

Full length article

LLMOA: A novel large language model assisted hyper-heuristic optimization algorithm

Rui Zhong ^a, Abdelazim G. Hussien ^{b,c}, Jun Yu ^d, Masaharu Munetomo ^a

^a Information Initiative Center, Hokkaido University, Sapporo, Japan

^b Department of Computer and Information Science, Linköping University, Linköping, Sweden

^c Faculty of Science, Fayoum University, Fayoum, Egypt

^d Institute of Science and Technology, Niigata University, Niigata, Japan

ARTICLE INFO

Dataset link: <https://github.com/RuiZhong961230/LLMOA>

Keywords:

Hyper-heuristic algorithm (HHA)
Large language model (LLM)
Prompt engineering
Low-level heuristics (LLHs)
Evolutionary computation (EC)

ABSTRACT

This work presents a novel approach, the large language model assisted hyper-heuristic optimization algorithm (LLMOA), tailored to address complex optimization challenges. Comprising two essential components – the high-level component and the low-level component – LLMOA leverages the LLM (i.e., Gemini) with prompt engineering in its high-level component to construct optimization sequences automatically and intelligently. Furthermore, we propose novel elite-based local search operators as low-level heuristics (LLHs), which draw inspiration from the proximate optimality principle (POP). These local search operators cooperated with well-known mutation and crossover operators from differential evolution (DE), at a total of ten efficient and versatile search operators, forming the whole LLHs. To assess the competitiveness of LLMOA, we conducted comprehensive numerical experiments across CEC2014, CEC2020, CEC2022, and ten engineering optimization problems, benchmarking against eleven state-of-the-art optimizers. Our experimental findings and statistical analyses underscore the powerfulness and effectiveness of LLMOA. Moreover, ablation experiments reveal the pivotal role of integrating the LLM Gemini and prompt engineering as the high-level component. Conclusively, this study provides a feasible avenue to introduce LLM to the evolutionary computation (EC) community. The research's source code is available for download at <https://github.com/RuiZhong961230/LLMOA>.

1. Introduction

In recent decades, evolutionary computation (EC) has emerged as a prominent optimization paradigm, garnering significant attention from scholars and researchers [1,2]. EC offers a stochastic optimization toolkit inspired by the principles of biological evolution and provides a robust approach to addressing intricate optimization problems. Motivated by the fundamental concept of survival of the fittest, EC techniques simulate the iterative process of natural selection and continually refining candidate solutions [3,4]. These techniques aim to converge towards optimal or near-optimal solutions within expansive search spaces.

As the EC community experiences rapid growth, researchers are broadening their horizons to draw inspiration from a diverse array of organisms and phenomena. Examples include the grey wolf [5], gorilla [6], slime mold [7], fire hawk [8], crayfish [9], dung beetle [10], walrus [11], rime-ice [12], snow ablation [13], Fick's law [14], multiplayer battle game [15], among others. However, criticisms have

arisen regarding the barbaric growth of metaphor-based optimization techniques. Aranha et al. [16] and Christian et al. [17–19] argue against what they perceive as the excessive reliance on metaphorical frameworks. They contend that many of these purportedly “novel” algorithms merely repackage existing search operators under a new metaphor, rather than introducing innovative optimization techniques or paradigms. Furthermore, Claus and Felipe have developed a repository called EC Bestiary¹ as a response to this trend. They advocate for a shift towards a more mathematical approach in the literature, suggesting that the EC community should move beyond its current metaphor-rich period of development. Therefore, exploring alternative avenues for advancing the EC community and moving beyond the current metaphor-rich era is crucial.

Fortunately, the hyper-heuristic (HH) framework offers an advanced and high-level methodology for enhancing optimization accuracy through integrating existing search operators [20]. Unlike traditional

* Corresponding author at: Faculty of Science, Fayoum University, Fayoum, Egypt.

E-mail addresses: zhongrui@iic.hokudai.ac.jp (R. Zhong), abdelazim.hussien@liu.se (A.G. Hussien), yujun@ie.niigata-u.ac.jp (J. Yu), munetomo@iic.hokudai.ac.jp (M. Munetomo).

¹ <https://github.com/fcampelo/EC-Bestiary>

meta-heuristic algorithms (MHAs) which are typically designed by domain experts with fixed optimization patterns and sequences, the HH framework provides flexibility in optimizing these static sequences. For instance, differential evolution (DE) employs selection, mutation, and crossover operators [21,22], while particle swarm optimization (PSO) repeats velocity and location update operators [23,24]. However, from the perspective of the HH framework, these static optimization sequences can also be optimized: Daliri et al. [25] introduced a pioneering World algorithm, which leverages reinforcement learning techniques to generalize HH approaches. This innovative algorithm dynamically switches between exploration and exploitation phases, drawing from a well-established meta-heuristic pool. The decision-making process for strategy selection is guided by reward signals, continually updating the record in tackling the traveling salesman problem (TSP) across a substantial dataset of 10,000 real cities. Song et al. [26] proposed the hyper-heuristic memetic algorithm (HHMA), an effective solution for the distributed assembly permutation flowshop scheduling problem (DAPFSP) aimed at minimizing the maximum completion time. HHMA integrates the estimation of distribution algorithm (EDA) as its high-level component during the global search phase to identify promising product sequences for subsequent exploitation. Additionally, a suite of efficient search operators, collaborating under the critical-products-based referenced local search (CP-RLS) method, serve as low-level heuristics (LLHs). Each LLH is equipped with a simulated annealing acceptance criterion to mitigate premature convergence. Comparative experiments against state-of-the-art algorithms for the DAPFSP underscore the efficacy of HHMA. Zhang et al. [27] proposed the Q-learning-based hyper-heuristic evolutionary algorithm (QHHEA) tailored specifically for addressing distributed flexible job-shop scheduling problems with crane transportation (DFJSPC). This approach introduces a novel left-shift decoding scheme to enhance machine processing utilization and crane transportation resources. The Q-learning-based high-level component dynamically selects the most suitable low-level heuristic (LLH) from a pre-designed set, leveraging valuable information derived from LLH efficacy. Statistical analyses affirm the competitive performance of QHHEA when pitted against several state-of-the-art algorithms in solving DFJSPC. However, the majority of research efforts are focused on combinatorial optimization problems, leaving the continuous domain of being relatively neglected [28, 29].

Furthermore, the continuous advancements in artificial intelligence (AI) and natural language processing (NLP) have catalyzed the emergence of large language models (LLMs), promoting a new era of innovation across various domains, spanning from text generation to complex problem-solving. Notable among these LLMs are Bing Chat [30], LLaMA [31], ChatGPT [32], and Gemini [33]. Their proliferation has significantly impacted diverse fields including art, healthcare, medicine, and beyond, showcasing remarkable successes [34–36]. Motivated by the unprecedented achievements of LLMs and recognizing the potential benefits of integrating LLMs into the EC community, this paper introduces a novel approach: the large language model assisted hyper-heuristic optimization algorithm (LLMOA). Leveraging the extensive knowledge encapsulated within pre-trained LLMs, we design ten effective and easily implementable search operators, termed LLHs. To facilitate this integration, we employ Gemini, a renowned LLM, as the high-level component of LLMOA. Through meticulous prompt engineering and design, this approach aims to transfer the linguistic capacity of LLMs to enhance the performance and adaptability of hyper-heuristic algorithm (HHA) in solving complex optimization problems. Specifically, the main contribution of this paper is summarized as follows:

- We propose a large language model assisted hyper-heuristic optimization algorithm (LLMOA) for continuous optimization. The designed LLHs module comprises ten effective search operators, and standard prompt engineering-guided Gemini is employed as the high-level component.

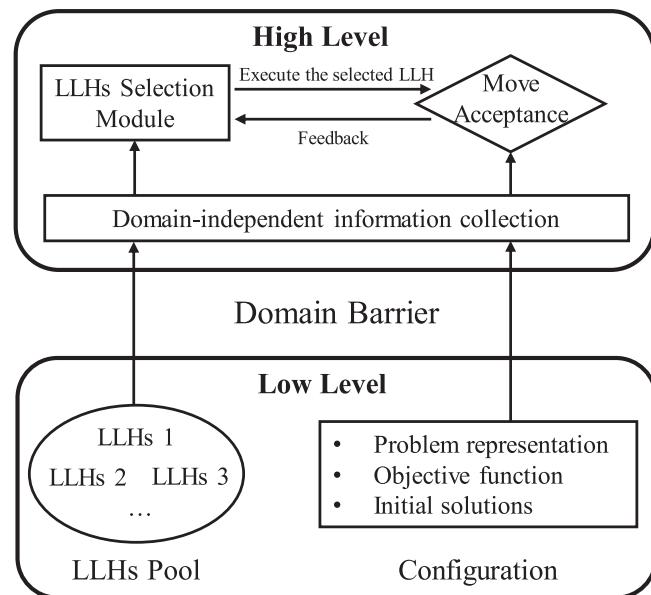


Fig. 1. A representative architecture of the HH framework [37,38].

- Comprehensive numerical experiments on CEC2014, CEC2020, CEC2022, and ten engineering optimization problems are conducted to evaluate the performance of LLMOA. Eleven state-of-the-art well-known optimizers are used as competitors.
- We conduct ablation experiments to investigate the effectiveness of introducing LLM independently.
- The experimental results and statistical analyses confirm the efficiency and effectiveness of our proposed LLMOA.

The remainder of this paper is organized as follows: Section 2 presents the related works including HHA, a brief review of LLM-assisted EC, and a literature review of HHA. Section 3 presents the details of our proposed LLMOA. The detailed numerical experiments outcomes as long as statistical analyses are demonstrated in Section 4, as well as the performance analyses. Finally, Section 5 concludes the paper.

2. Related works

2.1. Hyper-Heuristic Algorithm (HHA)

The hyper-heuristic optimization algorithm (HHA) is a high-level automatic optimization approach that instead of directly solving optimization problems, focuses on selecting or generating low-level heuristics (LLHs), and the selected or generated LLHs are employed to search for better individuals iteratively. The primary objective of the HHA is to identify or create sequences of LLHs that demonstrate exceptional robustness and scalability across various problem domains. Fig. 1 presents a general architecture of the HH framework.

A general HHA framework consists of 2 key components namely: the high-level & low-level components. The former component includes overarching mechanisms or strategies responsible for guiding the LLHs generation or selection. It offers a framework for decision-making and adjustment during the optimization process. Conversely, the latter component comprises LLHs applied to problem agents to produce solutions.

2.2. Literature review of LLM-assisted EC

The emergence of LLMs presents a promising avenue for advancing the EC community. By integrating LLMs with EC approaches, researchers can unlock new possibilities and capabilities for both LLMs

and EC approaches: Pluhacek et al. [39] leveraged GPT-4 to combine the advantages of cuckoo search (CS), whale optimization algorithm (WOA), particle swarm optimization (PSO), grey wolf optimizer (GWO), artificial bee colony (ABC), and self-organizing migrating algorithm (SOMA) to propose a novel hybrid algorithm for two distinct use-case scenarios. They primarily concluded that using LLMs to generate a hybrid approach is promising. LLMs successfully adhered to the instructions with minimal errors and generated functional code that aligns well with the algorithm descriptions and pseudo-codes. Zhong et al. [40] harnessed the remarkable capabilities of ChatGPT in text generation and reasoning to propose MA termed zoological search optimization (ZSO). This approach was achieved through the standard CRISPE prompt engineering framework (i.e., Role and Capacity, Personality, Statement, Insight, and Experiment). By conducting comprehensive experimental evaluations and rigorous statistical analyses against twenty expert-designed MHAs, their work revealed light on the potential of integrating LLMs into the EC community. Yang et al. [41] proposed Optimization by PROmpting (OPRO), a straightforward method for utilizing LLMs as optimizers, where the optimization task is described in natural language. At each optimization iteration, the LLM generates offspring solutions based on a prompt containing previously generated solutions along with their respective evaluations. Subsequently, these novel individuals are evaluated and incorporated into the prompt for subsequent iterations. The performance of OPRO was investigated in linear regression tasks and the traveling salesman problems (TSPs) and achieved improvements of up to 8% in GSM8K datasets and up to 50% in Big-Bench Hard tasks. Similarly, Liu et al. [42] pioneered the application of LLMs as an evolutionary combinatorial optimizer. The primary advantage of this approach lies in its remarkable efficiency, requiring minimal domain expertise and human intervention, while also eliminating the consumption of additional model training. Termed LLM-driven evolutionary algorithm (LMEA), their proposal directs the LLM to select parent solutions from the existing generation and executes mutation and crossover operations to generate offspring solutions in each iteration. Notably, LMEA incorporates a self-adaptation mechanism regulating the LLM's temperature, enabling a dynamic balance between exploration and exploitation. Encouragingly, LMEA demonstrated competitiveness with traditional heuristics in producing high-quality solutions for TSP instances featuring up to 20 nodes. For further exploration of LLM-assisted EC approaches, interested readers can refer to [43].

2.3. Evolution of HHA

HHA as a flexible optimization technique has received attention from many scholars and researchers. Daliri et al. [25] proposed a World HHA to solve continuous and discrete NP-hard optimization tasks. An infinite pool of meta-heuristics is designed that contains exploration and exploitation search operators, and a novel reinforcement learning approach is employed as the high-level component of the World algorithm. Which automatically switches between exploration and exploitation based on the reward and selection. Numerical experiments including traveling salesman problems (TSPs) confirm the state-of-the-art performance of the World algorithm. Tu et al. [44] presented a deep reinforcement learning-based HHA (DRLHHA) to handle online packing problems, where the information of real-time packing and the distributional information of random parameters of the problem are combined as the reward of the high-level component of DRLHHA. Simulation results validate that DRLHHA is significantly better than advanced approaches with 2%–19% improvements. Zhong et al. [45] inspired by the success history adaptation concept from success history adaptive DE (SHADE) that the specific search operator that successfully evolves individuals may have potential for other individuals. Based on this hypothesis, the DE architecture-based HHA (DEA^2H^2) is proposed to deal with numerical optimization challenges. The comprehensive

experiments in CEC2020 and CEC2022 against well-known optimization techniques demonstrate the competitiveness of DEA^2H^2 . Zhao et al. [46] introduced a Q-learning selected HHA (QLSHH) for complex optimization, where the Q-learning cooperating with the reward that is designed based on the environment selects the LLHs intelligently and automatically. Focused on the characteristics of the problem, four efficient LLHs are proposed. Optimization experiments on CEC2017 and CEC2020 against nine famous optimizers demonstrate that QLSHH is significantly better than the competitor algorithms on 50% of the functions. Tyasnurita et al. [47] focused on the challenging open vehicle routing problem (OVRP) and utilized a time delay neural network (NN) as the offline learning method to select or generate LLHs. The performance of the proposed approach is investigated in OVRP instances with different sizes, and the experimental results demonstrate the competitiveness compared with the state-of-the-art optimization algorithms. More details about HHA can refer to [48].

From the literature review of HHA, many advanced HHA approaches adopt reinforcement learning approaches or neural networks as the high-level components to automatically select the LLHs and construct the optimization sequence. In the meantime, LLHs, as intelligent decision-making systems, have achieved great success in many research domains, and we believe that they can also perform well as the high-level component of HHA, which is the motivation of this research.

3. Our proposal: LLMOA

This section first presents the framework of LLMOA. Subsequently, the design of the high-level component and low-level heuristics (LLHs) are described in Sections 3.2 and 3.3, respectively. Finally, Section 3.4 analyzes the computational complexity of LLMOA. An overview of LLMOA is also illustrated in Fig. 2.²

3.1. The framework of LLMOA

As the flowchart of LLMOA demonstrated in Fig. 2, the optimization process begins with the initialization of the population, parameters, and search operator pool. Next, we integrate prompt engineering techniques with the LLM, specifically Gemini, to sample the search operators and construct the optimization sequence. Subsequently, the generation of offspring individuals using the constructed optimization sequence and the activation of the selection operator follows a sequential order similar to most evolutionary algorithms (EAs). Finally, the optimization is terminated when the optimal solution is found or the computational resources are exhausted. In the following context, we introduce the design of the high-level and the low-level components in our proposed LLMOA. The theoretical computational complexity is also provided subsequently.

3.2. High-level component

This section presents the design of the high-level component design of LLMOA. A demonstration of the high-level component is presented in Fig. 3. Initially, we design the prompt using standard prompt engineering, and the prompt is inputted into Gemini and outputs the optimization sequence indexes. We assemble the optimization sequence with designed LLHs to further drive the population to optimize the problem, and the optimization results serve as feedback to update the prompt. This interaction is activated at the beginning of each iteration. Subsequently, the designed prompt is introduced as follows.

Specifically, our designed prompt is presented in Fig. 4 and consists of four components.

² The symbol of LLM is downloaded from <https://pixabay.com/illustrations/chip-ai-artificial-intelligence-8530784> as the copyright-free image.

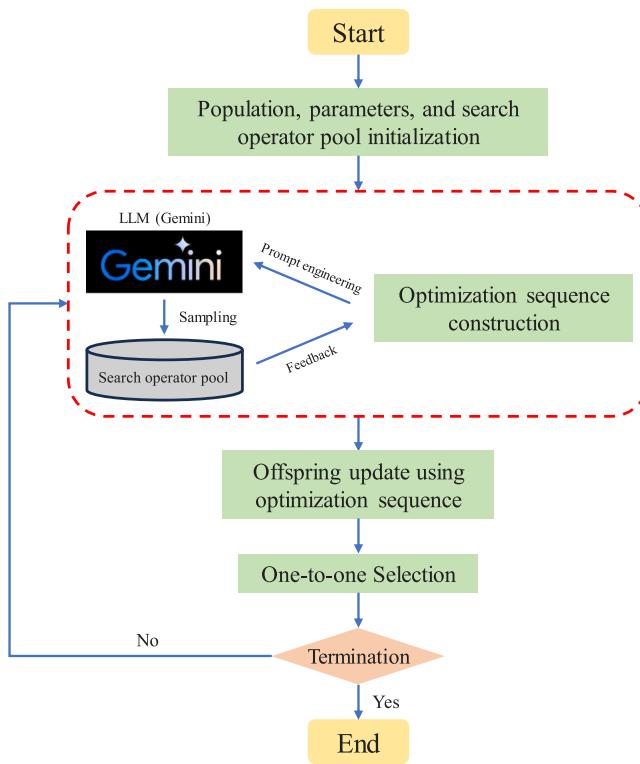


Fig. 2. The flowchart of LLMOA.

- Role definition:** This part defines the role of the LLM, which plays a crucial part in refining domain knowledge and expertise, thereby enhancing the responsiveness of the LLM's outputs.
- Problem instruction:** This part comprehensively describes the optimization problem, presenting its intricacies and challenges. Moreover, this part outlines the specifications given the desired solution properties, providing a clear roadmap for achieving the objective.
- In-context sample:** The solution (i.e., the optimization sequence in the previous iteration) is provided as the demonstration for the LLM. Additionally, the improvement of the population optimized by this optimization sequence is also provided, showcasing the evolution of the optimization process and serving as instructive examples for the LLM.
- Required output:** The format and specifications of the required output are identified in this part.

The problem description includes crucial attributes such as the type of problem (i.e., minimization) and the number of alternative search operators available. The required output should match the population size, with each element falling within the range greater than 0 and smaller than the total number of operators (i.e., in the range of [1, 10] in this research). If the hallucination phenomenon occurs in the LLM (i.e., an example in Fig. 4 is that 13 is out of the range), the feasibility check and repair operator is activated to re-generate a random integer to ensure that each element corresponds to the LLH. This well-designed prompt serves as a guiding framework for the LLM, facilitating the generation of new and effective optimization sequences.

3.3. Low-level heuristics

Low-level heuristics (LLHs) include a variety of search operators and is crucial for population updating. In this context, we utilize ten efficient and flexible search operators. These operators are summarized as follows:

Elite-based local search operator: The proximate optimality principle is often interpreted as “similar solutions have similar fitness” [49]. Therefore, we infer that the neighbourhoods of elite individuals may exhibit superior fitness values. Consequently, the search preference around these elite individuals may be promising. Inspired by this principle, we design an elite-based local search operator, as formulated in Eq. (1).

$$\mathbf{x}_i^{t+1} = \mathbf{e}^t + \boldsymbol{\delta} \quad (1)$$

where \mathbf{x}_i represents the i th individual and the superscript t and $t+1$ denote the iteration number. The symbol e denotes the elite individual that is randomly selected from the sub-population possessing the top 10% fitness values. $\boldsymbol{\delta}$ signifies the perturbation used in constructing new offspring individuals. Furthermore, three kinds of random generators are employed in the elite-based local search operator to construct the perturbation: uniform distribution, normal distribution, and Lévy flight, which are presented in Eqs. (2), (3), and (4), respectively.

$$\delta_j \sim U(-1, 1) \quad (2)$$

$$\delta_j \sim N(0, 1) \quad (3)$$

$$\begin{aligned} \delta_j &\sim \frac{u}{|v|^{\frac{1}{\beta}}} \\ u &\sim N(0, \sigma^2), u \sim N(0, 1) \\ \sigma &= \left\{ \frac{\Gamma(1+\beta) \sin(\frac{\pi\beta}{2})}{\beta \Gamma(\frac{1+\beta}{2}) 2^{\frac{\beta-1}{2}}} \right\}^{\frac{1}{\beta}} \\ \Gamma(z) &= \int_0^{+\infty} t^{z-1} e^{-t} dt \end{aligned} \quad (4)$$

where $U(-1, 1)$ represents a uniform distribution in $[-1, 1]$, $N(0, 1)$ denotes a normal distribution with the expectation of 0 and standard deviation of 1.

Mutation operators from DE: Mutation operators in DE play a crucial role in generating diverse solutions within the population, and the LLHs of LLMOA inherit several well-known and efficient mutation strategies: DE/best/1, DE/rand/1, DE/cur/1, DE/cur-to-best/1, and DE/cur-to-pbest/1. The formulations of these strategies are presented in Eq. (5).

$$\begin{aligned} \mathbf{x}_i^{t+1} &= \mathbf{x}_{best}^t + F \cdot (\mathbf{x}_{r1}^t - \mathbf{x}_{r2}^t) \\ \mathbf{x}_i^{t+1} &= \mathbf{x}_{r1}^t + F \cdot (\mathbf{x}_{r2}^t - \mathbf{x}_{r3}^t) \\ \mathbf{x}_i^{t+1} &= \mathbf{x}_i^t + F \cdot (\mathbf{x}_{r1}^t - \mathbf{x}_{r2}^t) \\ \mathbf{x}_i^{t+1} &= \mathbf{x}_i^t + F \cdot (\mathbf{x}_{best}^t - \mathbf{x}_i^t) + F \cdot (\mathbf{x}_{r1}^t - \mathbf{x}_{r2}^t) \\ \mathbf{x}_i^{t+1} &= \mathbf{x}_i^t + F \cdot (\mathbf{x}_{pbest}^t - \mathbf{x}_i^t) + F \cdot (\mathbf{x}_{r1}^t - \mathbf{x}_{r2}^t) \end{aligned} \quad (5)$$

where \mathbf{x}_{best}^t is the best individual in the t th iteration and \mathbf{x}_i^t is the i th individual in the t th iteration. \mathbf{x}_{r1}^t , \mathbf{x}_{r2}^t , and \mathbf{x}_{r3}^t are three mutually different individuals. F denotes the scaling factor which follows the normal distribution in $N(0.5, 0.3)$ as suggested in [50].

Crossover operators from DE: Crossover operators in DE are employed to combine information from the selected individuals, which determines how the trial vectors are integrated into the population to produce the offspring individuals. For the sake of simplicity, the designed LLHs contain two efficient variants of binary crossover operators: the binary crossover with a random individual and the binary crossover with the elite individual, as formulated in Eqs. (6) and (7), respectively.

$$\mathbf{x}_{i,j}^{t+1} = \begin{cases} \mathbf{x}_{r1,j}^t, & \text{if } r < Cr \text{ or } j = jrand \\ \mathbf{x}_{i,j}^t, & \text{otherwise} \end{cases} \quad (6)$$

$$\mathbf{x}_{i,j}^{t+1} = \begin{cases} \mathbf{e}_j^t, & \text{if } r < Cr \text{ or } j = jrand \\ \mathbf{x}_{i,j}^t, & \text{otherwise} \end{cases} \quad (7)$$

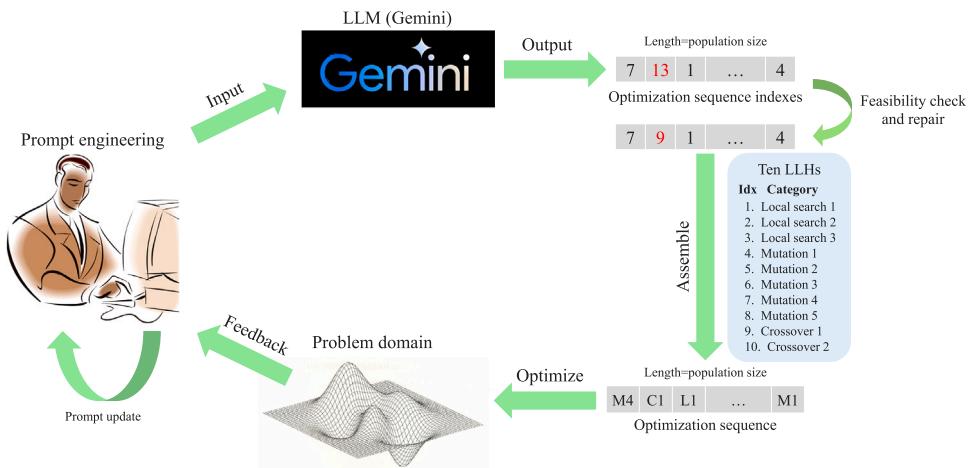


Fig. 3. A demonstration of the high-level component.

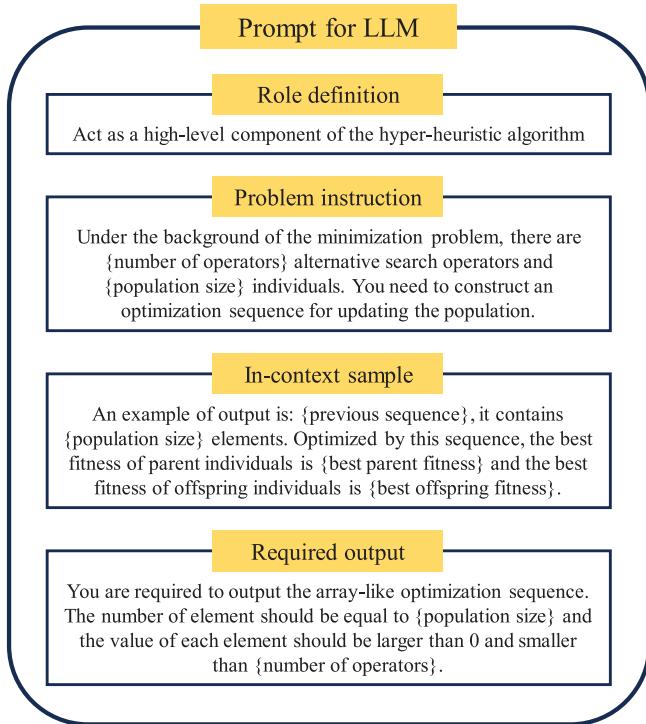


Fig. 4. The overview of the prompt. The contents within “{}” in the prompt will be replaced by the real data during optimization.

where r is a random number in the range of $(0, 1)$, Cr represents the crossover rate that follows a normal distribution in $N(0.5, 0.3)$ as recommended in [50], and $jrand$ is a random integer in $\{1, 2, \dots, D\}$.

Additionally, another component in LLMOA is the selection mechanism. Here, the one-to-one survival mechanism is employed in LLMOA, as formulated in Eq. (8).

$$x_i^{t+1} = \begin{cases} x_i^{t+1}, & \text{if } f(x_i^{t+1}) < f(x_i^t) \\ x_i^t, & \text{otherwise} \end{cases} \quad (8)$$

Simply, the one-to-one survival mechanism compares the parent individual with the corresponding offspring individual, and the better individual survives. In summary, the pseudocode of LLMOA is presented in Algorithm 1.

Algorithm 1: LLMOA

```

Input: Dimension:  $D$ , Population size:  $N$ , Search operator pool:  $SP$ , Maximum iteration:  $T$ 
Output: Optimum:  $x_{best}^t$ 
1 Function LLMOA( $D, T_{max}$ ):
2   Generate a random population  $X$ 
3    $t = 0$ 
4    $x_{best}^t \leftarrow \text{best}(X)$ 
5   while  $t < T$  do
6     Construct optimization sequence  $OS$  using LLM and
     prompt in Fig. 4.
7     Ensure each element in  $OS$  is an integer within  $[1, 10]$ .
8     Sort the population and maintain the elites.
9     for  $i = 1$  to  $N$  do
10      Determine the  $i^{th}$  search operator by  $SP$  and  $OS$ .
11      Construct the offspring individual  $x_i^{t+1}$  by the  $i^{th}$ 
       search operator
12      Boundary repair for the offspring individual  $x_i^{t+1}$ 
13      One-to-one greedy selection using Eq. (8)
14    end
15     $t \leftarrow t + 1$ 
16     $x_{best}^t \leftarrow \text{best}(X)$ 
17 end
18 return  $x_{best}^t$ 
```

3.4. Computational complexity analysis

At the beginning of the computational complexity analysis, we assume that the dimension size is D , the population size is N , the maximum iteration is T , and the necessary computational resource for interacting with Gemini is K . We analyse the computational complexity of LLMOA following the procedures in Algorithm 1.

The computational complexity of the first step of the population initialization is $O(N \times D)$. Then, LLMOA enters the main loop from Algorithm 1 Line 6 to Line 16, and the computational complexity of procedures in one iteration is summarized as follows:

- Interaction with Gemini: $O(K)$.
- Population sort and elite maintenance: $O(N \times \log N)$.
- Offspring construction: $O(N \times D)$.
- Selection: $O(N)$.

Table 1

Computational complexity of representative MHAs.

MHAs	Computational complexity
GA [51]	$O(T \times N \times (\log N + D))$
PSO [52]	$O(T \times N \times D)$
DE [53]	$O(T \times N \times D)$
slime mold algorithm (SMA) [7]	$O(T \times N \times (\log N + D))$
grey wolf optimizer (GWO) [5]	$O(T \times N \times (\log N + D))$
RIME [12]	$O(T \times N \times D)$

In summary, the computational complexity of LLMOA is computed in Eq. (9).

$$\begin{aligned} & O(N \times D + T \times (K + N \times \log N + N \times D + N)) \\ & := O(T \times (K + N \times \log N + N \times D)) \end{aligned} \quad (9)$$

For a clearer comparison, we present the computational complexities of basic MHAs in Table 1.

Since the sort operator and interaction with Gemini are involved in LLMOA, it consumes more computational complexity than the listed MHAs. This shortage limits the scalability and adaptability of LLMOA in expensive optimization tasks.

4. Numerical experiments

This section presents the experiment outcomes in detail. Section 4.1 summarizes the experimental settings including experimental settings/configurations and implementation, benchmark mathematical functions, and comparison optimizers and their configurations. Section 4.2 shows the statistical analyses and experimental results of the comparison, ablation, and investigation experiments on various benchmark mathematical functions. Finally, Section 4.2 presents the simulation experiments in engineering problems.

4.1. Experimental settings

4.1.1. Experimental environments and implementation

Our numerical experiments were conducted using Python 3.11 on a Lenovo Legion R9000P laptop running Windows 11. The laptop is equipped with an AMD Ryzen 7 5800H processor with Radeon Graphics, running at 3.20 GHz, and has 16 GB of RAM. Gemini³ was utilized as the LLM throughout the entirety of the experiments.

4.1.2. Benchmark functions

We adopted three well-known benchmarks from the IEEE Congress on Evolutionary Computation (CEC) and one set of engineering simulation optimization problems to thoroughly evaluate the performance of our proposed LLMOA. The outline of these benchmarks is summarized as follows:

- CEC2014 benchmark functions⁴: This benchmark suite contains thirty diverse benchmark functions and is provided by OpFuNu library [54].
- CEC2020 benchmark functions⁵: This benchmark suite contains ten diverse benchmark functions and is provided by OpFuNu library [54].
- CEC2022 benchmark functions⁶: This benchmark suite contains twelve diverse benchmark functions and is provided by OpFuNu library [54].

Table 2

Description of different 10 constrained engineering problems.

Name	Abb.	No. of Dim	No. of constraints
Corrugated Bulkhead Problem	CBHD	4	6
Cantilever Beam Problem	CBD	5	1
Gear Train Problem	GTD	4	0
Compression Spring Problem	CSP	4	4
Reinforced Concrete Beam Problem	RCB	3	2
I Beam Problem	IBD	4	2
Speed Reducer Problem	SRD	7	11
Three Bar Truss Problem	TBTB	2	3
Welded Beam Problem	WBP	4	7
Tubular Column Problem	TCD	2	6

- Engineering simulation optimization suite [55]: This benchmark suite contains ten diverse engineering optimization problems, primarily summarized in Table 2. This benchmark suite is provided by ENOPPY library [56].

The detailed summary and description of these benchmark functions can be found in Appendix A.

4.1.3. Compared optimizers and parameters

To evaluate the competitiveness of our proposed LLMOA, we adopt PSO [52], DE [53], covariance matrix adaptation evolution strategy (CMA-ES) [57], self-adaptive DE (SaDE) [50], comprehensive learning PSO (CL-PSO) [58], success history adaptive DE (SHADE) [59], SHADE with linear population reduction (L-SHADE) [60], hierarchical PSO with time-varying acceleration coefficients (HPSO-TVAC) [61], phasor PSO (PPSO) [62], weighted DE [63], and diversity-maintained multi-trial vector DE (DMDE) [64]. Table 3 summarizes the detailed parameters of these competitor algorithms. Additionally, the population size of algorithms except for L-SHADE is set to 100, and the maximum fitness evaluation (FE) is fixed at $1000 \times D$ (D = Dimension size). To ensure statistical robustness, each algorithm is independently executed 30 times.

As the original versions of these algorithms cannot handle constrained optimization problems, it is common for engineering optimization tasks to involve constraints. Therefore, we enhance these EAs by incorporating the static penalty function [65], as formulated in Eq. (10).

$$F(\mathbf{x}_i) = f(\mathbf{x}_i) + w \cdot \sum_{i=1}^m (\max(0, g_i(\mathbf{x}_i))) \quad (10)$$

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

4.2. Experimental results

This section presents the experimental results and statistical analyses on diverse optimization tasks. Sections 4.2.1, 4.2.2, and 4.2.3 summarize the results on CEC2014, CEC2020, and CEC2022, respectively. Section 4.2.4 presents the results of ablation experiments on CEC2014, CEC2020, and CEC2022 benchmark functions.

4.2.1. Results and analyses on CEC2014

Table 4 summarizes the statistical results on CEC2014 benchmark functions, while the experimental results and statistical analyses are detailed in Appendix C. Additionally, to quantitatively identify the exploration and exploitation behaviour of LLMOA during optimization,

³ <https://gemini.google.com/>

⁴ <https://github.com/P-N-Suganthan/CEC2014>

⁵ <https://github.com/P-N-Suganthan/2020-Bound-Constrained-Opt-Benchmark>

⁶ <https://github.com/P-N-Suganthan/2022-SO-BO>

Table 3
Parameters of competitor algorithms.

MHAs	Parameters	Value
DE	Mutation scheme	DE/cur-to-rand/1
	Scaling factor F	0.8
	Crossover rate Cr	0.9
PSO	Inertia factor w	1
	Acceleration coefficients c_1 and c_2	2.05 and 2.05
	Max. and min. speed	2 and -2
CMA-ES	Standard deviation σ	1.3
SaDE	μ_F and σ_F	0.5 and 0.3
	μ_{Cr} and σ_{Cr}	0.5 and 0.1
	Mutation scheme	DE/rand/1 and DE/cur-to-best/1
CL-PSO	Local coefficient c_{local}	1.2
	Max. and min. weight	0.9 and 0.4
SHADE	μ_F and μ_{Cr}	0.5 and 0.5
L-SHADE	μ_F and μ_{Cr}	0.5 and 0.5
	Population size	18 × D
HPSO-TVAC	c_{init} and c_{final}	0.5 and 0
PPSO	Parameter-free	
WDE	Parameter-free	
DMDE	Archive limitation k	200
	Angle parameter θ	45°
LLMOA	Scaling factor F	$N(0.5, 0.3)$
	Gemini version	gemini-pro

Table 4
Statistical results on CEC2014 benchmark functions.

Dim.	PSO	DE	CMA-ES	SaDE	CL-PSO	SHADE	L-SHADE	HPSO-TVAC	PPSO	WDE	DMDE	LLMOA
30-D	30/0/0	29/0/1	27/1/2	18/8/4	20/8/2	12/11/7	14/10/6	15/6/9	29/1/0	18/6/6	16/10/4	-
50-D	29/1/0	27/2/1	26/1/3	16/5/9	20/7/3	13/6/11	13/6/11	17/4/9	26/4/0	17/6/7	15/7/8	-

we present the plots of exploration and exploitation proportion as described in [66], as formulated in Eq. (11).

$$\begin{aligned} Div^t &= \frac{1}{D} \sum_{d=1}^D \frac{1}{N} \sum_{i=1}^N |\mathbf{x}_{mean,d}^t - \mathbf{x}_{i,d}^t| \\ Exploration &= \frac{Div^t}{Div_{max}} \\ Exploitation &= \frac{|Div^t - Div_{max}|}{Div_{max}} \end{aligned} \quad (11)$$

where D and N are the dimension size and the population size. \mathbf{x}_{mean}^t and \mathbf{x}_i^t denote the mean of the population and the i th individual. Similar to the standard deviation, Div^t is employed to measure the dispersion of the population in relation to the mean. Figs. 5 and 6 present convergence curves, boxplots, and the proportion between exploration and exploitation on CEC2014 representative functions, where the detailed figures are in Appendix C.

Besides, to identify the statistical significance, we collect the optima from 30 trial runs, then, the Friedman test is applied. If statistical significance exists, we apply the Mann–Whitney U test for each pair of optimizers and modify the obtained p -value by the Holm multiple comparison test. Symbols +, \approx , and – refer to LLMOA as substantially better, without significance, or substantially worse than the competitor EA. The best fitness value is highlighted in bold.

Through the statistical analysis summary, our proposed LLMOA is significantly better than classic optimizers PSO, DE, and CMA-ES in most instances and is highly competitive with state-of-the-art variants of DE and PSO on CEC2014 benchmark functions. We owe the superiority of LLMOA to the following two factors.

- (i). Intelligent LLM-assisted high-level component: The incorporation of the intelligent LLM Gemini brings exceptional domain knowledge and expertise in optimization to our proposed LLMOA. This assists in the intelligent construction of the sequence of heuristics, enhancing its adaptability and effectiveness.

- (ii). Efficient and effective LLHs: The LLHs form the foundation of LLMOA and are meticulously designed to be highly competitive. Their efficiency and effectiveness contribute significantly to the algorithm's overall performance across diverse benchmark functions.

However, we observe that LLMOA does not perform well on some multimodal and composite functions. Specifically, in f_{27} and f_{28} , LLMOA is significantly worse than most competitor algorithms. Although this phenomenon can also be explained by the NFLT [67], we infer that this deterioration is due to the limited local search heuristics (LLHs), which may not be effective for these specific composite problems. This issue can be addressed by incorporating more well-designed search operators in future research.

Overall, the competitive performance of LLMOA has been confirmed through comprehensive numerical experiments. An additional benefit of LLMOA is its ease of implementation and user-friendly characteristics. Unlike the complex internal mechanisms in MHAs, such as success-history-based parameter adaptation in SHADE and L-SHADE, we assign this challenge to Gemini and use prompt engineering techniques to instruct it. As a result, the proposed LLMOA offers high scalability and reproducibility, making it accessible even to amateurs in the EC community.

4.2.2. Results and analyses on CEC2020

Table 5 summarizes the statistical results on CEC2020 benchmark functions, while the experimental results and statistical analyses are detailed in Appendix D. Figs. 7 and 8 present convergence curves, boxplots, and the proportion between exploration and exploitation on CEC2020 representative functions, where the detailed figures are in Appendix D.

The experimental results and convergence curves demonstrate the competitiveness of LLMOA on CEC2020 benchmark functions, highlighting its success in integrating LLMs like Gemini into the hyper-heuristic framework. However, a unique phenomenon observed in

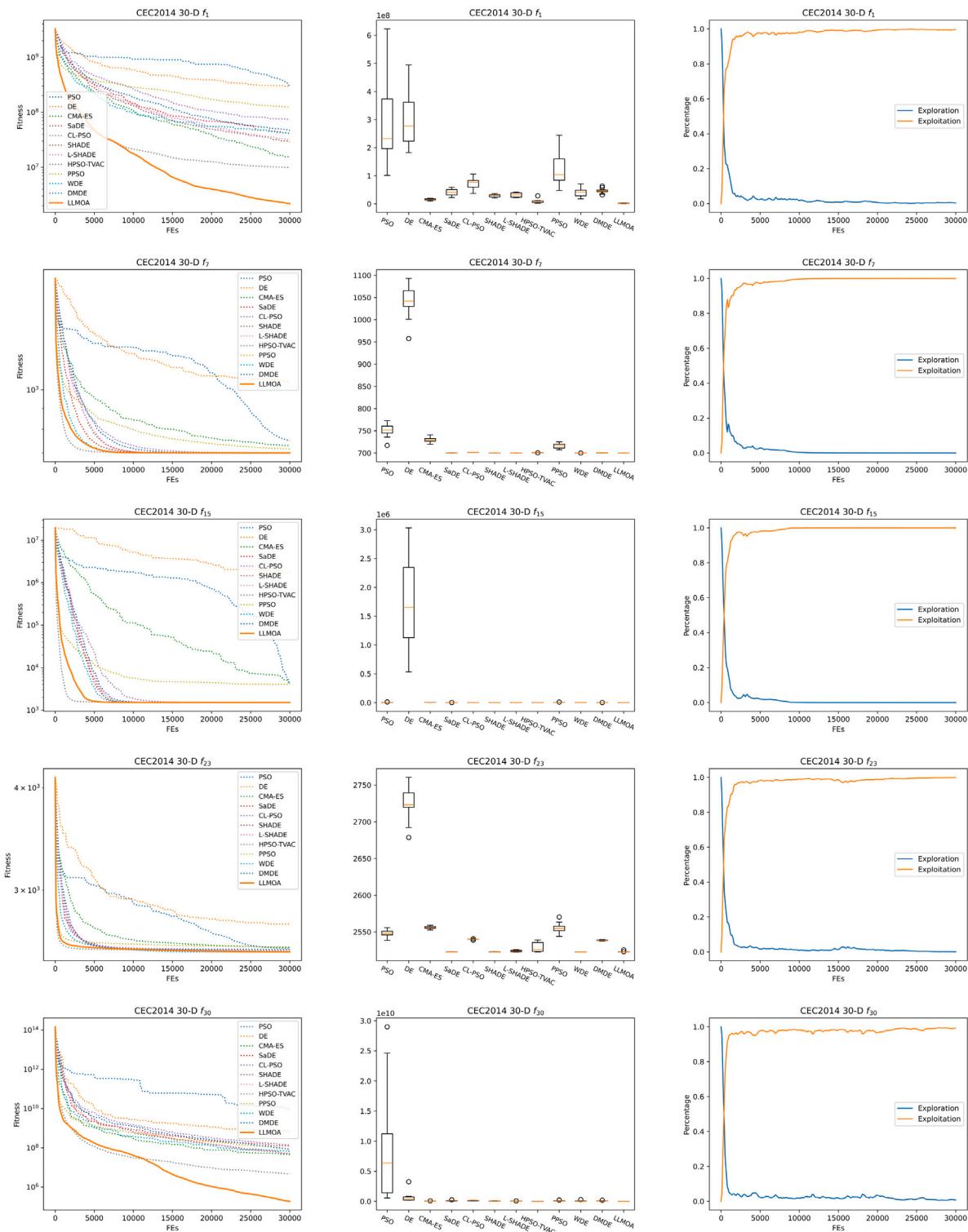


Fig. 5. Convergence curves and box plots of optimizers, and proportion between exploration and exploitation of LLMOA on 30-D CEC2014 representative functions (i.e., f_1 : unimodal function; f_4 , f_7 , and f_{15} : multimodal functions; f_{23} and f_{30} : composite functions).

Table 5

Statistical results on CEC2020 benchmark functions.

Dim.	PSO	DE	CMA-ES	SaDE	CL-PSO	SHADE	L-SHADE	HPSO-TVAC	PPSO	WDE	DMDE	LLMOA
30-D	10/0/0	10/0/0	9/0/1	6/2/2	9/0/1	6/2/2	6/2/2	7/0/3	10/0/0	6/1/3	8/1/1	–
50-D	10/0/0	10/0/0	9/0/1	3/2/5	9/0/1	3/1/6	3/1/6	9/0/1	10/0/0	3/1/6	8/1/1	–

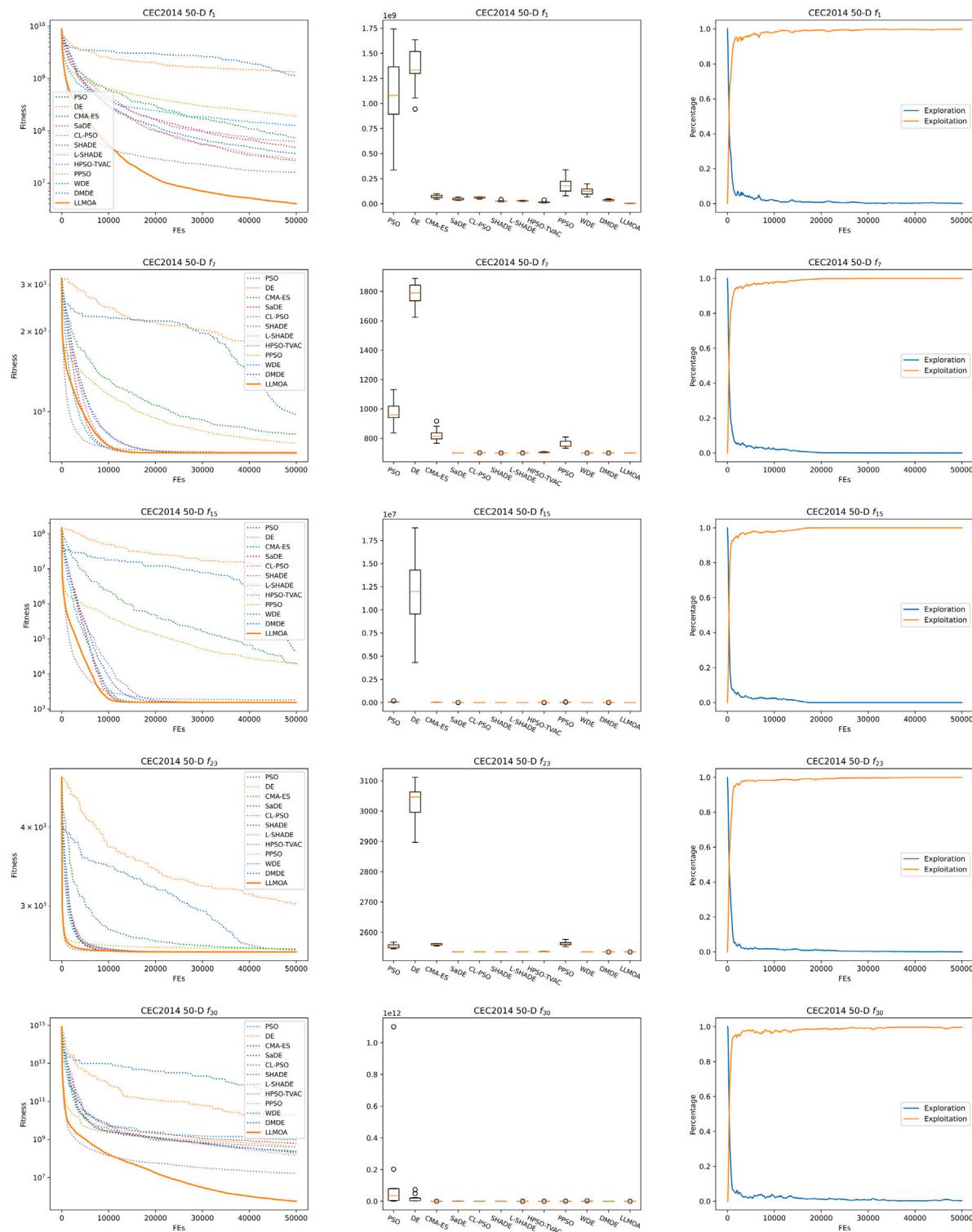


Fig. 6. Convergence curves and box plots of optimizers, and proportion between exploration and exploitation of LLMOA on 50-D CEC2014 representative functions.

Table 6

Statistical results on CEC2022 benchmark functions.

Dim.	PSO	DE	CMA-ES	SaDE	CL-PSO	SHADE	L-SHADE	HPSO-TVAC	PPSO	WDE	DMDE	LLMOA
10-D	11/1/0	11/0/1	8/2/2	7/3/2	9/3/0	5/5/2	6/4/2	5/2/5	11/1/0	7/2/3	7/5/0	–
20-D	11/1/0	10/1/1	10/1/1	7/2/3	9/3/0	7/3/2	7/3/2	5/2/5	10/2/0	7/2/3	7/3/2	–

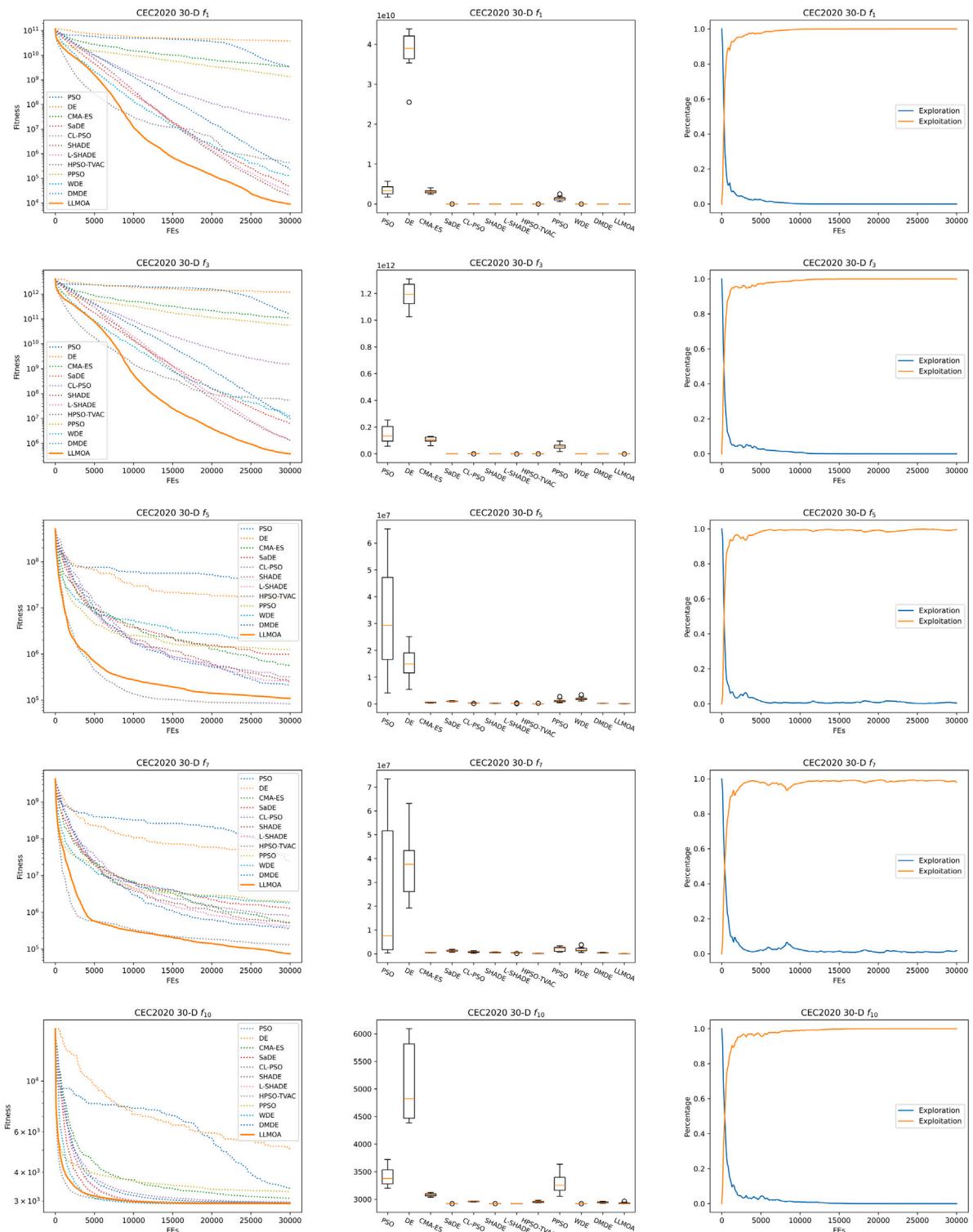


Fig. 7. Convergence curves and box plots of optimizers, and proportion between exploration and exploitation of LLMOA on 30-D CEC2020 representative functions (i.e., f_1 : unimodal function; f_3 : multimodal function; f_5 and f_7 : hybrid functions; f_{10} : composite function).

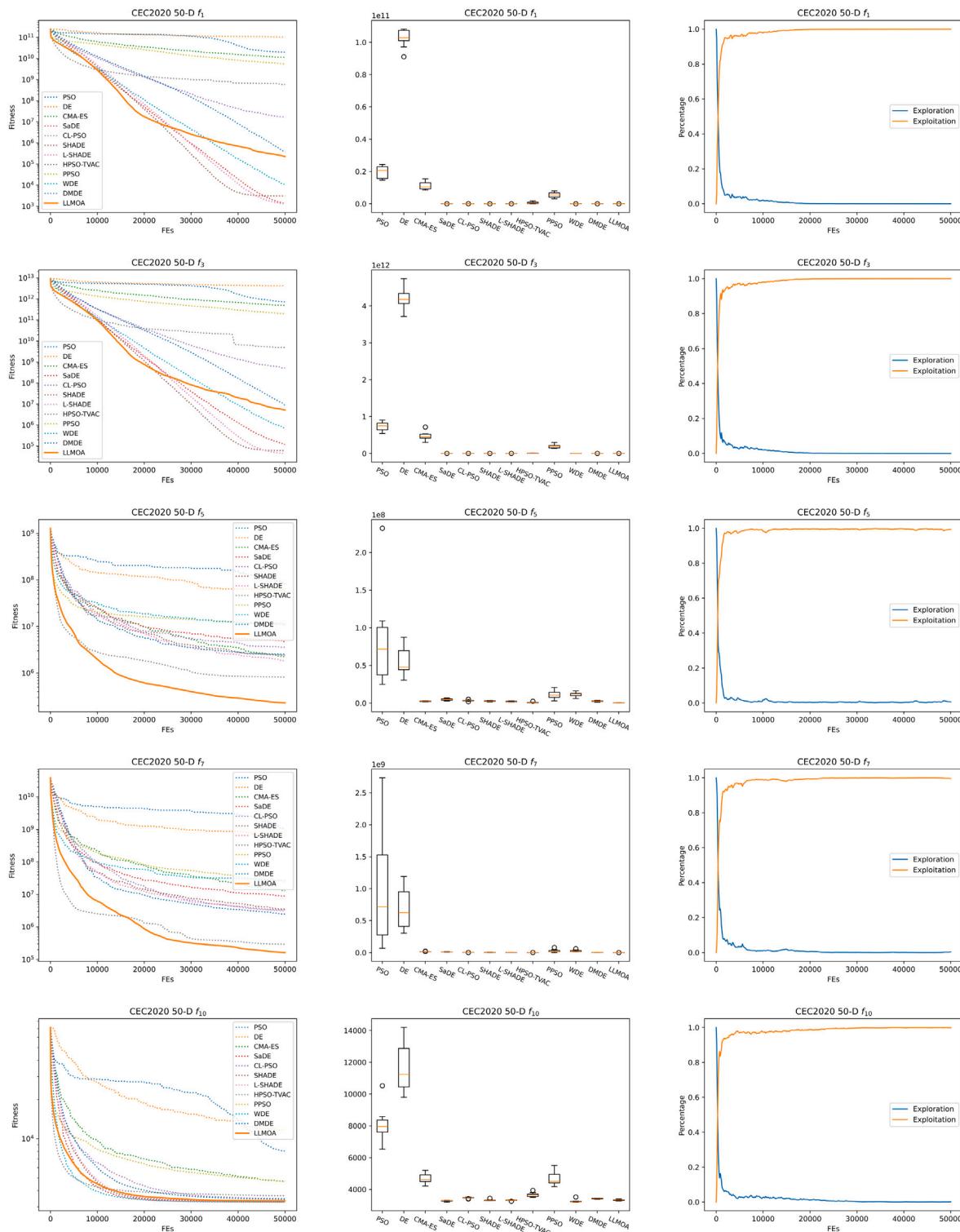


Fig. 8. Convergence curves and box plots of optimizers, and proportion between exploration and exploitation of LLMOA on 50-D CEC2020 representative functions.

CEC2020 is the significant performance degradation of LLMOA as the problem dimension increases from 30 to 50. In contrast, advanced DE-based optimizers maintain their superiority and effectiveness. This degradation is attributed to the curse of dimensionality, which severely affects the optimization performance of MHAs. Given the proven effectiveness and efficiency of DE-based optimizers that most champion algorithms in CEC competitions are developed based on DE, incorporating the expert-designed characteristics from optimizers such as search

operators and parameter adaptation mechanisms into LLMOA could further enhance its optimization performance in future research.

4.2.3. Results and analyses on CEC2022

Table 6 summarizes the statistical results on CEC2020 benchmark functions, while the experimental results and statistical analyses are detailed in [Appendix E](#). **Figs. 9 and 10** present convergence curves, boxplots, and the proportion between exploration and exploitation on

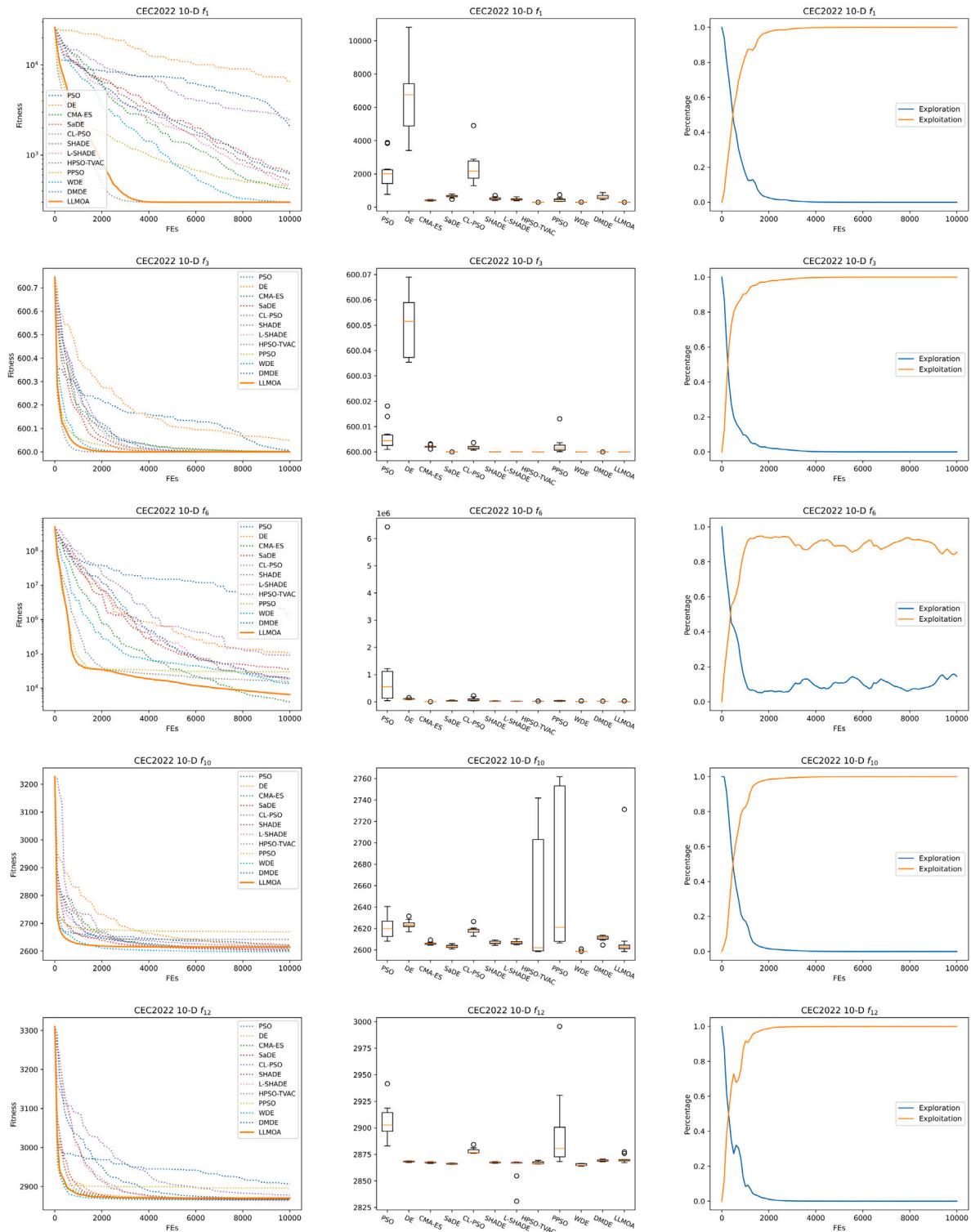


Fig. 9. Convergence curves and box plots of optimizers, and proportion between exploration and exploitation of LLMOA on 10-D CEC2022 representative functions (i.e., f_1 : unimodal function; f_3 : multimodal function; f_6 : hybrid functions; f_{10} and f_{12} : composite functions).

CEC2022 representative functions, where the detailed figures are in Appendix E.

Compared to CEC2014 and CEC2020, the CEC2022 benchmark suite contains only 10-D and 20-D benchmark functions. These relatively small-scale optimization problems are designed to evaluate the capacity of algorithms to find the optimum within a limited computational

budget. From the summarized experimental results and convergence curves, it is evident that LLMOA exhibits excellent convergence speed and optimization accuracy. In most instances, the optimum found by LLMOA is very close to the real optimum, underscoring the effectiveness and efficiency of our proposed LLMOA. These results demonstrate

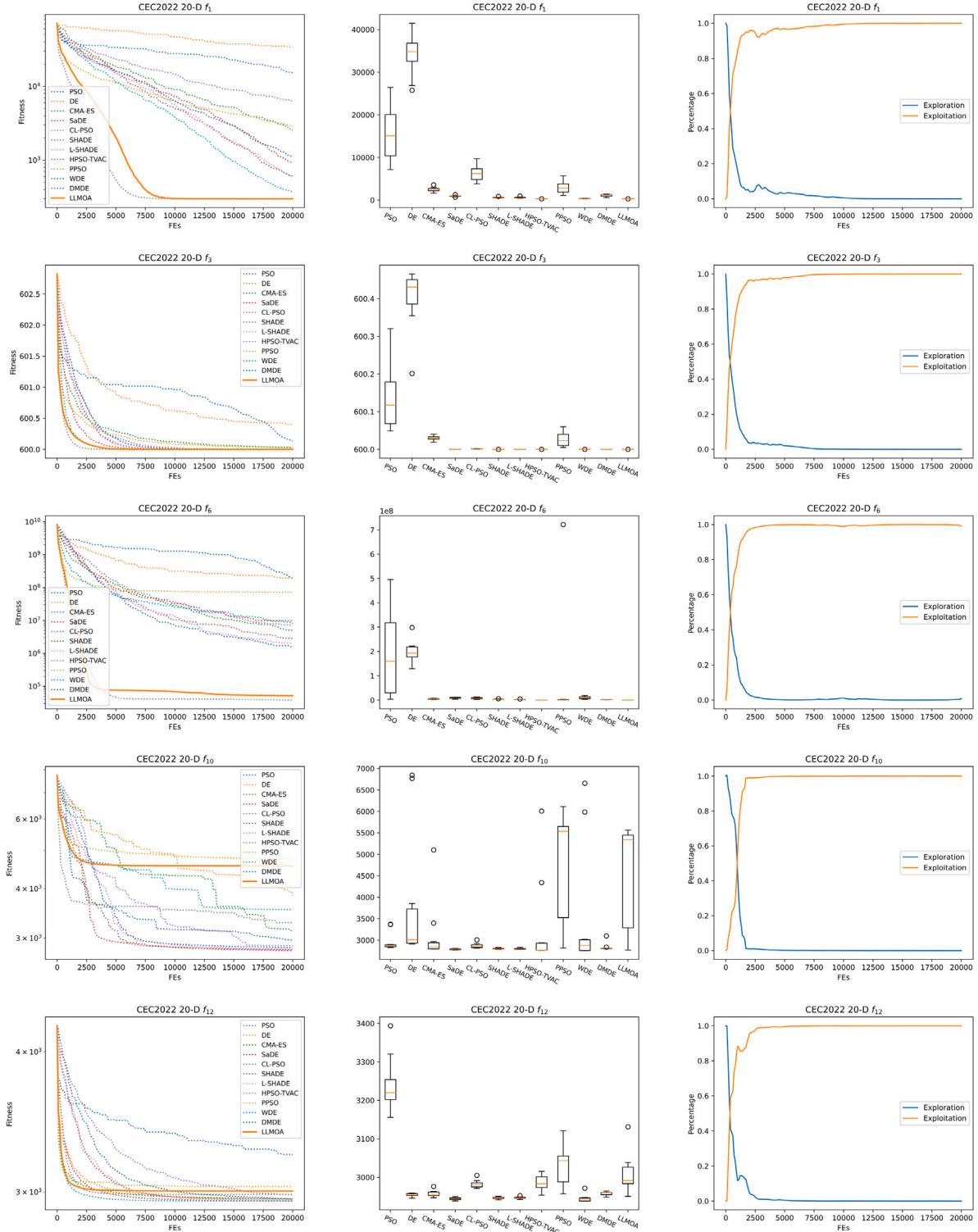


Fig. 10. Convergence curves and box plots of optimizers, and proportion between exploration and exploitation of LLMOA on 20-D CEC2022 representative functions.

that LLMOA is not only capable of approaching rapid convergence but also adept at achieving high-quality solutions in constrained settings.

4.2.4. Ablation experiments and analyses

To investigate the independent contribution of the LLM in acting as the high-level component of HHA, we conduct the ablation experiments on CEC2014, CEC2020, and CEC2022 benchmark functions. Three baseline methods are employed as the high-level component: simple

random (SR) [68,69], random permutation (RP) [70], and success-history adaptation (SHA) [45]. These high-level strategies are integrated with the proposed LLHs in this study to form HHA_{SR} , HHA_{RP} , and HHA_{SHA} , respectively. The experimental findings and statistical evaluations are presented in Table 7, while the detailed results are summarized in Appendix F.

The experimental results confirm that the integration of Gemini can achieve approximately equal or significantly better performance

Table 7
Ablation experiments on CEC benchmarks.

Bench.	Dim.	HHA _{SR}	HHA _{RP}	HHA _{SHA}	LLMOA
CEC2014	30-D	13/16/1	15/15/0	0/29/1	–
	50-D	17/13/0	18/12/0	3/27/0	–
CEC2020	30-D	8/2/0	7/3/0	1/8/1	–
	50-D	10/0/0	10/0/0	1/9/0	–
CEC2022	10-D	5/7/0	6/6/0	2/9/1	–
	20-D	2/10/0	4/8/0	0/11/1	–

Table 8
Performance investigation of various LLMs in LLMOA on CEC benchmarks.

Bench.	Dim.	LLMOA _{ERNIE}	LLMOA _{ChatGPT}	LLMOA _{Gemini}
CEC2014	30-D	0/30/0	1/28/1	–
	50-D	1/29/0	0/29/1	–
CEC2020	30-D	0/10/0	0/10/0	–
	50-D	1/9/0	0/10/0	–
CEC2022	10-D	0/12/0	0/11/1	–
	20-D	0/12/0	0/12/0	–

compared to HHA_{SR}, HHA_{RP}, and HHA_{SHA} in most instances. As one of the latest mechanisms in designing high-level components, LLMOA is only significantly worse than HHA_{SHA} in 30-D CEC2014 f_1 , 30-D CEC2020 f_9 , 10-D CEC2022 f_3 , and 20-D CEC2022 f_1 . In most cases, LLMOA demonstrates highly competitive performance with HHA_{SHA}. These ablation experiments underscore the significance and efficacy of integrating Gemini and prompt engineering into LLMOA.

4.2.5. Investigation of various LLMs as the high-level component

Furthermore, there are many famous LLMs except for Gemini, such as ERNIE and ChatGPT. This section investigates the performance of LLMOA when adopting different LLMs as the high-level component. Here, LLMOA_{ERNIE} denotes that the ernie-bot-8k developed by Baidu [71] is employed as the high-level component, while LLMOA_{ChatGPT} uses the gpt-3.5-turbo as the engineer [32]. Table 8 summarizes the statistical results on CEC benchmarks, while the detailed results are presented in Appendix G. Based on the experimental results and comprehensive statistical analyses, we conclude that the performance of our proposed LLMOA on CEC benchmark optimization tasks exhibits no statistically significant differences across various LLMs. The similar performance among LLMOA_{ERNIE}, LLMOA_{ChatGPT}, and LLMOA_{Gemini} highlights the robustness and consistency of LLMOA, which indicates that it is not sensitive to the choice of LLM as the high-level component. Such stability further reinforces the suitability of LLMOA for diverse optimization problems.

4.3. Constrained engineering optimization: results & analyses

To prove the powerfulness of the developed approach, 10 different engineering constrained problems have been used, namely: Corrugated Bulkhead Problem (CBHP), Cantilever Beam Problem (CBP), Gear Train Design Problem (GTP), Compression Spring Design Problem (CSP), Reinforced Concrete Beam Problem (RCBP), I-Beam Design Problem (IBDP), Speed Reducer Problem (SRP), Three-Bar Truss Design Problem (TBTDP), Welded Beam Problem (WBP) and Tubular Column Design Problem (TCDP).

4.3.1. Corrugated bulkhead problem

The mathematical formulation of the Corrugated Bulkhead Problem can be found in B.1. The similar superiority and competitiveness of LLMOA can also be observed in the Corrugated Bulkhead Problem, where LLMOA performed the best. The robustness of LLMOA is observable in the presented results and the box plot. Although the second-best algorithm, HPSO-TVAC, is competitive with LLMOA and demonstrates a high level of robustness, our proposed LLMOA still outperforms HPSO-TVAC with statistical significance. Fig. 11 shows the boxplot and convergence curve for all compared algorithms (see Tables 9 and 10).

4.3.2. Cantilever beam problem

The mathematical formulation of the Cantilever Beam Problem can be found in B.2. Table 11 presents the experimental results and statistical analyses of eleven well-known optimization algorithms and our proposed LLMOA. Table 12 summarizes the optimal solutions found by these optimizers on the Cantilever Beam Problem. Our proposed LLMOA achieved the best performance, as evaluated by the metrics of mean and optimum, with limited computational budgets. Furthermore, the robustness of LLMOA is evident from the standard deviation and the box plot across multiple trial runs. Therefore, LLMOA proves to be an excellent optimizer for addressing the Cantilever Beam Problem. Fig. 12 shows the boxplot and convergence curve for all compared algorithms.

4.3.3. Gear train design problem

The mathematical formulation of the Gear Train Design Problem can be found in B.3. In the Gear Train Design Problem, PPSO found the optimum in every independent trial run and achieved the best performance as evaluated by the mean and standard deviation metrics. From Table 13, our proposed LLMOA was the second-best optimizer among twelve competitors, with no statistical significance between PPSO and LLMOA. Additionally, the optimum presented in Table 14 for both PPSO and LLMOA have identical fitness values, demonstrating the remarkable applicability and competitiveness of our proposed LLMOA. Fig. 13 shows the boxplot and convergence curve for all compared algorithms (see Table 13).

4.3.4. Compression spring design problem

The mathematical formulation of the Compression Spring Design Problem can be found in B.4. PPSO and LLMOA both achieved the best performance in the Compression Spring Design Problem, with a standard deviation of 0. These results indicate that PPSO and LLMOA consistently found the optimum in multiple trial runs. The statistical outcomes and experimental analyses confirm the competitiveness and robustness of our proposed LLMOA compared to state-of-the-art optimizers. Fig. 14 shows the boxplot and convergence curve for all compared algorithms (see Tables 15 and 16).

4.3.5. Reinforced concrete beam problem

The mathematical formulation of the Reinforced Concrete Beam Problem can be found in B.5. In the Reinforced Concrete Beam Problem, HPSO-TVAC consistently identified the optimum solution in every independent trial, achieving superior performance as indicated by mean and standard deviation metrics. Our proposed LLMOA ranked as the second-best optimizer among twelve competitors, showing no statistically significant difference compared to HPSO-TVAC. Furthermore, the optimum values listed in Table 18 for both HPSO-TVAC and LLMOA are identical in terms of fitness, highlighting the exceptional applicability and competitiveness of our LLMOA. Fig. 15 shows the boxplot and convergence curve for all compared algorithms. (see Table 17).

4.3.6. I-beam design problem

The mathematical formulation of the I-Beam Design Problem can be found in B.6. In the I-Beam Design Problem, LLMOA performs approximately on par with WDE and PPSO and is significantly better than the rest of the competitor algorithms. Additionally, the standard deviations of PPSO and LLMOA are 0, while the standard deviation of WDE is very small, demonstrating their exceptional competitiveness and robustness. The rapid convergence speed of LLMOA, visualized in Fig. 16, further confirms its remarkable search behaviours.

It is noteworthy that while PPSO is not an efficient optimizer as evaluated by the standard CEC2017, CEC2020, and CEC2022 benchmarks, it exhibits high-level competitiveness in engineering applications. This phenomenon also aligns with NFLT as suggested in [67] (see Tables 19 and 20).

Table 9
Results outcomes & statistical analyses in CBHP.

Optimizer	mean	std	best	worst
PSO	7.424088e+00	+ 1.962953e-01	7.128307e+00	7.692603e+00
DE	6.937650e+00	+ 3.009130e-02	6.885362e+00	7.005797e+00
CMA-ES	6.874979e+00	+ 5.277392e-03	6.868750e+00	6.886037e+00
SaDE	6.864468e+00	+ 1.021072e-02	6.846111e+00	6.883159e+00
CL-PSO	7.378772e+00	+ 3.066998e-01	6.938689e+00	8.014867e+00
SHADE	6.872230e+00	+ 8.568610e-03	6.852859e+00	6.882175e+00
L-SHADE	6.881896e+00	+ 1.209845e-02	6.863073e+00	6.894093e+00
HPSO-TVAC	6.842959e+00	+ 6.706739e-07	6.842958e+00	6.842961e+00
PPSO	6.968523e+00	+ 9.346717e-02	6.851168e+00	7.106737e+00
WDE	6.843121e+00	+ 1.050629e-04	6.843029e+00	6.843385e+00
DMDE	6.931091e+00	+ 7.253400e-02	6.862804e+00	7.124950e+00
LLMOA	6.842959e+00	8.795318e-07	6.842958e+00	6.842961e+00

Table 10
The optimum comparison of optimizers in CBHP.

Optimizer	x_1	x_2	x_3	x_4	$f(X)$
PSO	49.0058929	34.4400938	59.961216	1.09035572	7.12830655279963
DE	55.91375755	34.11016929	57.63062519	1.05479776	6.885361554285388
CMA-ES	56.2464485	34.26737472	57.71959277	1.05171332	6.868750220144203
SaDE	57.68253267	34.13807704	57.66635184	1.05045648	6.846110707792844
CL-PSO	58.28530687	33.90636451	57.54876581	1.06658042	6.93868857115105
SHADE	57.71306258	34.15113198	57.75198451	1.05124767	6.849803666526612
L-SHADE	57.70694568	34.14644596	57.72153212	1.05051382	6.8455090207194065
HPSO-TVAC	57.69229221	34.14762185	57.69230066	1.05	6.842958342204411
PPSO	56.3629558	34.13862627	57.69322395	1.05001429	6.8511681700130715
WDE	57.54252407	34.16275234	57.13203632	1.05084004	6.862803560473003
DMDE	57.69217903	34.1478327	57.69234735	1.05000938	6.843029303664572
LLMOA	57.69230644	34.14762049	57.69230762	1.05	6.842958026993829

Table 11
Results outcomes & statistical analyses in CBP.

Optimizer	mean	std	best	worst
PSO	2.245917e+00	+ 2.529386e-01	1.727746e+00	2.601542e+00
DE	1.991760e+00	+ 2.613664e-01	1.690454e+00	2.613015e+00
CMA-ES	1.533355e+00	+ 4.562528e-02	1.463810e+00	1.634877e+00
SaDE	1.341662e+00	+ 7.307357e-04	1.340409e+00	1.343154e+00
CL-PSO	1.485766e+00	+ 8.930905e-02	1.401818e+00	1.660418e+00
SHADE	1.343333e+00	+ 1.413807e-03	1.341562e+00	1.345754e+00
L-SHADE	1.345690e+00	+ 2.916471e-03	1.341557e+00	1.350492e+00
HPSO-TVAC	1.340016e+00	+ 4.791613e-05	1.339964e+00	1.340100e+00
PPSO	3.481719e+00	+ 1.198011e+00	1.956401e+00	5.576156e+00
WDE	1.340136e+00	+ 9.172721e-05	1.339973e+00	1.340304e+00
DMDE	1.347589e+00	+ 2.932155e-03	1.343571e+00	1.353188e+00
LLMOA	1.339960e+00	8.534547e-06	1.339957e+00	1.339986e+00

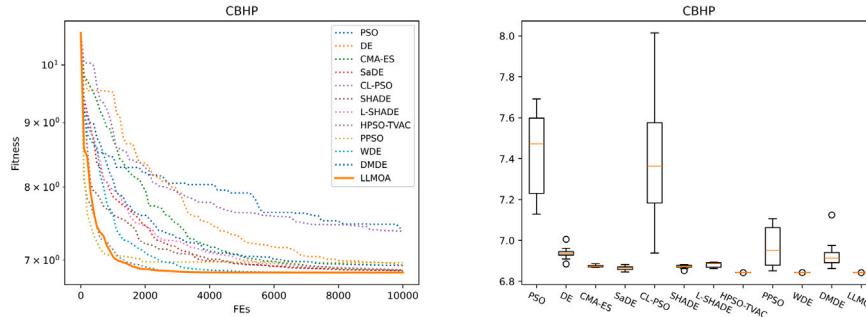


Fig. 11. Convergence curves and box plots of optimizers on CBHP.

4.3.7. Speed reducer problem

The mathematical formulation of the Speed Reducer Problem can be found in B.7. Table 21 presents the experimental results and statistical analyses of twelve well-known optimization algorithms and our proposed LLMOA. Table 22 summarizes the optimal solutions found by these optimizers on the Speed Reducer Problem. Our proposed LLMOA achieved the best performance, as evaluated by the metrics of mean and optimum, with limited computational budgets. Furthermore, the

robustness of LLMOA is evident from the standard deviation and the box plot across multiple trial runs. Therefore, LLMOA proves to be an excellent optimizer for addressing the Speed Reducer Problem. Fig. 17 shows the boxplot and convergence curve for all compared algorithms.

4.3.8. Three-bar truss design problem

The mathematical formulation of the Three-Bar Truss Design Problem can be found in B.8. Table 23 showcases the experimental outcomes

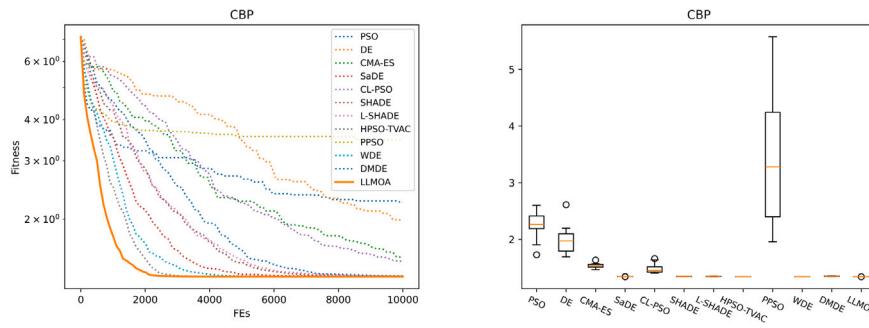


Fig. 12. Convergence curves and box plots of optimizers on CBP.

Table 12
The optimum comparison of optimizers in CBP.

Optimizer	x_1	x_2	x_3	x_4	$f(X)$
PSO	5.19642794	5.58575196	4.72007383	7.77687828	4.40909813
DE	8.16411474	4.69515161	4.12311453	6.80780304	3.30042563
CMA-ES	5.86738852	5.12413562	4.55120035	4.37475766	3.54101105
SaDE	6.04262963	5.31089695	4.50180426	3.50881375	2.11677404
CL-PSO	6.75876264	5.98042919	3.83356814	3.33441016	2.55785588
SHADE	6.01805616	5.20270572	4.57831375	3.49130715	2.19656988
L-SHADE	6.00146676	5.31260719	4.45353039	3.51616361	2.19892916
HPSO-TVAC	6.02725385	5.31167908	4.48474977	3.49722582	2.15287331
PPSO	5.60902493	5.95280589	3.57530868	13.66742071	2.54802429
WDE	5.97570832	5.40939575	4.61914452	3.43637063	2.09096829
DMDE	6.02035861	5.31260084	4.49549836	3.50451522	2.14095139
LLMOA	6.01773802	5.30937869	4.49471543	3.49932166	2.15250966

Table 13
Results outcomes & statistical analyses in GTP.

Optimizer	mean	std	best	worst
PSO	1.279767e-08 +	2.391043e-08	3.022592e-12	7.631617e-08
DE	3.517936e-11 +	4.573915e-11	8.943311e-13	1.543077e-10
CMA-ES	1.503739e-11 +	2.206908e-11	1.380495e-13	7.578548e-11
SaDE	3.350526e-11 +	4.190060e-11	1.584842e-12	1.280347e-10
CL-PSO	3.005213e-11 +	3.228653e-11	4.803331e-14	9.564966e-11
SHADE	8.702860e-13 +	8.349890e-13	1.076082e-15	2.292919e-12
L-SHADE	7.719467e-12 +	1.164320e-11	2.439210e-17	3.916808e-11
HPSO-TVAC	3.192705e-13 +	3.233009e-13	7.098397e-17	8.922830e-13
PPSO	0.000000e+00 ≈	0.000000e+00	0.000000e+00	0.000000e+00
WDE	4.780490e-12 +	6.447177e-12	1.343052e-14	2.258277e-11
DMDE	4.178890e-12 +	6.324141e-12	1.774107e-14	2.109789e-11
LLMOA	8.088906e-33	2.300112e-32	0.000000e+00	7.703720e-32

Table 14
The optimum comparison of optimizers in GTP.

Optimizer	x_1	x_2	x_3	x_4	$f(X)$
PSO	40.59375679	19.53999077	17.29090701	57.68641496	3.0225919151022713e-12
DE	49.82758739	25.98685987	12.	43.37687324	8.943310512549171e-13
CMA-ES	48.24440667	20.22794409	19.92197739	57.89379224	1.3804945605539826e-13
SaDE	58.26904767	16.77572134	30.01276918	59.8892728	1.5848417085849555e-12
CL-PSO	38.43193376	28.18216708	12.	60.99	4.8033312339420615e-14
SHADE	49.97601136	19.61393274	21.13250888	57.48446396	2.1071133157927097e-14
L-SHADE	50.71580923	21.49276358	12.8681276	37.79723721	1.1777347194797394e-15
HPSO-TVAC	50.83405252	20.06727721	22.27679042	60.95119995	7.098396948332857e-17
PPSO	60.34768776	20.04485818	17.32778702	39.89159062	0.0
WDE	48.72103367	25.44693545	14.93178225	54.05388858	1.7741066179337994e-14
DMDE	42.91966556	13.60115161	16.09926593	35.36071002	1.3430524359058748e-14
LLMOA	46.50925217	28.63912007	12.58377691	53.70654604	0.0

and statistical analyses of twelve well-known optimization algorithms alongside our proposed LLMOA. Table 24 provides a summary of the optimal solutions discovered by these optimizers for the Three-Bar Truss Design Problem. Our LLMOA attained the highest performance, as indicated by mean and optimum metrics, even with restricted computational resources. Moreover, the robustness of LLMOA is highlighted by the standard deviation and box plot results from multiple trial runs. Thus, LLMOA proves to be an outstanding optimizer for the Three-Bar

Truss Design Problem. Fig. 18 shows the boxplot and convergence curve for all compared algorithms.

4.3.9. Welded beam problem

The mathematical formulation of the Welded Beam Problem can be found in B.9. Table 25 displays the experimental results and statistical analyses of eleven renowned optimization algorithms alongside our

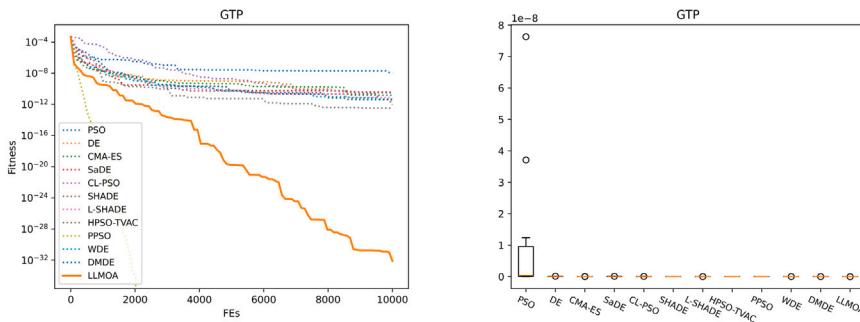


Fig. 13. Convergence curves and box plots of optimizers on GTP.

Table 15
Results outcomes & statistical analyses in CSP.

Optimizer	mean	std	best	worst
PSO	8.602518e-03 +	1.368032e-03	6.849755e-03	1.197090e-02
DE	6.076142e-03 +	1.440351e-14	6.076142e-03	6.076142e-03
CMA-ES	6.076142e-03 +	1.319069e-11	6.076142e-03	6.076142e-03
SaDE	6.076142e-03 +	5.592890e-11	6.076142e-03	6.076142e-03
CL-PSO	6.285306e-03 +	2.061086e-04	6.113929e-03	6.785599e-03
SHADE	6.076157e-03 +	1.943473e-08	6.076142e-03	6.076194e-03
L-SHADE	6.076169e-03 +	1.930417e-08	6.076142e-03	6.076201e-03
HPSO-TVAC	6.076142e-03 ≈	1.146152e-16	6.076142e-03	6.076142e-03
PPSO	6.076142e-03 +	0.000000e+00	6.076142e-03	6.076142e-03
WDE	6.076142e-03 +	1.017374e-14	6.076142e-03	6.076142e-03
DMDE	6.076348e-03 +	2.860861e-07	6.076155e-03	6.077172e-03
LLMOA	6.076142e-03	0.000000e+00	6.076142e-03	6.076142e-03

Table 16
The optimum comparison of optimizers in CSP.

Optimizer	x_1	x_2	x_3	$f(X)$
PSO	0.05068653	0.60091989	2.43683482	0.00684975510259421
DE	0.05	0.60761419	2.	0.006076141915888035
CMA-ES	0.05	0.60761419	2.	0.006076141918323448
SaDE	0.05	0.60761419	2.	0.006076141923882993
CL-PSO	0.05	0.61139288	2.	0.006113928754987095
SHADE	0.05	0.6076142	2.00000002	0.00607614201748021
L-SHADE	0.05	0.60761426	2.	0.006076142571563121
HPSO-TVAC	0.05	0.60761419	2.	0.006076141915887938
PPSO	0.05	0.60761419	2.	0.00607614191588792
WDE	0.05	0.60761551	2.	0.006076155136502324
DMDE	0.05	0.60761419	2.	0.0060761419158901035
LLMOA	0.05	0.60761419	2.	0.00607614191588792

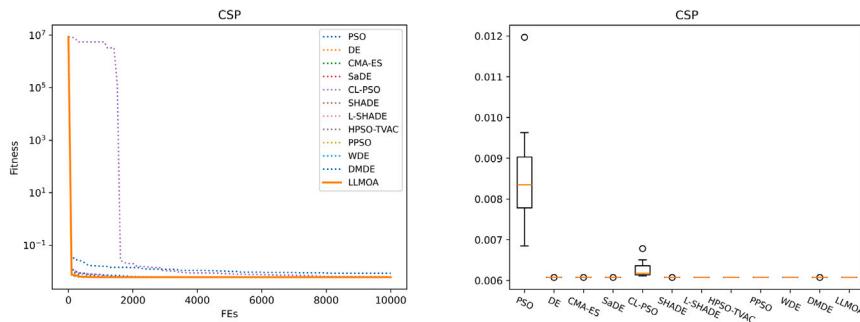


Fig. 14. Convergence curves and box plots of optimizers on CSP.

proposed LLMOA. Table 26 provides a summary of the optimal solutions discovered by these optimizers for the Welded Beam Problem. Our LLMOA demonstrated the second-best performance, following WDE, based on evaluations of mean and optimum metrics, despite operating within constrained computational budgets. Additionally, the robustness of LLMOA is underscored by the standard deviation and box plot analyses across multiple trial runs. Therefore, LLMOA emerges as a highly effective optimizer for tackling the challenges posed by the

Welded Beam Problem. Fig. 19 shows the boxplot and convergence curve for all compared algorithms.

4.3.10. Tubular column design problem

The mathematical formulation of Tubular Column Design Problem can be founded in B.10. Table 27 presents the experimental outcomes and statistical analyses of twelve well-known optimization algorithms, including our proposed LLMOA. Table 28 summarizes the optimal

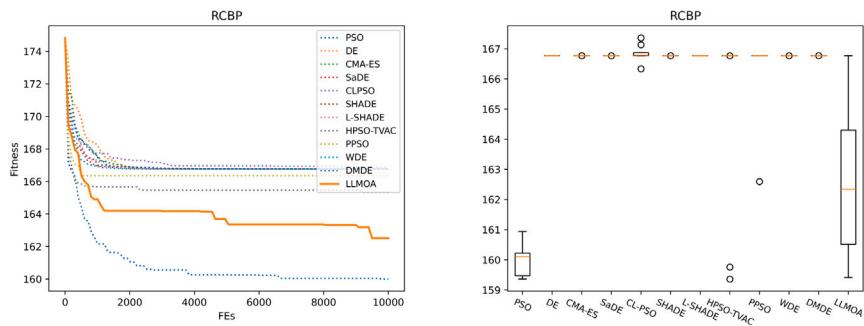


Fig. 15. Convergence curves and box plots of optimizers on RCBP.

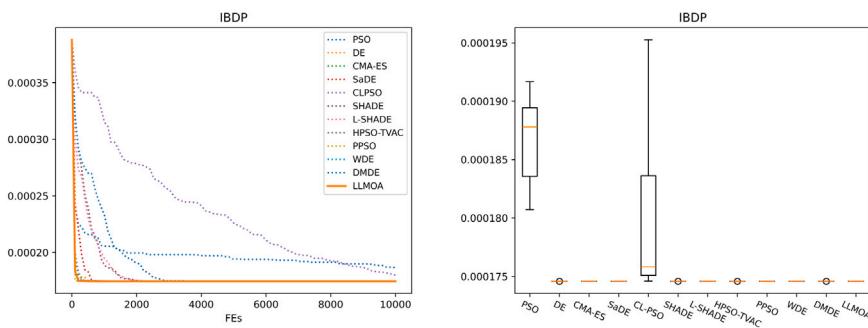


Fig. 16. Convergence curves and box plots of optimizers on IBDP.

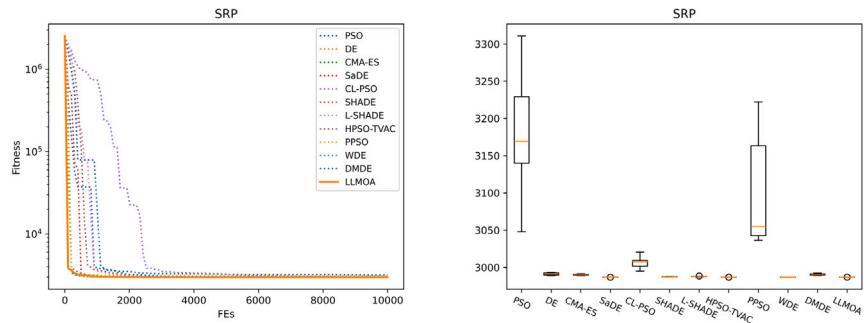


Fig. 17. Convergence curves and box plots of optimizers on SRP.

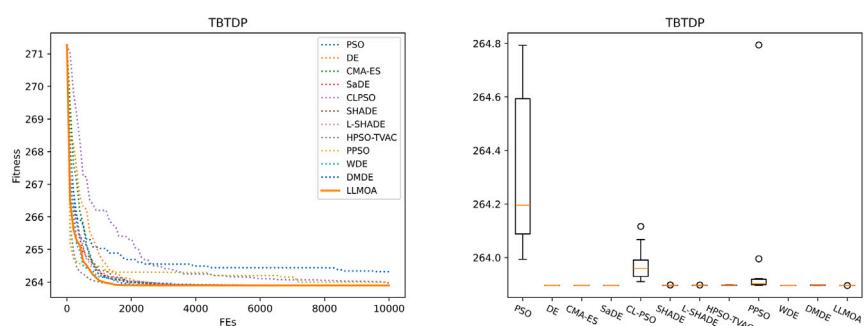


Fig. 18. Convergence curves and box plots of optimizers on TBTDP.

Table 17
Results outcomes & statistical analyses in RCBP.

Optimizer	mean	std	best	worst
PSO	1.599965e+02 –	5.121515e-01	1.593646e+02	1.609372e+02
DE	1.667700e+02 +	1.220421e-11	1.667700e+02	1.667700e+02
CMA-ES	1.667704e+02 +	1.003925e-03	1.667700e+02	1.667734e+02
SaDE	1.667700e+02 +	4.138899e-07	1.667700e+02	1.667700e+02
CL-PSO	1.668469e+02 +	2.515921e-01	1.663385e+02	1.673640e+02
SHADE	1.667701e+02 +	1.184150e-04	1.667700e+02	1.667704e+02
L-SHADE	1.667702e+02 +	1.669845e-04	1.667700e+02	1.667705e+02
HPSO-TVAC	1.653279e+02 +	2.885535e+00	1.593600e+02	1.667700e+02
PPSO	1.663525e+02 +	1.252455e+00	1.625951e+02	1.667700e+02
WDE	1.667700e+02 +	2.820370e-07	1.667700e+02	1.667700e+02
DMDE	1.667706e+02 +	4.584065e-04	1.667701e+02	1.667716e+02
LLMOA	1.625206e+02	2.344671e+00	1.594122e+02	1.667700e+02

Table 18
The optimum comparison of optimizers in RCBP.

Optimizer	x_1	x_2	x_3	$f(X)$
PSO	8.4	30.	7.50025551	159.36459911381905
DE	7.8	31.	7.75	166.77000000000035
CMA-ES	7.8	31.	7.75	166.77000001434595
SaDE	7.8	31.	7.75	166.7700000337324
CL-PSO	8.4	30.	7.88769483	166.33850694717228
SHADE	7.8	31.	7.75000008	166.77000156727476
L-SHADE	7.8	31.	7.75000005	166.7700009770908
HPSO-TVAC	8.4	30.	7.5	159.36
PPSO	8.4	30.	7.67973051	162.5951491432242
WDE	7.8	31.	7.7500047	166.77008736959536
DMDE	7.8	31.	7.75	166.77000000000868
LLMOA	8.4	30.	7.50290116	159.41222091784823

Table 19
Results outcomes & statistical analyses in IBDP.

Optimizer	mean	std	best	worst
PSO	1.8680355189e-04 +	3.853382e-06	1.807342e-04	1.916762e-04
DE	1.7458205142e-04 ≈	2.609697e-19	1.745821e-04	1.745821e-04
CMA-ES	1.7458205144e-04 +	1.368768e-14	1.745821e-04	1.745821e-04
SaDE	1.7458205143e-04 +	4.873485e-15	1.745821e-04	1.745821e-04
CL-PSO	1.8017328762e-04 +	7.453386e-06	1.745946e-04	1.952676e-04
SHADE	1.7458205626e-04 +	2.806117e-12	1.745821e-04	1.745821e-04
L-SHADE	1.7458205628e-04 +	3.789258e-12	1.745821e-04	1.745821e-04
HPSO-TVAC	1.7458205142e-04 ≈	3.804376e-19	1.745821e-04	1.745821e-04
PPSO	1.7458205142e-04 ≈	0.0000000e+00	1.745821e-04	1.745821e-04
WDE	1.7458205142e-04 ≈	3.428548e-20	1.745821e-04	1.745821e-04
DMDE	1.7458210775e-04 +	6.232631e-11	1.745821e-04	1.745823e-04
LLMOA	1.7458205142e-04	0.0000000e+00	1.745821e-04	1.745821e-04

Table 20
The optimum comparison of optimizers in IBDP.

Optimizer	x_1	x_2	x_3	x_4	$f(X)$
PSO	49.7098142	79.36203483	0.94161218	4.97989953	0.00018073420864426573
DE	50.	80.	1.76470588	5.	0.00017458205142468371
CMA-ES	50.	80.	1.76470588	5.	0.00017458205142949847
SaDE	50.	80.	1.76470588	5.	0.00017458205142495715
CL-PSO	50.	80.	1.75749403	5.	0.00017459461808586437
SHADE	50.	80.	1.76470588	5.	0.00017458205142947068
L-SHADE	50.	80.	1.76470588	5.	0.00017458205142537075
HPSO-TVAC	50.	80.	1.76470588	5.	0.00017458205142468377
PPSO	50.	80.	1.76470588	5.	0.0001745820514246837
WDE	50.	80.	1.76470483	5.	0.00017458205326583275
DMDE	50.	80.	1.76470588	5.	0.0001745820514246837
LLMOA	50.	80.	1.76470588	5.	0.0001745820514246837

solutions these optimizers found for the Tubular Column Design Problem. Our LLMOA achieved the best performance, as shown by the mean and optimum metrics, even with limited computational resources. Additionally, the robustness of LLMOA is evident from the standard deviation and box plot results across multiple trial runs. Therefore, LLMOA proves to be an excellent optimizer for the Tubular Column Design Problem. Fig. 20 shows the boxplot and convergence curve for all compared algorithms.

Based on the experimental results and statistical analyses, the competitiveness of LLMOA across diverse engineering optimization tasks is evident. LLMOA emerges as the top-performing algorithm in seven out of ten instances. In the meantime, it is worth noting that DE-based optimizers (i.e., SaDE, SHADE, L-SHADE, WDE, and DMDE), renowned for their remarkable performance across CEC benchmark functions, do not consistently perform well in real-world applications. This phenomenon can be attributed to the No Free Lunch Theorem (NFLT) [67],

Table 21
Results outcomes & statistical analyses in SRP.

Optimizer	mean	std	best	worst
PSO	3.180525e+03 +	7.071867e+01	3.048101e+03	3.310799e+03
DE	2.991316e+03 +	1.631451e+00	2.989104e+03	2.993615e+03
CMA-ES	2.990167e+03 +	7.711532e-01	2.989118e+03	2.991346e+03
SaDE	2.987004e+03 +	2.771877e-02	2.986942e+03	2.987035e+03
CL-PSO	3.007082e+03 +	7.340675e+00	2.995176e+03	3.020619e+03
SHADE	2.987522e+03 +	1.638167e-01	2.987341e+03	2.987861e+03
L-SHADE	2.987908e+03 +	2.673894e-01	2.987504e+03	2.988507e+03
HPSO-TVAC	2.986898e+03 ≈	2.297782e-02	2.986885e+03	2.986963e+03
PPSO	3.098980e+03 +	7.524922e+01	3.036622e+03	3.222263e+03
WDE	2.986886e+03 ≈	6.732939e-04	2.986885e+03	2.986888e+03
DMDE	2.990546e+03 +	1.092935e+00	2.989154e+03	2.992486e+03
LLMOA	2.986901e+03	4.271065e-02	2.986884e+03	2.987029e+03

Table 22
The optimum comparison of optimizers in SRP.

Optimizer	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	$f(X)$
PSO	3.53410854	0.70202854	17.00333832	7.52034546	7.93658781	3.36311147	5.31830989	3048.101418063313	
DE	3.50242163	0.7	17.	7.3	7.72478907	3.32176236	5.28769841	2989.1043739329925	
CMA-ES	3.50145867	0.70002789	17.	7.3	7.72411234	3.32289009	5.28769607	2989.118272812213	
SaDE	3.50002896	0.70000001	17.	7.30174443	7.71580711	3.32025004	5.28665747	2986.9417324348233	
CL-PSO	3.5006262	0.7	17.	7.68365973	7.86783845	3.32127762	5.28837913	2995.1759610367185	
SHADE	3.50014854	0.7	17.0001793	7.3	7.71670091	3.32025682	5.28668848	2987.0455783927305	
L-SHADE	3.50020137	0.7	17.00027968	7.3	7.71633953	3.32024844	5.28668945	2987.0742745231055	
HPSO-TVAC	3.50000057	0.70000004	17.	7.30001461	7.71532597	3.32017814	5.28665449	2986.885260833517	
PPSO	3.6	0.7	17.	7.4848948	8.10801988	3.32066563	5.2867905	3036.6218705032784	
WDE	3.50304202	0.700004874	17.	7.3	7.73709191	3.32044795	5.28713899	2989.154107450529	
DMDE	3.50000047	0.7	17.00000132	7.3	7.71533962	3.32017736	5.28665453	2986.8852823098855	
LLMOA	3.50000012	0.7	17.	7.30000056	7.71532316	3.32017684	5.28665447	2986.8843932240543	

Table 23
Results outcomes & statistical analyses in TBTDP.

Optimizer	mean	std	best	worst
PSO	2.643224e+02 +	2.780816e-01	2.639937e+02	2.647934e+02
DE	2.638958e+02 +	2.799473e-06	2.638958e+02	2.638959e+02
CMA-ES	2.638958e+02 +	1.216568e-06	2.638958e+02	2.638958e+02
SaDE	2.638959e+02 +	4.488338e-05	2.638959e+02	2.638960e+02
CL-PSO	2.639776e+02 +	6.350277e-02	2.639101e+02	2.641168e+02
SHADE	2.638967e+02 +	6.745390e-04	2.638960e+02	2.638982e+02
L-SHADE	2.638966e+02 +	6.795873e-04	2.638959e+02	2.638979e+02
HPSO-TVAC	2.638966e+02 +	7.384675e-04	2.638959e+02	2.638981e+02
PPSO	2.640022e+02 +	2.656705e-01	2.638970e+02	2.647947e+02
WDE	2.638958e+02 +	1.528077e-08	2.638958e+02	2.638958e+02
DMDE	2.638971e+02 +	7.022771e-04	2.638961e+02	2.638980e+02
LLMOA	2.638958e+02	3.448007e-11	2.638958e+02	2.638958e+02

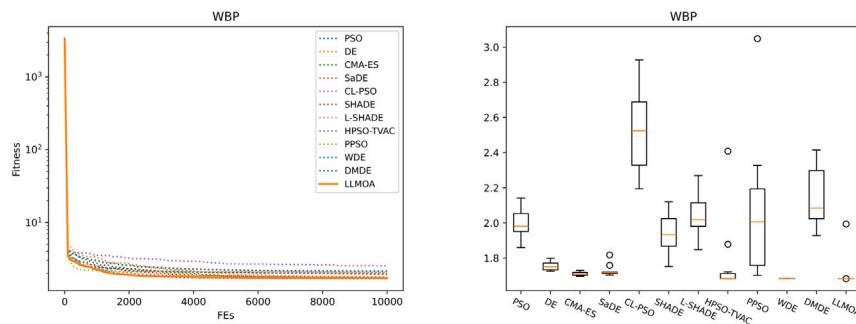


Fig. 19. Convergence curves and box plots of optimizers on WBP.

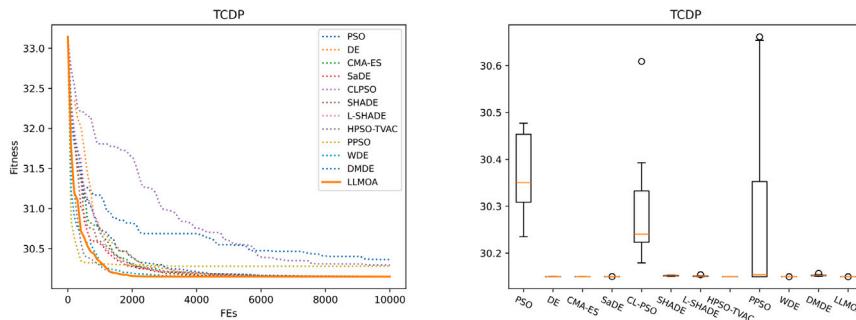


Fig. 20. Convergence curves and box plots of optimizers on TCDP.

Table 24
The optimum comparison of optimizers in TBTDP.

Optimizer	x_1	x_2	$f(X)$
PSO	0.78215048	0.42768147	263.9937106457624
DE	0.78865292	0.40831113	263.8958447688327
CMA-ES	0.78867817	0.40823972	263.89584358081225
SaDE	0.78862096	0.40840184	263.89587591223295
CL-PSO	0.79093741	0.40199261	263.9101430571122
SHADE	0.78872814	0.40809848	263.89585450700963
L-SHADE	0.78853189	0.40865363	263.8958618700631
HPSO-TVAC	0.78873508	0.40807923	263.89589287538934
PPSO	0.78741925	0.4118121	263.8970074685495
WDE	0.78891785	0.4075642	263.8960850338838
DMDE	0.78867495	0.40824882	263.8958433774264
LLMOA	0.78867513	0.4082483	263.89584337646835

which suggests that no single optimization algorithm can universally outperform others across all problem domains. Therefore, the further development of LLMOA and hyper-heuristic algorithms may provide an avenue for promoting a general methodology for optimization.

5. Conclusion & future work

This paper presents a pioneering approach termed large language model assisted hyper-heuristic algorithm (LLMOA) designed for tackling complex optimization challenges. In the high-level component, LLM Gemini and prompt engineering are employed to intelligently construct optimization sequences, while in the low-level heuristics (LLHs), we employ ten efficient and easily implemented search operators to update solutions. Our experimental results and statistical analyses confirm the competitiveness of LLMOA across various optimization tasks, including CEC2014, CEC2020, CEC2022, and engineering optimization problems when competing against state-of-the-art optimizers. Moreover, ablation experiments underscore the significance of integrating Gemini and prompt engineering into the high-level component of LLMOA.

Furthermore, we identify certain limitations of LLMOA, such as the extra computational resources and relative inferiority compared to SHADE in specific benchmark functions of CEC2014 and CEC2020. Nevertheless, this research presents a promising avenue by introducing LLM to the EC community, thereby contributing to the advancement of general optimization techniques. Future research should focus on reducing the computational overhead associated with LLMOA by optimizing the implementation of LLM Gemini and the prompt engineering process, as well as exploring more efficient search operators in low-level heuristics.

Investigating the scalability of LLMOA to handle larger and more complex optimization problems, including high-dimensional optimization tasks and large-scale real-world problems, is another critical area. Combining LLMOA with other optimization techniques, such as evolutionary algorithms or machine learning models, could lead to hybrid approaches that leverage the strengths of multiple methods, providing more robust and versatile solutions to diverse optimization challenges.

CRediT authorship contribution statement

Rui Zhong: Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Abdelazim G. Hussien:** Writing – original draft, Visualization, Methodology, Formal analysis, Data curation. **Jun Yu:** Writing – review & editing, Writing – original draft, Validation. **Masaharu Munetomo:** Writing – review & editing, Supervision, Resources.

Declaration of competing interest

Authors has no conflict of interest.

Acknowledgements

This work was supported by JSPS, Japan KAKENHI Grant Number 21A402 and 24K15098, and JST SPRING Grant Number JPMJSP2119

Appendix A. Functions details

Tables 29, 30, and 31 summarize the basic information of CEC2014, CEC2020, and CEC2022 benchmark functions. The mathematical models of engineering optimization problems are demonstrated from Eq. (13) to Eq. (20). Note that all tasks are minimization problems.

Appendix B. Engineering problem details

B.1. Corrugated Bulkhead Problem (CBHP)

: The mathematical model of CBHP is presented in Eq. (12).

$$\begin{aligned} \min f(X) &= \frac{5.885x_4(x_1 + x_3)}{x_1 + \sqrt{|x_3^2 - x_2^2|}} \\ \text{s.t. } g_1(X) &= -x_4x_2(0.4x_1 + \frac{x_3}{6}) + 8.94(x_1 + \sqrt{|x_3^2 - x_2^2|}) \leq 0 \\ g_2(X) &= -x_4x_2^2(0.2x_1 + \frac{x_3}{12}) + 2.2(8.94(x_1 + \sqrt{|x_3^2 - x_2^2|}))^{4/3} \leq 0 \\ g_3(X) &= -x_4 + 0.0156x_1 + 0.15 \leq 0 \\ g_4(X) &= -x_4 + 0.0156x_3 + 0.15 \leq 0 \\ g_5(X) &= -x_4 + 1.05 \leq 0 \\ g_6(X) &= -x_3 + x_2 \leq 0 \end{aligned} \quad (12)$$

where $0 \leq x_1, x_2, x_3 \leq 100$

$$0 \leq x_4 \leq 5$$

B.2. Cantilever Beam Problem (CBP)

: The mathematical model of CBP is presented in Eq. (13).

$$\begin{aligned} \min f(X) &= 0.0624(x_1 + x_2 + x_3 + x_4 + x_5) \\ \text{s.t. } g(X) &= \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \\ \text{where } 0.01 &\leq x_i \leq 100, i \in \{1, 2, 3, 4, 5\} \end{aligned} \quad (13)$$

Table 25
Results outcomes & statistical analyses in WBP.

Optimizer	mean	std	best	worst
PSO	2.000291e+00 +	8.616484e-02	1.859097e+00	2.141591e+00
DE	1.754236e+00 +	2.423020e-02	1.725620e+00	1.798515e+00
CMA-ES	1.710787e+00 ≈	1.025549e-02	1.696358e+00	1.730076e+00
SaDE	1.728483e+00 +	3.333475e-02	1.702927e+00	1.818044e+00
CL-PSO	2.521648e+00 +	2.308713e-01	2.194454e+00	2.927043e+00
SHADE	1.940363e+00 +	1.211720e-01	1.752666e+00	2.120733e+00
L-SHADE	2.048445e+00 +	1.192010e-01	1.847187e+00	2.269164e+00
HPSO-TVAC	1.779222e+00 +	2.174934e-01	1.682675e+00	2.408063e+00
PPSO	2.071612e+00 +	3.862458e-01	1.701872e+00	3.048503e+00
WDE	1.683521e+00 ≈	2.947096e-04	1.683043e+00	1.683949e+00
DMDE	2.145979e+00 +	1.635062e-01	1.928014e+00	2.414697e+00
LLMOA	1.713857e+00	9.352205e-02	1.682574e+00	1.994423e+00

Table 26
The optimum comparison of optimizers in WBP.

Optimizer	x_1	x_2	x_3	x_4	$f(X)$
PSO	0.16527366	4.20431689	9.22361616	0.21443433	1.8590965518281852
DE	0.18609057	3.63749773	9.07619527	0.20599382	1.725619795232447
CMA-ES	0.19860549	3.30178479	9.06354019	0.20578008	1.69635822494173
SaDE	0.19726942	3.3222856	9.08325152	0.20609673	1.7029272070577661
CL-PSO	0.26309537	2.89779088	6.94111401	0.34962597	2.1944542450287234
SHADE	0.20917615	3.14994706	8.9437908	0.2103535	1.7045345317429093
L-SHADE	0.20497262	3.23436898	8.91462349	0.21164144	1.7144688256472622
HPSO-TVAC	0.2056887	3.16115773	9.03667455	0.20573012	1.6826754811644884
PPSO	0.21066508	3.11579784	8.93014199	0.21066508	1.701871699261414
WDE	0.25785105	2.73092398	7.83205181	0.2740121	1.9280142218261207
DMDE	0.20565688	3.16219329	9.0352076	0.20580001	1.6830429868515826
LLMOA	0.20572942	3.1600682	9.03662184	0.20572975	1.6825738592162611

Table 27
Results outcomes & statistical analyses in TCDP.

Optimizer	mean	std	best	worst
PSO	3.0362921692e+01 +	8.686667e-02	3.023512e+01	3.047706e+01
DE	3.0149903875e+01 +	1.115591e-04	3.014976e+01	3.015010e+01
CMA-ES	3.0149791778e+01 +	3.600231e-05	3.014975e+01	3.014987e+01
SaDE	3.0149883972e+01 +	8.500704e-05	3.014978e+01	3.015008e+01
CL-PSO	3.0294341462e+01 +	1.221178e-01	3.017909e+01	3.060910e+01
SHADE	3.0151906211e+01 +	9.003242e-04	3.015037e+01	3.015323e+01
L-SHADE	3.0151086193e+01 +	1.126875e-03	3.014997e+01	3.015402e+01
HPSO-TVAC	3.0149738069e+01 +	2.306449e-08	3.014974e+01	3.014974e+01
PPSO	3.0281224253e+01 +	2.031596e-01	3.014974e+01	3.066169e+01
WDE	3.0149738496e+01 +	2.448756e-07	3.014974e+01	3.014974e+01
DMDE	3.0152751255e+01 +	1.634916e-03	3.015069e+01	3.015677e+01
LLMOA	3.0149738037e+01	2.087366e-11	3.014974e+01	3.014974e+01

Table 28
The optimum comparison of optimizers in TCDP.

Optimizer	x_1	x_2	$f(X)$
PSO	7.11803655	0.22935486	30.235124953007073
DE	7.10298263	0.2290471	30.14975774979825
CMA-ES	7.10297499	0.22904742	30.149746963115557
SaDE	7.1029716	0.22904804	30.149776224179085
CL-PSO	7.09587693	0.22990268	30.179092991632604
SHADE	7.10297832	0.22904801	30.149802450373507
L-SHADE	7.10298896	0.22904743	30.149807613380087
HPSO-TVAC	7.10297476	0.2290473	30.149738044011414
PPSO	7.10297477	0.2290473	30.149738070728816
WDE	7.10337102	0.22903679	30.150688475401306
DMDE	7.10297477	0.2290473	30.149738180804253
LLMOA	7.10297476	0.2290473	30.149738037056824

B.3. Gear Train Design Problem (GTP)

: The mathematical model of GTP is presented in Eq. (14).

$$\min f(X) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4} \right)^2 \quad (14)$$

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

B.4. Compression Spring Design Problem (CSP)

: The mathematical model of CSP is presented in Eq. (15).

$$\begin{aligned} \min f(X) &= (x_3 + 2)x_2 x_1^2 \\ \text{s.t. } g_1(X) &= 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \\ g_2(X) &= \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} \leq 0 \\ g_3(X) &= 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0 \\ g_4(X) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0 \end{aligned} \quad (15)$$

where $0.05 \leq x_1 \leq 2$

$$0.25 \leq x_2 \leq 1.3$$

$$2 \leq x_3 \leq 15$$

Table 29
Summary of the CEC2014 Suite.

Num.	Function	Category	Optimal
f_1	Ro High Conditioned Elliptic Function	Uni.	100
f_2	Ro Bent Cigar Function		200
f_3	Ro Discus Function		300
f_4	Sh & Ro Rosenbrock's Function	Multi.	400
f_5	Sh & Ro Ackley's Function		500
f_6	Sh & Ro Weierstrass Function		600
f_7	Sh & Ro Griewank's Function		700
f_8	Shifted Rastrigin's Function		800
f_9	Sh & Ro Rastrigin's Function		900
f_{10}	Shifted Schwefel's Function		1000
f_{11}	Sh & Ro Schwefel's Function		1100
f_{12}	Sh & Ro Katsuura Function		1200
f_{13}	Sh & Ro HappyCat Function		1300
f_{14}	Sh & Ro HGBat Function		1400
f_{15}	Sh & Ro Expanded Griewank's plus Rosenbrock's Function		1500
f_{16}	Sh & Ro Expanded Scaffer's F6 Function		1600
f_{17}	Hfunction 1 (n = 3)	Hybrid.	1700
f_{18}	Hfunction 2 (n = 3)		1800
f_{19}	Hfunction 3 (n = 4)		1900
f_{20}	Hfunction 4 (n = 4)		2000
f_{21}	Hfunction 5 (n = 5)		2100
f_{22}	Hfunction 6 (n = 5)		2200
f_{23}	CFunction 1 (n = 5)	Comp.	2300
f_{24}	CFunction 2 (n = 3)		2400
f_{25}	CFunction 3 (n = 3)		2500
f_{26}	CFunction 4 (n = 5)		2600
f_{27}	CFunction 5 (n = 5)		2700
f_{28}	CFunction 6 (n = 5)		2800
f_{29}	CFunction 7 (n = 3)		2900
f_{30}	CFunction 8 (n = 3)		3000

Search space: $[-100, 100]^D$

Sh: Shifted

Ro: Rotated

Hfunction: Hybrid function

CFunction: Composition function

Table 30

Summary of the CEC2020 benchmark functions.

Num.	Function	Category	Optimal
f_1	Sh & Ro Bent Cigar Function	Uni.	100
f_2	Sh & Ro Schwefel's function		1100
f_3	Sh & Ro Lunacek bi-Rastrigin function	Multi.	700
f_4	Expanded Rosenbrock's plus Griewangk's function		1900
f_5	Hfunction 1 (N = 3)		1700
f_6	Hfunction 2 (N = 4)	Hybrid.	1600
f_7	Hfunction 3 (N = 5)		2100
f_8	CFunction 1 (N = 3)		2200
f_9	CFunction 2 (N = 4)	Comp.	2400
f_{10}	CFunction 3 (N = 5)		2500

Search range: $[-100, 100]^D$

Sh: Shifted

Ro: Rotated

Hfunction: Hybrid function

CFunction: Composition function

B.5. Reinforced Concrete Beam Problem (RCBP)

: The mathematical model of RCBP is presented in Eq. (16).

$$\min f(X) = 2.9x_1 + 0.6x_2x_3$$

$$\text{s.t. } g_1(X) = \frac{x_2}{x_3} - 4 \leq 0$$

$$g_2(X) = 180 + 7.375 \frac{x_1^2}{x_3} - x_1x_2 \leq 0 \quad (16)$$

where $x_1 \in \{6.0, 6.16, 6.32, 6.6, 7.0, 7.11, 7.2, 7.8, 7.9, 8.0, 8.4\}$

$$x_2 \in \{28, 29, \dots, 39, 40\}$$

$$5 \leq x_3 \leq 10$$

Table 31

Summary of the CEC2022 benchmark functions.

Fun.	Description	Feature	Optimum
f_1	Sh & full Ro Zakharov	Uni.	300
f_2	Sh & full Ro Rosenbrock		400
f_3	Sh & full Ro Expanded Schaffer f_6	Basic.	600
f_4	Sh & full Ro Non-Continuous Rastrigin		800
f_5	Sh & full Ro Levy		900
f_6	Hfunction 1 (N = 3)		1800
f_7	Hfunction 2 (N = 6)	Hybrid.	2000
f_8	Hfunction 3 (N = 5)		2200
f_9	CFunction 1 (N = 5)		2300
f_{10}	CFunction 2 (N = 4)	Comp.	2400
f_{11}	CFunction 3 (N = 5)		2600
f_{12}	CFunction 3 (N = 6)		2700

Search range: $[-100, 100]^D$

Sh: Shifted

Ro: Rotated

Hfunction: Hybrid function

CFunction: Composition function

B.6. I-Beam Design Problem (IBDP)

: The mathematical model of IBDP is presented in Eq. (17).

$$\begin{aligned} \min f(X) &= \frac{5000}{x_3(x_2 - 2x_4)^3/12 + (x_1x_4^3/6) + 2x_1x_4(x_2 - x_4/2)^2} \\ \text{s.t. } g_1(X) &= 2x_1x_3 + x_3(x_2 - 2x_4) - 300 \leq 0 \\ g_2(X) &= \frac{180000x_2}{x_3(x_2 - 2x_4)^3 + 2x_1x_3(4x_4^2 + 3x_2(x_2 - 2x_4))} + \\ &\quad \frac{15000x_1}{(x_2 - 2x_4)x_3^2 + 2x_3x_1^3} - 56 \leq 0 \end{aligned} \quad (17)$$

where $10 \leq x_1 \leq 50$

$10 \leq x_2 \leq 80$

$0.9 \leq x_3, x_4 \leq 5$

B.7. Speed Reducer Problem (SRP)

: The mathematical model of SRP is presented in Eq. (18).

$$\begin{aligned} \min f(X) &= 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0924) \\ &\quad - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \\ \text{s.t. } g_1(X) &= \frac{27}{x_1x_2^2x_3} - 1 \leq 0 \\ g_2(X) &= \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0 \\ g_3(X) &= \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leq 0 \\ g_4(X) &= \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leq 0 \\ g_5(X) &= \frac{\sqrt{(745x_4/x_2x_3)^2 + 16900000}}{110x_6^3} - 1 \leq 0 \\ g_6(X) &= \frac{\sqrt{(745x_5/x_2x_3)^2 + 157500000}}{85x_7^3} - 1 \leq 0 \\ g_7(X) &= \frac{x_2x_3}{40} - 1 \leq 0 \\ g_8(X) &= \frac{5x_2}{x_1} - 1 \leq 0 \\ g_9(X) &= \frac{x_1}{12x_2} - 1 \leq 0 \\ g_{10}(X) &= \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0 \\ g_{11}(X) &= \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0 \end{aligned}$$

where $2.6 \leq x_1 \leq 2.6$

$0.7 \leq x_2 \leq 0.8$

$17 \leq x_3 \leq 28$

$7.3 \leq x_4x_5 \leq 8.3$

$2.9 \leq x_6 \leq 3.9$

$5 \leq x_7 \leq 5.5$

B.8. Three-Bar Truss Design Problem (TBTDP)

: The mathematical model of TBTDP is presented in Eq. (19).

$$\begin{aligned} \min f(X) &= (2\sqrt{2}x_1 + x_2) \cdot l \\ \text{s.t. } g_1(X) &= \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \\ g_2(X) &= \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \\ g_3(X) &= \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0 \\ l &= 100 \text{ cm}, P = 2 \text{ kN/cm}^3, \sigma = 2 \text{ kN/cm}^3 \end{aligned} \quad (19)$$

where $0 \leq x_1, x_2 \leq 1$

B.9. Welded Beam Problem (WBP)

: The mathematical model of WBP is presented in Eq. (20). Some implicit constraints can be found in [72].

$$\begin{aligned} \min f(X) &= 1.10471x_1x_2 + 0.04811x_3x_4(14 + x_2) \\ \text{s.t. } g_1(X) &= \tau(X) - \tau_{max} \leq 0 \\ g_2(X) &= \sigma(X) - \sigma_{max} \leq 0 \\ g_3(X) &= \delta(X) - \delta_{max} \leq 0 \\ g_4(X) &= x_1 - x_4 \leq 0 \\ g_5(X) &= P - P_c(X) \leq 0 \\ g_6(X) &= 0.125 - x_1 \leq 0 \\ g_7(X) &= 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0 \end{aligned} \quad (20)$$

where $0.1 \leq x_1, x_4 \leq 2$

$0.1 \leq x_2, x_3 \leq 10$

B.10. Tubular Column Design Problem (TCDP)

: The mathematical model of TCDP is presented in Eq. (21).

$$\begin{aligned} \min f(X) &= 9.8x_1x_2 + 2x_1 \\ \text{s.t. } g_1(X) &= \frac{P}{\pi x_1 x_2 \sigma_y} - 1 \leq 0 \\ g_2(X) &= \frac{8PL^2}{\pi^3 E x_1 x_2 (x_1^2 + x_2^2)} - 1 \leq 0 \\ \text{where } 2 \leq x_1 &\leq 14 \\ 0.2 \leq x_2 &\leq 0.8 \end{aligned} \quad (21)$$

Appendix C. Detailed experimental results on CEC2014

Tables 32 and 33 summarize the experimental results of optimizers on CEC2014. Convergence curves, proportion changes between exploration and exploitation, and box plots on CEC2014 representative functions are presented in Figs. 21, 22, 23, 24, 25, and 26.

Appendix D. Detailed experimental results on CEC2020

Tables 34 and 35 summarize the experimental results of optimizers on CEC2020. Convergence curves, proportion changes between exploration and exploitation, and box plots on CEC2020 are presented in Figs. 27, 28, 29, 30, 31, and 32.

Table 32

Results and analyses on 30-D CEC2014 benchmark functions.

F	PSO	DE	CMA-ES	SaDE	CL-PSO	SHADE	L-SHADE	HPSO-TVAC	PPSO	WDE	DMDE	LLMOA	
f_1	mean	3.004e+08 +	2.986e+08 +	1.521e+07 +	4.128e+07 +	7.279e+07 +	2.906e+07 +	3.186e+07 +	9.946e+06 +	1.233e+08 +	4.085e+07 +	4.650e+07 +	2.183e+06
	std	1.556e+08	9.976e+07	2.977e+06	1.128e+07	1.829e+07	5.068e+06	7.963e+06	7.045e+06	5.638e+07	1.487e+07	9.045e+06	7.172e+05
f_2	mean	6.953e+09 +	4.020e+10 +	3.220e+09 +	7.415e+04 +	4.695e+07 +	3.203e+04 ≈	3.580e+04 ≈	9.555e+05 +	1.810e+09 +	4.593e+04 +	6.060e+05 +	2.257e+04
	std	2.768e+09	3.634e+09	7.148e+08	3.500e+04	1.251e+07	9.148e+03	1.570e+04	1.117e+06	8.369e+08	4.210e+04	3.640e+05	1.435e+04
f_3	mean	8.425e+04 +	6.828e+04 +	5.850e+03 +	9.884e+02 +	7.717e+03 +	7.408e+02 +	8.063e+02 +	1.904e+04 +	7.850e+04 +	4.492e+02 +	1.222e+03 +	3.991e+02
	std	2.972e+04	4.889e+03	8.303e+02	1.175e+02	1.271e+03	1.244e+02	2.038e+02	9.626e+03	2.075e+04	4.824e+01	3.128e+02	1.132e+02
f_4	mean	1.855e+03 +	4.162e+03 +	8.139e+02 +	5.231e+02 +	6.032e+02 +	5.014e+02 ≈	5.006e+02 ≈	5.297e+02 +	7.419e+02 +	4.933e+02 ≈	5.481e+02 +	5.026e+02
	std	1.216e+03	6.080e+02	5.729e+01	3.975e+01	1.310e+01	2.787e+01	2.821e+01	3.148e+01	5.525e+01	2.255e+01	2.132e+01	2.585e+01
f_5	mean	5.210e+02 +	5.209e+02 +	5.210e+02 +	5.209e+02 +	5.209e+02 +	5.207e+02 +	5.210e+02 +	5.205e+02 +	5.210e+02 +	5.207e+02 +	5.203e+02	5.203e+02
	std	5.584e-02	6.546e-02	3.964e-02	7.134e-02	7.994e-02	3.667e-02	4.519e-02	4.099e-02	1.085e-01	4.527e-02	4.908e-02	1.973e-01
f_6	mean	6.400e+02 +	6.374e+02 +	6.378e+02 +	6.293e+02 ≈	6.271e+02 ≈	6.244e+02 ≈	6.246e+02 +	6.167e+02 +	6.379e+02 +	6.147e+02 -	6.253e+02 ≈	6.284e+02
	std	6.449e-01	2.271e+00	2.398e+00	1.063e+00	1.218e+00	1.689e+00	9.873e-01	2.363e+00	2.741e+00	8.492e+00	1.301e+00	3.657e+00
f_7	mean	7.503e+02 +	1.040e+03 +	7.295e+02 +	7.001e+02 +	7.014e+02 +	7.000e+02 ≈	7.001e+02 +	7.008e+02 +	7.162e+02 +	7.000e+02 ≈	7.005e+02 +	7.000e+02
	std	1.479e+01	3.648e+01	5.887e+00	6.333e-02	8.878e-02	2.769e-02	2.084e-02	1.695e-01	5.909e+00	2.175e-02	1.122e-01	2.936e-02
f_8	mean	1.099e+03 +	1.101e+03 +	1.050e+03 +	8.859e+02 ≈	9.156e+02 +	8.559e+02 +	8.634e+02 +	8.793e+02 ≈	9.654e+02 +	9.846e+02 +	8.682e+02 ≈	8.933e+02
	std	2.563e+01	1.779e+01	1.127e+01	7.732e+00	1.033e+01	7.173e+00	5.282e+00	1.807e+01	2.016e+01	1.239e+01	3.822e+00	2.810e+01
f_9	mean	1.218e+03 +	1.296e+03 +	1.169e+03 +	1.075e+03 +	1.091e+03 +	1.048e+03 ≈	1.048e+03 ≈	9.857e+02 +	1.120e+03 +	1.119e+03 +	1.058e+03 +	1.044e+03
	std	4.356e+01	2.003e+01	9.523e+00	9.523e+00	1.237e+01	9.847e+00	1.146e+01	1.250e+01	3.478e+01	1.096e+01	4.757e+00	2.165e+01
f_{10}	mean	8.307e+03 +	6.485e+03 +	8.027e+03 +	4.357e+03 +	4.744e+03 +	2.917e+03 +	3.201e+03 ≈	3.159e+03 +	5.357e+03 +	6.949e+03 +	3.417e+03 ≈	3.511e+03
	std	1.652e+02	2.432e+02	3.427e+02	1.822e+02	4.504e+02	3.126e+02	2.119e+02	5.939e+02	6.658e+02	3.908e+02	9.602e+01	7.020e+02
f_{11}	mean	8.705e+03 +	8.701e+03 +	8.653e+03 +	7.394e+03 +	6.870e+03 +	5.913e+03 +	5.922e+03 +	5.607e+03 +	6.312e+03 +	8.565e+03 +	6.170e+03 +	5.566e+03
	std	3.873e+02	2.469e+02	2.510e+02	2.431e+02	1.932e+02	3.676e+02	2.140e+02	1.597e+03	5.600e+02	2.673e+02	1.055e+02	4.871e+02
f_{12}	mean	1.203e+03 +	1.203e+03 +	1.203e+03 +	1.202e+03 +	1.202e+03 +	1.201e+03 +	1.201e+03 +	1.203e+03 +	1.202e+03 +	1.203e+03 +	1.201e+03 +	1.200e+03
	std	3.374e-01	3.674e-01	1.969e-01	3.127e-01	2.162e-01	1.408e-01	1.535e-01	4.815e-01	4.635e-01	5.113e-01	1.388e-01	9.827e-02
f_{13}	mean	1.301e+03 +	1.305e+03 +	1.301e+03 +	1.300e+03 +	1.300e+03 +	1.300e+03 +	1.301e+03 +	1.301e+03 +	1.301e+03 +	1.301e+03 +	1.300e+03 +	1.300e+03
	std	6.081e-01	4.087e-01	1.307e-01	3.304e-02	4.261e-02	4.654e-02	6.561e-02	1.586e-01	5.442e-01	6.896e-02	5.595e-02	9.861e-02
f_{14}	mean	1.414e+03 +	1.510e+03 +	1.408e+03 +	1.400e+03 +	1.400e+03 +	1.400e+03 ≈	1.400e+03 +	1.400e+03 ≈	1.403e+03 +	1.400e+03 +	1.400e+03 ≈	1.400e+03
	std	6.116e+00	1.223e+01	1.793e+00	8.528e-02	2.621e-02	3.212e-02	2.461e-02	1.655e-01	4.948e+00	2.072e-01	2.795e-02	1.516e-01
f_{15}	mean	4.201e+03 +	1.720e+06 +	4.151e+03 +	1.516e+03 +	1.525e+03 +	1.514e+03 +	1.514e+03 +	1.523e+03 +	4.049e+03 +	1.520e+03 +	1.516e+03 +	1.517e+03
	std	3.959e+03	7.394e+05	7.216e+02	7.201e-01	1.902e+00	1.228e+00	1.045e+00	1.215e+01	2.438e+03	6.463e-01	1.003e+00	4.899e+00
f_{16}	mean	1.613e+03 +	1.614e+03 +	1.613e+03 +	1.613e+03 +	1.612e+03 +	1.612e+03 +	1.612e+03 +	1.613e+03 +	1.613e+03 +	1.612e+03 +	1.612e+03 +	1.612e+03
	std	2.262e-01	1.094e-01	2.511e-01	2.998e-01	2.257e-01	2.440e-01	1.228e-01	4.635e-01	4.707e-01	2.143e-01	2.549e-01	4.352e-01
f_{17}	mean	3.094e+07 +	1.299e+07 +	7.445e+05 +	1.769e+06 +	1.928e+06 +	1.112e+06 +	1.317e+06 +	1.383e+05 +	4.090e+06 +	2.060e+06 +	1.623e+06 +	7.872e+04
	std	1.591e+07	4.664e+06	2.582e+05	9.273e+05	3.808e+05	4.971e+05	6.702e+04	7.593e+05	5.640e+05	4.578e+05	5.105e+04	4.951e+04
f_{18}	mean	8.576e+08 +	4.358e+08 +	1.747e+07 +	1.381e+07 +	4.336e+06 +	2.829e+06 +	1.459e+06 +	6.191e+04 +	1.725e+05 +	3.931e+07 +	1.235e+06 +	4.951e+04
	std	5.096e+08	1.655e+08	8.647e+06	4.913e+06	1.806e+06	8.809e+05	3.767e+05	1.298e+04	3.940e+04	2.128e+07	3.300e+05	3.662e+04
f_{19}	mean	5.141e+07 +	1.231e+07 +	5.579e+04 +	4.921e+04 +	1.295e+04 +	4.094e+03 +	2.896e+03 +	1.932e+03 +	3.251e+03 +	2.622e+05 +	2.846e+03 +	1.938e+03
	std	9.507e+07	7.768e+06	3.106e+04	4.640e+04	6.184e+03	1.207e+03	5.770e+02	3.181e+01	1.690e+03	2.827e+05	4.530e+02	2.731e+01
f_{20}	mean	2.311e+11 +	3.207e+10 +	1.840e+06 +	1.109e+06 +	2.731e+06 +	9.921e+04 +	7.772e+04 +	8.601e+04 +	3.395e+07 +	1.605e+07 +	8.100e+04 +	5.640e+04
	std	4.100e+11	2.226e+10	1.571e+06	3.838e+05	2.293e+06	1.797e+04	1.359e+04	3.393e+04	1.008e+08	3.911e+07	1.181e+04	2.130e+04
f_{21}	mean	1.099e+08 +	2.834e+07 +	7.685e+05 +	1.028e+06 +	5.727e+05 +	3.190e+05 +	2.121e+05 +	3.403e+04 +	1.356e+07 +	2.340e+06 +	1.964e+05 +	3.995e+04
	std	6.125e+07	1.394e+07	1.802e+05	3.637e+05	2.311e+05	1.800e+05	8.357e+04	1.228e+04	3.335e+07	8.154e+05	9.470e+04	2.091e+04
f_{22}	mean	1.237e+08 +	4.546e+07 +	4.794e+03 +	6.103e+03 +	5.208e+03 +	4.108e+03 +	3.898e+03 +	3.482e+03 +	6.437e+03 +	2.641e+04 +	3.997e+03 +	2.651e+03
	std	2.300e+08	5.033e+07	4.793e+02	1.160e+03	2.817e+02	3.716e+02	1.752e+02	5.055e+02	1.308e+03	4.180e+04	2.862e+02	2.421e+02
+/≈/-		30/0/0	29/0/1	27/1/2	18/8/4	20/8/2	12/11/7	14/10/6	15/6/9	29/1/0	18/6/6	16/10/4	-

Appendix E. Detailed experimental results on CEC2022

Tables 36 and **37** summarize the experimental results of optimizers on CEC2022. Convergence curves, proportion changes between exploration and exploitation, and box plots on CEC2022 are presented in Figs. 33, 34, 35, 36, 37, and 38.

Appendix F. Detailed results of ablation experiments

Tables 38, 39, 40, 41, 42, and 43 summarize the results of ablation experiments against HHA_{SR}, HHA_{RP}, and HHA_{SHA} on CEC benchmarks.

Appendix G. Detailed results of investigation experiments of various LLMs

Tables 44, 45, 46, 47, 48, and 49 summarize the results of investigation experiments of various LLMs as the high-level component in LLMOA on CEC benchmarks.

Data availability

The source code of this research can be downloaded from [The source code of this research can be downloaded from The source code of this research can be downloaded from <https://github.com/RuiZhong961230/LLMOA>.](https://github.com/RuiZhong961230/LLMOA)

Table 33

Results and analyses on 50-D CEC2014 benchmark functions.

F	PSO	DE	CMA-ES	SaDE	CL-PSO	SHADE	L-SHADE	HPSO-TVAC	PPSO	WDE	DMDE	LLMOA	
f_1	mean	1.091e+09 + 1.350e+09 +	7.255e+07 + 4.779e+07 +	6.117e+07 + 2.662e+07 +	2.885e+07 + 1.602e+07 +	1.906e+08 + 1.263e+08 +	3.600e+07 + 4.051e+06						
	std	3.943e+08 2.082e+08	1.723e+07 9.973e+06	6.474e+06 5.847e+06	3.201e+06 8.518e+06	8.476e+07 3.773e+07	7.022e+06 2.701e+05						
f_2	mean	2.824e+10 + 1.026e+11 +	1.258e+10 + 4.390e+03 -	1.510e+07 + 4.592e+03 -	5.427e+03 - 2.338e+08 +	6.555e+09 + 1.831e+04 -	2.701e+05 2.788e+05						
	std	6.543e+09 8.157e+09	2.549e+09 4.837e+03	3.515e+06 3.098e+03	5.189e+03 5.189e+03	2.134e+08 2.677e+09	8.995e+03 1.753e+05						
f_3	mean	2.009e+05 + 3.283e+05 +	2.404e+04 + 5.208e+04 +	1.049e+04 + 1.465e+04 +	1.339e+04 + 3.743e+04 +	1.468e+05 + 5.505e+04 +	1.155e+04 + 3.602e+03						
	std	1.925e+04 4.794e+04	5.157e+03 8.358e+03	2.628e+03 2.080e+03	2.159e+03 7.226e+03	3.460e+04 9.239e+03	2.315e+03 9.585e+02						
f_4	mean	4.996e+03 + 1.750e+04 +	1.924e+03 + 5.266e+02 ≈	6.496e+02 + 5.246e+02 ≈	5.606e+02 ≈ 7.863e+02 +	1.348e+03 + 4.982e+02 +	6.264e+02 + 5.406e+02						
	std	2.062e+03 5.067e+03	3.612e+02 2.949e+01	1.595e+01 2.623e+01	2.392e+01 9.318e+01	2.254e+02 1.594e+00	1.863e+01 2.804e+01						
f_5	mean	5.212e+02 + 5.212e+02 +	5.212e+02 + 5.211e+02 +	5.210e+02 + 5.207e+02 +	5.207e+02 + 5.212e+02 +	5.208e+02 + 5.212e+02 +	5.208e+02 + 5.203e+02						
	std	3.520e-02 2.466e-02	3.298e-02 4.794e-02	8.987e-02 4.946e-02	3.281e-02 4.491e-02	6.096e-02 6.696e-02	3.258e-02 4.142e-02						
f_6	mean	6.734e+02 + 6.602e+02 ≈	6.624e+02 + 6.554e+02 ≈	6.493e+02 + 6.446e+02 ≈	6.456e+02 + 6.373e+02 ≈	6.696e+02 + 6.287e+02 ≈	6.461e+02 + 6.564e+02						
	std	2.118e+00 4.418e+00	5.807e+00 2.319e+00	2.144e+00 2.004e+00	1.537e+00 3.920e+00	3.141e+00 1.294e+01	1.222e+00 4.359e+00						
f_7	mean	9.751e+02 + 1.779e+03 +	8.229e+02 + 7.000e+02 -	7.011e+02 + 7.000e+02 -	7.001e+02 - 7.048e+02 +	7.607e+02 + 7.000e+02 -	7.004e+02 ≈ 7.003e+02						
	std	9.101e+01 8.307e+01	4.465e+01 1.423e-02	3.457e-02 4.485e-02	4.208e-02 2.592e+00	2.376e+01 1.920e-02	1.532e-01 1.294e-01						
f_8	mean	1.350e+03 + 1.422e+03 +	1.288e+03 + 9.849e+02 ≈	1.010e+03 ≈ 9.013e+02 +	9.203e+02 - 9.835e+02 ≈	1.120e+03 ≈ 1.165e+03 ≈	9.248e+02 - 1.007e+03						
	std	5.353e+01 2.857e+01	1.257e+01 7.256e+00	9.390e+00 5.992e+00	6.914e+00 2.956e+01	4.309e+01 1.768e+01	7.569e+00 2.608e+01						
f_9	mean	1.475e+03 + 1.721e+03 +	1.400e+03 + 1.249e+03 +	1.260e+03 + 1.181e+03 ≈	1.178e+03 ≈ 1.084e+03 +	1.372e+03 + 1.327e+03 +	1.201e+03 + 1.201e+03 ≈						
	std	1.005e+02 2.936e+01	3.228e+01 1.548e+01	2.123e+01 1.219e+01	1.630e+01 3.415e+01	5.754e+01 1.529e+01	8.801e+00 4.654e+01						
f_{10}	mean	1.502e+04 + 1.274e+04 +	1.448e+04 + 8.716e+03 +	8.298e+03 + 4.796e+03 +	5.249e+03 + 5.524e+03 -	9.315e+03 + 1.312e+04 +	5.221e+03 - 6.800e+03						
	std	3.858e+02 3.151e+02	4.310e+02 4.218e+02	6.911e+02 2.184e+02	4.115e+02 7.448e+02	9.236e+02 4.588e+02	3.181e+02 9.310e+02						
f_{11}	mean	1.516e+04 + 1.515e+04 +	1.533e+04 + 1.365e+04 +	1.165e+04 + 1.030e+04 +	1.013e+04 + 8.788e+03 +	1.147e+04 + 1.515e+04 +	1.068e+04 + 1.7928e+03						
	std	5.220e+02 2.184e+02	3.265e+02 2.529e+02	5.381e+02 3.114e+02	1.961e+02 3.187e+03	8.653e+02 3.016e+02	3.165e+02 1.023e+03						
f_{12}	mean	1.204e+03 + 1.204e+03 +	1.204e+03 + 1.203e+03 +	1.202e+03 + 1.201e+03 +	1.201e+03 + 1.204e+03 +	1.202e+03 + 1.204e+03 +	1.201e+03 + 1.201e+03 +						
	std	3.288e-01 3.486e-01	3.354e-01 2.548e-01	2.340e-01 1.505e-01	1.522e-01 3.431e-01	5.564e-01 1.807e-01	1.401e-01 1.432e-01						
f_{13}	mean	1.303e+03 + 1.306e+03 +	1.301e+03 ≈ 1.300e+03 +	1.301e+03 ≈ 1.301e+03 ≈									
	std	1.021e+00 2.893e-01	3.978e-01 5.519e-02	2.281e-02 2.677e-02	6.672e-02 2.938e-02	1.046e-01 9.488e-02	5.336e-02 4.552e-02						
f_{14}	mean	1.462e+03 + 1.719e+03 +	1.433e+03 + 1.400e+03 -	1.400e+03 ≈ 1.401e+03 ≈	1.400e+03 ≈ 1.401e+03 ≈	1.411e+03 + 1.400e+03 +	1.401e+03 + 1.401e+03 ≈						
	std	1.681e+01 2.460e+01	1.620e+01 3.171e-02	5.558e-02 1.248e-01	1.331e-01 2.374e-01	7.382e-00 2.277e-01	1.210e-01 1.998e-01						
f_{15}	mean	4.289e+04 + 1.201e+07 +	1.900e+04 + 1.532e+03 ≈	1.544e+03 + 1.526e+03 +	1.526e+03 + 1.528e+03 ≈	1.793e+03 + 1.996e+04 +	1.537e+03 + 1.549e+03 +						
	std	5.077e+04 4.421e+06	9.903e+03 1.395e+00	2.422e+00 1.930e+00	1.864e+00 1.328e+02	1.590e+04 9.462e-01	6.280e+00 4.502e+00						
f_{16}	mean	1.623e+03 + 1.623e+03 +	1.623e+03 + 1.622e+03 ≈	1.622e+03 + 1.621e+03 +	1.621e+03 + 1.621e+03 +	1.622e+03 ≈ 1.623e+03 +	1.621e+03 - 1.622e+03						
	std	1.426e-01 1.376e-01	1.028e-01 1.231e-01	4.044e-01 4.211e-01	2.160e-01 2.032e-01	4.211e-01 6.228e-01	2.549e-01 2.400e-01						
f_{17}	mean	7.434e+07 + 9.732e+07 +	7.306e+06 + 7.7151e+06 +	4.241e+06 + 2.912e+06 +	3.388e+06 + 3.621e+06 +	4.621e+05 + 1.049e+07 +	1.740e+07 + 3.076e+06 +						
	std	3.360e+07 9.599e+07	1.654e+07 2.116e+06	2.535e+06 8.409e+05	8.607e+05 7.383e+05	2.351e+05 6.270e+06	1.249e+06 9.708e+04						
f_{18}	mean	1.281e+09 + 2.436e+09 +	1.257e+08 + 3.462e+07 +	3.017e+07 + 3.729e+06 +	3.281e+06 + 3.281e+06 +	1.133e+05 + 1.133e+05 +	2.778e+08 + 3.554e+06 +						
	std	1.291e+09 1.018e+09	2.849e+07 6.357e+06	8.969e+06 9.063e+05	8.767e+05 8.767e+05	1.284e+06 7.654e+07	6.751e+05 6.751e+05						
f_{19}	mean	3.939e+08 + 3.166e+08 +	1.419e+06 + 1.367e+05 +	8.414e+04 + 5.167e+03 +	4.669e+03 + 3.534e+03 +	7.284e+04 + 3.278e+06 +	3.826e+03 + 2.021e+03						
	std	4.373e+08 2.384e+08	4.980e+05 7.819e+04	4.100e+04 4.126e+03	1.926e+03 2.032e-01	4.211e-01 6.228e-01	2.549e-01 2.400e-01						
f_{20}	mean	5.945e+13 + 4.162e+12 +	1.835e+12 + 1.807e+07 +	2.027e+07 + 1.517e+05 +	1.410e+05 + 1.273e+05 +	5.538e+07 + 2.459e+09 +	1.690e+05 + 1.315e+05						
	std	9.599e+13 4.836e+12	1.327e+09 1.288e+07	1.337e+07 2.906e+04	2.121e+04 1.536e+08	2.126e+09 4.409e+04	3.222e+04 1.9708e+04						
f_{21}	mean	3.893e+08 + 2.785e+08 +	7.106e+06 + 4.025e+06 +	4.404e+06 + 9.045e+05 +	9.045e+05 + 1.026e+06 +	6.032e+05 + 9.889e+06 +	1.523e+07 + 1.138e+06 +						
	std	4.069e+08 1.617e+08	8.183e+06 1.197e+06	1.204e+06 1.835e+05	1.835e+05 2.378e+05	3.487e+05 4.309e+06	7.613e+06 4.146e+05						
f_{22}	mean	2.063e+12 + 3.503e+11 +	1.937e+06 + 1.475e+05 +	1.347e+04 + 5.564e+03 +	5.176e+03 + 4.046e+03 +	8.425e+08 + 6.969e+08 +	1.696e+08 + 5.143e+03 +						
	std	3.025e+12 2.426e+11	2.111e+06 1.500e+05	6.346e+03 4.999e+02	4.152e+02 4.152e+02	2.527e+09 3.707e+08	4.182e+02 2.102e+02						
+/≈/-		29/1/0	27/2/1	26/1/3	16/5/9	20/7/3	13/6/11	13/6/11	17/4/9	26/4/0	17/6/7	15/7/8	-

Table 34

Results and analyses on 30-D CEC2020 benchmark functions.

Func.	PSO	DE	CMA-ES	SaDE	CL-PSO	SHADE	L-SHADE	HPSO-TVAC	PPSO	WDE	DMDE	LLMOA
f_1	mean	3.517e+09 + 3.825e+10 +	3.194e+09 + 4.794e+04 +	2.359e+07 + 2.025e+04 +	2.908e+04 + 2.035e+03 ≈	4.386e+05 + 2.535e+03 ≈	1.204e+05 + 2.535e+03 ≈	2.276e+05 + 2.535e+03 ≈				
	std	1.334e+09 5.030e+09	4.758e+08 2.602e+04	5.530e+06 7.807e+03	1.793e+04 5.788e+05	5.349e+08 5.349e+08	9.514e+04 1.025e+05	6.740e+03 6.740e+03				
f_2	mean	9.573e+11 + 3.623e+12 +	3.447e+11 + 6.004e+06 +	2.956e+09 + 2.114e+06 +	2.599e+06 + 2.114e+06 +	3.816e+07 + 2.224e+11 +	6.087e+06 + 2.695e+07 +	2.695e+07 + 2.695e+07 +				
	std	9.552e+11 3.374e+11	5.446e+10 5.207e+06	6.897e+08 9.576e+05	1.612e+06 1.612e+06	4.092e+07 7.898e+10	5.304e+06 5.304e+06	1.551e+07 5.943e+05				
f_3	mean	1.498e+11 + 1.193e+12 +	1.069e+11 + 6.441e+06 +	1.518e+09 + 1.306e+06 +	1.278e+06 + 1.278e+06 +	5.563e+07 + 5.563e+07 +	1.273e+07 + 1.273e+07 +	9.806e+06 + 9.806e+06 +				
	std	6.724e+10 8.835e+10	2.036e+10 4.342e+06	3.496e+08 5.104e+05	1.131e+06 1.131e+06	9.519e+07 9.519e+07	2.299e+10 2.299e+10	1.273e+07 1.273e+07				

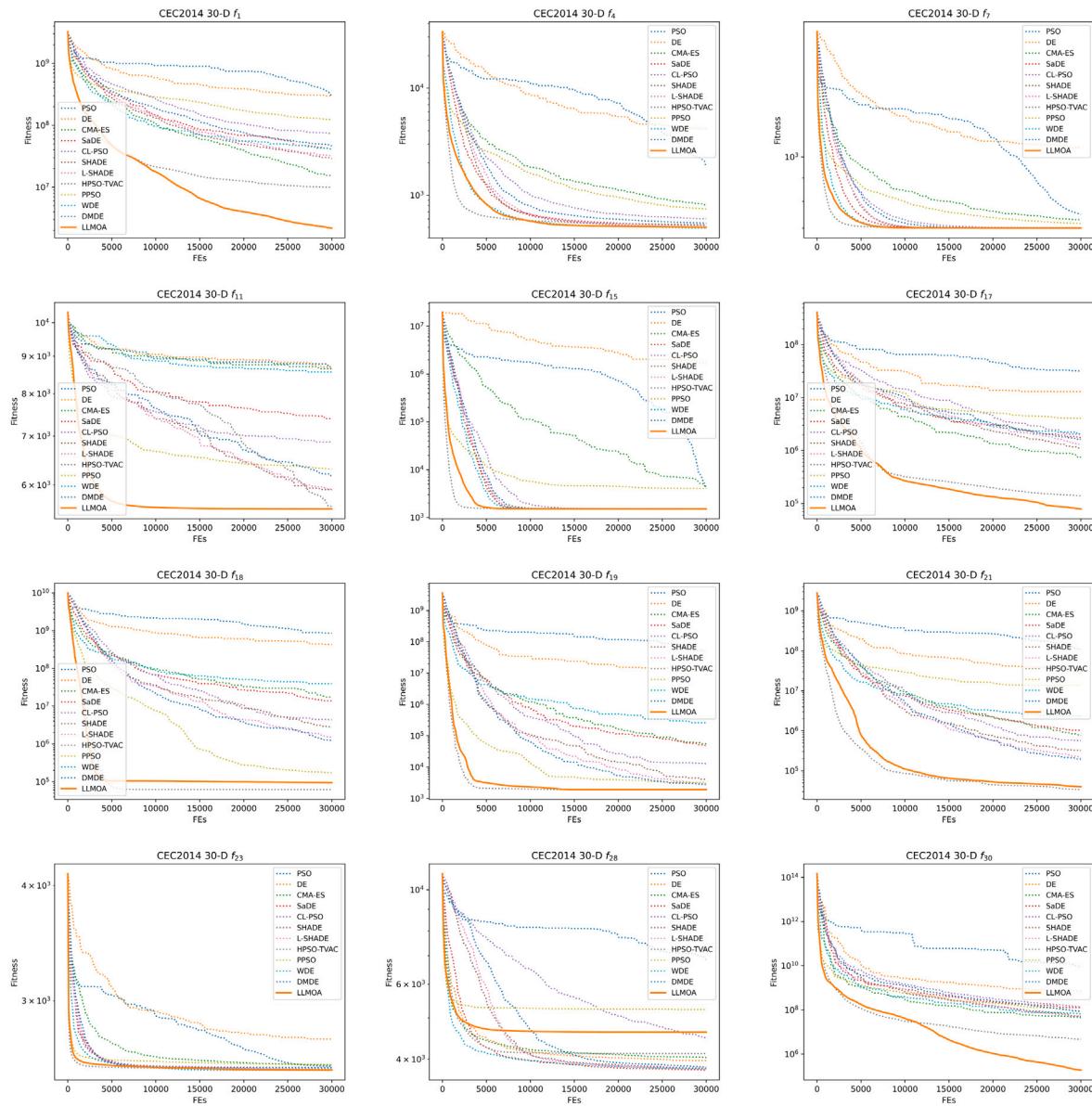


Fig. 21. Convergence curves on 30-D CEC2014 representative functions. f_1 : Unimodal function; f_4 , f_7 , f_{11} and f_{15} : Multimodal functions; f_{17} , f_{18} , f_{19} , and f_{21} : Hybrid functions; f_{23} , f_{28} , and f_{30} : Composite functions.

Table 35
Results and analyses on 50-D CEC2020 benchmark functions.

Func.	PSO	DE	CMA-ES	SaDE	CL-PSO	SHADE	L-SHADE	HPSO-TVAC	PPSO	WDE	DMDE	LLMOA
f_1	mean 1.971e+10 + 3.655e+09	1.027e+11 + 5.229e+09	1.126e+10 + 2.468e+09	1.344e+03 - 2.042e+03	1.668e+07 + 3.696e+06	3.098e+03 - 5.118e+03	1.318e+03 - 1.644e+03	5.847e+08 + 5.171e+08	5.491e+09 + 1.637e+09	1.081e+04 - 7.696e+03	3.704e+05 + 1.810e+05	2.292e+05 1.720e+05
f_2	mean 2.224e+12 + 3.747e+11	1.175e+13 + 1.016e+12	1.452e+12 + 5.949e+11	4.810e+05 - 3.239e+05	2.362e+09 + 3.564e+08	3.994e+05 - 1.377e+05	4.634e+05 - 2.384e+05	5.855e+10 + 8.245e+10	6.457e+11 + 1.901e+11	1.115e+06 - 5.571e+05	4.486e+07 + 2.032e+07	2.271e+07 1.323e+07
f_3	mean 7.288e+11 + 1.183e+11	4.220e+12 + 2.894e+11	4.862e+11 + 2.894e+11	1.198e+05 - 1.283e+05	5.176e+08 - 6.156e+08	6.156e+04 - 7.370e+04	4.497e+04 - 5.260e+04	4.938e+09 + 5.366e+10	1.968e+11 + 4.864e+05	7.007e+05 - 5.406e+06	9.037e+06 + 5.369e+06	5.213e+06 5.369e+06
f_4	mean 1.866e+05 + 4.716e+05	1.212e+06 + 5.280e+05	4.918e+03 + 1.664e+03	1.932e+03 ≈ 1.446e+00	1.942e+03 + 1.605e+00	1.927e+03 + 1.591e+00	1.927e+03 - 1.925e+00	2.096e+03 + 7.939e+01	1.158e+04 + 8.203e+03	1.937e+03 ≈ 1.976e+00	1.948e+03 + 5.294e+00	1.938e+03 7.660e+00
f_5	mean 8.190e+07 + 5.847e+07	5.507e+07 + 1.754e+07	2.268e+06 + 4.223e+05	4.727e+06 + 1.275e+06	3.609e+06 + 9.131e+05	2.415e+06 + 5.610e+05	1.846e+06 + 4.910e+05	8.233e+05 + 5.791e+05	1.091e+07 + 5.582e+06	1.120e+07 + 2.904e+06	2.542e+06 + 7.056e+05	2.281e+05 9.048e+04
f_6	mean 4.760e+06 + 8.356e+06	1.082e+08 + 1.787e+07	2.718e+04 + 8.219e+03	1.043e+04 + 4.162e+03	1.075e+04 + 1.696e+03	7.794e+03 + 2.068e+03	7.018e+03 + 1.266e+03	8.158e+03 + 3.150e+03	6.383e+05 + 7.215e+05	1.873e+04 + 1.118e+04	7.452e+03 + 8.915e+02	4.851e+03 2.075e+03
f_7	mean 9.831e+08 + 8.696e+08	6.789e+08 + 3.127e+08	1.309e+07 + 3.989e+06	8.978e+06 + 2.178e+06	3.350e+06 + 1.106e+06	3.526e+06 + 1.594e+06	3.087e+06 + 7.663e+05	2.919e+05 + 2.013e+05	2.673e+07 + 2.156e+07	2.590e+07 + 1.429e+07	2.494e+06 + 4.511e+05	1.628e+05 6.225e+04
f_8	mean 4.320e+03 + 1.130e+03	3.236e+03 + 5.552e+01	2.659e+03 - 5.013e+01	2.399e+03 - 4.417e+00	2.413e+03 - 6.138e+00	2.394e+03 - 2.307e+00	2.396e+03 - 4.017e+00	2.614e+03 - 6.133e+01	5.218e+03 + 2.230e+03	2.424e+03 - 9.276e+00	2.398e+03 - 2.888e+00	2.933e+03 1.420e+02
f_9	mean 2.219e+04 + 2.898e+03	2.755e+04 + 1.095e+03	1.131e+04 + 6.505e+02	2.601e+03 - 8.358e-02	2.939e+03 + 5.204e+01	2.644e+03 - 1.325e+02	2.662e+03 - 1.378e+02	4.380e+03 + 2.386e+03	1.242e+04 + 4.185e+03	2.603e+03 - 3.655e-01	2.723e+03 ≈ 2.301e+02	2.806e+03 2.687e+02
f_{10}	mean 8.043e+03 + 1.039e+03	1.164e+04 + 1.527e+03	4.703e+03 + 3.218e+02	3.276e+03 ≈ 3.185e+01	3.470e+03 + 4.358e+01	3.320e+03 ≈ 2.166e+01	3.314e+03 ≈ 1.305e+02	3.654e+03 + 4.262e+02	4.699e+03 + 8.802e+01	3.432e+03 + 2.590e+01	3.334e+03 3.540e+01	-
+/-	10/0/0	10/0/0	9/0/1	3/2/5	9/0/1	3/1/6	3/1/6	9/0/1	10/0/0	3/1/6	8/1/1	-

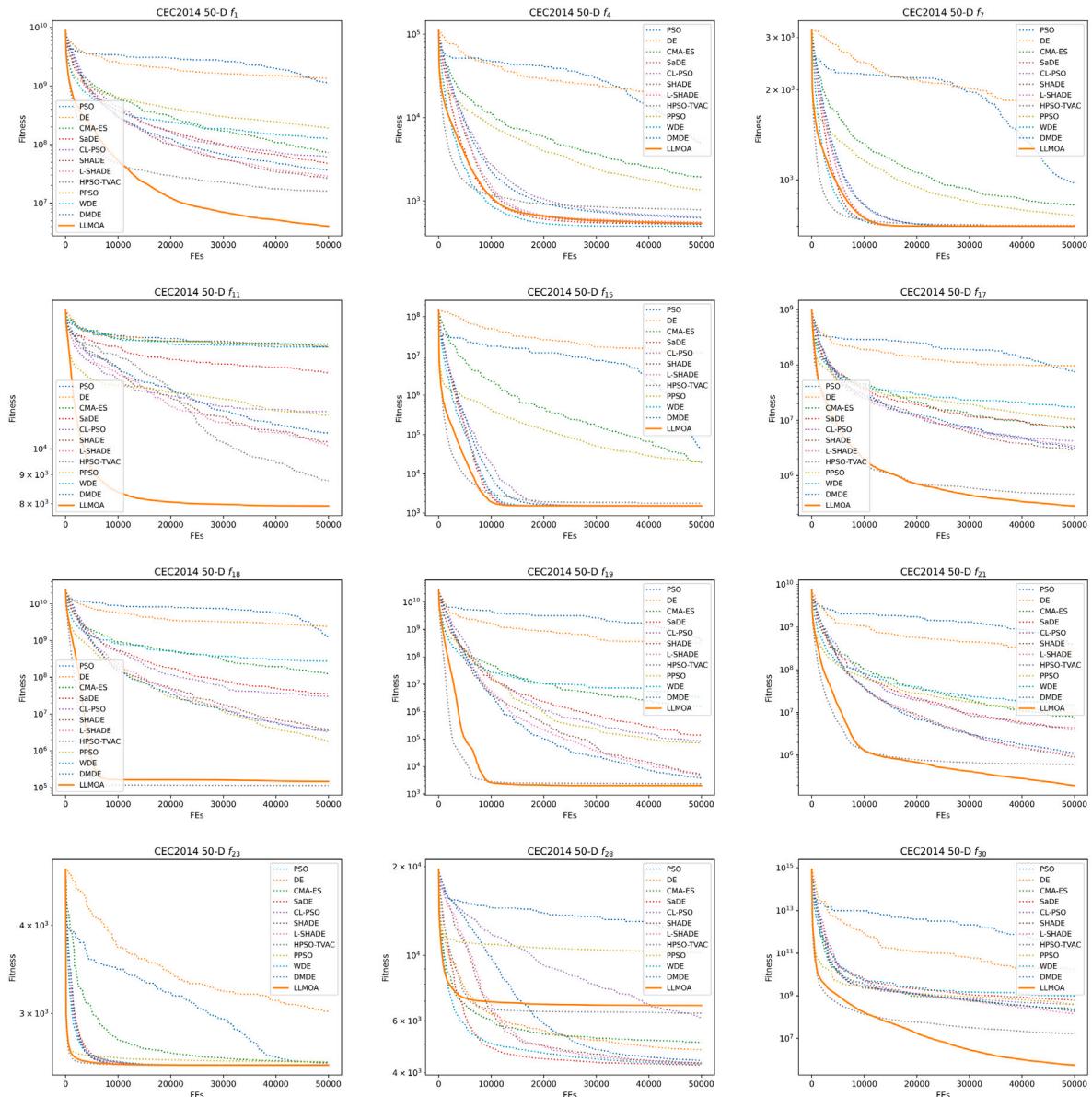


Fig. 22. Convergence curves on 50-D CEC2014 representative functions.

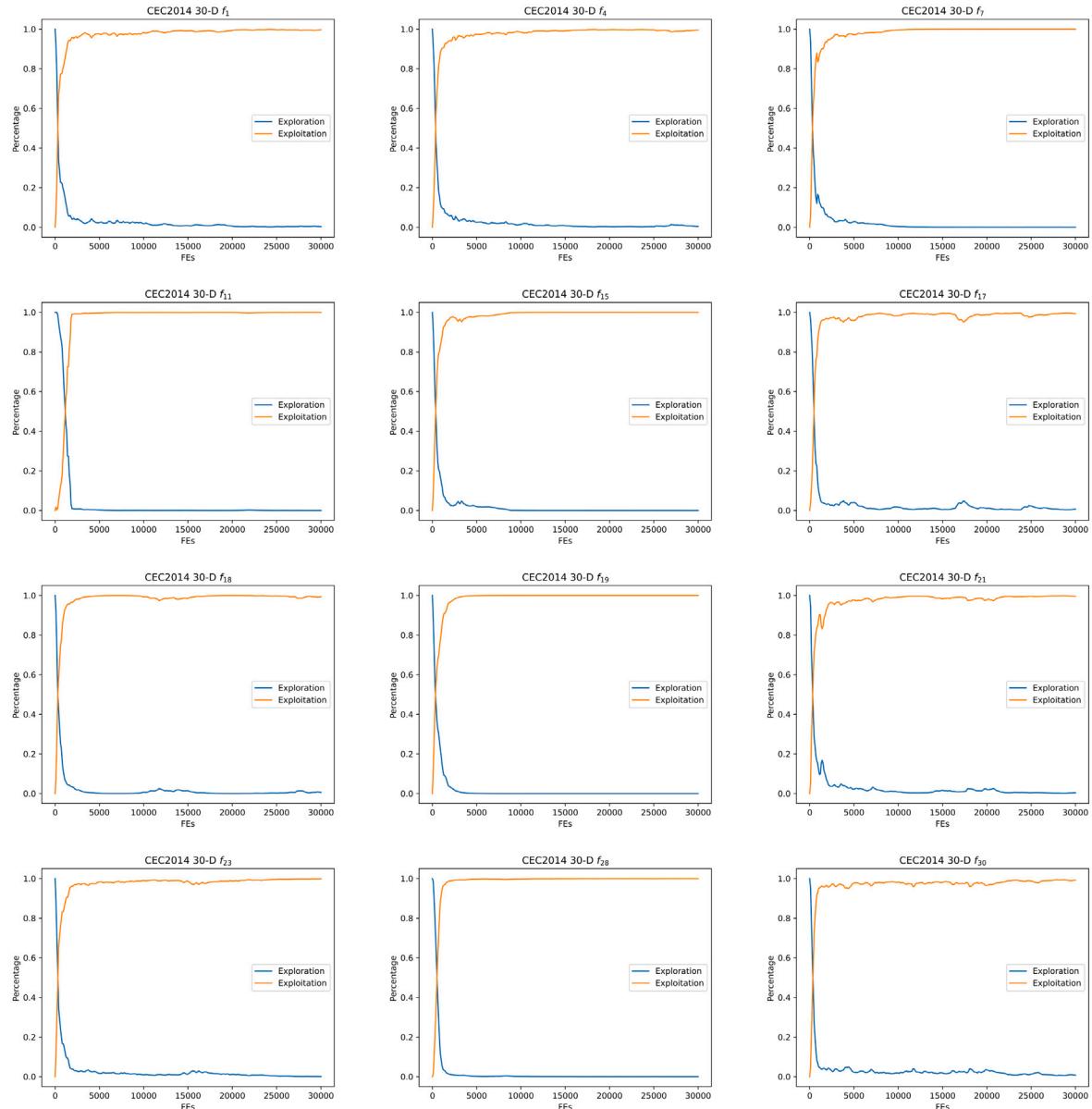


Fig. 23. Plots of exploration and exploitation proportion on 30-D CEC2014 representative functions.

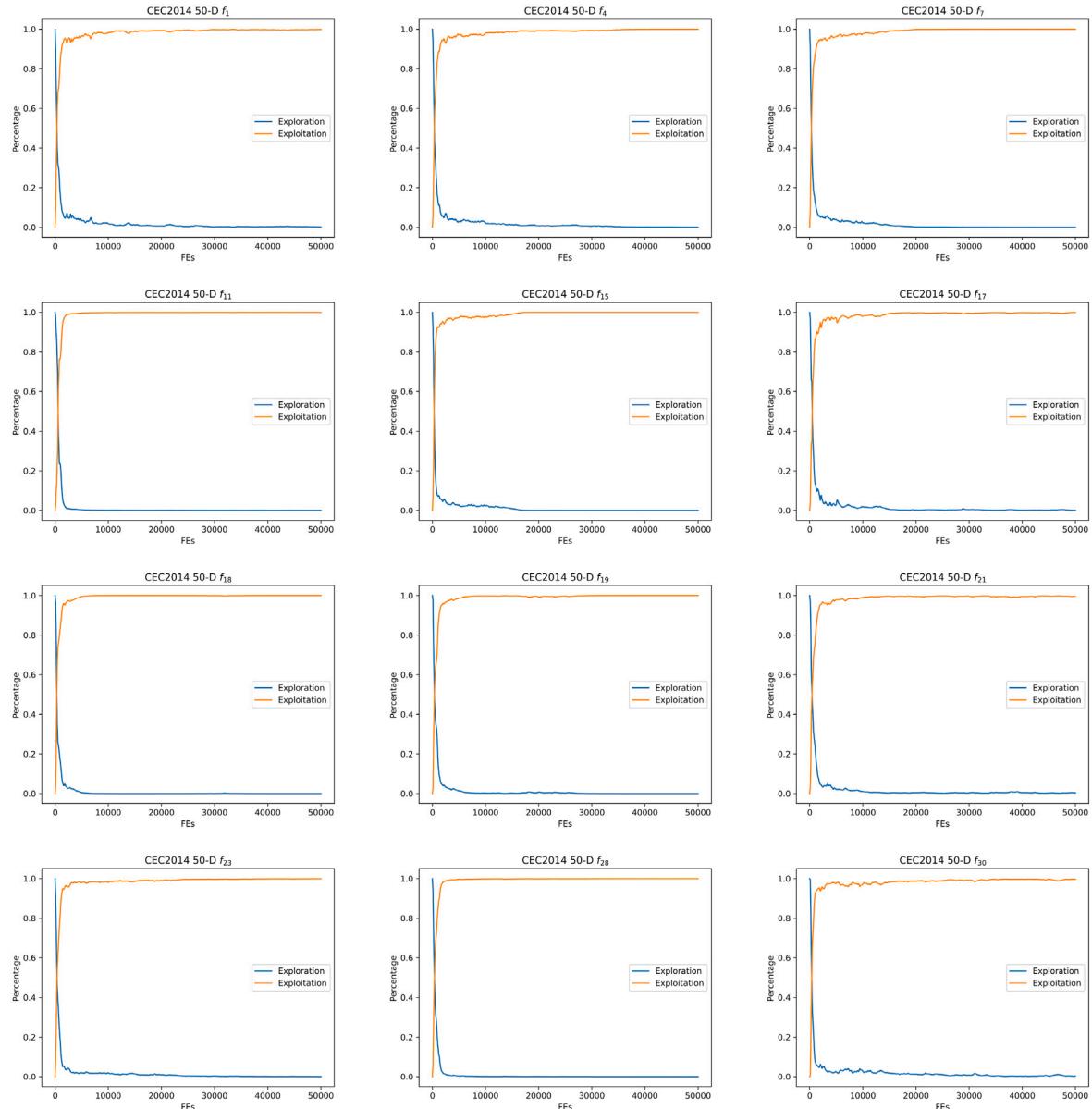


Fig. 24. Plots of exploration and exploitation proportion on 50-D CEC2014 representative functions.

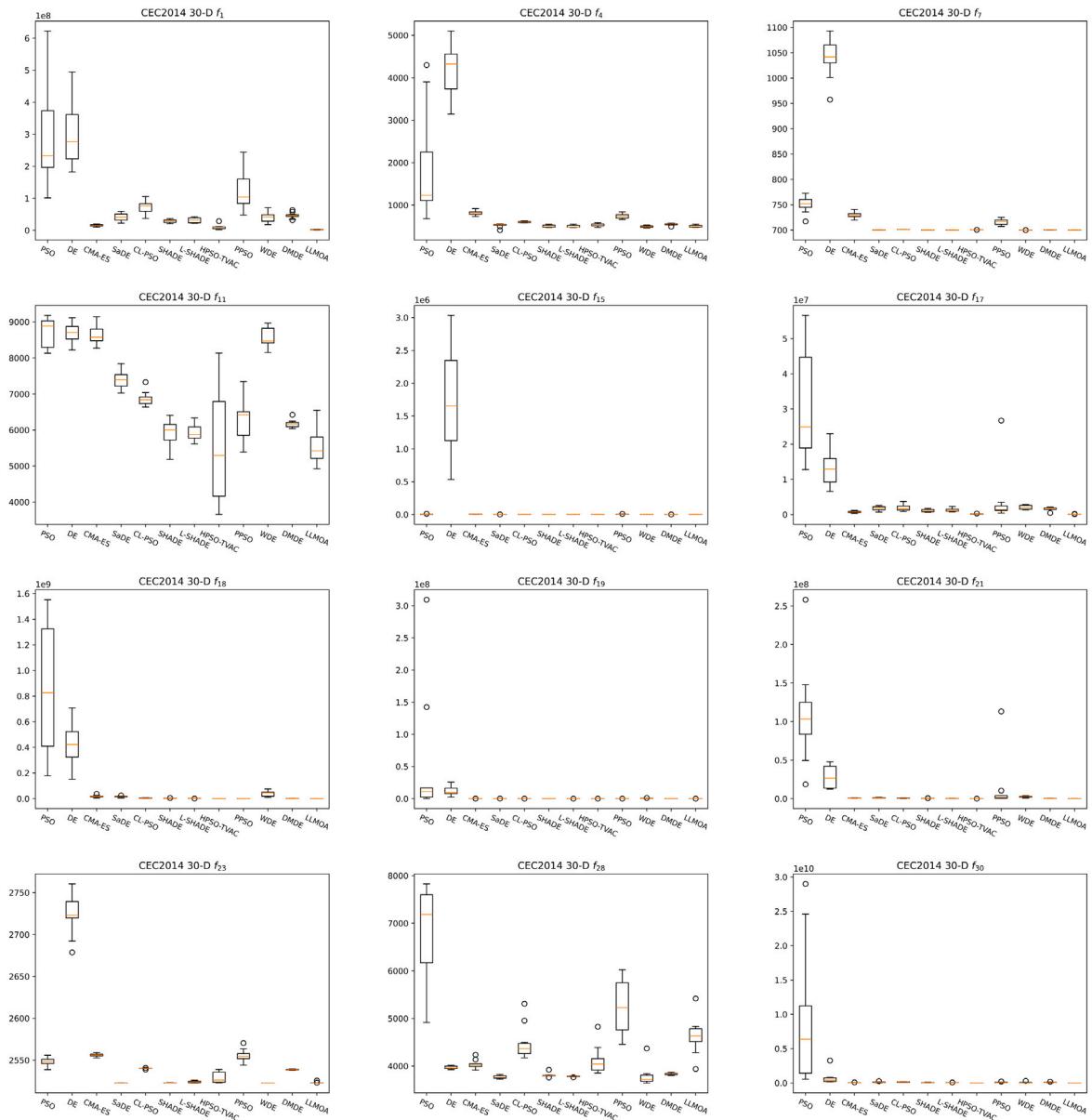


Fig. 25. Box plots of optimizers on 30-D CEC2014 representative functions.

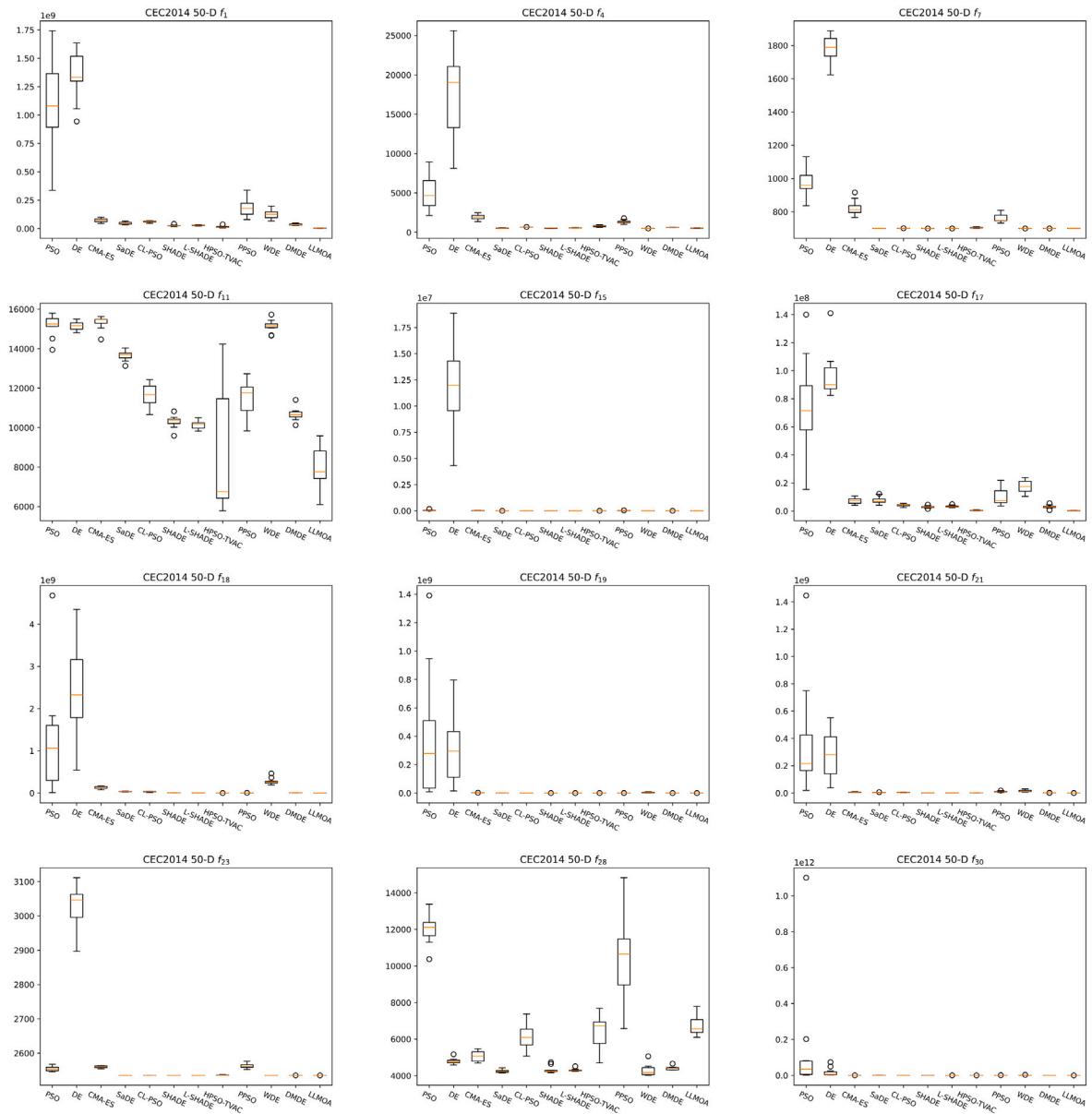


Fig. 26. Box plots of optimizers on 50-D CEC2014 representative functions.

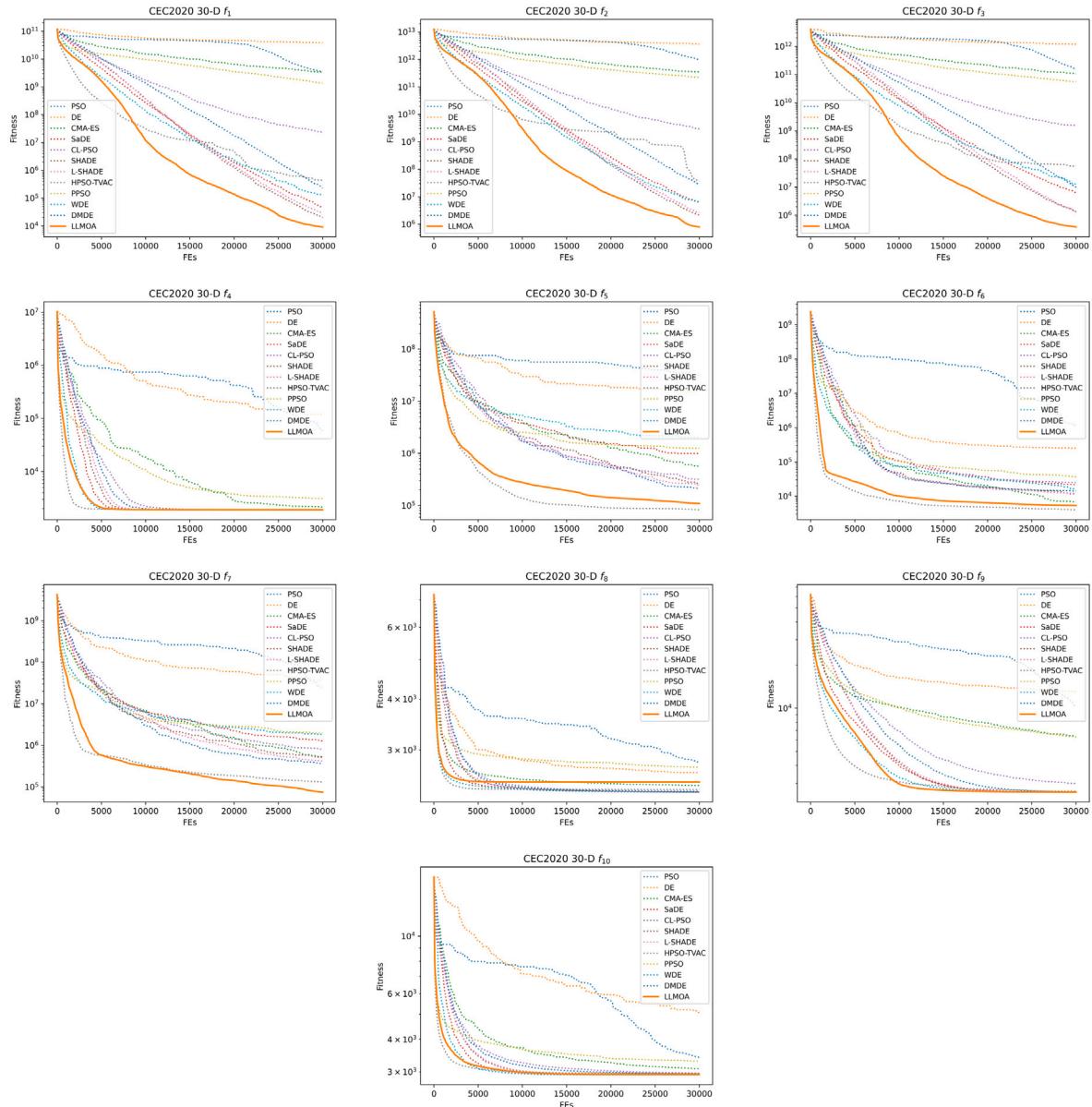


Fig. 27. Convergence graphs on 30-D CEC2020 mathematical functions.

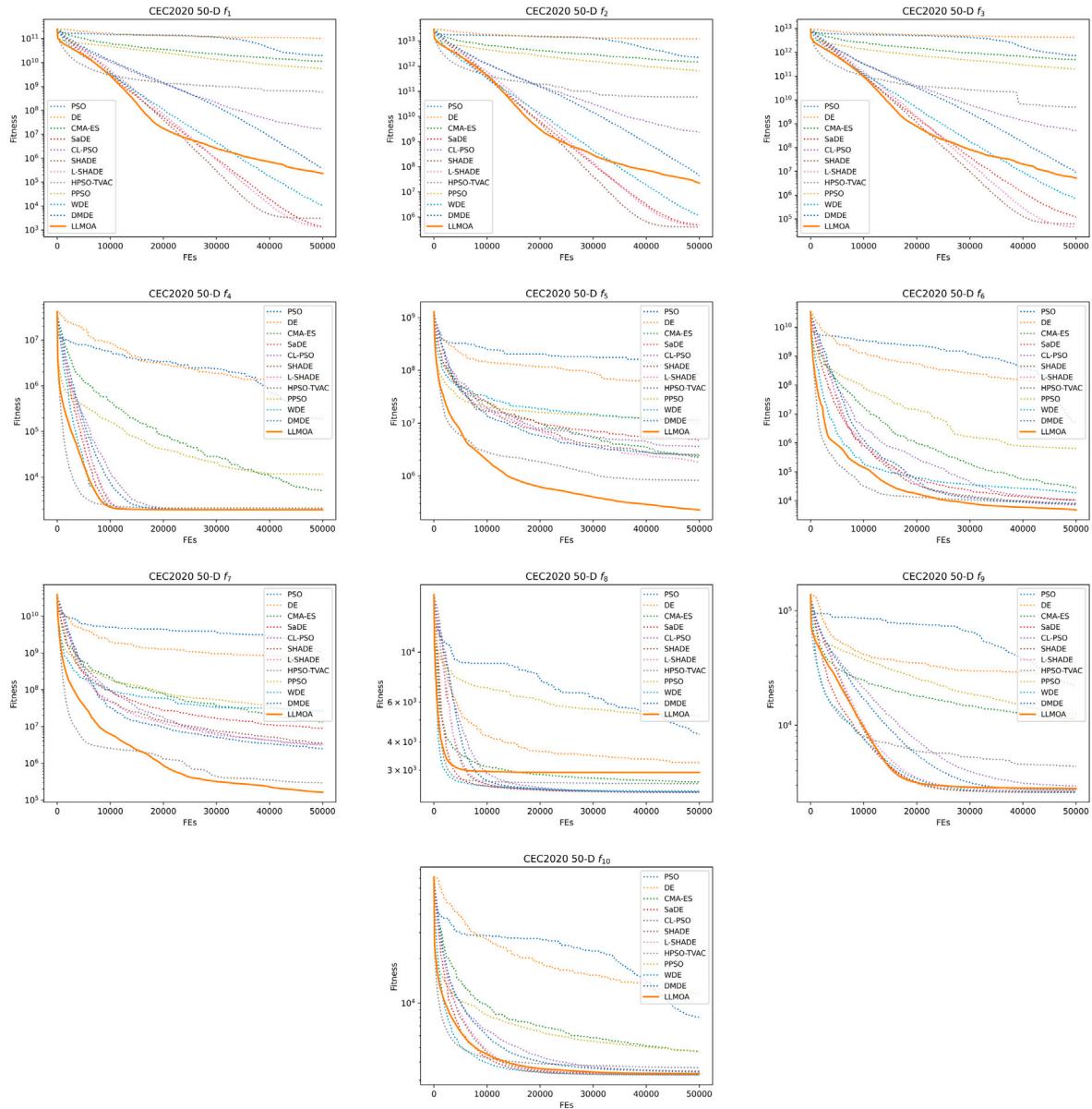


Fig. 28. Convergence graphs on 50-D CEC2020 mathematical functions.

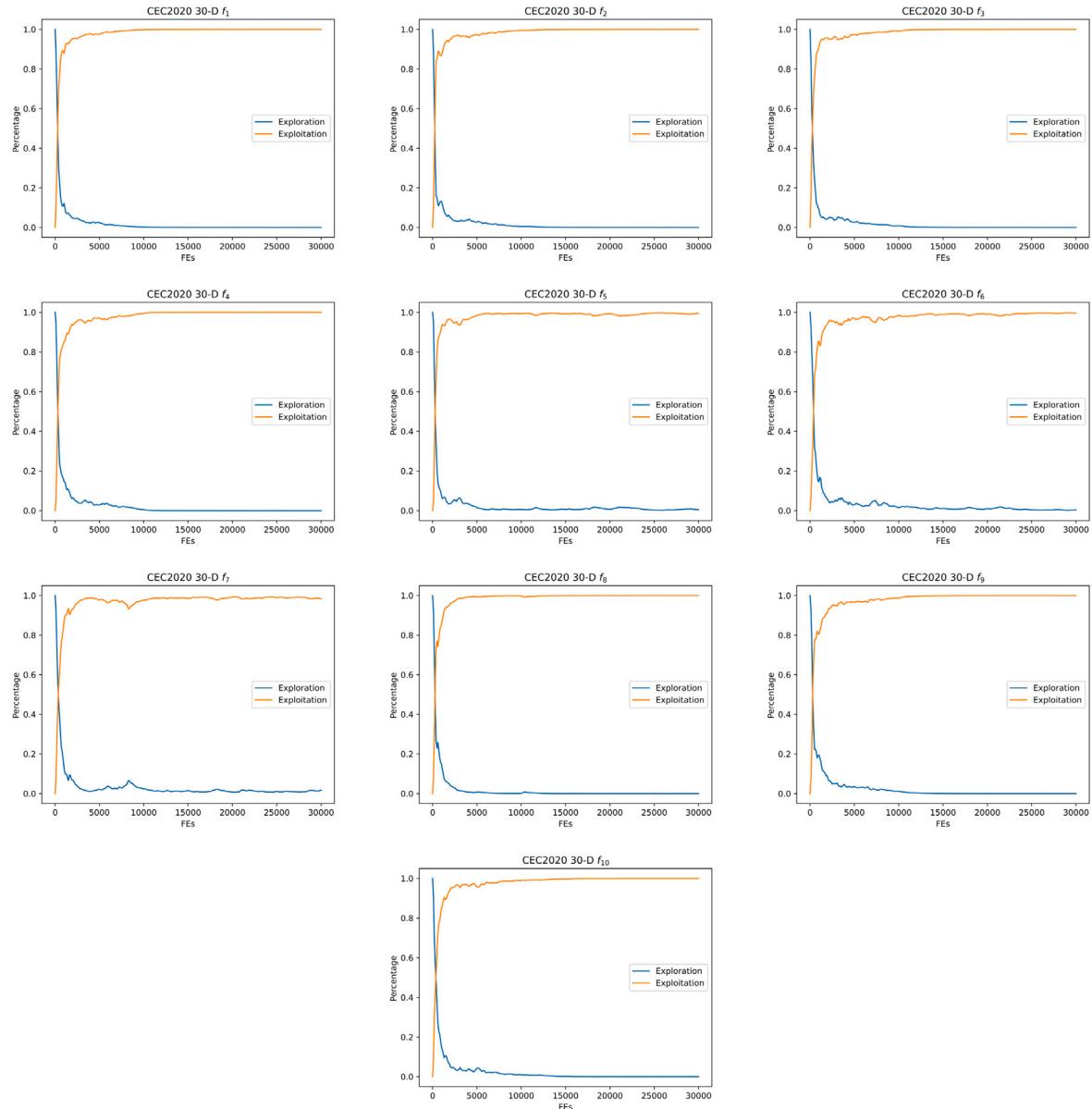


Fig. 29. Plots of exploration and exploitation proportion on 30-D CEC2020.

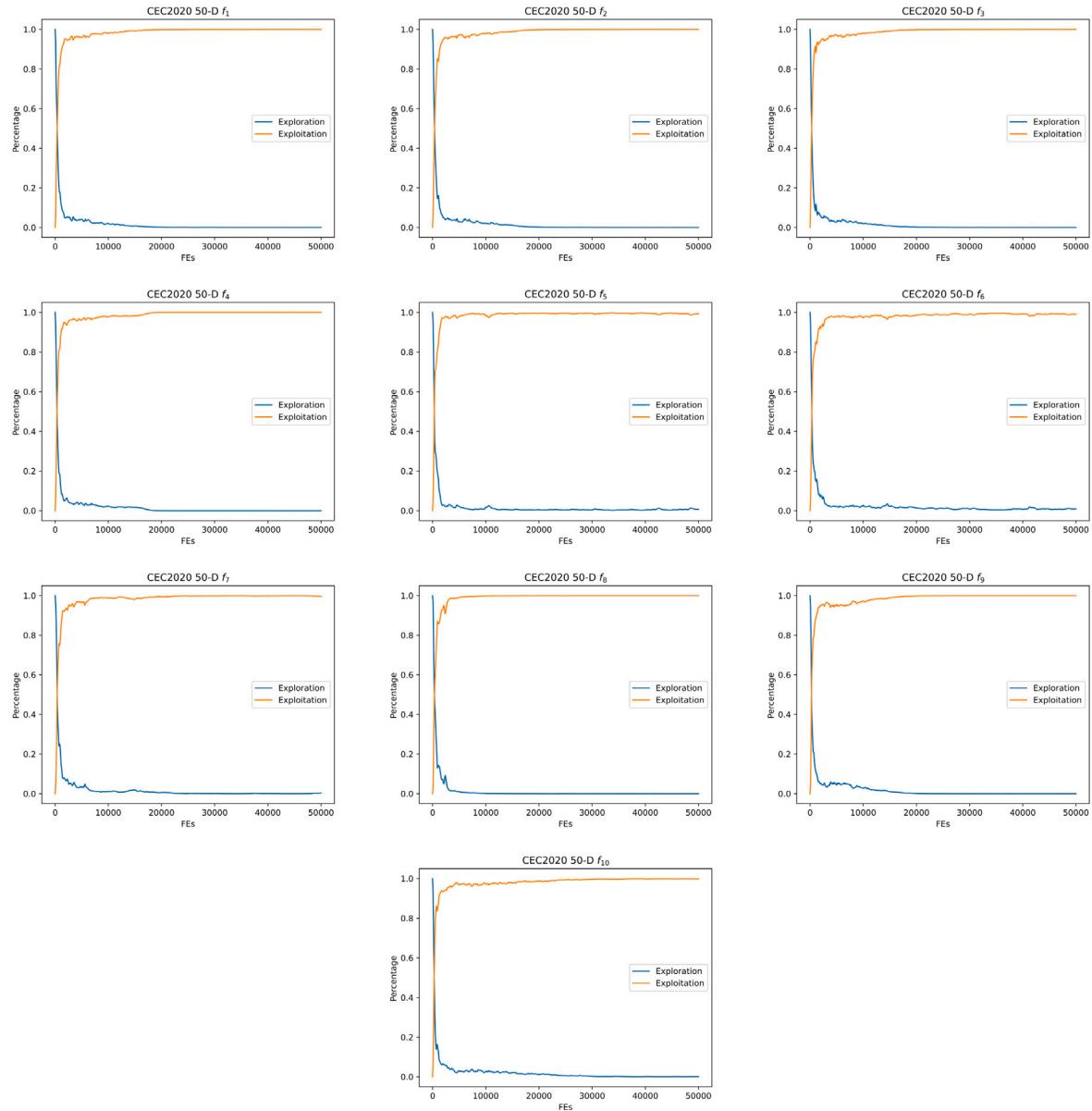


Fig. 30. Plots of exploration and exploitation proportion on 50-D CEC2020.

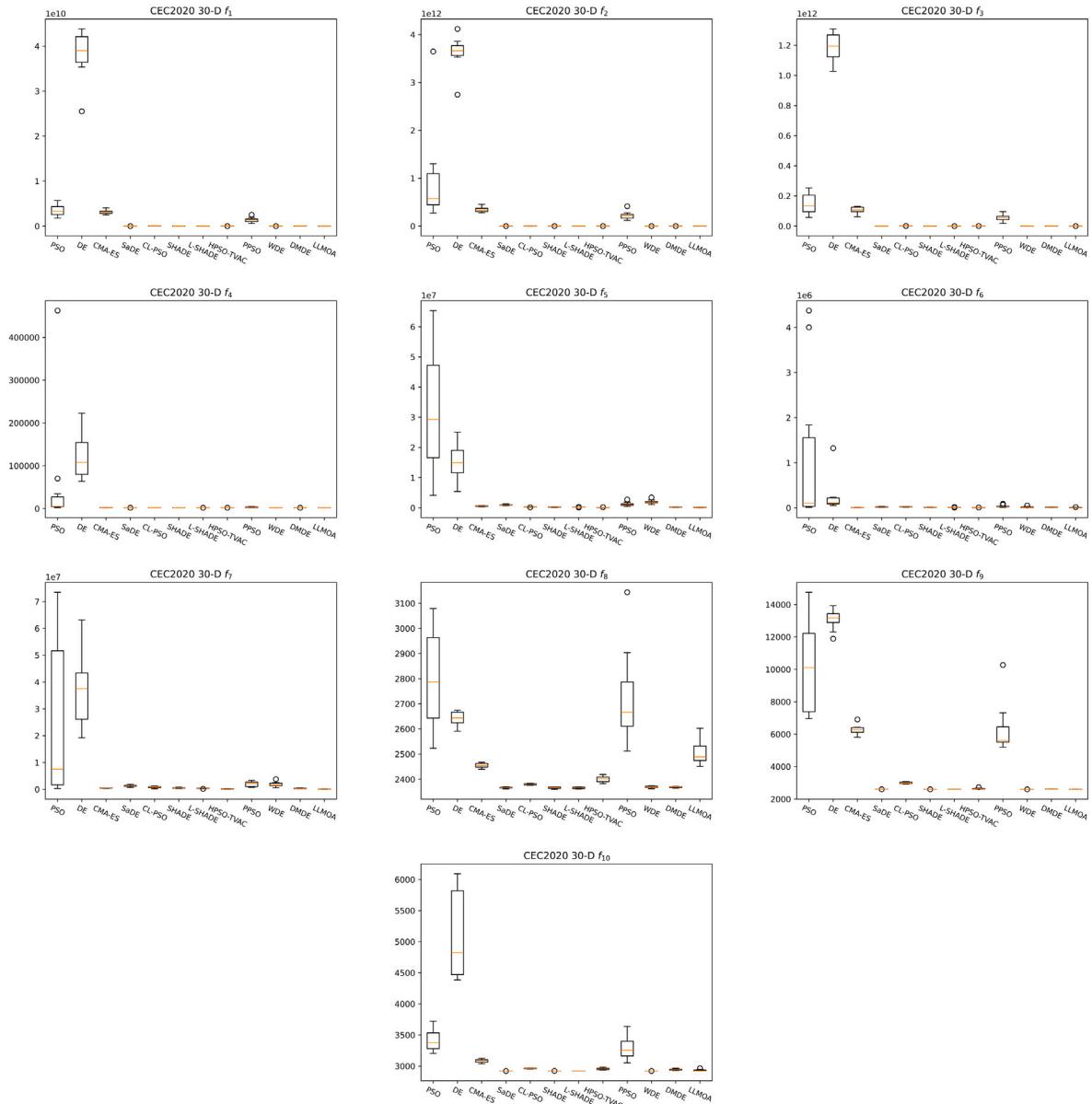


Fig. 31. Box plots of optimizers on 30-D CEC2020.

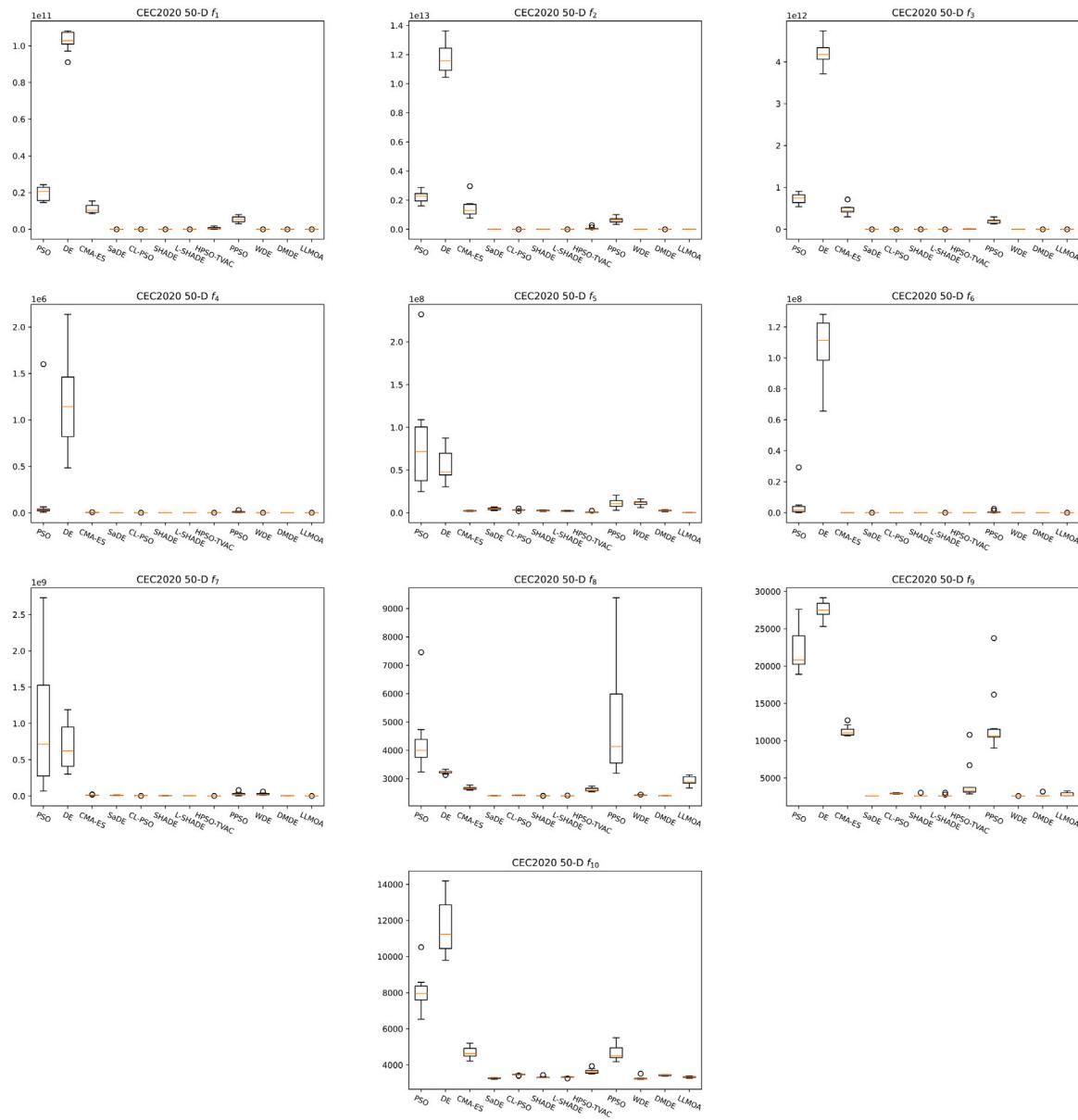


Fig. 32. Box plots of optimizers on 50-D CEC2020.

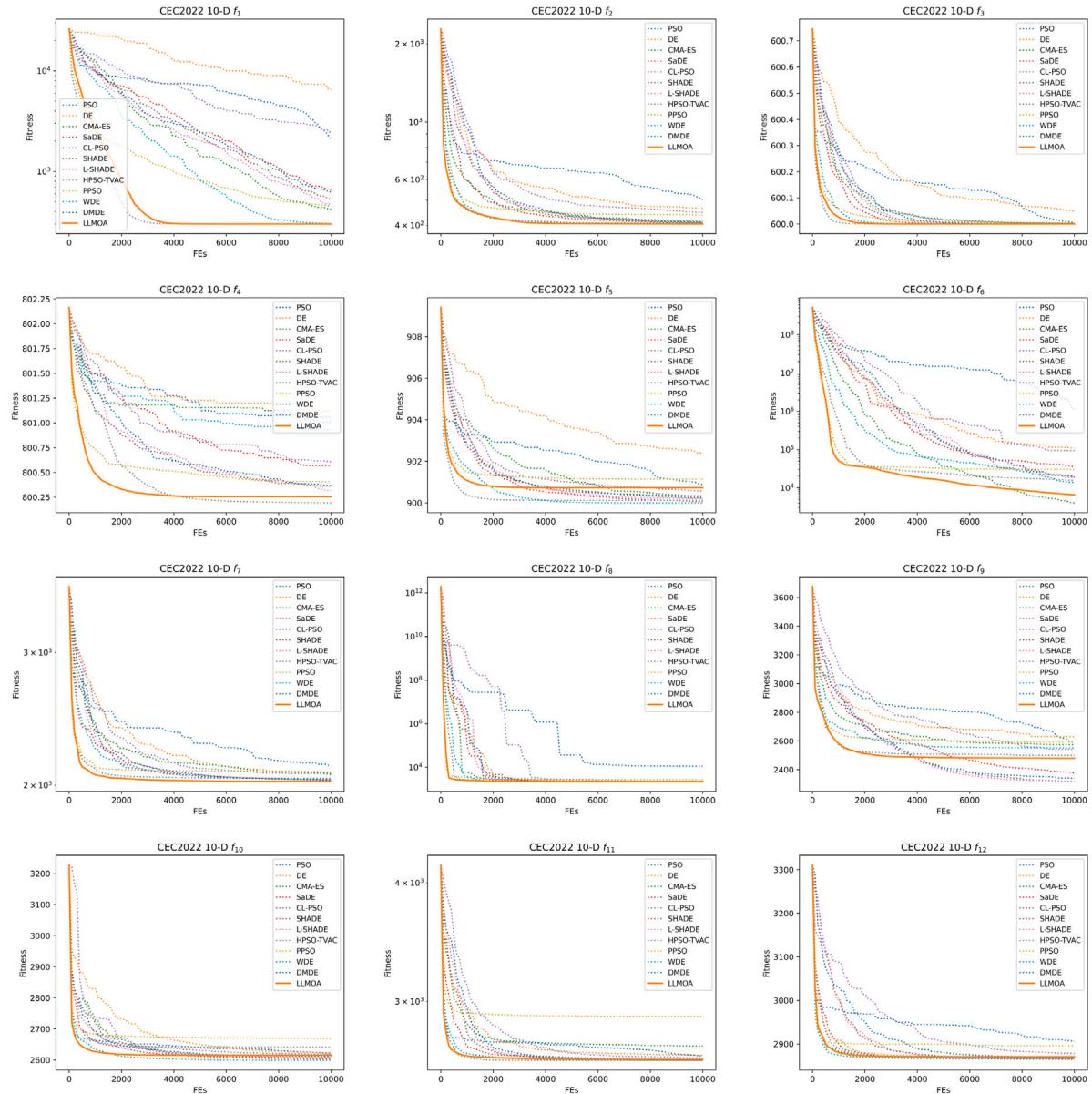


Fig. 33. Convergence graphs on 10-D CEC2022 mathematical functions.

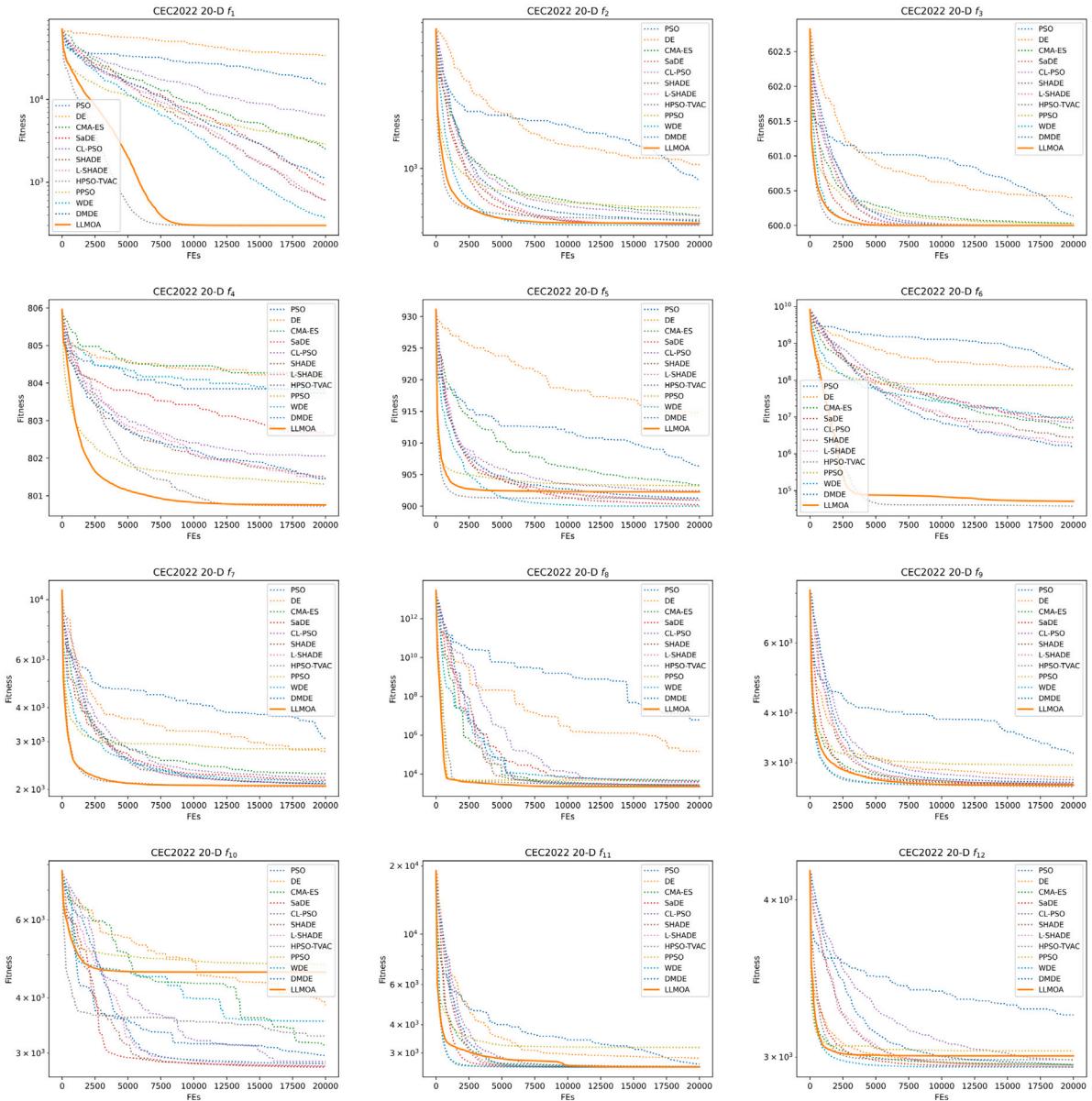


Fig. 34. Convergence graphs on 20-D CEC2022 mathematical functions.

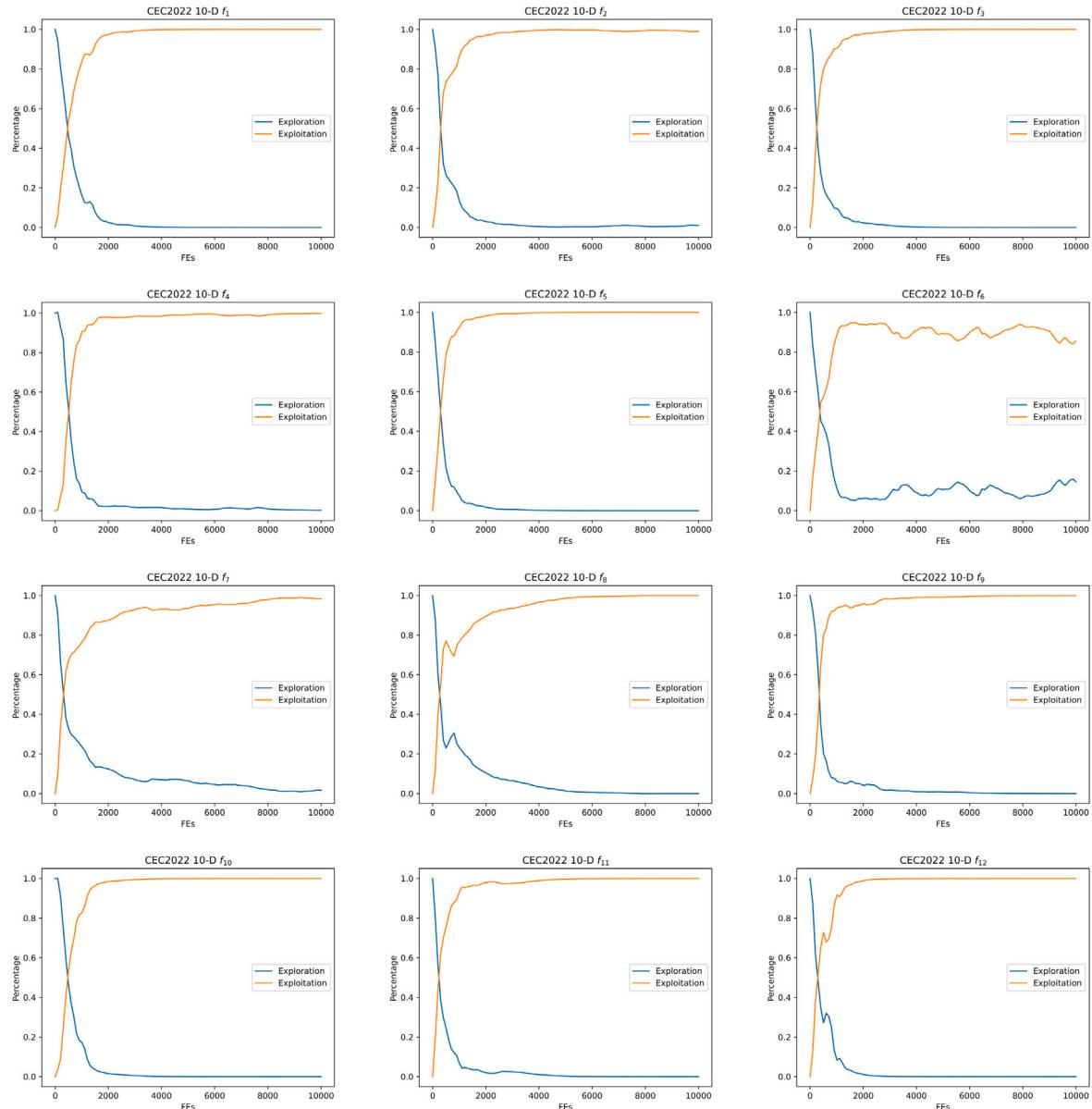


Fig. 35. Plots of exploration and exploitation proportion on 10-D CEC2022 benchmark functions.

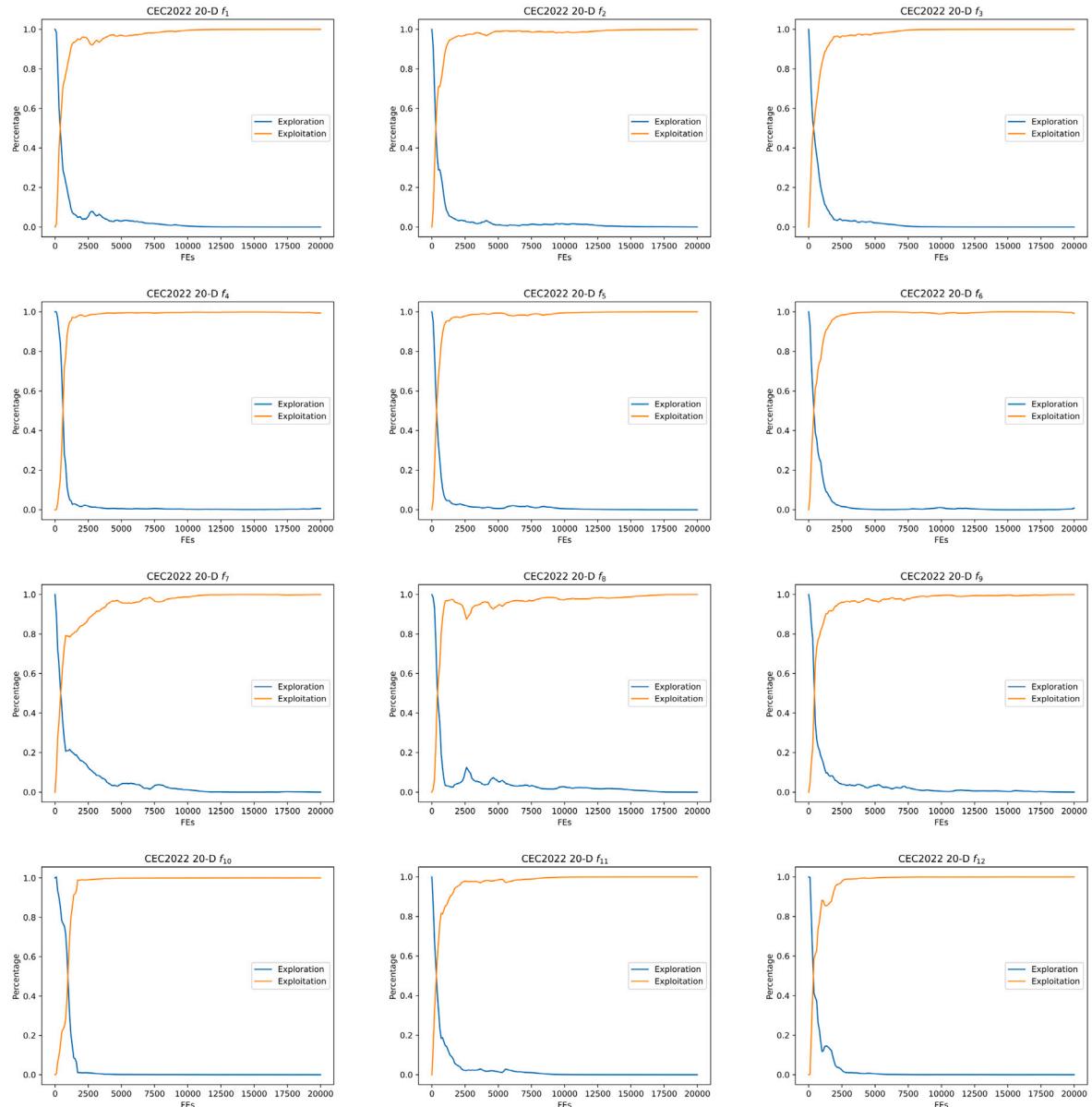


Fig. 36. Plots of exploration and exploitation proportion on 20-D CEC2022 benchmark functions.

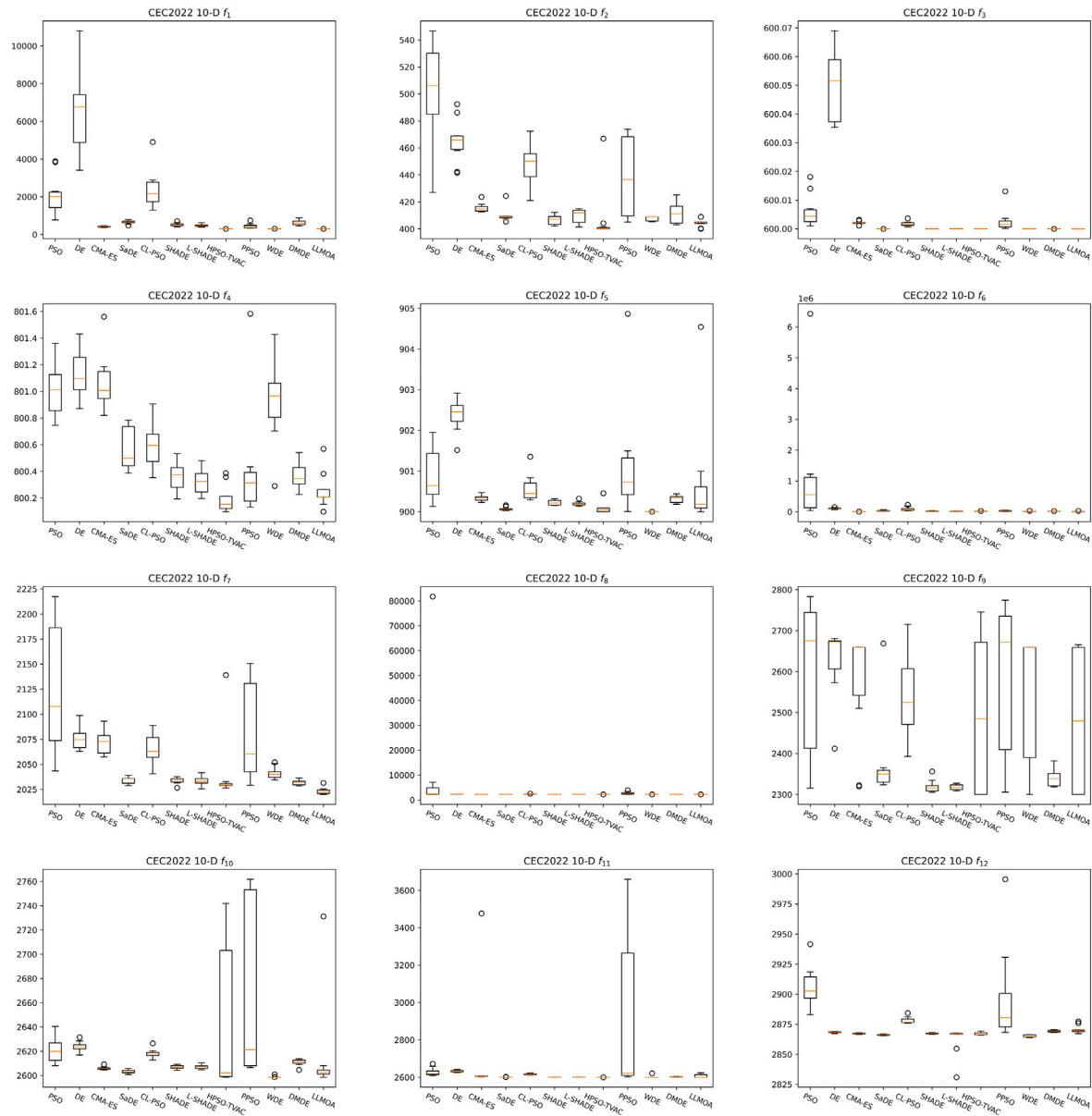


Fig. 37. Box plots of optimizers on 10-D CEC2022.

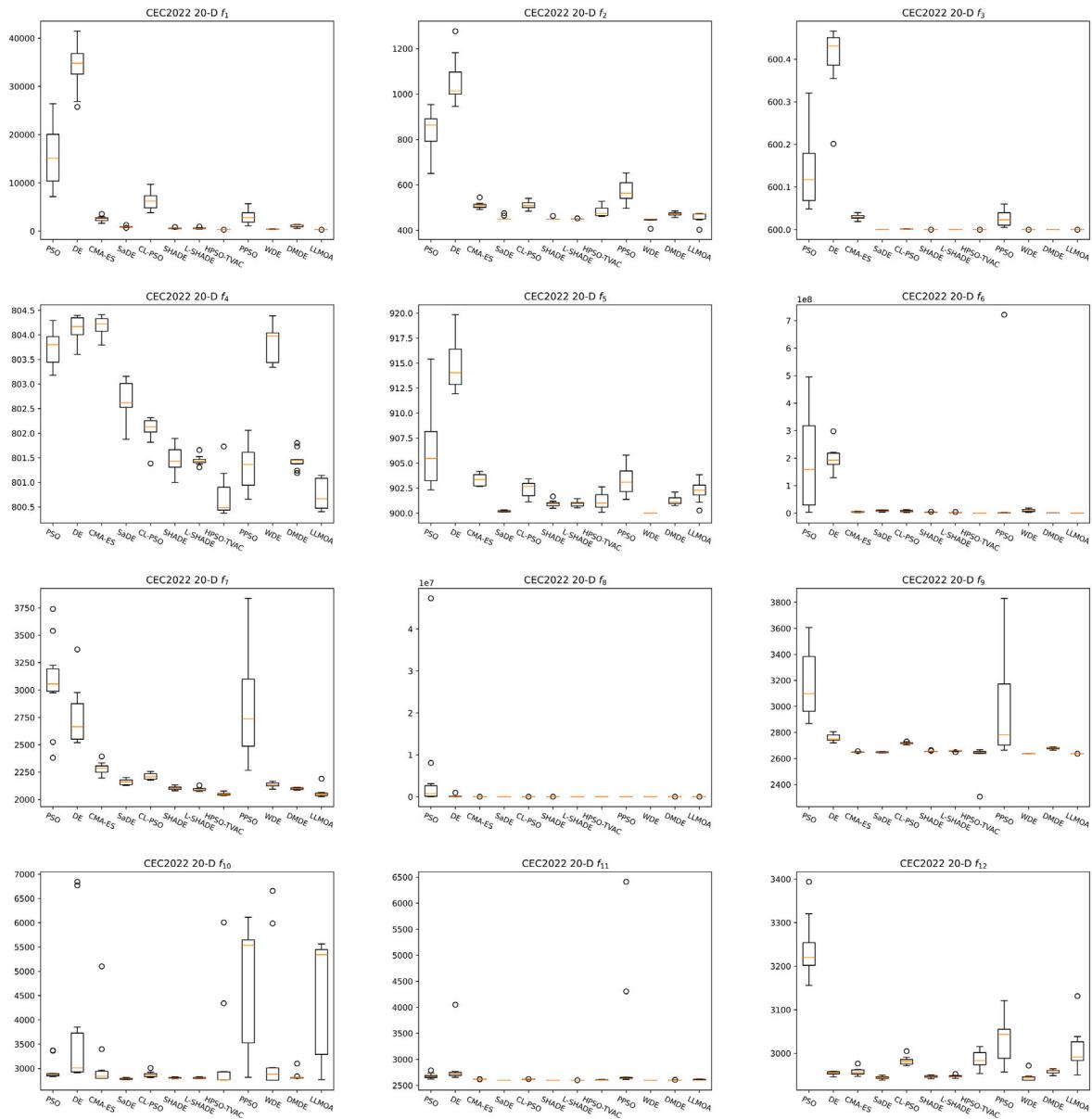


Fig. 38. Box plots of optimizers on 20-D CEC2022.

Table 36
Results and analyses on 10-D CEC2022 benchmark functions.

Func.	PSO	DE	CMA-ES	SaDE	CL-PSO	SHADE	L-SHADE	HPSO-TVAC	PPSO	WDE	DMDE	LLMOA	
f_1	mean std	2.095e+03 + 1.016e+03	6.559e+03 + 2.068e+03	4.210e+02 + 2.888e+01	6.491e+02 + 8.668e+01	2.394e+03 + 9.840e+02	5.208e+02 + 9.075e+01	4.745e+02 + 6.421e+01	3.000e+02 - 3.614e-10	4.554e+02 + 1.211e+02	3.027e+02 + 1.920e+00	6.234e+02 + 1.333e+02	3.000e+02 + 1.043e-06
f_2	mean std	5.024e+02 + 3.572e+01	4.651e+02 + 1.534e+01	4.156e+02 + 3.195e+00	4.100e+02 + 4.922e+00	4.489e+02 + 1.459e+01	4.067e+02 ≈ 3.386e+00	4.097e+02 + 4.938e+00	4.073e+02 + 1.992e+01	4.384e+02 + 2.859e+01	4.076e+02 ≈ 1.651e+00	4.112e+02 + 7.252e+00	4.046e+02 + 2.770e+00
f_3	mean std	6.000e+02 + 5.291e-03	6.000e+02 + 1.169e-02	6.000e+02 + 6.009e-04	6.000e+02 + 3.569e-07	6.000e+02 + 7.853e-04	6.000e+02 + 5.514e-06	6.000e+02 + 3.683e-06	6.000e+02 + 0.000e+00	6.000e+02 + 3.631e-03	6.000e+02 + 2.848e-09	6.000e+02 + 1.006e-05	6.000e+02 + 4.814e-12
f_4	mean std	8.010e+02 + 1.835e-01	8.011e+02 + 1.776e-01	8.011e+02 + 2.050e-01	8.006e+02 + 1.528e-01	8.006e+02 + 1.802e-01	8.004e+02 ≈ 9.828e-02	8.003e+02 ≈ 8.900e-02	8.002e+02 - 9.727e-02	8.004e+02 + 4.048e-01	8.004e+02 + 2.947e-01	8.004e+02 + 8.832e-02	8.003e+02 + 1.265e-01
f_5	mean std	9.009e+02 ≈ 5.920e-01	9.024e+02 + 3.974e-01	9.003e+02 ≈ 6.720e-02	9.001e+02 ≈ 4.141e-02	9.006e+02 ≈ 3.167e-01	9.002e+02 ≈ 6.700e-02	9.001e+02 ≈ 5.228e-02	9.011e+02 ≈ 1.727e-01	9.000e+02 - 1.336e+00	9.003e+02 ≈ 8.117e-06	9.003e+02 ≈ 8.757e-02	9.007e+02 + 1.310e+00
f_6	mean std	1.138e+06 + 1.820e+06	1.083e+05 + 2.246e+04	4.040e+03 ≈ 4.194e+02	3.519e+04 + 1.519e+04	9.330e+04 + 6.165e+04	2.016e+04 + 4.298e+03	1.841e+04 + 5.653e+03	1.565e+04 + 5.603e+03	3.032e+04 + 1.472e+04	1.398e+04 + 9.699e+03	1.895e+04 + 1.910e+03	6.541e+03 + 7.785e+03
f_7	mean std	2.123e+03 + 6.167e+01	2.075e+03 + 1.034e+01	2.071e+03 + 1.081e+01	2.034e+03 + 3.273e+00	2.066e+03 + 1.415e+01	2.034e+03 + 3.092e+00	2.040e+03 + 4.709e+00	2.080e+03 + 3.290e+01	2.041e+03 + 4.678e+01	2.032e+03 + 5.597e+00	2.023e+03 + 2.570e+00	2.023e+03 + 3.270e+00
f_8	mean std	1.109e+04 + 2.362e+04	2.259e+03 + 7.772e+00	2.229e+03 + 1.677e+00	2.232e+03 + 3.426e+00	2.315e+03 + 7.816e+01	2.231e+03 + 4.352e+00	2.229e+03 + 4.528e+00	2.224e+03 + 6.114e+00	2.226e+03 + 5.228e+02	2.223e+03 + 4.108e+00	2.221e+03 + 3.576e+00	2.221e+03 + 7.387e+00
+/-	11/1/0	11/0/1	8/2/2	7/3/2	9/3/0	5/5/2	6/4/2	5/2/5	11/1/0	7/2/3	7/5/0	-	

Table 37
Results and analyses on 20-D CEC2022 benchmark functions.

Func.	PSO	DE	CMA-ES	SaDE	CL-PSO	SHADE	L-SHADE	HPSO-TVAC	PPSO	WDE	DMDE	LLMOA	
f_1	mean std	1.538e+04 + 6.072e+03	3.412e+04 + 4.679e+03	2.544e+03 + 5.214e+02	9.132e+02 + 1.503e+02	6.359e+03 + 1.829e+03	6.059e+02 + 1.028e+02	5.967e+02 + 1.507e+02	3.000e+02 + 1.628e-02	2.868e+03 + 1.367e+03	3.743e+02 + 3.546e+01	1.099e+03 + 2.674e+02	3.000e+02 + 1.052e-03
f_2	mean std	8.330e+02 + 9.590e+01	1.058e+03 + 1.390e+01	5.102e+02 + 8.622e+00	4.534e+02 ≈ 1.769e+01	5.123e+02 + 4.172e+00	4.509e+02 ≈ 1.235e+00	4.504e+02 ≈ 2.186e+01	4.826e+02 + 5.109e+01	5.713e+02 + 1.246e+01	4.440e+02 - 1.246e+01	4.724e+02 ≈ 7.950e+00	4.580e+02 + 2.130e+01
f_3	mean std	6.001e+02 + 8.122e-02	6.004e+02 + 7.511e-02	6.000e+02 + 5.501e-03	6.000e+02 + 6.834e-08	6.000e+02 + 3.718e-04	6.000e+02 + 3.482e-07	6.000e+02 + 5.299e-07	6.000e+02 + 4.079e-08	6.000e+02 + 1.786e-02	6.000e+02 + 3.230e-09	6.000e+02 + 2.036e-06	6.000e+02 + 1.747e-09
f_4	mean std	8.037e+02 + 3.679e-01	8.041e+02 + 2.398e-01	8.042e+02 + 1.859e-01	8.027e+02 + 3.847e-01	8.021e+02 + 2.665e-01	8.015e+02 + 2.683e-01	8.014e+02 + 8.930e-02	8.007e+02 ≈ 4.217e-01	8.013e+02 + 4.228e-01	8.038e+02 + 3.643e-01	8.015e+02 + 1.796e-01	8.008e+02 + 3.043e-01
f_5	mean std	9.063e+02 + 3.848e+00	9.148e+02 + 2.399e+00	9.032e+02 + 5.623e-01	9.002e+02 - 7.131e-02	9.024e+02 ≈ 7.644e-01	9.009e+02 - 3.105e-01	9.009e+02 - 2.603e-01	9.012e+02 - 7.724e-01	9.032e+02 + 1.393e+00	9.000e+02 - 2.787e-05	9.013e+02 - 4.333e-01	9.023e+02 - 1.009e+00
f_6	mean std	1.880e+08 + 1.727e+08	1.956e+08 + 4.568e+07	4.650e+06 + 1.627e+06	8.182e+06 + 3.121e+06	7.250e+06 + 9.771e+05	2.849e+06 + 9.298e+05	1.934e+06 + 7.826e+03	3.823e+04 - 2.164e+08	7.271e+07 + 4.986e+06	9.920e+06 + 2.049e+06	1.568e+06 + 4.098e+05	5.170e+04 + 2.244e+04
f_7	mean std	3.062e+03 + 3.844e+02	2.754e+03 + 2.549e+02	2.285e+03 + 5.277e+01	2.161e+03 + 2.521e+01	2.212e+03 + 2.741e+01	2.103e+03 + 1.644e+01	2.092e+03 + 1.580e+01	2.050e+03 + 1.211e+01	2.022e+03 + 1.484e+02	2.134e+03 + 2.196e+01	2.099e+03 + 9.126e+00	2.059e+03 + 4.540e+01
f_8	mean std	6.134e+06 + 1.394e+07	1.429e+05 + 2.773e+05	2.430e+03 + 1.000e+02	4.031e+03 + 1.101e+03	3.720e+03 + 5.345e+02	2.750e+03 + 1.721e+02	2.643e+03 + 1.821e+02	2.411e+03 + 1.420e+02	4.741e+03 + 1.125e+03	4.443e+03 + 8.236e+02	2.726e+03 + 1.997e+02	2.261e+03 + 7.610e+01
f_9	mean std	3.172e+03 + 2.568e+02	2.758e+03 + 2.663e+01	2.648e+03 + 3.469e+00	2.647e+03 + 3.508e+00	2.718e+03 + 7.642e+00	2.656e+03 + 3.338e+00	2.657e+03 + 3.681e+00	2.615e+03 - 1.035e+02	2.962e+03 + 3.556e+02	2.636e+03 + 3.421e-01	2.678e+03 + 7.006e+00	2.636e+03 + 1.755e-01
f_{10}	mean std	2.962e+03 ≈ 2.038e+02	3.852e+03 ≈ 1.503e+03	3.121e+03 ≈ 6.831e+02	2.789e+03 ≈ 5.317e+01	2.869e+03 ≈ 5.770e+01	2.804e+03 ≈ 1.077e+01	2.804e+03 ≈ 1.260e+01	3.279e+03 ≈ 1.020e+03	4.759e+03 ≈ 1.257e+03	3.544e+03 ≈ 1.401e+03	2.840e+03 ≈ 8.721e+01	4.572e+03 ≈ 1.192e+03
f_{11}	mean std	2.683e+03 + 4.687e+01	2.846e+03 + 4.029e+02	2.620e+03 + 1.646e+00	2.600e+03 ≈ 2.848e-02	2.618e+03 + 3.260e+00	2.600e+03 ≈ 3.926e-02	2.600e+03 ≈ 4.187e-02	2.604e+03 + 4.689e+00	3.178e+03 + 1.188e+03	2.600e+03 ≈ 1.609e-03	2.603e+03 ≈ 2.476e+00	2.607e+03 ≈ 8.628e+00
f_{12}	mean std	3.240e+03 + 6.763e+01	2.955e+03 - 4.155e+00	2.944e+03 - 3.354e+00	2.983e+03 ≈ 9.535e+00	2.947e+03 - 2.575e+00	2.947e+03 - 2.677e+00	2.985e+03 - 1.929e+01	3.034e+03 ≈ 4.989e+01	2.945e+03 - 9.494e+00	2.958e+03 - 4.621e+00	3.006e+03 + 4.961e+01	3.006e+03 + 2.958e+00
+/-	11/1/0	10/1/1	10/1/1	7/2/3	9/3/0	7/3/2	7/3/2	5/2/5	10/2/0	7/2/3	7/3/2	-	

Table 38

Findings from the ablation experiments conducted on 30-D CEC2014 benchmark functions.

Func	HHA _{SR}		HHA _{RP}		HHA _{SHA}		LLMOA	
	mean	std	mean	std	mean	std	mean	std
f_1	8.868e+06 +	2.873e+06	6.772e+06 +	3.624e+06	1.444e+06 −	6.471e+05	2.183e+06	7.172e+05
f_2	2.980e+06 +	1.636e+06	1.962e+06 +	1.039e+06	2.258e+04 ≈	1.660e+04	2.257e+04	1.435e+04
f_3	5.040e+03 +	2.532e+03	4.481e+03 +	2.631e+03	3.787e+02 ≈	5.385e+01	3.991e+02	1.132e+02
f_4	5.315e+02 +	2.053e+01	5.493e+02 +	5.662e+01	4.906e+02 ≈	1.880e+01	5.026e+02	2.585e+01
f_5	5.205e+02 ≈	1.414e−01	5.204e+02 ≈	1.738e−01	5.203e+02 ≈	3.140e−01	5.203e+02	1.973e−01
f_6	6.315e+02 +	2.588e+00	6.292e+02 ≈	4.683e+00	6.310e+02 ≈	3.311e+00	6.284e+02	3.657e+00
f_7	7.008e+02 +	1.866e−01	7.009e+02 +	1.948e−01	7.000e+02 ≈	3.344e−02	7.000e+02	2.936e−02
f_8	8.962e+02 ≈	2.848e+01	9.200e+02 +	2.156e+01	9.130e+02 ≈	5.080e+01	8.933e+02	2.810e+01
f_9	1.024e+03 ≈	3.108e+01	1.052e+03 ≈	3.013e+01	1.062e+03 ≈	6.487e+01	1.044e+03	2.165e+01
f_{10}	3.882e+03 ≈	1.034e+03	3.927e+03 ≈	8.387e+02	3.759e+03 ≈	8.395e+02	3.511e+03	7.020e+02
f_{11}	4.993e+03 −	5.511e+02	5.154e+03 ≈	4.993e+02	5.282e+03 ≈	5.364e+02	5.566e+03	4.871e+02
f_{12}	1.200e+03 ≈	1.187e−01	1.201e+03 +	2.148e−01	1.200e+03 ≈	2.556e−01	1.200e+03	9.827e−02
f_{13}	1.301e+03 ≈	1.211e−01	1.301e+03 +	1.380e−01	1.301e+03 ≈	1.415e−01	1.300e+03	9.861e−02
f_{14}	1.400e+03 ≈	1.240e−01	1.400e+03 ≈	1.189e−01	1.400e+03 ≈	2.065e−01	1.400e+03	1.516e−01
f_{15}	1.530e+03 +	8.913e+00	1.522e+03 ≈	4.271e+00	1.518e+03 ≈	4.765e+00	1.516e+03	4.899e+00
f_{16}	1.612e+03 ≈	5.485e−01	1.612e+03 ≈	4.368e−01	1.613e+03 ≈	4.655e−01	1.613e+03	4.352e−01
f_{17}	1.658e+05 +	1.380e+05	1.581e+05 +	9.156e+04	8.283e+04 ≈	3.897e+04	7.872e+04	5.105e+04
f_{18}	8.352e+04 ≈	3.133e+04	9.364e+04 ≈	4.233e+04	8.033e+04 ≈	3.548e+04	9.451e+04	3.662e+04
f_{19}	2.610e+03 ≈	1.605e+03	2.059e+03 ≈	2.984e+02	2.022e+03 ≈	2.038e+02	1.938e+03	2.731e+01
f_{20}	1.050e+05 ≈	3.835e+04	1.075e+05 +	3.925e+04	6.612e+04 ≈	2.285e+04	5.640e+04	2.130e+04
f_{21}	2.391e+05 +	2.384e+05	1.281e+05 +	7.871e+04	4.101e+04 ≈	2.813e+04	3.995e+04	2.091e+04
f_{22}	2.679e+03 ≈	1.456e+02	2.752e+03 ≈	2.895e+02	2.476e+03 ≈	2.270e+02	2.651e+03	2.421e+02
f_{23}	2.537e+03 +	4.229e+00	2.538e+03 +	3.211e+00	2.523e+03 ≈	6.663e−02	2.523e+03	8.642e−01
f_{24}	2.643e+03 +	1.437e+01	2.647e+03 +	8.586e+00	2.641e+03 ≈	1.012e+01	2.636e+03	6.864e+00
f_{25}	2.727e+03 ≈	6.229e+00	2.729e+03 ≈	8.204e+00	2.725e+03 ≈	5.792e+00	2.723e+03	6.865e+00
f_{26}	2.701e+03 ≈	1.950e−01	2.711e+03 ≈	2.986e+01	2.731e+03 ≈	4.556e+01	2.701e+03	1.876e−01
f_{27}	3.647e+03 ≈	3.619e+02	3.724e+03 ≈	2.351e+02	3.651e+03 ≈	3.136e+02	3.704e+03	2.210e+02
f_{28}	4.678e+03 ≈	6.733e+02	4.842e+03 ≈	4.259e+02	4.615e+03 ≈	3.057e+02	4.632e+03	3.656e+02
f_{29}	7.296e+05 +	1.237e+06	6.265e+06 +	1.015e+07	1.080e+04 ≈	4.211e+03	1.635e+06	4.838e+06
f_{30}	5.401e+06 +	7.628e+06	3.703e+06 +	4.017e+06	7.777e+05 ≈	1.982e+06	1.867e+05	1.635e+05
+/-/-	13/16/1		15/15/0		0/29/1		-	

Table 39

Findings from the ablation experiments conducted on 50-D CEC2014 benchmark functions.

Func	HHA _{SR}		HHA _{RP}		HHA _{SHA}		LLMOA	
	mean	std	mean	std	mean	std	mean	std
f_1	1.249e+07 +	4.751e+06	1.510e+07 +	7.799e+06	3.820e+06 ≈	1.343e+06	4.051e+06	1.260e+06
f_2	1.008e+07 +	2.778e+06	1.394e+07 +	3.540e+06	2.512e+05 ≈	2.820e+05	2.788e+05	1.688e+05
f_3	1.647e+04 +	4.851e+03	1.684e+04 +	5.649e+03	5.822e+03 +	2.119e+03	3.602e+03	9.585e+02
f_4	6.321e+02 +	6.309e+01	6.084e+02 +	6.645e+01	5.349e+02 ≈	6.173e+01	5.406e+02	2.804e+01
f_5	5.206e+02 +	1.834e−01	5.205e+02 +	1.708e−01	5.205e+02 ≈	3.141e−01	5.203e+02	2.056e−01
f_6	6.560e+02 ≈	5.954e+00	6.563e+02 ≈	3.996e+00	6.563e+02 ≈	3.726e+00	6.564e+02	4.359e+00
f_7	7.011e+02 +	2.516e−02	7.012e+02 +	7.891e−02	7.003e+02 ≈	2.474e−01	7.003e+02	1.294e−01
f_8	1.032e+03 +	2.537e+01	1.085e+03 +	1.053e+02	1.014e+03 ≈	3.952e+01	1.007e+03	2.608e+01
f_9	1.187e+03 ≈	5.807e+01	1.261e+03 +	1.188e+02	1.186e+03 ≈	4.998e+01	1.181e+03	4.654e+01
f_{10}	7.260e+03 ≈	9.930e+02	6.453e+03 ≈	8.359e+02	6.170e+03 ≈	7.178e+02	6.800e+03	9.310e+02
f_{11}	8.573e+03 ≈	8.530e+02	8.566e+03 ≈	8.990e+02	8.284e+03 ≈	9.167e+02	7.928e+03	1.023e+03
f_{12}	1.201e+03 ≈	3.033e−01	1.201e+03 ≈	1.926e−01	1.201e+03 ≈	3.891e−01	1.201e+03	1.432e−01
f_{13}	1.301e+03 ≈	6.716e−02	1.301e+03 ≈	1.235e−01	1.301e+03 ≈	1.351e−01	1.301e+03	1.046e−01
f_{14}	1.401e+03 ≈	2.024e−01	1.401e+03 ≈	2.443e−01	1.401e+03 ≈	2.318e−01	1.401e+03	1.998e−01
f_{15}	1.563e+03 +	2.072e+01	1.583e+03 +	2.622e+01	1.542e+03 +	7.783e+00	1.536e+03	4.502e+00
f_{16}	1.622e+03 ≈	5.849e−01	1.622e+03 ≈	7.875e−01	1.622e+03 ≈	7.121e−01	1.622e+03	4.418e−01
f_{17}	8.454e+05 +	4.668e+05	7.190e+05 +	2.243e+05	2.928e+05 ≈	1.110e+05	2.837e+05	9.708e+04
f_{18}	1.293e+05 ≈	2.708e+04	1.645e+05 ≈	4.081e+04	1.400e+05 ≈	3.371e+04	1.465e+05	3.404e+04
f_{19}	2.856e+03 +	1.954e+03	1.998e+03 ≈	9.286e+01	2.074e+03 ≈	3.268e+02	2.021e+03	1.834e+02
f_{20}	2.051e+05 +	6.733e+04	2.297e+05 +	5.786e+04	1.302e+05 ≈	3.579e+04	1.315e+05	3.222e+04
f_{21}	6.794e+05 +	4.414e+05	8.965e+05 +	6.026e+05	1.769e+05 ≈	1.593e+05	1.953e+05	1.422e+05
f_{22}	3.480e+03 +	5.934e+02	3.792e+03 +	6.460e+02	3.037e+03 +	2.362e+02	2.775e+03	2.102e+02
f_{23}	2.536e+03 +	4.819e−01	2.538e+03 +	2.842e+00	2.535e+03 ≈	6.623e−02	2.535e+03	2.595e−02
f_{24}	2.731e+03 ≈	2.715e+01	2.736e+03 +	1.730e+01	2.716e+03 ≈	1.710e+01	2.713e+03	1.525e+01
f_{25}	2.756e+03 +	1.048e+01	2.754e+03 ≈	1.305e+01	2.749e+03 ≈	1.309e+01	2.744e+03	1.097e+01
f_{26}	2.741e+03 ≈	4.897e+01	2.791e+03 +	3.008e+01	2.711e+03 ≈	2.988e+01	2.741e+03	4.869e+01
f_{27}	4.557e+03 ≈	1.724e+02	4.603e+03 ≈	2.054e+02	4.454e+03 ≈	1.166e+02	4.514e+03	7.991e+01
f_{28}	6.714e+03 ≈	8.388e+02	6.863e+03 ≈	7.591e+02	6.508e+03 ≈	1.000e+03	6.768e+03	5.403e+02
f_{29}	7.565e+05 +	5.016e+05	1.138e+07 +	1.725e+07	8.707e+04 ≈	8.009e+04	5.576e+04	2.259e+04
f_{30}	4.874e+06 +	4.272e+06	1.194e+07 +	7.539e+06	4.563e+05 ≈	2.567e+05	5.648e+05	3.021e+05
+/-/-	17/13/0		18/12/0		3/27/0		-	

Table 40

Findings from the ablation experiments conducted on 30-D CEC2020 benchmark functions.

Func	HHA _{SR}		HHA _{RP}		HHA _{SHA}		LLMOA	
	mean	std	mean	std	mean	std	mean	std
f_1	1.367e+06 +	7.395e+05	1.793e+06 +	1.179e+06	1.011e+04 ≈	1.066e+04	9.100e+03	6.740e+03
f_2	2.871e+08 +	1.425e+08	2.066e+08 +	1.434e+08	2.872e+05 ≈	2.009e+05	7.839e+05	5.943e+05
f_3	8.785e+07 +	7.037e+07	7.645e+07 +	3.363e+07	2.103e+05 ≈	1.640e+05	3.775e+05	2.272e+05
f_4	1.927e+03 +	6.294e+00	1.925e+03 ≈	8.393e+00	1.916e+03 ≈	4.973e+00	1.918e+03	4.503e+00
f_5	2.065e+05 +	1.245e+05	2.022e+05 +	6.090e+04	1.123e+05 ≈	4.157e+04	1.089e+05	4.719e+04
f_6	1.689e+04 +	1.122e+04	9.316e+03 +	4.321e+03	2.633e+03 ≈	5.885e+02	5.307e+03	4.747e+03
f_7	1.982e+05 +	7.609e+04	2.695e+05 +	1.393e+05	1.306e+05 +	7.537e+04	7.460e+04	4.391e+04
f_8	2.562e+03 ≈	8.165e+01	2.569e+03 ≈	1.063e+02	2.623e+03 ≈	2.195e+02	2.503e+03	4.510e+01
f_9	2.755e+03 +	1.504e+02	2.729e+03 +	9.988e+01	2.603e+03 –	1.601e+00	2.604e+03	2.134e+00
f_{10}	2.937e+03 ≈	1.728e+01	2.939e+03 ≈	1.183e+01	2.931e+03 ≈	9.212e+00	2.932e+03	1.352e+01
+/-/-	8/2/0		7/3/0		1/8/1		–	

Table 41

Findings from the ablation experiments conducted on 50-D CEC2020 benchmark functions.

Func	HHA _{SR}		HHA _{RP}		HHA _{SHA}		LLMOA	
	mean	std	mean	std	mean	std	mean	std
f_1	8.429e+06 +	2.499e+06	1.231e+07 +	5.914e+06	2.233e+05 ≈	3.414e+05	2.292e+05	1.720e+05
f_2	1.123e+09 +	5.005e+08	1.547e+09 +	5.561e+08	2.810e+07 +	5.607e+07	2.271e+07	1.323e+07
f_3	2.901e+08 +	8.171e+07	4.422e+08 +	2.001e+08	6.210e+06 ≈	8.033e+06	5.213e+06	5.369e+06
f_4	1.966e+03 +	3.298e+01	1.977e+03 +	2.036e+01	1.939e+03 ≈	7.687e+00	1.938e+03	7.660e+00
f_5	6.989e+05 +	1.958e+05	8.039e+05 +	4.095e+05	2.597e+05 ≈	1.203e+05	2.281e+05	9.048e+04
f_6	2.030e+04 +	7.338e+03	2.369e+04 +	7.427e+03	3.620e+03 ≈	6.769e+02	4.851e+03	2.075e+03
f_7	5.902e+05 +	2.692e+05	6.994e+05 +	3.442e+05	1.870e+05 ≈	5.216e+04	1.628e+05	6.225e+04
f_8	3.337e+03 +	4.428e+02	3.889e+03 +	1.244e+03	3.226e+03 ≈	3.621e+02	2.933e+03	1.420e+02
f_9	3.208e+03 +	5.563e+02	3.113e+03 +	3.579e+02	2.730e+03 ≈	2.077e+02	2.806e+03	2.687e+02
f_{10}	3.433e+03 +	8.393e+01	3.496e+03 +	1.066e+02	3.382e+03 ≈	1.490e+02	3.334e+03	3.540e+01
+/-/-	10/0/0		10/0/0		1/9/0		–	

Table 42

Findings from the ablation experiments conducted on 10-D CEC2022 benchmark functions.

Func	HHA _{SR}		HHA _{RP}		HHA _{SHA}		LLMOA	
	mean	std	mean	std	mean	std	mean	std
f_1	3.024e+02 +	6.834e+00	3.088e+02 +	2.593e+01	3.000e+02 ≈	4.104e-06	3.000e+02	1.043e-06
f_2	4.063e+02 ≈	3.341e+00	4.155e+02 +	1.972e+01	4.051e+02 ≈	3.673e+00	4.046e+02	2.770e+00
f_3	6.000e+02 +	8.960e-07	6.000e+02 +	1.169e-07	6.000e+02 –	2.491e-13	6.000e+02	4.814e-12
f_4	8.003e+02 ≈	1.314e-01	8.003e+02 ≈	1.324e-01	8.003e+02 ≈	1.002e-01	8.003e+02	1.265e-01
f_5	9.009e+02 +	7.230e-01	9.008e+02 ≈	9.246e-01	9.007e+02 ≈	9.469e-01	9.007e+02	1.310e+00
f_6	1.941e+04 +	1.165e+04	1.093e+04 +	6.437e+03	9.372e+03 ≈	9.797e+03	6.541e+03	7.785e+03
f_7	2.022e+03 ≈	8.782e-01	2.045e+03 ≈	4.127e+01	2.022e+03 ≈	1.905e+00	2.023e+03	3.270e+00
f_8	2.223e+03 ≈	3.091e+00	2.222e+03 ≈	1.681e+00	2.222e+03 ≈	1.152e+00	2.221e+03	7.387e+00
f_9	2.541e+03 +	1.987e+02	2.530e+03 +	1.888e+02	2.626e+03 +	1.087e+02	2.480e+03	1.802e+02
f_{10}	2.622e+03 ≈	5.047e+01	2.619e+03 ≈	4.218e+01	2.628e+03 ≈	7.048e+01	2.615e+03	3.870e+01
f_{11}	2.608e+03 ≈	9.884e+00	2.609e+03 +	1.036e+01	2.609e+03 +	1.084e+01	2.606e+03	9.906e+00
f_{12}	2.880e+03 ≈	1.348e+01	2.882e+03 ≈	2.395e+01	2.876e+03 ≈	1.596e+01	2.871e+03	3.137e+00
+/-/-	5/7/0		6/6/0		2/9/1		–	

Table 43

Findings from the ablation experiments conducted on 20-D CEC2022 benchmark functions.

Func	HHA _{SR}		HHA _{RP}		HHA _{SHA}		LLMOA	
	mean	std	mean	std	mean	std	mean	std
f_1	3.041e+02 +	1.041e+01	3.041e+02 +	9.177e+00	3.000e+02 -	2.950e-04	3.000e+02	1.052e-03
f_2	4.573e+02 \approx	2.326e+01	4.785e+02 \approx	3.089e+01	4.464e+02 \approx	2.324e+01	4.580e+02	2.130e+01
f_3	6.000e+02 +	5.101e-06	6.000e+02 +	6.400e-06	6.000e+02 \approx	1.838e-09	6.000e+02	1.747e-09
f_4	8.009e+02 \approx	3.362e-01	8.008e+02 \approx	2.918e-01	8.008e+02 \approx	4.572e-01	8.008e+02	3.043e-01
f_5	9.018e+02 \approx	1.230e+00	9.030e+02 \approx	2.119e+00	9.029e+02 \approx	1.878e+00	9.023e+02	1.009e+00
f_6	5.932e+04 \approx	2.021e+04	5.876e+04 \approx	1.296e+04	5.715e+04 \approx	3.380e+04	5.170e+04	2.244e+04
f_7	2.069e+03 \approx	2.940e+01	2.102e+03 +	4.555e+01	2.061e+03 \approx	2.114e+01	2.059e+03	4.540e+01
f_8	2.240e+03 \approx	8.394e+00	2.245e+03 \approx	1.310e+01	2.239e+03 \approx	7.333e+00	2.261e+03	7.610e+01
f_9	2.666e+03 +	1.920e+01	2.630e+03 +	1.111e+02	2.636e+03 \approx	2.185e-01	2.636e+03	1.755e-01
f_{10}	4.097e+03 \approx	1.128e+03	4.027e+03 \approx	1.076e+03	4.534e+03 \approx	1.277e+03	4.572e+03	1.192e+03
f_{11}	2.763e+03 \approx	4.613e+02	2.612e+03 \approx	1.238e+01	2.604e+03 \approx	6.481e+00	2.607e+03	8.628e+00
f_{12}	2.999e+03 \approx	2.770e+01	3.004e+03 \approx	5.007e+01	3.036e+03 \approx	1.083e+02	3.006e+03	4.961e+01
+/-	2/10/0		4/8/0		0/11/1		-	

Table 44

Findings from the investigation experiments conducted on 30-D CEC2014 benchmark functions.

Func	LLMOA _{ERNIE}		LLMOA _{ChatGPT}		LLMOA _{Gemini}	
	mean	std	mean	std	mean	std
f_1	2.206e+06 \approx	7.183e+05	1.824e+06 \approx	9.654e+05	2.183e+06	7.172e+05
f_2	1.383e+04 \approx	9.184e+03	1.128e+04 \approx	7.604e+03	2.257e+04	1.435e+04
f_3	4.031e+02 \approx	8.181e+01	3.860e+02 \approx	8.290e+01	3.991e+02	1.132e+02
f_4	4.996e+02 \approx	2.388e+01	4.817e+02 \approx	3.204e+01	5.026e+02	2.585e+01
f_5	5.202e+02 \approx	1.842e-01	5.201e+02 -	6.672e-02	5.203e+02	1.973e-01
f_6	6.279e+02 \approx	4.219e+00	6.277e+02 \approx	2.410e+00	6.284e+02	3.657e+00
f_7	7.000e+02 \approx	2.196e-02	7.000e+02 \approx	2.679e-02	7.000e+02	2.936e-02
f_8	8.954e+02 \approx	1.644e+01	8.945e+02 \approx	2.197e+01	8.933e+02	2.810e+01
f_9	1.040e+03 \approx	1.472e+01	1.036e+03 \approx	4.040e+01	1.044e+03	2.165e+01
f_{10}	3.637e+03 \approx	4.889e+02	3.232e+03 \approx	7.456e+02	3.511e+03	7.020e+02
f_{11}	4.980e+03 \approx	4.768e+02	5.554e+03 \approx	4.749e+02	5.566e+03	4.871e+02
f_{12}	1.200e+03 \approx	1.928e-01	1.200e+03 \approx	6.677e-02	1.200e+03	9.827e-02
f_{13}	1.301e+03 \approx	1.618e-01	1.301e+03 \approx	1.281e-01	1.301e+03	9.861e-02
f_{14}	1.400e+03 \approx	1.568e-01	1.400e+03 \approx	1.884e-01	1.400e+03	1.516e-01
f_{15}	1.518e+03 \approx	4.337e+00	1.516e+03 \approx	4.047e+00	1.517e+03	4.899e+00
f_{16}	1.613e+03 \approx	4.515e-01	1.613e+03 \approx	6.876e-01	1.613e+03	4.352e-01
f_{17}	7.772e+04 \approx	3.328e+04	1.132e+05 +	4.985e+04	7.872e+04	5.105e+04
f_{18}	7.598e+04 \approx	2.600e+04	9.578e+04 \approx	3.390e+04	9.451e+04	3.662e+04
f_{19}	1.959e+03 \approx	7.886e+01	2.093e+03 \approx	3.094e+02	1.938e+03	2.731e+01
f_{20}	6.464e+04 \approx	3.749e+04	6.727e+04 \approx	3.464e+04	5.640e+04	2.130e+04
f_{21}	3.890e+04 \approx	1.556e+04	3.692e+04 \approx	9.897e+03	3.995e+04	2.091e+04
f_{22}	2.651e+03 \approx	2.062e+02	2.527e+03 \approx	2.096e+02	2.651e+03	2.421e+02
f_{23}	2.523e+03 \approx	4.027e-02	2.523e+03 \approx	3.104e-01	2.523e+03	8.642e-01
f_{24}	2.637e+03 \approx	1.288e+01	2.635e+03 \approx	1.453e+01	2.636e+03	6.864e+00
f_{25}	2.722e+03 \approx	6.099e+00	2.723e+03 \approx	4.302e+00	2.723e+03	6.865e+00
f_{26}	2.721e+03 \approx	3.971e+01	2.711e+03 \approx	2.978e+01	2.701e+03	1.876e-01
f_{27}	3.666e+03 \approx	1.975e+02	3.694e+03 \approx	2.290e+02	3.704e+03	2.210e+02
f_{28}	4.381e+03 \approx	2.523e+02	4.419e+03 \approx	3.210e+02	4.632e+03	3.656e+02
f_{29}	5.921e+05 \approx	1.737e+06	1.845e+06 \approx	3.668e+06	1.635e+06	4.838e+06
f_{30}	5.032e+05 \approx	9.009e+05	2.743e+05 \approx	3.868e+05	1.867e+05	1.635e+05
+/-	0/30/0		1/28/1		-	

Table 45
Findings from the investigation experiments conducted on 50-D CEC2014 benchmark functions.

Func	LLMOA _{ERNIE}		LLMOA _{ChatGPT}		LLMOA _{Gemini}	
	mean	std	mean	std	mean	std
f_1	4.906e+06 ≈	1.574e+06	4.273e+06 ≈	1.606e+06	4.051e+06	1.260e+06
f_2	2.002e+05 ≈	7.997e+04	2.457e+05 ≈	2.124e+05	2.788e+05	1.688e+05
f_3	3.898e+03 ≈	1.445e+03	2.966e+03 ≈	1.281e+03	3.602e+03	9.585e+02
f_4	5.686e+02 ≈	2.453e+01	5.425e+02 ≈	3.968e+01	5.406e+02	2.804e+01
f_5	5.203e+02 ≈	2.768e-01	5.203e+02 ≈	2.741e-01	5.203e+02	2.056e-01
f_6	6.556e+02 ≈	5.414e+00	6.563e+02 ≈	3.839e+00	6.564e+02	4.359e+00
f_7	7.004e+02 ≈	2.011e-01	7.003e+02 ≈	9.898e-02	7.003e+02	1.294e-01
f_8	1.028e+03 ≈	1.767e+01	1.016e+03 ≈	4.363e+01	1.007e+03	2.608e+01
f_9	1.197e+03 ≈	3.089e+01	1.207e+03 ≈	6.513e+01	1.181e+03	4.654e+01
f_{10}	6.190e+03 ≈	1.111e+03	6.483e+03 ≈	1.048e+03	6.800e+03	9.310e+02
f_{11}	8.678e+03 ≈	6.366e+02	8.105e+03 ≈	8.398e+02	7.928e+03	1.023e+03
f_{12}	1.201e+03 ≈	2.161e-01	1.201e+03 ≈	1.821e-01	1.201e+03	1.432e-01
f_{13}	1.301e+03 ≈	9.307e-02	1.301e+03 ≈	1.036e-01	1.301e+03	1.046e-01
f_{14}	1.401e+03 ≈	2.608e-01	1.400e+03 ≈	2.205e-01	1.401e+03	1.998e-01
f_{15}	1.540e+03 ≈	9.097e+00	1.539e+03 ≈	7.462e+00	1.536e+03	4.502e+00
f_{16}	1.622e+03 ≈	4.643e-01	1.622e+03 ≈	4.318e-01	1.622e+03	4.418e-01
f_{17}	2.774e+05 ≈	1.334e+05	2.597e+05 ≈	6.522e+04	2.837e+05	9.708e+04
f_{18}	1.229e+05 ≈	3.576e+04	1.242e+05 ≈	4.103e+04	1.465e+05	3.404e+04
f_{19}	2.026e+03 ≈	1.643e+02	1.978e+03 ≈	6.384e+01	2.021e+03	1.834e+02
f_{20}	1.193e+05 ≈	3.682e+04	1.385e+05 ≈	3.227e+04	1.315e+05	3.222e+04
f_{21}	2.012e+05 ≈	1.794e+05	1.644e+05 ≈	8.071e+04	1.953e+05	1.422e+05
f_{22}	2.817e+03 ≈	3.076e+02	2.887e+03 ≈	2.601e+02	2.775e+03	2.102e+02
f_{23}	2.535e+03 ≈	1.015e-02	2.535e+03 ≈	5.299e-02	2.535e+03	2.595e-02
f_{24}	2.716e+03 ≈	1.179e+01	2.713e+03 ≈	1.493e+01	2.713e+03	1.525e+01
f_{25}	2.744e+03 ≈	1.049e+01	2.753e+03 ≈	1.560e+01	2.744e+03	1.097e+01
f_{26}	2.780e+03 +	3.985e+01	2.760e+03 ≈	4.878e+01	2.741e+03	4.869e+01
f_{27}	4.412e+03 ≈	4.458e+02	4.508e+03 ≈	1.339e+02	4.514e+03	7.991e+01
f_{28}	6.546e+03 ≈	7.479e+02	6.021e+03 −	6.529e+02	6.768e+03	5.403e+02
f_{29}	5.052e+04 ≈	2.830e+04	4.559e+04 ≈	1.528e+04	5.576e+04	2.259e+04
f_{30}	6.185e+05 ≈	2.133e+05	1.249e+06 ≈	1.381e+06	5.648e+05	3.021e+05
+/-/-	1/29/0		0/29/1		-	

Table 46
Findings from the investigation experiments conducted on 30-D CEC2020 benchmark functions.

Func	LLMOA _{ERNIE}		LLMOA _{ChatGPT}		LLMOA _{Gemini}	
	mean	std	mean	std	mean	std
f_1	6.672e+03 ≈	4.761e+03	1.214e+04 ≈	1.177e+04	9.100e+03	6.740e+03
f_2	2.714e+06 ≈	5.206e+06	1.344e+06 ≈	9.331e+05	7.839e+05	5.943e+05
f_3	3.048e+05 ≈	2.244e+05	4.658e+05 ≈	2.812e+05	3.775e+05	2.272e+05
f_4	1.916e+03 ≈	4.228e+00	1.917e+03 ≈	4.994e+00	1.918e+03	4.503e+00
f_5	9.068e+04 ≈	5.874e+04	8.024e+04 ≈	2.954e+04	1.089e+05	4.719e+04
f_6	2.647e+03 ≈	3.391e+02	2.401e+03 ≈	7.399e+02	5.307e+03	4.747e+03
f_7	8.025e+04 ≈	4.279e+04	8.118e+04 ≈	3.966e+04	7.460e+04	4.391e+04
f_8	2.546e+03 ≈	7.124e+01	2.487e+03 ≈	5.676e+01	2.503e+03	4.510e+01
f_9	2.629e+03 ≈	7.487e+01	2.594e+03 ≈	3.077e+01	2.604e+03	2.134e+00
f_{10}	2.930e+03 ≈	6.080e+00	2.925e+03 ≈	2.636e+00	2.932e+03	1.352e+01
+/-/-	0/10/0		0/10/0		-	

Table 47
Findings from the investigation experiments conducted on 50-D CEC2020 benchmark functions.

Func	LLMOA _{ERNIE}		LLMOA _{ChatGPT}		LLMOA _{Gemini}	
	mean	std	mean	std	mean	std
f_1	3.180e+05 ≈	4.141e+05	1.493e+05 ≈	1.187e+05	2.292e+05	1.720e+05
f_2	3.768e+07 ≈	4.048e+07	3.357e+07 ≈	2.162e+07	2.271e+07	1.323e+07
f_3	9.020e+06 ≈	8.460e+06	7.970e+06 ≈	6.349e+06	5.213e+06	5.369e+06
f_4	1.946e+03 +	1.048e+01	1.939e+03 ≈	7.745e+00	1.938e+03	7.660e+00
f_5	2.994e+05 ≈	1.114e+05	2.635e+05 ≈	1.161e+05	2.281e+05	9.048e+04
f_6	4.094e+03 ≈	9.183e+02	4.381e+03 ≈	1.605e+03	4.851e+03	2.075e+03
f_7	1.486e+05 ≈	5.388e+04	1.861e+05 ≈	8.166e+04	1.628e+05	6.225e+04
f_8	3.062e+03 ≈	1.671e+02	3.200e+03 ≈	4.302e+02	2.933e+03	1.420e+02
f_9	2.775e+03 ≈	2.256e+02	2.715e+03 ≈	1.807e+02	2.806e+03	2.687e+02
f_{10}	3.385e+03 ≈	7.648e+01	3.357e+03 ≈	5.214e+01	3.334e+03	3.540e+01
+/-/-	1/9/0		0/10/0		-	

Table 48

Findings from the investigation experiments conducted on 10-D CEC2022 benchmark functions.

Func	LLMOA _{ERNIE}		LLMOA _{ChatGPT}		LLMOA _{Gemini}	
	mean	std	mean	std	mean	std
f_1	3.000e+02 ≈	1.512e-05	3.000e+02 ≈	3.309e-06	3.000e+02	1.043e-06
f_2	4.057e+02 ≈	3.272e+00	4.124e+02 ≈	1.976e+01	4.046e+02	2.770e+00
f_3	6.000e+02 ≈	7.860e-13	6.000e+02 ≈	4.533e-13	6.000e+02	4.814e-12
f_4	8.004e+02 ≈	2.884e-01	8.004e+02 ≈	1.670e-01	8.003e+02	1.265e-01
f_5	9.003e+02 ≈	2.563e-01	9.003e+02 ≈	2.931e-01	9.007e+02	1.310e+00
f_6	6.746e+03 ≈	4.230e+03	7.408e+03 ≈	7.438e+03	6.541e+03	7.785e+03
f_7	2.032e+03 ≈	2.935e+01	2.019e+03 ≈	5.344e+00	2.023e+03	3.270e+00
f_8	2.222e+03 ≈	4.987e-01	2.231e+03 ≈	2.992e+01	2.221e+03	7.387e+00
f_9	2.553e+03 ≈	1.655e+02	2.589e+03 ≈	1.443e+02	2.480e+03	1.802e+02
f_{10}	2.616e+03 ≈	3.705e+01	2.630e+03 ≈	5.625e+01	2.615e+03	3.870e+01
f_{11}	2.694e+03 ≈	2.611e+02	2.604e+03 ≈	8.753e+00	2.606e+03	9.906e+00
f_{12}	2.870e+03 ≈	5.688e+00	2.867e+03 –	1.909e+00	2.871e+03	3.137e+00
+/-/-	0/12/0		0/11/1		–	

Table 49

Findings from the investigation experiments conducted on 20-D CEC2022 benchmark functions.

Func	LLMOA _{ERNIE}		LLMOA _{ChatGPT}		LLMOA _{Gemini}	
	mean	std	mean	std	mean	std
f_1	3.000e+02 ≈	1.417e-03	3.000e+02 ≈	4.292e-03	3.000e+02	1.052e-03
f_2	4.434e+02 ≈	2.108e+01	4.529e+02 ≈	9.726e+00	4.580e+02	2.130e+01
f_3	6.000e+02 ≈	1.433e-08	6.000e+02 ≈	4.781e-09	6.000e+02	1.747e-09
f_4	8.007e+02 ≈	2.519e-01	8.008e+02 ≈	3.045e-01	8.008e+02	3.043e-01
f_5	9.013e+02 ≈	9.282e-01	9.013e+02 ≈	8.558e-01	9.023e+02	1.009e+00
f_6	5.757e+04 ≈	2.503e+04	5.062e+04 ≈	2.184e+04	5.170e+04	2.244e+04
f_7	2.051e+03 ≈	2.052e+01	2.044e+03 ≈	9.643e+00	2.059e+03	4.540e+01
f_8	2.237e+03 ≈	8.686e+00	2.238e+03 ≈	1.272e+01	2.261e+03	7.610e+01
f_9	2.636e+03 ≈	1.062e-01	2.636e+03 ≈	3.531e-01	2.636e+03	1.755e-01
f_{10}	3.928e+03 ≈	9.808e+02	3.696e+03 ≈	1.113e+03	4.572e+03	1.192e+03
f_{11}	2.602e+03 ≈	4.574e+00	2.603e+03 ≈	6.775e+00	2.607e+03	8.628e+00
f_{12}	2.984e+03 ≈	1.995e+01	2.979e+03 ≈	1.511e+01	3.006e+03	4.961e+01
+/-/-	0/12/0		0/12/0		–	

References

- [1] R. Zhong, E. Zhang, M. Munetomo, Cooperative coevolutionary surrogate ensemble-assisted differential evolution with efficient dual differential grouping for large-scale expensive optimization problems, *Complex Intell. Syst.* 10 (2023) 2129–2149.
- [2] Y. Zhang, S. Li, Y. Wang, Y. Yan, J. Zhao, Z. Gao, Self-adaptive enhanced learning differential evolution with surprisingly efficient decomposition approach for parameter identification of photovoltaic models, *Energy Convers. Manage.* 308 (2024) 118387.
- [3] A.P. Piotrowski, J.J. Napiorkowski, A.E. Piotrowska, Particle swarm optimization or differential evolution—A comparison, *Eng. Appl. Artif. Intell.* 121 (2023) 106008.
- [4] Y.-J. Zhang, Y.-F. Wang, Y.-X. Yan, J. Zhao, Z.-M. Gao, LMRAOA: An improved arithmetic optimization algorithm with multi-leader and high-speed jumping based on opposition-based learning solving engineering and numerical problems, *Alex. Eng. J.* 61 (2022) 12367–12403.
- [5] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61.
- [6] B. Abdollahzadeh, F. Soleimani Gharehchopogh, S. Mirjalili, Artificial Gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems, *Int. J. Intell. Syst.* 36 (10) (2021) 5887–5958.
- [7] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comput. Syst.* 111 (2020) 300–323.
- [8] M. Azizi, S. Talatahari, A.H. Gandomi, Fire Hawk optimizer: A novel metaheuristic algorithm, *Artif. Intell. Rev.* 56 (1) (2023) 287–363.
- [9] H. Jia, H. Rao, C. Wen, S. Mirjalili, Crayfish optimization algorithm, *Artif. Intell. Rev.* 1 (2023) 1.
- [10] J.-K. Xue, B. Shen, Dung beetle optimizer: a new meta-heuristic algorithm for global optimization, *J. Supercomput.* 79 (2023) 7305–7336.
- [11] M. Han, Z. Du, K.F. Yuen, H. Zhu, Y. Li, Q. Yuan, Walrus optimizer: A novel nature-inspired metaheuristic algorithm, *Expert Syst. Appl.* 239 (2024) 122413.
- [12] H. Su, D. Zhao, A.A. Heidari, L. Liu, X. Zhang, M. Mafarja, H. Chen, RIME: A physics-based optimization, *Neurocomputing* 532 (2023) 183–214.
- [13] L. Deng, S. Liu, Snow ablation optimizer: A novel metaheuristic technique for numerical optimization and engineering design, *Expert Syst. Appl.* 225 (2023) 120069.
- [14] F.A. Hashim, R.R. Mostafa, A.G. Hussien, S. Mirjalili, K.M. Sallam, Fick's law algorithm: A physical law-based algorithm for numerical optimization, *Knowl.-Based Syst.* 260 (2023) 110146.
- [15] Y. Xu, R. Zhong, C. Zhang, J. Yu, Multiplayer battle game-inspired optimizer for complex optimization problems, *Cluster Comput.* (2024) 1–25.
- [16] C. Aranha, C. Villalón, F. Campelo, M. Dorigo, R. Ruiz, M. Sevaux, K. Sörensen, T. Stützle, Metaphor-based metaheuristics, a call for action: the elephant in the room, *Swarm Intell.* 16 (2021) 1–6.
- [17] C. Camacho, M. Dorigo, T. Stützle, 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.* 13 (2019).
- [18] C.L. Camacho Villalón, T. Stützle, M. Dorigo, Grey wolf, firefly and bat algorithms: Three widespread algorithms that do not contain any novelty, in: *Swarm Intelligence*, Springer International Publishing, Cham, 2020, pp. 121–133.
- [19] C.L. Camacho Villalón, M. Dorigo, T. Stützle, Exposing the grey wolf, moth-flame, whale, firefly, bat, and antlion algorithms: six misleading optimization techniques inspired by bestial metaphors, *Int. Trans. Oper. Res.* 30 (6) (2023) 2945–2971.
- [20] R. Zhong, E. Zhang, M. Munetomo, Evolutionary multi-mode slime mold optimization: a hyper-heuristic algorithm inspired by slime mold foraging behaviors, *J. Supercomput.* (2024).
- [21] C. Huang, X. Zhou, X. Ran, J. Wang, H. Chen, W. Deng, Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning, *Eng. Appl. Artif. Intell.* 121 (2023) 105942.
- [22] Q. Gu, S. Li, Z. Liao, Solving nonlinear equation systems based on evolutionary multitasking with neighborhood-based speciation differential evolution, *Expert Syst. Appl.* 238 (2024) 122025.
- [23] Y. Wang, Z. Wang, G.-G. Wang, Hierarchical learning particle swarm optimization using fuzzy logic, *Expert Syst. Appl.* 232 (2023) 120759.
- [24] Y. Zhang, Elite archives-driven particle swarm optimization for large scale numerical optimization and its engineering applications, *Swarm Evol. Comput.* 76 (2023) 101212.
- [25] A. Daliri, M. Alimoradi, M. Zabihimayvan, R. Sadeghi, World hyper-heuristic: A novel reinforcement learning approach for dynamic exploration and exploitation, *Expert Syst. Appl.* 244 (2024) 122931.
- [26] H.-B. Song, Y.-H. Yang, J. Lin, J.-X. Ye, An effective hyper heuristic-based memetic algorithm for the distributed assembly permutation flow-shop scheduling problem, *Appl. Soft Comput.* 135 (2023) 110022.

- [27] Z.-Q. Zhang, F.-C. Wu, B. Qian, R. Hu, L. Wang, H.-P. Jin, A Q-learning-based hyper-heuristic evolutionary algorithm for the distributed flexible job-shop scheduling problem with crane transportation, *Expert Syst. Appl.* 234 (2023) 121050.
- [28] J.M. Cruz-Duarte, I. Amaya, J.C. Ortiz-Bayliss, S.E. Conant-Pablos, H. Terashima-Marin, A primary study on hyper-heuristics to customise metaheuristics for continuous optimisation, in: 2020 IEEE Congress on Evolutionary Computation, CEC, 2020, pp. 1–8.
- [29] J.M. Tapia-Avitia, J.M. Cruz-Duarte, I. Amaya, J.C. Ortiz-Bayliss, H. Terashima-Marin, N. Pillay, 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, 2022, pp. 1–8.
- [30] D. Kelly, Y. Chen, S.E. Cornwell, N.S. Delellis, A. Mayhew, S. Onaoalapo, V.L. Rubin, Bing chat: The future of search engines? *Proc. Assoc. Inform. Sci. Technol.* 60 (1) (2023) 1007–1009.
- [31] C. Wu, W. Lin, X. Zhang, Y. Zhang, W. Xie, Y. Wang, PMC-LLaMA: toward building open-source language models for medicine, *J. Am. Med. Inform. Assoc.* (2024) ocae045.
- [32] S. Biswas, Role of chat GPT in public health, *Ann. Biomed. Eng.* 51 (2023) 3.
- [33] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A.M. Dai, A. Hauth, et al., Gemini: A family of highly capable multimodal models, 2023, [arXiv:2312.11805](https://arxiv.org/abs/2312.11805).
- [34] B. Paranjape, S. Lundberg, S. Singh, H. Hajishirzi, L. Zettlemoyer, M.T. Ribeiro, ART: Automatic multi-step reasoning and tool-use for large language models, 2023, [arXiv:2303.09014](https://arxiv.org/abs/2303.09014).
- [35] A.J. Thirunavukarasu, D.S.J. Ting, K. Elangovan, L. Gutierrez, T.F. Tan, D.S.W. Ting, Large language models in medicine, *Nat. Med.* 29 (8) (2023) 1930–1940.
- [36] S. Harrer, Attention is not all you need: the complicated case of ethically using large language models in healthcare and medicine, *EBioMedicine* 90 (2023).
- [37] S.S. Choong, L.-P. Wong, C.P. Lim, Automatic design of hyper-heuristic based on reinforcement learning, *Inform. Sci.* 436–437 (2018) 89–107.
- [38] R. Zhong, J. Yu, A novel evolutionary status guided hyper-heuristic algorithm for continuous optimization, *Cluster Comput.* (2024) 1–30.
- [39] M. Pluhacek, A. Kazikova, T. Kadavy, A. Viktorin, R. Senkerik, Leveraging large language models for the generation of novel metaheuristic optimization algorithms, in: Proceedings of the Companion Conference on Genetic and Evolutionary Computation, in: GECCO '23 Companion, Association for Computing Machinery, New York, NY, USA, 2023, pp. 1812–1820.
- [40] R. Zhong, Y. Xu, C. Zhang, J. Yu, Leveraging large language model to generate a novel metaheuristic algorithm with CRISPE framework, *Cluster Comput.* 27 (2024) 13835–13869.
- [41] C. Yang, X. Wang, Y. Lu, H. Liu, Q.V. Le, D. Zhou, X. Chen, Large language models as optimizers, 2023, [arXiv:2309.03409](https://arxiv.org/abs/2309.03409).
- [42] S. Liu, C. Chen, X. Qu, K. Tang, Y.-S. Ong, Large language models as evolutionary optimizers, 2023, [arXiv:2310.19046](https://arxiv.org/abs/2310.19046).
- [43] X. Wu, S. hao Wu, J. Wu, L. Feng, K.C. Tan, Evolutionary computation in the era of large language model: Survey and roadmap, 2024, [arXiv:2401.10034](https://arxiv.org/abs/2401.10034).
- [44] C. Tu, R. Bai, U. Aickelin, Y. Zhang, H. Du, A deep reinforcement learning hyper-heuristic with feature fusion for online packing problems, *Expert Syst. Appl.* 230 (2023) 120568.
- [45] R. Zhong, J. Yu, DEA2h2: Differential evolution architecture based adaptive hyper-heuristic algorithm for continuous optimization, *Cluster Comput.* (2024) 1–28.
- [46] F. Zhao, Y. Liu, N. Zhu, T. Xu, Jonrinaldi, A selection hyper-heuristic algorithm with Q-learning mechanism, *Appl. Soft Comput.* 147 (2023) 110815.
- [47] R. Tyasurita, E. Özcan, J.H. Drake, S. Asta, Constructing selection hyper-heuristics for open vehicle routing with time delay neural networks using multiple experts, *Knowl.-Based Syst.* 295 (2024) 111731.
- [48] T. Dokeroglu, T. Kucukyilmaz, E.-G. Talbi, Hyper-heuristics: A survey and taxonomy, *Comput. Ind. Eng.* 187 (2024) 109815.
- [49] J.A. McCall, L.A. Christie, A.E. Brownlee, Generating easy and hard problems using the proximate optimality principle, in: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, in: GECCO Companion '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 767–768.
- [50] A. Qin, P. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: 2005 IEEE Congress on Evolutionary Computation, vol. 2, 2005, pp. 1785–1791.
- [51] L.M. Schmitt, Theory of genetic algorithms, *Theoret. Comput. Sci.* 259 (1) (2001) 1–61.
- [52] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.
- [53] R. Storn, On the usage of differential evolution for function optimization, in: Proceedings of North American Fuzzy Information Processing, 1996, pp. 519–523.
- [54] T. Nguyen, A framework of Optimization Functions using Numpy (OpFuNu) for optimization problems, Zenodo, 2020.
- [55] R. Zhong, J. Yu, C. Zhang, M. Munetomo, SRIME: a strengthened RIME with latin hypercube sampling and embedded distance-based selection for engineering optimization problems, *Neural Comput. Appl.* 36 (2024) 6721–6740.
- [56] N.V. Thieu, ENOPPY: A Python Library for Engineering Optimization Problems, Zenodo, 2023.
- [57] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, *Evolut. Comput.* 9 (2) (2001) 159–195.
- [58] J. Liang, A. Qin, P. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Trans. Evol. Comput.* 10 (3) (2006) 281–295.
- [59] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 71–78.
- [60] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation, CEC, 2014, pp. 1658–1665.
- [61] M. Ghasemi, J. Aghaei, M. Hadipour, New self-organising hierarchical PSO with jumping time-varying acceleration coefficients, *Electron. Lett.* 53 (20) (2017) 1360–1362.
- [62] M. Ghasemi, E. Akbari, A. Rahimnejad, E. Razavi, S. Ghavidel, L. Li, Phasor particle swarm optimization: a simple and efficient variant of PSO, *Soft Comput.* 23 (2019).
- [63] P. Çivicioglu, E. Beşdok, M.A. Günen, U.H. Atasever, Weighted differential evolution algorithm for numerical function optimization: a comparative study with Cuckoo search, artificial bee colony, adaptive differential evolution, and backtracking search optimization algorithms, *Neural Comput. Appl.* 32 (2020) 3923–3937.
- [64] M.H. Nadimi-Shahroki, H. Zamani, DMDE: Diversity-maintained multi-trial vector differential evolution algorithm for non-decomposition large-scale global optimization, *Expert Syst. Appl.* 198 (2022) 116895.
- [65] C.A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Comput. Methods Appl. Mech. Engrg.* 191 (11) (2002) 1245–1287.
- [66] K. Li, H. Huang, S. Fu, C. Ma, Q. Fan, Y. Zhu, A multi-strategy enhanced northern Goshawk optimization algorithm for global optimization and engineering design problems, *Comput. Methods Appl. Mech. Engrg.* 415 (2023) 116199.
- [67] D. Wolpert, W. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1 (1) (1997) 67–82.
- [68] W.G. Jackson, E. Özcan, J.H. Drake, Late acceptance-based selection hyper-heuristics for cross-domain heuristic search, in: 2013 13th UK Workshop on Computational Intelligence, UKCI, 2013, pp. 228–235.
- [69] R. Zhong, J. Yu, C. Zhang, M. Munetomo, Surrogate ensemble-assisted hyper-heuristic algorithm for expensive optimization problems, *Int. J. Comput. Intell. Syst.* 16 (2023) 169.
- [70] P. Cowling, G. Kendall, E. Soubeiga, Hyperheuristics: A tool for rapid prototyping in scheduling and optimisation, in: Applications of Evolutionary Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 1–10.
- [71] L. Zou, S. Zhang, H. Cai, D. Ma, S. Cheng, S. Wang, D. Shi, Z. Cheng, D. Yin, Pre-trained language model based ranking in Baidu search, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 4014–4022.
- [72] H. Bayzidi, S. Talatahari, M. Saraei, C.-P. Lamarche, Social network search for solving engineering optimization problems, *Comput. Intell. Neurosci.* 2021 (2021) 1–32.