, DEA^{2}H^2 navigates the trade-off between the exploitation of known promising regions and the exploration of unexplored areas, thereby achieving competitive performance across diverse optimization scenarios.

Table 11 Results of the ablation experiments on 30-D and 50-D CEC2020 benchmark functions

Func		30-D		50-D	
		DEA^2H^2	DEA^2H^2-rand	DEA^2H^2	DEA^2H^2-rand
f_1	Mean	**3.77e+03**	3.76e+06 +	**1.37e+04**	1.72e+07 +
	SD	4.78e+03	2.20e+06	2.49e+04	7.09e+06
f_2	Mean	**7.01e+05**	3.90e+08 +	**2.25e+06**	2.42e+09 +
	SD	6.86e+05	2.90e+08	4.95e+06	1.35e+09
f_3	Mean	**2.79e+05**	1.21e+08 +	**1.46e+06**	6.54e+08 +
	SD	3.21e+05	5.21e+07	5.61e+06	2.12e+08
f_4	Mean	**1.92e+03**	1.94e+03 +	**1.96e+03**	1.97e+03 +
	SD	7.85e+00	1.34e+01	1.67e+01	1.82e+01
f_5	Mean	**6.80e+04**	3.37e+05 +	**1.35e+05**	1.27e+06 +
	SD	2.60e+04	2.25e+05	6.51e+04	6.73e+05
f_6	Mean	**3.11e+03**	1.80e+04 +	**3.58e+03**	2.69e+04 +
	SD	1.80e+03	1.01e+04	3.42e+03	1.26e+04
f_7	Mean	**3.47e+04**	4.31e+05 +	**1.14e+05**	1.59e+06 +
	SD	2.30e+04	2.46e+05	4.09e+04	1.62e+06
f_8	Mean	**2.43e+03**	2.47e+03 +	**2.71e+03**	2.87e+03 +
	SD	1.86e+01	3.64e+01	3.62e+02	1.91e+02
f_9	Mean	**2.64e+03**	2.79e+03 +	**2.85e+03**	3.11e+03 +
	SD	1.47e+02	1.28e+02	5.27e+02	3.40e+02
f_{10}	Mean	**2.93e+03**	2.95e+03 +	**3.33e+03**	3.58e+03 +
	SD	1.58e+01	1.93e+01	6.48e+01	1.77e+02
+/≈/−		–	10/0/0	–	10/0/0
Ave. rank		**1.0**	2.0	**1.0**	2.0

While the DEA^2H^2 demonstrates commendable performance on the aforementioned functions, its efficacy on the remaining composite functions is not satisfactory. In contrast, other DE-derived optimizers, such as SHADE and L-SHADE, exhibit significantly superior performance on these functions. This observation prompts us to explore potential factors contributing to this discrepancy in performance. We hypothesize that the performance difference observed between DEA^2H^2 and other DE-derived optimizers on composite functions may be attributed to the intricacies of the parameters within these functions. While DEA^2H^2 focuses on determining the optimization sequence, the performance of composite functions may be heavily influenced by the specific configurations of parameters embedded within them. These parameters are meticulously designed to reflect complex problem characteristics and can significantly impact the optimization process. In light of this observation, we recognize the importance of investigating the role of meticulously designed parameters in composite functions and their impact on optimization algorithms' performance. In our future research endeavors, we plan to conduct numerical experiments to empirically validate this hypothesis. By systematically analyzing the influence of various parameter configurations on algorithm performance, we aim to gain deeper insights into the underlying factors contributing to the effectiveness of optimization algorithms on composite functions.

As the problem dimensionality increases, a notable deficiency of DEA^2H^2 compared to other DE-derived optimizers becomes apparent. This challenge is primarily attributed to the curse of dimensionality, a well-established concept in optimization theory [43]. The curse of dimensionality asserts that as the dimensionality of the problem increases linearly, the search space expands exponentially. Consequently, efficiently exploring and evaluating the entire solution space becomes increasingly challenging, leading to prolonged optimization times and potentially prohibitive computational costs. As discussed earlier, we hypothesize that well-designed parameters also play a crucial role in addressing challenges posed by high-dimensional search spaces. In the context of DEA^2H^2, the effectiveness of parameter configurations becomes particularly significant as the problem dimensionality increases. Well-designed parameters can facilitate more effective exploration and exploitation strategies, thereby mitigating the adverse effects of the curse of dimensionality on optimization performance. In our future research endeavors, we aim to address the challenge posed by high-dimensional optimization problems in DEA^2H^2 by introducing advanced parameter adaptation schemes [44, 45] and leveraging cooperative coevolution frameworks [46, 47]. These adaptations are envisioned to extend the algorithm's capabilities and enhance its performance in high-dimensional search spaces. Parameter adaptation schemes will enable the algorithm to dynamically adjust its parameters based on the characteristics of the optimization landscape, thereby improving its adaptability and robustness. Additionally, leveraging cooperative coevolution frameworks will facilitate the decomposition of high-dimensional problems into smaller, more manageable subcomponents, allowing for more efficient exploration and exploitation of the solution space. By introducing these adaptations, we anticipate significant improvements in DEA^2H^2's performance and scalability, enabling it to tackle more complex optimization tasks efficiently. These advancements will not only enhance the algorithm's applicability to real-world problems but also contribute to the broader understanding of optimization algorithms in high-dimensional settings.

5.4 Performance analysis on CEC2022 benchmark functions

The CEC2022 benchmark suite represents one of the latest and most comprehensive collections of benchmark functions, encompassing twelve well-established functions. This suite offers a rigorous evaluation platform for assessing the performance of optimization algorithms across various problem domains and complexities. DEA^2H^2 exhibits high competitiveness across the diverse set of benchmark functions included in the CEC2022 suite. Its robustness and scalability are evident in its consistent performance across these functions, highlighting its potential as a reliable optimization tool. While we won't delve into the superior performance of DEA^2H^2 on the CEC2022 benchmarks here, it's worth highlighting a common deficiency observed in composite functions f_{10} to f_{12}. This deficiency has been also found in the CEC2020 benchmark suite, suggesting that the inherent structure of DEA^2H^2 may not be optimally suited for effectively handling composite functions. The recurrence of this deficiency across multiple benchmark suites underscores the significance of further investigation into this limitation.

5.5 Performance analysis on engineering problems

The eight engineering optimization problems selected for this study are renowned mathematical models that offer valuable insights into the performance of optimization algorithms in real-world scenarios. These problems are carefully chosen to reflect diverse challenges encountered in engineering design and simulation, thereby providing a comprehensive evaluation platform for assessing the efficacy of optimization algorithms. In real-world applications, stability and robustness are paramount considerations for ensuring the reliability and effectiveness of optimization algorithms. To provide a comprehensive assessment, we present both the best and worst optima observed across 30 trial runs for each engineering optimization problem. This approach allows for a thorough examination of the algorithm's performance under varying conditions and highlights its stability and robustness in practical scenarios. Through rigorous experimental evaluations and statistical analyses, our proposed DEA^2H^2 algorithm showcases strong scalability and applicability in addressing real-world simulation problems. This algorithm consistently delivers competitive results across the diverse set of engineering optimization problems considered in this study, demonstrating its versatility and effectiveness in handling complex optimization challenges.

As a high-level, flexible, and easily implemented approach, DEA^2H^2 holds promise for effectively tackling real-world optimization challenges encountered in various practical applications. Its robust performance, coupled with its scalability and applicability across diverse problem domains, positions DEA^2H^2 as a viable solution for addressing complex optimization problems in engineering design, simulation, and other practical domains.

5.6 Performance analysis on ablation experiments

In our study, we designed and executed ablation experiments on 30-dimensional and 50-dimensional CEC2020 benchmark functions. These experiments were crafted to rigorously evaluate the efficacy of the success-history-based high-level component in isolation, aiming to shed light on its ability to intelligently construct the sequence of heuristics. Through meticulous analysis of the experimental results, complemented by thorough statistical analyses, we obtained unequivocal confirmation of our hypothesis. The results, coupled with extensive statistical analyses, underscored the pivotal role of success-history information in guiding the construction of the optimization sequence. These findings validate the effectiveness of the success-history-based high-level component. By incorporating historical success information, the algorithm can intelligently adapt and refine its optimization strategy, leading to more efficient exploration and exploitation of the solution space.

6 Conclusion

This paper introduces a novel DE-architecture-based adaptive hyper-heuristic algorithm named DEA^2H^2, designed specifically for continuous optimization tasks. DEA^2H^2 incorporates ten efficient search operators categorized into local search, differential mutation, and crossover operators. The integration of versatile search operators ensures the algorithm's general scalability and robustness across diverse optimization tasks.

Inspired by the parameter adaptation strategy of scaling factor F and crossover rate Cr in SHADE, we introduce a success-history-based high-level component in DEA^2H^2 to construct the optimization sequence. This component preserves a specific search operator if a parent individual successfully evolves an offspring individual using it; otherwise, the corresponding search operator in success-history memory will be replaced randomly. The effectiveness of this simple yet efficient component is

independently confirmed through ablation experiments conducted in this study.

In comprehensive numerical experiments involving CEC2020, CEC2022 benchmark functions, and eight engineering problems, DEA^{2}H^2 is evaluated against fifteen well-known MAs. The experimental results and statistical analyses validate the efficiency and robustness of DEA^{2}H^2 across diverse optimization tasks. However, we also identify deficiencies in DEA^{2}H^2 regarding composite functions and high-dimensional search spaces. We hypothesize that a better parameter adaptation strategy may be more crucial than optimization sequence construction in addressing these challenges.

Therefore, in future research, we aim to further investigate the components of DEA^{2}H^2 and enhance its performance by integrating advanced parameter adaptation strategies and cooperative coevolution frameworks. These enhancements will contribute to the algorithm's capability to tackle composite functions and high-dimensional search spaces more effectively.

Acknowledgements This work was supported by JST SPRING Grant Number JPMJSP2119.

Author contributions RZ: conceptualization, methodology, investigation, writing—original draft, writing—review & editing, and funding acquisition. JY: investigation, methodology, formal analysis, writing—review & editing, and project administration.

Data availability The source code of this research can be downloaded from https://github.com/RuiZhong961230/DEA2H2.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

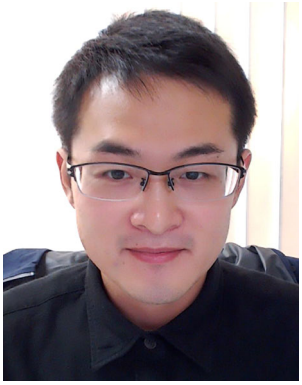
1. Deng, L., Liu, S.: A multi-strategy improved slime mould algorithm for global optimization and engineering design problems. Comput. Methods Appl. Mech. Eng. **404**, 115764 (2023). https://doi.org/10.1016/j.cma.2022.115764
2. Shoukat, R., Xiaoqiang, Z.: Upstream logistics optimization from Shanghai, China to Kasur, Pakistan: an implementation of mixed-integer linear programming. Transp. Res. Rec. **2678**(1), 539–554 (2024). https://doi.org/10.1177/03611981231171157
3. Zhao, S., Zhang, T., Cai, L., Yang, R.: Triangulation topology aggregation optimizer: a novel mathematics-based meta-heuristic algorithm for continuous optimization and engineering applications. Expert Syst. Appl. **238**, 121744 (2024). https://doi.org/10.1016/j.eswa.2023.121744
4. Deng, L., Liu, S.: Snow ablation optimizer: a novel metaheuristic technique for numerical optimization and engineering design. Expert Syst. Appl. **225**, 120069 (2023). https://doi.org/10.1016/j.eswa.2023.120069
5. Zhong, R., Peng, F., Yu, J., Munetomo, M.: Q-learning based vegetation evolution for numerical optimization and wireless sensor network coverage optimization. Alex. Eng. J. **87**, 148–163 (2024). https://doi.org/10.1016/j.aej.2023.12.028
6. Deng, L., Liu, S.: An enhanced slime mould algorithm based on adaptive grouping technique for global optimization. Expert Syst. Appl. **222**, 119877 (2023). https://doi.org/10.1016/j.eswa.2023.119877
7. Deng, L., Liu, S.: Incorporating q-learning and gradient search scheme into jaya algorithm for global optimization. Artif. Intell. Rev. (2023). https://doi.org/10.1007/s10462-023-10613-1
8. Sörensen, K.: Metaheuristics—the metaphor exposed. Int. Trans. Oper. Res. (2013). https://doi.org/10.1111/itor.12001
9. Aranha, C., Villalón, C., Campelo, F., Dorigo, M., Ruiz, R., Sevaux, M., Sörensen, K., Stützle, T.: Metaphor-based metaheuristics, a call for action: the elephant in the room. Swarm Intell. **16**, 1–6 (2021). https://doi.org/10.1007/s11721-021-00202-9
10. Camacho Villalón, C.L., Stützle, T., Dorigo, M.: Grey wolf, firefly and bat algorithms: Three widespread algorithms that do not contain any novelty. In: Swarm Intelligence, pp. 121–133. Springer, Cham (2020)
11. Weyland, D.: A rigorous analysis of the harmony search algorithm: How the research community can be misled by a "novel'' methodology. Int. J. Appl. Metaheuristic Comput. **1**(2), 50–60 (2010). https://doi.org/10.4018/jamc.2010040104
12. Camacho, C., Dorigo, M., Stützle, T.: The intelligent water drops algorithm: why it cannot be considered a novel algorithm: a brief discussion on the use of metaphors in optimization. Swarm Intell. (2019). https://doi.org/10.1007/s11721-019-00165-y
13. Zhong, R., Yu, J., Chao, Z., Munetomo, M.: Surrogate ensemble-assisted hyper-heuristic algorithm for expensive optimization problems. Int. J. Comput. Intell. Syst. (2023). https://doi.org/10.1007/s44196-023-00346-y
14. Zhao, F., Liu, Y., Zhu, N., Xu, T.: Jonrinaldi: a selection hyper-heuristic algorithm with q-learning mechanism. Appl. Soft Comput. **147**, 110815 (2023). https://doi.org/10.1016/j.asoc.2023.110815
15. Kelvin Ching Wei Lim, L.-P.W., Chin, J.F.: Simulated-annealing-based hyper-heuristic for flexible job-shop scheduling. Eng. Optim. **55**(10), 1635–1651 (2023). https://doi.org/10.1080/0305215X.2022.2106477
16. Zhao, F., Di, S., Cao, J., Tang, J.: Jonrinaldi: a novel cooperative multi-stage hyper-heuristic for combination optimization problems. Complex Syst. Model. Simul. **1**(2), 91–108 (2021). https://doi.org/10.23919/CSMS.2021.0010
17. Zhang, Q., Gao, H., Zhan, Z.-H., Li, J., Zhang, H.: Growth optimizer: a powerful metaheuristic algorithm for solving continuous and discrete global optimization problems. Knowl.-Based Syst. **261**, 110206 (2023). https://doi.org/10.1016/j.knosys.2022.110206
18. Qin, W., Zhuang, Z., Huang, Z., Huang, H.: A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem. Comput. Ind. Eng. **156**, 107252 (2021). https://doi.org/10.1016/j.cie.2021.107252
19. Tapia-Avitia, J.M., Cruz-Duarte, J.M., Amaya, I., Ortiz-Bayliss, J.C., Terashima-Marin, H., Pillay, N.: A primary study on hyper-heuristics powered by artificial neural networks for customising population-based metaheuristics in continuous optimisation problems. In: 2022 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2022). https://doi.org/10.1109/CEC55065.2022.9870275
20. Zhong, R., Zhang, E., Munetomo, M.: Evolutionary multi-mode slime mold optimization: a hyper-heuristic algorithm inspired by slime mold foraging behaviors. J. Supercomput. (2024). https://doi.org/10.1007/s11227-024-05909-0

21. Tanabe, R., Fukunaga, A.: Success-history based parameter adaptation for differential evolution. In: 2013 IEEE Congress on Evolutionary Computation, pp. 71–78 (2013). https://doi.org/10.1109/CEC.2013.6557555
22. Choong, S.S., Wong, L.-P., Lim, C.P.: Automatic design of hyper-heuristic based on reinforcement learning. Inf. Sci. **436–437**, 89–107 (2018). https://doi.org/10.1016/j.ins.2018.01.005
23. Meng, Z., Chen, Y.: Differential evolution with exponential crossover can be also competitive on numerical optimization. Appl. Soft Comput. **146**, 110750 (2023). https://doi.org/10.1016/j.asoc.2023.110750
24. Nguyen, T.: A framework of optimization functions using Numpy (OpFuNu) for optimization problems. Zenodo (2020). https://doi.org/10.5281/zenodo.3620960
25. Thieu, N.V.: ENOPPY: a Python library for engineering optimization problems. Zenodo (2023). https://doi.org/10.5281/zenodo.7953206
26. Bayzidi, H., Talatahari, S., Saraee, M., Lamarche, C.-P.: Social network search for solving engineering optimization problems. Comput. Intell. Neurosci. **2021**, 1–32 (2021). https://doi.org/10.1155/2021/8548639
27. Srinivas, M., Patnaik, L.M.: Genetic algorithms: a survey. Computer **27**(6), 17–26 (1994). https://doi.org/10.1109/2.294849
28. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, pp. 1942–19484 (1995). https://doi.org/10.1109/ICNN.1995.488968
29. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**, 341–359 (1997). https://doi.org/10.1023/A:1008202821328
30. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evol. Comput. **9**(2), 159–195 (2001). https://doi.org/10.1162/106365601750190398
31. Zhang, J., Sanderson, A.C.: Jade: adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput. **13**(5), 945–958 (2009). https://doi.org/10.1109/TEVC.2009.2014613
32. Tanabe, R., Fukunaga, A.S.: Improving the search performance of shade using linear population size reduction. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 1658–1665 (2014). https://doi.org/10.1109/CEC.2014.6900380
33. Yang, X.-S., Deb, S.: Cuckoo search via lévy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 210–214 (2009). https://doi.org/10.1109/NABIC.2009.5393690
34. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014). https://doi.org/10.1016/j.advengsoft.2013.12.007
35. Mirjalili, S.: Sca: a sine cosine algorithm for solving optimization problems. Knowl.-Based Syst. **96**, 120–133 (2016). https://doi.org/10.1016/j.knosys.2015.12.022
36. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Adv. Eng. Softw. **95**, 51–67 (2016). https://doi.org/10.1016/j.advengsoft.2016.01.008
37. Chopra, N., Mohsin Ansari, M.: Golden jackal optimization: a novel nature-inspired optimizer for engineering applications. Expert Syst. Appl. **198**, 116924 (2022). https://doi.org/10.1016/j.eswa.2022.116924
38. Hashim, F.A., Mostafa, R.R., Hussien, A.G., Mirjalili, S., Sallam, K.M.: Fick's law algorithm: a physical law-based algorithm for numerical optimization. Knowl.-Based Syst. **260**, 110146 (2023). https://doi.org/10.1016/j.knosys.2022.110146
39. Ahmadianfar, I., Heidari, A.A., Noshadian, S., Chen, H., Gandomi, A.H.: Info: an efficient optimization algorithm based on weighted mean of vectors. Expert Syst. Appl. **195**, 116516 (2022). https://doi.org/10.1016/j.eswa.2022.116516
40. Su, H., Zhao, D., Heidari, A.A., Liu, L., Zhang, X., Mafarja, M., Chen, H.: Rime: a physics-based optimization. Neurocomputing **532**, 183–214 (2023). https://doi.org/10.1016/j.neucom.2023.02.010
41. Holm, S.: A simple sequentially rejective multiple test procedure. Scand. J. Stat. **6**(2), 65–70 (1979)
42. Jackson, W.G., Özcan, E., Drake, J.H.: Late acceptance-based selection hyper-heuristics for cross-domain heuristic search. In: 2013 13th UK Workshop on Computational Intelligence (UKCI), pp. 228–235 (2013). https://doi.org/10.1109/UKCI.2013.6651310
43. Köppen, M.: The curse of dimensionality. In: 5th Online World Conference on Soft Computing in Industrial Applications (WSC5), vol. 1, pp. 4–8 (2000)
44. Gu, Q., Li, S., Liao, Z.: Solving nonlinear equation systems based on evolutionary multitasking with neighborhood-based speciation differential evolution. Expert Syst. Appl. **238**, 122025 (2024). https://doi.org/10.1016/j.eswa.2023.122025
45. Layeb, A.: Differential evolution algorithms with novel mutations, adaptive parameters, and weibull flight operator. Soft. Comput. (2024). https://doi.org/10.1007/s00500-023-09561-3
46. Zhong, R., Zhang, E., Munetomo, M.: Cooperative coevolutionary differential evolution with linkage measurement minimization for large-scale optimization problems in noisy environments. Complex Intell. Syst. **9**, 4439–4456 (2023). https://doi.org/10.1007/s40747-022-00957-6
47. Bull, L., Liu, H.: On cooperative coevolution and global crossover. IEEE Trans. Evol. Comput. (2024). https://doi.org/10.1109/TEVC.2024.3355776

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Rui Zhong received a Bachelor degree from Huazhong Agricultural University, China in 2019, and a Master degree from Kyushu University, Japan in 2022. He is currently pursuing a Ph.D. at Hokkaido University, Japan. His research interests include evolutionary computation, large-scale global optimization, and meta/hyperheuristics.



Jun Yu received a Bachelor degree from Northeastern University, China in 2014, and a Master degree and a doctorate from Kyushu University, Japan in 2017 and 2019, respectively. He is currently an Assistant Professor at Niigata University, Japan. His research interests include evolutionary computation, artificial neural networks, and machine learning.