Cluster Computing
https://doi.org/10.1007/s10586-024-04593-2

A novel evolutionary status guided hyper-heuristic algorithm for continuous optimization

Rui Zhong[1] · Jun Yu[2]

Received: 18 March 2024 / Revised: 19 May 2024 / Accepted: 23 May 2024

Abstract
This paper proposes a novel evolutionary status guided hyper-heuristic algorithm named ES-HHA for continuous optimization. A representative hyper-heuristic algorithm consists of two components: the low-level component and the high-level component. In the low-level component, to balance the exploitation and exploration during optimization, we design an exploitative operator pool and an explorative operator pool as low-level heuristics (LLHs), where the former is constructed using local search based operators, and the latter consists of various mutation operators from differential evolution (DE). In the high-level component, we design a probabilistic selection function based on the fitness distance correlation (FDC) and the population diversity (PD). Since these two metrics can reflect the complexity of the fitness landscape and the status of the evolutionary swarm, the integration of these two metrics is expected to determine the sequence of heuristics automatically and intelligently. To evaluate the performance of our proposal, we implement comprehensive numerical experiments on CEC2014, CEC2022, and eight engineering optimization tasks. A total of 14 famous optimization approaches are adopted as competitors. Furthermore, the ablation experiment is conducted to evaluate the high-level component independently, while the sensitivity experiment contributes to determining the optimal hyper-parameter setting. The experimental results and statistical analysis show that ES-HHA is competitive, and the evolutionary status guided probabilistic selection function can determine the optimization intelligently.

Keywords Hyper-heuristic algorithm · Evolutionary status · Low-level heuristics (LLHs) · Fitness distance correlation (FDC) · Population diversity (PD)

1 Introduction

Hyper-heuristics is not a novel terminology, which can be traced back to the early 1960s [1] as a probabilistic learning approach and first debut in [2] to describe the principle of "heuristics to choose heuristics". As an "off-the-peg" technique rather than "made-to-measure" meta-heuristics [3], the hyper-heuristic algorithm is a kind of high-level automatic optimization technique that manipulates a set of low-level heuristics (LLHs) to construct the optimal sequence of heuristics by domain knowledge, whereas most of the meta-heuristics are directly implemented to the search space of the problem.

A representative architecture of the hyper-heuristic algorithm demonstrated in Fig. 1 consists of two dependent components: the high-level component and the low-level component. The high-level component plays a decision-maker role in constructing the optimization sequence, and the low-level component contains a set of LLHs, problem representation, the objective function(s), and initial solutions.

The hyper-heuristics community has achieved unprecedented development since 2000. Bai et al. [6] formulated the shelf space allocation problem as a variant of the NP-hard multi-knapsack problem and proposed a simulated annealing-based hyper-heuristic algorithm to solve several instances with various scales and space ratios. Koulinas

✉ Jun Yu
yujun@ie.niigata-u.ac.jp

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

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

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

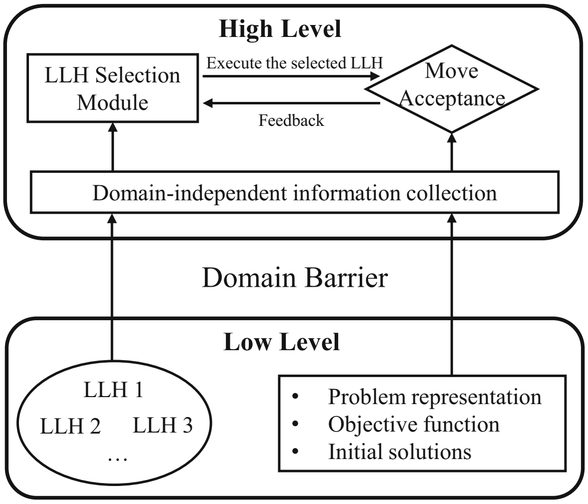


Fig. 1 A representative architecture of the hyper-heuristic algorithm [4, 5]

et al. [7] designed a particle swarm optimization-based hyper-heuristic algorithm (PSO-HH) for the classic resource-constrained project scheduling problem. Each particle is an eight-dimensional vector, which represents eight corresponding LLHs. As an online learning method, PSO-HH consumed an extra computational budget to construct the optimal optimization sequence. Tsai et. al [8] adopted the hyper-heuristic algorithm to the cloud computing systems for scheduling. Diversity detection and improvement detection as operators were employed to dynamically determine the potential LLHs that can construct better solution candidates. Zhang et al. [9] proposed a deep reinforcement learning-based hyper-heuristics framework with a data-driven heuristic selection module to deal with combinatorial optimization problems with uncertainty. Double deep Q-learning which can offer excellent performance and training stability was employed as the selection module to dominate the LLHs, and ten easy-implemented search operators partially inspired by the manual heuristics in [10] were employed as the LLHs. In multi-objective optimization, hyper-heuristics also make a critical difference: Burke et al. [11] designed a multi-objective hyper-heuristic algorithm based on the tabu search technique and this approach showed the competitiveness in the space allocation and timetabling problems. Kumari et al. [12] presented a fast multi-objective hyper-heuristic genetic algorithm (MHypGA) for solving the multi-objective software module clustering problem, twelve basic search operators which were the variants of the selection, recombination, and mutation in GA were employed as the LLHs, equal probabilistic selection function determined the optimization sequence like the high-level component. Mahmud et al. [13] focused on solving integrated supply chain scheduling problems and proposed a self-adaptive multi-operator and multi-objective hyper-heuristic (SA(MO)$_2$H), where the LLHs utilized the advantage of four solution updating heuristics and the construction of optimization sequence was intelligently guided by the reinforcement learning technique. Cao et al. [14] extended the basic skeleton of multi-objective simulated annealing based on re-seed (MOSA/R) to structural damage identification. Four re-seed strategies were introduced as LLHs, reinforcement learning hyper-heuristic strategy including heuristics selection and credit assignment was adopted to select between different heuristics autonomously. Zhong et al. [15] regarded surrogate-assisted estimation as a kind of search operator and embedded this technique into the LLHs of the hyper-heuristic framework, while the learning-free probabilistic selection function was employed as the high-level component. In the meantime, most hyper-heuristics approaches were proposed for combinatorial problems while only a few have dealt with continuous optimization scenarios [16, 17]. Therefore, developing efficient hyper-heuristic algorithms with high performance for continuous optimization becomes an emerging research topic.

This paper proposes a novel evolutionary status guided hyper-heuristic algorithm (ES-HHA) for continuous optimization. In the low-level component, we design an exploitative operator pool and an explorative operator pool, each pool contains four efficient and easy-implemented search operators. In the high-level component, the fitness distance correlation (FDC) [18] and the population diversity (PD) [19] are two crucial metrics to reflect the complexity of the optimization problem and the status of the evolutionary swarm. Therefore, we design a probabilistic selection function based on these two performance indicators. Comprehensive numerical experiments are conducted to evaluate the performance of our proposed ES-HHA thoroughly. More specifically, the contributions of this paper are as follows:

(1) Under the background that most hyper-heuristic approaches focus on discrete space, we propose a novel evolutionary status guided hyper-heuristic algorithm (ES-HHA) to solve continuous optimization problems.
(2) In the low-level component of ES-HHA, we design an exploitative operator pool that consists of four local search based operators, and an explorative operator pool that consists of four mutation operators from DE. The cooperation between two operator pools is expected to balance the exploitation and exploration well during optimization.
(3) In the high-level component of ES-HHA, we design a probabilistic selection function based on FDC and PD, since these metrics can reflect the status of

optimization and may contribute to determining the optimization sequence with knowledge from the fitness landscape and swarm.

The remainder of this paper is organized as follows: Sect. 2 introduced the related works including fitness distance correlation and population diversity. Section 3 provides a detailed introduction to our proposal: ES-HHA. Section 4 covers numerical experiments and statistical results of the optimization. Section 5 discusses the performance of ES-HHA and lists some open topics for future research. Finally, Sect. 6 concludes this paper.

2 Related works

2.1 Fitness distance correlation

Fitness distance correlation (FDC) was first proposed in [18] as a metric to evaluate the difficulty of optimization problems. Supposing the global optimum is known, FDC calculates the correlation between sample-optimum distance and fitness to further describe the characteristics of the fitness landscape. A dataset of the sample $\mathscr{X} = \{X_1, X_2, \ldots, X_n\}$ and the corresponding fitness value are $\mathscr{F} = \{f_1, f_2, \ldots, f_n\}$, the Euclidean distance $\mathscr{D} = \{d_1, d_2, \ldots, d_n\}$ between samples and the global optimum can be calculated by Eq. (1).

$$d_i = \sqrt{\sum_{j=1}^{D} (X_i^j - X_{best}^j)^2} \tag{1}$$

where X_{best} is the location of the global optimum. Then, the computation of FDC is expressed by Eq. (2).

$$FDC = \frac{\sum_{i=1}^{n} (f_i - \bar{f})(d_i - \bar{d})}{\sqrt{\sum_{i=1}^{n} (f_i - \bar{f})^2} \sqrt{\sum_{i=1}^{n} (d_i - \bar{d})^2}} \tag{2}$$

$\bar{f}$ and $\bar{d}$ are the arithmetic mean of $\mathscr{F}$ and $\mathscr{D}$ respectively. Simply, FDC is an extension of the Pearson Correlation Coefficient in optimization theory, and the value range of FDC is $[-1, 1]$.

The investigation [20] suggested that the problems can be roughly classified into three categories based on the FDC given the known global optimum. Here, we take the minimization problem as an example. A strong negative correlation means the solution that is further from the optimum has better fitness, and the problem in this case has deceptive characteristics. A strong positive correlation implies the solution that is closer to the optimum has better fitness, and this kind of problem has a simple fitness landscape. A weak correlation indicates the problem has a complex fitness landscape, which means the problem is difficult to solve.

However, the global optimum is usually unknown in practice, and a common approach is to adopt the current best solution in $\mathscr{X}$ to approximately represent the global optimum. In this case, the deceptive feature of the problem cannot be inferred by the FDC because the global optimum is unknown, and the deceptiveness is no longer considered practically [21].

Here, we demonstrate the calculation of FDC on 2-D space with eight common benchmark functions, and the information is summarized in Table 1. The datasets for computation are randomly generated with a scale of 10000. Figure 2 draws the contour figures and fitness-distance relationship figures of corresponding benchmark functions.

The contour and fitness-distance relationship figures in Fig. 2 are consistent with the definition of FDC that a strong positive correlation means the problem is simple and vice versa. Therefore, the motivation of this research is to utilize FDC as one of the guides to navigate the selection of exploitative or explorative search preferences during optimization.

Furthermore, there have been many works on how FDC assists evolutionary algorithms: Li et al. [22] adopted a simplified FDC as the basis to select the mutation strategy for differential evolution (DE). More concretely, in a certain population with n individuals, the distances between each individual and optimum are sorted, and then, the fitness value of each individual and its latter individual is compared in turn. Finally, the simplified FDC can be calculated by $r = \frac{\theta}{(n-1)}$, where θ is the number of individuals inferior to the latter. Tan et al. [23] introduced the FDC and ruggedness of information entropy to DE to describe the characteristics of the fitness landscape, and the random forest was trained to establish the relationship between these two metrics and mutation operators. Luo et al. [24], Xu et al. [21], and Qiao et al. [20] extended the FDC to solve large-scale optimization problems based on cooperative co-evolution. FDC was employed to estimate the difficulty of each sub-component, and a more complex sub-component was allocated more computational resources. Therefore, FDC is reasonable and feasible to reflect the difficulty and complexity of the fitness landscape, and the integration of FDC to hyper-heuristic algorithms may contribute to constructing the optimization sequence automatically and intelligently.

2.2 Population diversity

Population diversity (PD) is a metric to evaluate the distribution of the population or swarm. Although different formulations of PD exist in many works [19, 25–27], the kernel of PD is to describe how far each individual lies

Table 1 Eight common benchmark functions with corresponding characteristics and FDC

Function	Formulation	Search space	FDC value
Schwefel	$f(x) = 418.9829D - \sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x_i \rvert})$	[−500, 500]	0.307
Styblinski–Tang	$f(x) = \frac{\sum_{i=1}^{D} x_i^4 - 16x_i^2 + 5x_i}{2}$	[−5, 5]	0.435
Rosenbrock	$f(x) = \sum_{i=1}^{D-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[−2.048, 2.048]	0.545
Rotated hyper ellipsoid	$f(x) = \sum_{i=1}^{D} \sum_{j=1}^{i} x_j^2$	[−65.536, 65.536]	0.685
Rastrigin	$f(x) = 10D + \sum_{i=1}^{D} x_i^2 - 10\cos(2\pi x_i)$	[−5.12, 5.12]	0.705
Ackley	$f(x) = 20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	[−32, 32]	0.776
Griewank	$f(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1$	[−600, 600]	0.973
Sphere	$f(x) = \sum_{i=1}^{D} x^2$	[−10, 10]	0.983

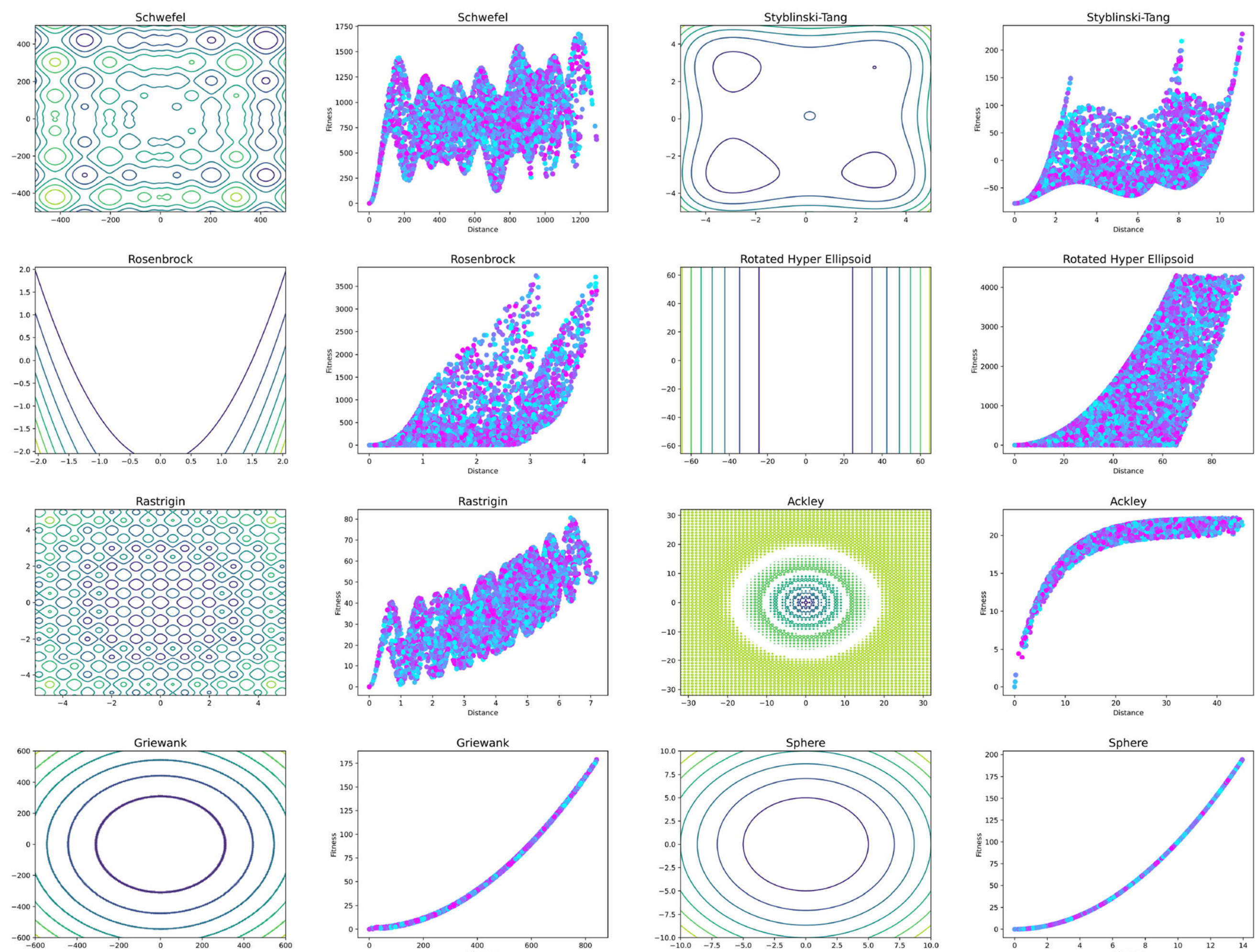


Fig. 2 The contour and fitness-distance relationship figures of benchmark functions in Table 1

from the mean of the population as similar to the definition of standard deviation.

$$PD = \frac{1}{n \times D} \sum_{i=1}^{n} \sum_{j=1}^{D} \frac{|X_{i,j} - \bar{X}_j|}{UB_j - LB_j} \tag{3}$$

where $\bar{X}$ represents the geometric centroid of the population. UB_j and LB_j are the upper and the lower bound of the search space in j^{th} dimension respectively, and the range of the PD is from 0 to 1.

A simple imagination of PD in the optimization process is that, in the initial period of optimization, the exploration behaviors are encouraged, and the PD should be maintained at a certain level to avoid optimization premature. In the late period of optimization, the individuals gather around and PD may be at a low level for better convergence. This idea follows the intuition of the optimization process, and actually, many approaches utilize PD as we explained: Poláková et al. [28] proposed a dynamic population size mechanism based on PD during the optimization process. If the current PD is larger than the threshold $r1$ or less than the threshold $r2$, this strategy suggests removing the worst individual or adding a random individual respectively. Yu et al. [29] used the PD metric to control the scaling factor of mutation strategies in DE automatically. If the optimization sticks into stagnation and the PD is low, the unique backward search is applied to jump out of the local optimum. Li et al. [30] designed a modified PD computation and embedded this metric in the population state evaluation (PSE) framework. When the population has a smaller value of modified PD than the adaptive threshold, then the dispersion operation is activated to enhance the diversity of the population.

3 Our proposal: ES-HHA

In this section, we introduce our proposed ES-HHA in detail. Section 3.1 provides the overview of ES-HHA, Sect. 3.2 and 3.3 introduce the low-level and the high-level component design of ES-HHA, respectively, and Sect. 3.4 introduces the selection mechanism of ES-HHA.

3.1 The overview of ES-HHA

Figure 3 shows the architecture of the proposed ES-HHA. The novel evolutionary status guided probabilistic selection function is designed to dominate the LLHs and construct the optimization sequence. An exploitative operator pool and an explorative operator pool contain various search operators representing distinct preferences. The determined search operators are activated to update the solution swarm, while the evolutionary status feedback obtained from the distribution of the solution swarm adjusts the weight between exploitation and exploration adaptively. Next, we will introduce the operator pool and the evolutionary status guided probabilistic selection function in the proposed ES-HHA.

3.2 The operator pool of ES-HHA

The design of LLHs is the most important in the low-level component of ES-HHA. However, innumerable search operators have been proposed and reported, and how to select the representative search operators to form the exploitation or exploration pool is an unsolved issue. The original intention of the hyper-heuristics algorithm is to develop a generic optimization methodology with easy-implemented LLHs [31]. Thus, we follow this opinion and adopt simple search operators, which can be derived from two categories: the exploitative operator and the explorative operator.

3.2.1 Exploitative operator pool

Local search is an efficient and simple exploitative search operator, which has been widely adopted as a heuristic approach to solve computational optimization problems [32]. In discrete space, local search is usually expressed as moving from solution to solution in the space of candidate solutions by applying local changes. In continuous search space, the concept of neighbor is infinite, and in many works of literature, Eq. (4) is adopted to describe this operation [33, 34].

$$X_i^{t+1} = X_i^t + \delta \tag{4}$$

where δ is a perturbation with an identical dimension to the i^{th} individual X_i, superscripts t and $t+1$ represent the iteration. Inspired by the proximate optimality principle (POP) [35], better solutions share similar structures, and the exploitative search around promising solutions may construct better offspring individuals. Therefore, we replace the X_i^t in Eq. (4) with the X_{best}^t, and this local search operator is only activated around the current best individual, which is regarded as an exploitative operation.

Another problem is how to design the perturbation δ. Here, we define four kinds of δ perturbation techniques: *Uniform*, *Normal*, *Lévy flight*, and *Differential*. The fusion of diverse perturbation generators can strengthen the robustness and accelerate the optimization convergence of ES-HHA. Furthermore, if one perturbation generator performs poorly, other operators can serve as compensation and prevent stagnation during optimization. Additionally, each strategy in the exploitation pool has the same probability of being selected.

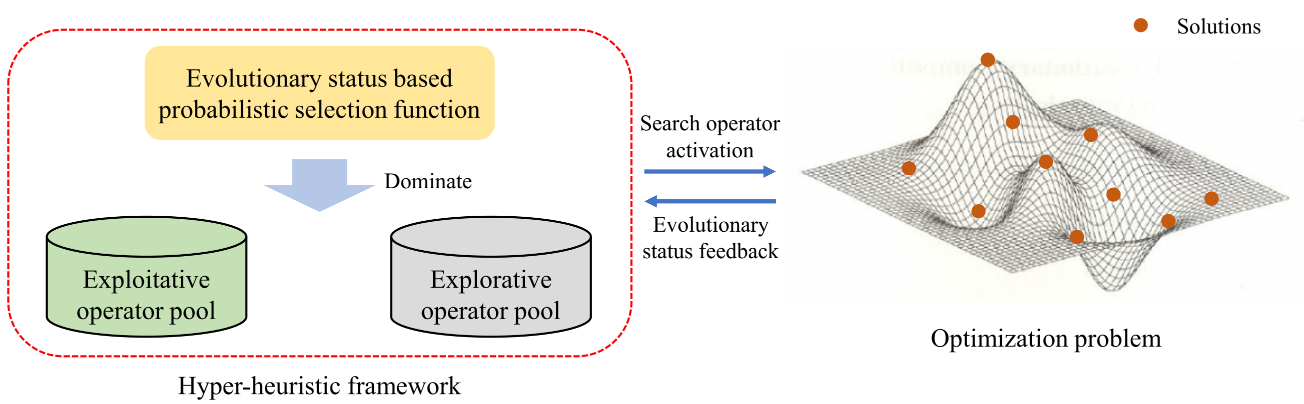


Fig. 3 The architecture of the proposed ES-HHA

Uniform: In the Uniform principle, δ_j is randomly sampled from a uniform distribution.

$$\delta_j \sim R \cdot U(-1, 1) \tag{5}$$

where $U(-1, 1)$ represents a random value sampled from a uniform distribution from $-1 to 1$, and R is a parameter to control the search radius.

Normal: Similar to the Uniform principle, δ determined by the Normal principle is randomly sampled from a standard normal distribution.

$$\delta_j \sim R \cdot N(0, 1) \tag{6}$$

Lévy flight: Lévy flight can be applied to describe the foraging behaviors of predators [36]: When sharks or other ocean predators cannot find food, they abandon Brownian motion, the random motion seen in swirling gas molecules, for Lévy flight, a mix of long trajectories and short, random movements found in turbulent fluids [37]. Equation (7) formulates the Lévy flight with the mathematical model.

$$\delta_j \sim \frac{u}{|v|^{\frac{1}{\beta}}} \tag{7}$$

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

$$u \sim N(0, \sigma^2), u \sim N(0, 1) \tag{8}$$

and the standard deviation σ is defined by

$$\sigma = \left\{ \frac{\Gamma(1+\beta) \sin\left(\frac{\pi\beta}{2}\right)}{\beta\Gamma\left(\frac{1+\beta}{2}\right) 2^{\frac{\beta-1}{2}}} \right\}^{\frac{1}{\beta}} \tag{9}$$

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

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt. \tag{10}$$

Differential: We adopt the DE/best/1 mutation operator as one of the exploitation operators in this study, which is expressed in Eq. (11).

$$\delta = F \cdot (X_{r1}^t - X_{r2}^t) \tag{11}$$

where $F = 0.8$ is the scaling factor. X_{r1}^t and X_{r2}^t are two different solutions randomly sampled from the population in the tth iteration.

3.2.2 Explorative operator pool

DE is one of the most well-studied stochastic optimization techniques, and the mutation operator of DE has been widely applied to realize the explorative search, such as simulating the foraging behaviors of animals [38, 39], physics and chemistry laws [40–42], and natural phenomena [43, 44]. The essence of this operator is to compute a scaled differential vector between sampled individuals and add this to the base vector. In effect, the spread of mutants in the search space depends on the actual spread of the population. Hence, the differential mutation acts as an implicit adaptation mechanism of the search range and direction [45]. In this research, we design an explorative operator pool involving four representative mutation operators from DE. Once the explorative operator pool is activated, each involved DE mutation operator has an equal probability of being adopted.

DE/rand/1: Eq. (12) shows the formulation of the DE/rand/1 operator.

$$X_i^{t+1} = X_{r1}^t + F \cdot (X_{r2}^t - X_{r3}^t) \tag{12}$$

where X_{r1}^t, X_{r2}^t, and X_{r3}^t are mutually different and randomly sampled from the t^{th} population.

DE/cur/1: The DE/cur/1 operator replaces the base vector in Eq. (12) with the current solution X_i^t, which is expressed in Eq. (13).

$$X_i^{t+1} = X_i^t + F \cdot (X_{r1}^t - X_{r2}^t) \quad (13)$$

DE/cur-to-best/1: Equation (14) defines the formulation of the DE/cur-to-best/1 operator.

$$X_i^{t+1} = X_i^t + F \cdot (X_{best}^t - X_i^t) - F \cdot (X_{r_1}^t - X_{r_2}^t) \quad (14)$$

where X_{best}^t is the optimum found so far.

DE/cur-to-pbest/1: The only difference between the DE/cur-to-best/1 operator and the DE/cur-to-pbest/1 is in the second term that X_{best}^t is replaced by X_{pbest}^t, as presented in Eq. (15).

$$X_i^{t+1} = X_i^t + F \cdot (X_{pbest}^t - X_i^t) - F \cdot (X_{r_1}^t - X_{r_2}^t) \quad (15)$$

where X_{pbest}^t represents the centroid of the best-p solutions.

3.3 The high-level component of ES-HHA

This section introduces the design of the evolutionary status guided probabilistic selection function. As we mentioned before, the performance indicators of FDC and PD can reflect the complexity of the problem and the distribution of the population. Thus, we believe that the cooperation of these two metrics can contribute to determining the optimization sequence automatically and intelligently. Here, we separate the level of these two metrics into four cases, which are shown in Fig. 4

Figure 4(a) shows the case that the FDC is high and PD is high. We can infer that the fitness landscape around the samples is simple and the optimization does not approach convergence, thus the exploitation behaviors are encouraged to search for a better solution and accelerate the convergence in this case.

Figure 4 (b) shows the case that the FDC is high and PD is low, and in this case, the fitness landscape around the samples is simple and optimization approaches convergence, individuals have a high probability to gather around on an optimum. However, we do not know whether individuals cluster in the global optimum or trap in the local optimum, and the balanced strategy between exploration and exploitation should be encouraged in this case.

Figure 4(c) shows the case that the FDC is low and PD is high. Similar to the case in Fig. 4(b) that we cannot simply determine which strategy should dominate in the search process, and our proposal prefers to adopt the balanced search operator in this case.

Figure 4(d) shows the case that the FDC is low and PD is low, and the fitness landscape around the samples is complex while individuals gather around. In this case, the direction of optimization may trap into a local optimum, and the exploration operation should be encouraged to allow the optimization to escape from the local optimum.

From the above explanation, we define the probability of adopting the exploitation operation (P_1) by Eq. (16).

$$P_1 = w_1 \cdot \mathrm{FDC} + (1 - w_1) \cdot \mathrm{PD} \quad (16)$$

where w_1 is a hyper-parameter to control the importance of each component, and the sensitivity experiment is conducted to determine the optimal hyper-parameter setting.

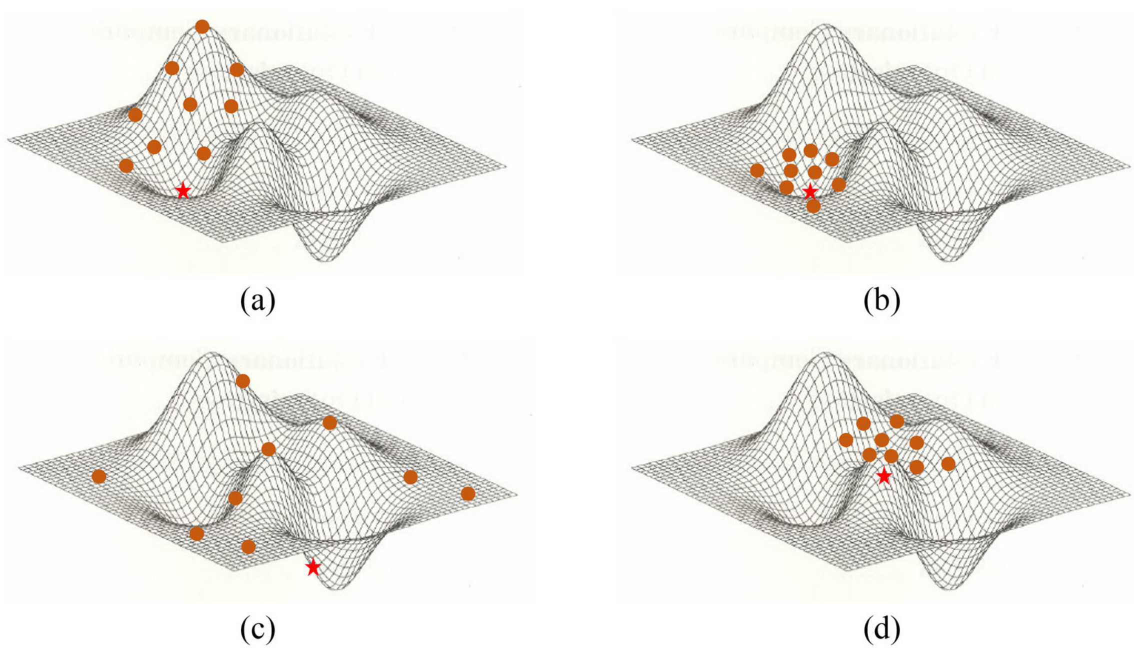


Fig. 4 Four instances of the FDC and the PD. The brown points represent the individuals and the red star point denotes the optimum found so far. (**a**) The FDC is high and the PD is high. (**b**) The FDC is high and the PD is low. (**c**) The FDC is low and the PD is high. (**d**) The FDC is low and the PD is low

Reasonably, the probability of adopting the exploration operation (P_2) is $1 - P_1$. Furthermore, the truncation function is applied to further limit the range of exploitation probability. If P_1 is larger than 0.9 or smaller than 0.1, we manually adjust the P_1 to 0.9 or 0.1, respectively.

3.4 Selection mechanism

We use the one-to-one greedy selection mechanism in our proposed ES-HHA. This strategy is widely adopted in many optimization approaches [46, 47]. Simply, the parent individual will be replaced by the offspring individual while the offspring individual has a better fitness value. Equation (17) describes this selection mechanism.

$$X_i^{t+1} = \begin{cases} X_i^{t+1}, & \textit{if } f(X_i^{t+1}) \textit{ is better} \\ X_i^t, & \textit{otherwise} \end{cases} \tag{17}$$

4 Numerical experiments

In this section, we implement a set of comprehensive experiments to evaluate the performance of our proposal. Section 4.1 introduces the experiment settings: experimental environments, benchmark functions, and compared methods with their parameters. Section 4.2 shows the experimental results.

4.1 Experiment settings

4.1.1 Experimental environments and implementation

All optimization methods are programmed by Python 3.11 and tested in Lenovo Legion R9000P, which is equipped with Windows 11, AMD Ryzen 7 5800 H with Radeon Graphics 3.20 GHz, and 16 GB RAM. All compared meta-heuristics algorithms are provided by the MEALPY library [48]. CEC2014 and CEC2022 benchmark functions are provided by the OpFuNu library [49], and eight engineering optimization problems are provided by the ENOPPY library [50].

4.1.2 Benchmark functions

To evaluate the overall performance of our proposed ES-HHA fairly, we implement optimization experiments on two kinds of optimization problems: (1). 30-D and 50-D CEC2014 benchmark functions and 10-D and 20-D CEC2022 benchmark functions, which are summarized in Table 2 and 3, respectively. (2). Ten real-world engineering optimization problems. Here, we only list the elementary information in Table 4, and the detailed formulation can be found in [51] and https://github.com/RuiZhong961230/ES-HHA.

Given engineering problems contain constraints and original optimization approaches cannot deal with constrained optimization problems, thus, we equip them with the static penalty function [52], which is defined by Eq. (18)

$$F(R_i) = f(R_i) + w \cdot \sum_{i=1}^{m} (\max(0, g_i(R_i))) \tag{18}$$

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

4.1.3 Compared methods and parameters

The first experiment is implemented to evaluate the overall optimization performance of ES-HHA. Here, we adopt four kinds of meta-heuristics as competitors: (1) classic optimizers including genetic algorithm (GA) [53], particle swarm optimization (PSO) [54], differential evolution (DE) [55], evolutionary strategy (ES) [56], and CMA-ES [57]. (2) highly cited optimizers including moth-flame optimization (MFO) [58], whale optimization algorithm (WOA) [59], and sine cosine algorithm (SCA) [60]. (3) newly proposed optimizers including human conception optimizer (HCO) [61], snow ablation optimizer (SAO) [62], and brown-bear optimization algorithm (BOA) [63]. (4) efficient DE-based variants including success-history adaptive DE with linear population size reduction (L-SHADE) [64], chaotic mutation and crossover with constraint factor DE (CCDE) [65], and co-evolutionary multi-swarm adaptive DE (ECMADE) [66]. For the fair comparison, the population size of all algorithms except for L-SHADE is set to 100, maximum fitness evaluations (FEs) for CEC benchmark functions and engineering problems are fixed at 1000*D and 20,000 respectively, 30 independent trial runs are implemented to avoid randomness, and the detailed parameter settings of all competitor algorithms are listed in Table 5.

The second experiment aims to investigate the effectiveness of the high-level component in ES-HHA. Three conventional hyper-heuristic frameworks are employed as competitor algorithms, and the descriptions are

Table 2 Summary of the CEC2014 Suite: Uni. = unimodal function, Multi. = multimodal function, Hybrid. = hybrid function, Comp. = composition function

Fun	Description	Feature	Optimum
f_1	Rotated high conditioned elliptic function	Uni.	100
f_2	Rotated bent cigar function		200
f_3	Rotated discus function		300
f_4	Shifted and rotated Rosenbrock's function	Multi.	400
f_5	Shifted and rotated Ackley's function		500
f_6	Shifted and rotated Weierstrass function		600
f_7	Shifted and rotated Griewank's function		700
f_8	Shifted Rastrigin's function		800
f_9	Shifted and rotated Rastrigin's function		900
f_{10}	Shifted Schwefel's function		1000
f_{11}	Shifted and rotated Schwefel's function		1100
f_{12}	Shifted and rotated Katsuura function		1200
f_{13}	Shifted and rotated HappyCat function		1300
f_{14}	Shifted and rotated HGBat function		1400
f_{15}	Shifted and rotated expanded Griewank's plus Rosenbrock's function		1500
f_{16}	Shifted and rotated expanded Scaffer's F6 function		1600
f_{17}	Hybrid function 1 (n = 3)	Hybrid.	1700
f_{18}	Hybrid function 2 (n = 3)		1800
f_{19}	Hybrid function 3 (n = 4)		1900
f_{20}	Hybrid function 4 (n = 4)		2000
f_{21}	Hybrid function 5 (n = 5)		2100
f_{22}	Hybrid function 6 (n = 5)		2200
f_{23}	Composition function 1 (n = 5)	Comp.	2300
f_{24}	Composition function 2 (n = 3)		2400
f_{25}	Composition function 3 (n = 3)		2500
f_{26}	Composition function 4 (n = 5)		2600
f_{27}	Composition function 5 (n = 5)		2700
f_{28}	Composition function 6 (n = 5)		2800
f_{29}	Composition function 7 (n = 3)		2900
f_{30}	Composition function 8 (n = 3)		3000
Search space: $[-100, 100]^D$			

summarized in Table 6. All compared hyper-heuristic frameworks manipulate the identical LLHs with ES-HHA.

The third experiment provides the sensitivity analysis of the proposed ES-HHA with the weight parameter w_1. Here, we fix the w_1 to $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and run ES-HHA 30 times independently.

4.2 Experimental results

4.2.1 Comparison experiments on CEC2014, CEC2022, and engineering problems

The experimental results on CEC2014, CEC2022, and engineering problems are provided in this section. To identify the statistical significance between every pair of algorithms, we first apply the Kruskal-Wallis test, and if the statistical significance exists, we further apply the Mann–Whitney U test to calculate the p value between every pair of competitor algorithms. Subsequently, the Holm multiple comparison test [71] is employed to correct the p-value obtained by the Mann–Whitney U test and identify the statistical significance level. Marks $+$, $\approx$, and $-$ are applied to denote that our proposed ES-HHA is significantly better, with no significance, and significantly worse with the compared method. The best value is colored in bold.

Tables 7 and 8 summarize the experimental results with 14 meta-heuristic algorithms, while Figs. 5 and 6 visualize the convergence curve of representative functions on CEC2014 (i.e., f_1 and f_2: unimodal functions; f_4, f_7, f_{10}, and f_{11}: multimodal functions; f_{17}, f_{21} and f_{22}: hybrid functions; f_{23}, f_{28}, and f_{30}: composition functions).

Table 3 Summary of the CEC2022 benchmark functions: Uni. = unimodal function, Basic. = basic function, Hybrid. = hybrid function, Comp. = composition function

Fun	Description	Feature	Optimum
f_1	Shifted and full rotated Zakharov	Uni	300
f_2	Shifted and full rotated Rosenbrock	Basic.	400
f_3	Shifted and full rotated expanded Schaffer f_6		600
f_4	Shifted and full rotated non-continuous Rastrigin		800
f_5	Shifted and full rotated Levy		900
f_6	Hybrid function 1 (N = 3)	Hybrid.	1800
f_7	Hybrid function 2 (N = 6)		2000
f_8	Hybrid function 3 (N = 5)		2200
f_9	Composition function 1 (N = 5)	Comp.	2300
f_{10}	Composition function 2 (N = 4)		2400
f_{11}	Composition function 3 (N = 5)		2600
f_{12}	Composition function 3 (N = 6)		2700

Table 4 Summary of eight engineering optimization problems

Name	Abbr	Dim	# of constraints
Welded beam problem	WBP	4	7
Speed reducer problem	SRD	7	11
Three bar truss problem	TBTD	2	3
Gear train problem	GTD	4	0
Cantilever beam problem	CBD	5	1
Tubular column problem	TCD	2	6
Corrugated bulkhead problem	CBHD	4	6
Reinforced concrete beam problem	RCB	3	2

Tables 9 and 10 summarize the experimental results and statistical analysis on CEC2022 benchmark functions. Similarly, the statistical significance is also identified between every pair of algorithms to demonstrate the competitiveness of ES-HHA. The convergence curves are visualized in Figs. 7 and 8.

Furthermore, to investigate the performance of ES-HHA in real-world applications, we apply ES-HHA to engineering simulation optimization problems. This section provides the corresponding experimental results. Table 11 summarizes the details and Fig. 9 demonstrates the convergence curves.

In this study, we conduct a set of comparison experiments on CEC2014 benchmark functions, CEC2022 benchmark functions, and eight engineering optimization problems to evaluate the performance of the proposed ES-HHA comprehensively. Through the experimental results and the statistical analysis, ES-HHA has significant competitiveness against 14 famous meta-heuristic algorithms. The following observation is summarized to conclude the experiments.

1. Since the f_1 to f_3 in the CEC2014 suite and f_1 in the CEC2022 suite are unimodal functions, only one global optimum exists in these problems. Therefore, these functions are allowed to evaluate the exploitation capacity of optimizers. The experimental results and statistical analysis demonstrate that ES-HHA is significantly better than the meta-heuristic algorithms in most instances except for certain cases when compared with state-of-the-art optimizers L-SHADE and ECMADE. We owe this superiority to the efficient exploitation and exploration pool design and effective evolutionary status guided probabilistic selection function design.
2. The rest of the functions are multimodal, hybrid, and composite, these functions have several optima and

Table 5 The parameter setting of all compared optimization methods

Alg	Parameters	Value
ES-HHA	Search radius R	1
	Scaling factor F	0.8
	Weight parameter w_1	0.5
GA	Crossover probability pc	0.9
	Mutation probability pm	0.01
PSO	Inertia factor w	1
	Coefficients c_1 and c_2	2.05
	Max. and min. speed	2 and -2
DE	Mutation strategy	DE/cur-to-rand/1/bin
	Scaling factor F	0.8
	Crossover rate Cr	0.9
ES	Survival rate λ	0.75
CMA-ES	σ	1.3
L-SHADE	μ_F and μ_{Cr}	0.5 and 0.5
	Population size	18 $\times D$
MFO	Constant b	1
	Adaptive parameter t	$[-1, 1]$
WOA	Constant b	1
SCA	Constant A	2
CCDE	Parameter-free	
HCO	Probability of fitness selection P_{fit}	0.65
	Weight factor w_1	0.1
	Constants C_1 and C_2	1.4
SAO	Degree-day factor DDF	[0.35, 0.6]
BOA	Parameter-free	
ECMADE	μ_{Cr}	0.9, 0.5, and 0.5
	p_1, p_2, and p_3	0.9, 0.8, and 0.8
	θ and α	1/13 and 0.8

Table 6 The compared hyper-heuristic algorithms and descriptions

Hyper-heuristics	Description
Simple random (SR) [67, 68]	Select the search operator with uniform probability
Random descent (RD) [69]	Initially select a search operator randomly and
	Repeat it as long as the improvement is achieved
Random permutation (RP) [70]	Randomly construct a sequence of heuristics

complex fitness landscapes, thus, they can evaluate the balanced ability between exploration and exploitation and the capacity to escape the local optimum. We first highlight the superiority of the proposed ES-HHA in hybrid functions from both the CEC2014 and CEC2022 suites. In CEC2014 $f_{17} - f_{22}$ and CEC2022 $f_6 - f_8$, ES-HHA has best performance on nearly most cases. Through the statistical analysis, ES-HHA is at least statistically equal or significantly better than the other 14 meta-heuristic algorithms. These results practically confirm the efficiency and effectiveness of our proposed ES-HHA. In the meantime, we notice that ES-HHA is not good at dealing with some composite functions such as $f_{23} - f_{25}$ and $f_{26} - f_{28}$ in the CEC2014

Table 7 The experimental results and statistical analyses on 30-D CEC2014 benchmark functions

Func	GA	PSO	DE	ES	CMA-ES	L-SHADE	MFO	WOA	SCA	CCDE	HCO	SAO	BOA	ECMADE	ES-HHA
f_1	1.07e+09 +	4.32e+08 +	2.87e+08 +	1.08e+08 +	1.55e+07 +	5.12e+07 +	1.20e+09 +	7.44e+07 +	1.20e+08 +	1.59e+08 +	2.78e+09 +	2.17e+08 +	4.77e+08 +	4.05e+07 +	**1.17e+06**
f_2	7.14e+10 +	5.57e+09 +	3.70e+10 +	3.57e+09 +	2.91e+09 +	1.75e+06 ≈	8.53e+10 +	8.00e+09 +	3.24e+09 +	9.94e+09 +	1.20e+11 +	1.40e+10 +	5.03e+10 +	2.24e+07 +	**1.51e+06**
f_3	1.36e+05 +	1.07e+05 +	7.11e+04 +	1.50e+05 +	5.59e+03 +	2.44e+03 −	2.89e+05 +	5.29e+04 +	5.65e+04 +	2.81e+04 +	9.15e+05 +	6.28e+04 +	5.47e+04 +	**1.31e+03** −	3.79e+03
f_4	1.27e+04 +	2.22e+03 +	4.18e+03 +	8.77e+02 +	8.57e+02 +	5.33e+02 +	1.64e+04 +	1.11e+03 +	8.94e+02 +	1.71e+03 +	1.00e+04 +	1.42e+03 +	7.28e+03 +	5.51e+02 +	**4.80e+02**
f_5	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	**5.20e+02** −	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02
f_6	6.41e+02 +	6.40e+02 +	6.37e+02 +	6.29e+02 +	6.38e+02 +	6.27e+02 ≈	6.39e+02 +	6.40e+02 +	6.25e+02 −	6.35e+02 +	6.35e+02 +	6.30e+02 +	6.32e+02 +	**6.24e+02** −	6.27e+02
f_7	1.32e+03 +	7.69e+02 +	1.03e+03 +	7.48e+02 +	7.28e+02 +	7.01e+02 +	1.38e+03 +	7.31e+02 +	7.36e+02 +	7.84e+02 +	9.78e+02 +	8.48e+02 +	1.16e+03 +	7.01e+02 +	**7.01e+02**
f_8	1.17e+03 +	1.10e+03 +	1.11e+03 +	1.02e+03 +	1.06e+03 +	**8.79e+02** −	1.21e+03 +	1.02e+03 +	9.81e+02 ≈	9.74e+02 ≈	1.02e+03 +	9.67e+02 ≈	1.02e+03 +	1.01e+03 +	9.73e+02
f_9	1.34e+03 +	1.21e+03 +	1.29e+03 +	1.23e+03 +	1.17e+03 ≈	**1.07e+03** −	1.43e+03 +	1.16e+03 ≈	1.11e+03 ≈	1.19e+03 +	1.14e+03 ≈	1.09e+03 −	1.15e+03 ≈	1.15e+03 ≈	1.14e+03
f_{10}	8.31e+03 +	8.43e+03 +	6.51e+03 +	4.81e+03 +	7.94e+03 +	**3.68e+03** −	7.61e+03 +	5.79e+03 +	6.05e+03 +	4.72e+03 −	8.27e+03 +	5.71e+03 +	6.31e+03 +	7.16e+03 +	5.27e+03
f_{11}	8.77e+03 +	8.78e+03 +	8.64e+03 +	**4.84e+03** −	8.57e+03 +	6.16e+03 +	8.02e+03 +	6.49e+03 +	7.46e+03 +	7.52e+03 +	8.47e+03 +	6.15e+03 +	6.52e+03 +	8.56e+03 +	5.42e+03
f_{12}	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	**1.20e+03**
f_{13}	1.31e+03 +	1.30e+03 +	1.30e+03 +	1.30e+03 +	1.30e+03 +	**1.30e+03** ≈	1.31e+03 +	1.30e+03 +	1.30e+03 ≈	1.30e+03 +	1.30e+03 +	1.30e+03 +	1.31e+03 +	1.30e+03 ≈	1.30e+03
f_{14}	1.59e+03 +	1.41e+03 +	1.51e+03 +	1.41e+03 +	1.41e+03 +	**1.40e+03** ≈	1.63e+03 +	1.42e+03 +	1.41e+03 +	1.43e+03 +	1.49e+03 +	1.44e+03 +	1.54e+03 +	1.40e+03 +	1.40e+03
f_{15}	2.28e+06 +	3.93e+03 +	1.82e+06 +	2.05e+03 +	3.67e+03 +	**1.52e+03** ≈	1.45e+07 +	3.98e+03 +	1.61e+03 +	6.74e+04 +	1.63e+07 +	5.08e+03 +	5.83e+04 +	1.52e+03 +	1.52e+03
f_{16}	1.61e+03 +	1.61e+03 +	1.61e+03 +	1.61e+03 +	1.61e+03 +	**1.61e+03** ≈	1.61e+03 +	1.61e+03 +	1.61e+03 +	1.61e+03 +	1.61e+03 +	1.61e+03 +	1.61e+03 ≈	1.61e+03 +	1.61e+03
f_{17}	6.80e+07 +	3.29e+07 +	1.42e+07 +	3.90e+06 +	7.04e+05 +	1.87e+06 +	5.81e+07 +	1.30e+07 +	4.54e+06 +	6.38e+06 +	2.93e+08 +	1.01e+07 +	1.77e+07 +	1.87e+06 +	**3.39e+04**
f_{18}	2.23e+09 +	6.31e+08 +	4.36e+08 +	6.51e+05 +	1.77e+07 +	3.41e+06 +	5.40e+08 +	1.15e+05 +	1.26e+07 +	1.59e+08 +	9.55e+09 +	5.24e+06 +	5.51e+08 +	4.33e+07 +	**6.72e+04**
f_{19}	1.92e+08 +	2.22e+07 +	9.46e+06 +	8.87e+03 +	4.43e+04 +	5.28e+03 +	1.31e+08 +	1.52e+04 +	2.54e+04 +	2.12e+06 +	2.78e+09 +	1.37e+04 +	7.97e+06 +	3.30e+05 +	**1.93e+03**
f_{20}	4.80e+13 +	7.82e+11 +	1.99e+10 +	2.04e+07 +	1.84e+06 +	2.17e+05 +	8.47e+13 +	9.04e+08 +	3.58e+07 +	3.68e+09 +	4.72e+15 +	1.28e+09 +	7.34e+11 +	2.15e+07 +	**3.24e+04**
f_{21}	3.58e+08 +	7.97e+07 +	2.64e+07 +	6.84e+06 +	6.88e+05 +	4.83e+05 +	5.52e+08 +	2.75e+06 +	5.69e+06 +	1.27e+07 +	2.51e+09 +	2.06e+07 +	8.85e+07 +	1.99e+06 +	**2.78e+04**
f_{22}	6.83e+10 +	4.97e+09 +	7.42e+07 +	5.92e+03 +	4.47e+03 +	4.17e+03 +	2.91e+12 +	7.46e+03 +	5.55e+03 +	7.21e+05 +	8.34e+13 +	5.74e+03 +	9.02e+04 +	1.45e+04 +	**2.53e+03**
f_{23}	2.92e+03 +	2.55e+03 +	2.73e+03 +	2.58e+03 +	2.55e+03 +	2.53e+03 +	3.35e+03 +	2.55e+03 +	2.54e+03 +	2.61e+03 +	2.62e+03 +	2.54e+03 +	2.54e+03 +	2.52e+03 +	**2.52e+03**
f_{24}	2.86e+03 +	2.65e+03 +	2.89e+03 +	2.67e+03 +	2.68e+03 +	2.61e+03 −	2.98e+03 +	**2.60e+03** −	2.60e+03 −	2.72e+03 +	2.65e+03 +	2.60e+03 −	**2.60e+03** −	2.61e+03 −	2.64e+03
f_{25}	2.80e+03 +	2.76e+03 +	2.75e+03 +	2.74e+03 +	2.71e+03 −	2.72e+03 ≈	2.83e+03 +	2.72e+03 ≈	2.72e+03 ≈	2.74e+03 +	2.74e+03 +	2.70e+03 −	2.70e+03 −	2.71e+03 −	2.73e+03
f_{26}	2.71e+03 −	2.70e+03 −	2.70e+03 −	2.75e+03 +	2.70e+03 −	**2.70e+03** −	2.71e+03 −	2.70e+03 −	2.70e+03 −	2.70e+03 −	2.75e+03 +	2.74e+03 ≈	2.72e+03 ≈	2.70e+03 −	2.73e+03

Table 7 (continued)

Func	GA	PSO	DE	ES	CMA-ES	L-SHADE	MFO	WOA	SCA	CCDE	HCO	SAO	BOA	ECMADE	ES-HHA
f_{27}	3.99e+03 +	3.85e+03 +	3.97e+03 +	3.69e+03 ≈	3.93e+03 +	**3.36e+03** −	4.00e+03 +	4.03e+03 +	3.55e+03 −	3.87e+03 +	4.21e+03 +	3.82e+03 +	3.81e+03 +	3.51e+03 −	3.71e+03
f_{28}	8.19e+03 +	6.73e+03 +	3.97e+03 −	6.61e+03 +	4.01e+03 −	3.84e+03 −	5.17e+03 −	5.83e+03 ≈	5.08e+03 −	3.88e+03 −	8.93e+03 +	5.48e+03 −	4.67e+03 −	**3.79e+03** −	5.88e+03
f_{29}	8.61e+08 +	3.37e+08 +	8.16e+07 +	6.74e+07 +	6.72e+06 +	2.14e+06 +	5.13e+08 +	5.42e+07 +	3.64e+07 +	8.20e+07 +	3.53e+09 +	1.18e+06 +	2.55e+06 +	4.92e+06 +	**1.05e+04**
f_{30}	6.24e+10 +	8.19e+09 +	5.15e+08 +	2.17e+08 +	4.25e+07 +	9.39e+07 +	4.52e+11 +	4.16e+08 +	5.72e+08 +	2.58e+08 +	9.44e+13 +	6.89e+08 +	6.15e+08 +	5.78e+07 +	**2.73e+04**
+/≈/−	28/1/1	28/1/1	27/1/2	26/2/2	25/2/3	14/8/8	27/1/2	24/3/3	20/5/5	25/2/3	28/2/0	23/3/4	23/4/3	20/3/7	–

f_1-f_3: unimodal functions; f_4-f_{16}: simple multimodal functions; $f_{17}-f_{22}$: hybrid functions; $f_{23}-f_{30}$: composition functions

suite comparing with WOA, SAO, BOA, and ECMADE. Moreover, L-SHADE, as one of the champions in the CEC2014, is significantly better than ES-HHA in many instances of 50-D CEC2014 from f_4 to f_{16} and f_{23} to f_{28}, and No Free Lunch Theorem (NFLT) [72] can be used to explain this phenomenon: NFLT emphasizes any pair of algorithms have the same average performance in all possible problems, thus if an algorithm is better than another algorithm in a category of the problem, it must degenerate on the other kinds of problems since it is the only way to achieve the same average performance.

3. Eight engineering optimization problems contain multiple constraints and complex fitness landscapes, and the optimization performance on these tasks can reflect the capacity of algorithms in real-world scenarios. The experimental results summarized in Table 11 further demonstrate the superiority of the proposed ES-HHA, confirming that ES-HHA is a promising optimization approach in real-world applications.

In conclusion, our proposal has achieved satisfactory performance verified by comprehensive numerical experiments.

4.2.2 Ablation experiments on CEC2014 benchmark functions

To investigate the effectiveness of the evolutionary status guided probabilistic selection function, the ablation experiment is conducted to compete with baseline hyper-heuristic frameworks. Tables 12 and 13 summarize the results on CEC2014 benchmark functions.

As the experimental results and statistical analysis of ablation experiments on CEC2014 benchmark functions, these experiments aim to investigate the effectiveness of the proposed evolutionary status guided probabilistic selection function. Three conventional hyper-heuristic frameworks are employed as the baseline: simple random (SR), random descent (RD), and random permutation (RP). Statistical analysis proves that the proposed high-level component of ES-HHA is competitive with the competitor baseline. In a total of 60 benchmark functions, the significant deterioration situation does not happen. Additionally, as the dimension of the problem increases, the superiority is also amplified. Therefore, our proposed evolutionary status guided probabilistic selection function is more efficient than the compared baseline hyper-heuristic frameworks, which can learn the regularity and knowledge from the optimization information and

Table 8 The experimental results and statistical analyses on 50-D CEC2014 benchmark functions

Func	GA	PSO	DE	ES	CMA-ES	L-SHADE	MFO	WOA	SCA	CCDE	HCO	SAO	BOA	ECMADE	ES-HHA
f_1	2.95e+09 +	9.49e+08 +	1.26e+09 +	8.22e+07 +	7.73e+07 +	4.97e+07 +	2.65e+09 +	1.39e+08 +	2.43e+08 +	7.28e+08 +	8.14e+09 +	4.97e+08 +	1.96e+09 +	1.42e+08 +	**5.90e+06**
f_2	1.66e+11 +	2.76e+10 +	1.03e+11 +	3.90e+09 +	1.22e+10 +	**4.53e+04** −	1.95e+11 +	2.73e+10 +	1.31e+10 +	3.16e+10 +	2.58e+11 +	4.81e+10 +	1.24e+11 +	3.94e+07 +	1.07e+07
f_3	2.52e+05 +	1.99e+05 +	3.22e+05 +	2.41e+05 +	2.24e+04 +	2.08e+04 +	4.42e+05 +	9.39e+04 +	1.25e+05 +	1.29e+05 +	7.31e+05 +	1.37e+05 +	1.14e+05 +	4.31e+04 +	**4.64e+03**
f_4	4.61e+04 +	7.95e+03 +	1.96e+04 +	1.10e+03 +	1.67e+03 +	5.47e+02 ≈	6.26e+04 +	3.17e+03 +	2.39e+03 +	4.02e+03 +	3.86e+04 +	8.20e+03 +	2.91e+04 +	5.42e+02 ≈	**5.34e+02**
f_5	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	**5.20e+02** −	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02 ≈	5.21e+02
f_6	6.75e+02 +	6.73e+02 +	6.64e+02 +	6.54e+02 ≈	6.60e+02 +	6.49e+02 −	6.68e+02 +	6.72e+02 +	6.51e+02 −	6.65e+02 +	6.67e+02 +	6.57e+02 ≈	6.62e+02 +	**6.39e+02** −	6.57e+02
f_7	2.33e+03 +	9.53e+02 +	1.77e+03 +	7.50e+02 +	8.24e+02 +	**7.00e+02** −	2.68e+03 +	9.27e+02 +	8.10e+02 +	9.81e+02 +	1.50e+03 +	1.15e+03 +	1.87e+03 +	7.01e+02 +	7.01e+02
f_8	1.55e+03 +	1.37e+03 +	1.42e+03 +	1.29e+03 +	1.29e+03 +	**9.53e+02** −	1.61e+03 +	1.24e+03 +	1.18e+03 ≈	1.15e+03 ≈	1.27e+03 +	1.19e+03 ≈	1.29e+03 +	1.21e+03 ≈	1.19e+03
f_9	1.80e+03 +	1.45e+03 +	1.72e+03 +	1.60e+03 +	1.41e+03 ≈	**1.22e+03** −	1.99e+03 +	1.41e+03 ≈	1.32e+03 −	1.50e+03 +	1.41e+03 ≈	1.33e+03 −	1.45e+03 +	1.36e+03 ≈	1.39e+03
f_{10}	1.48e+04 +	1.51e+04 +	1.26e+04 +	8.42e+03 ≈	1.47e+04 +	**6.25e+03** −	1.40e+04 +	1.03e+04 +	1.16e+04 +	8.96e+03 +	1.48e+04 +	1.09e+04 +	1.20e+04 +	1.34e+04 +	8.48e+03
f_{11}	1.54e+04 +	1.53e+04 +	1.51e+04 +	**8.31e+03** ≈	1.51e+04 +	1.06e+04 +	1.40e+04 +	1.17e+04 +	1.31e+04 +	1.33e+04 +	1.52e+04 +	1.18e+04 +	1.25e+04 +	1.51e+04 +	8.66e+03
f_{12}	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 −	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	1.20e+03 +	**1.20e+03**
f_{13}	1.31e+03 +	1.30e+03 +	1.31e+03 +	1.30e+03 ≈	1.30e+03 +	**1.30e+03** −	1.31e+03 +	1.30e+03 +	1.30e+03 +	1.30e+03 +	1.31e+03 +	1.30e+03 +	1.31e+03 +	1.30e+03 ≈	1.30e+03
f_{14}	1.81e+03 +	1.45e+03 +	1.71e+03 +	1.40e+03 +	1.43e+03 +	**1.40e+03** −	1.94e+03 +	1.44e+03 +	1.43e+03 +	1.49e+03 +	1.57e+03 +	1.50e+03 +	1.68e+03 +	1.40e+03 ≈	1.40e+03
f_{15}	2.30e+07 +	6.96e+04 +	1.09e+07 +	1.99e+03 +	2.30e+04 +	**1.53e+03** −	1.19e+08 +	5.09e+04 +	6.64e+03 +	7.97e+05 +	1.00e+08 +	1.50e+05 +	1.39e+06 +	1.54e+03 +	1.54e+03
f_{16}	1.62e+03 +	1.62e+03 +	1.62e+03 +	1.62e+03 +	1.62e+03 +	**1.62e+03** −	1.62e+03 +	1.62e+03 +	1.62e+03 +	1.62e+03 +	1.62e+03 +	1.62e+03 +	1.62e+03 ≈	1.62e+03 +	1.62e+03
f_{17}	1.96e+08 +	9.50e+07 +	8.51e+07 +	3.84e+06 +	6.54e+06 +	4.44e+06 +	1.72e+08 +	1.01e+07 +	1.23e+07 +	3.98e+07 +	9.09e+08 +	3.35e+07 +	8.58e+07 +	1.55e+07 +	**1.78e+05**
f_{18}	8.31e+09 +	2.25e+09 +	2.18e+09 +	3.09e+06 +	1.24e+08 +	9.58e+06 +	3.00e+09 +	2.15e+06 +	7.76e+07 +	1.17e+09 +	2.07e+10 +	5.19e+07 +	5.55e+09 +	2.23e+08 +	**1.57e+05**
f_{19}	4.30e+09 +	5.82e+08 +	3.12e+08 +	7.88e+03 +	1.27e+06 +	2.07e+04 +	1.69e+09 +	9.85e+04 +	1.43e+06 +	6.88e+07 +	2.24e+10 +	5.93e+06 +	1.51e+09 +	4.00e+06 +	**2.11e+03**
f_{20}	7.48e+14 +	3.03e+13 +	8.21e+12 +	3.16e+05 +	9.33e+08 +	6.63e+05 +	9.03e+14 +	1.05e+08 +	8.25e+08 +	6.31e+11 +	2.62e+16 +	5.36e+10 +	2.65e+13 +	8.13e+09 +	**1.16e+05**
f_{21}	1.78e+09 +	3.92e+08 +	3.17e+08 +	3.63e+06 +	6.85e+06 +	2.40e+06 +	1.44e+09 +	1.22e+07 +	2.25e+07 +	1.18e+08 +	7.79e+09 +	9.08e+07 +	7.39e+08 +	2.14e+07 +	**2.43e+05**
f_{22}	3.16e+13 +	8.37e+11 +	2.83e+11 +	7.55e+03 +	3.01e+06 +	6.12e+03 +	5.48e+13 +	7.97e+03 +	8.79e+05 +	5.12e+09 +	3.23e+15 +	2.25e+10 +	7.33e+11 +	7.07e+07 +	**2.58e+03**
f_{23}	3.39e+03 +	2.55e+03 +	3.06e+03 +	2.57e+03 +	2.56e+03 +	2.54e+03 ≈	4.28e+03 +	2.53e+03 −	2.54e+03 ≈	2.66e+03 +	2.62e+03 +	2.51e+03 +	**2.50e+03** −	2.54e+03 ≈	2.54e+03
f_{24}	3.16e+03 +	2.74e+03 +	3.25e+03 +	2.73e+03 +	2.76e+03 +	2.67e+03 −	3.38e+03 +	2.61e+03 −	2.60e+03 −	2.90e+03 +	2.72e+03 ≈	2.60e+03 −	**2.60e+03** −	2.68e+03 −	2.72e+03
f_{25}	2.95e+03 +	2.81e+03 +	2.82e+03 +	2.77e+03 +	2.73e+03 −	2.74e+03 ≈	2.96e+03 +	2.70e+03 −	2.72e+03 −	2.78e+03 +	2.77e+03 +	2.70e+03 −	**2.70e+03** −	2.73e+03 ≈	2.75e+03
f_{26}	2.76e+03 ≈	2.71e+03 ≈	2.71e+03 ≈	2.82e+03 +	2.73e+03 ≈	2.72e+03 ≈	2.71e+03 ≈	2.70e+03 ≈	2.74e+03 ≈	**2.70e+03** ≈	2.78e+03 ≈	2.78e+03 ≈	2.76e+03 ≈	2.77e+03 ≈	2.77e+03

Table 8 (continued)

Func	GA	PSO	DE	ES	CMA-ES	L-SHADE	MFO	WOA	SCA	CCDE	HCO	SAO	BOA	ECMADE	ES-HHA
f_{27}	5.15e+03 +	4.83e+03 +	4.85e+03 +	4.58e+03 +	4.70e+03 +	4.04e+03 −	4.81e+03 +	5.11e+03 +	4.29e+03 −	4.68e+03 +	5.28e+03 +	4.64e+03 +	4.83e+03 +	**3.91e+03** −	4.49e+03
f_{28}	1.45e+04 +	1.27e+04 +	4.74e+03 −	1.12e+04 +	5.04e+03 −	4.34e+03 −	7.25e+03 −	1.01e+04 +	9.60e+03 ≈	4.46e+03 −	1.59e+04 +	8.14e+03 −	8.62e+03 ≈	**4.32e+03** −	9.52e+03
f_{29}	4.21e+09 +	1.73e+09 +	4.24e+08 +	9.84e+07 +	5.95e+07 +	1.34e+06 +	2.75e+09 +	5.07e+07 +	1.51e+08 +	4.06e+08 +	1.21e+10 +	3.11e+03 −	**3.11e+03** −	3.16e+07 +	1.27e+05
f_{30}	4.09e+12 +	1.45e+11 +	1.31e+10 +	1.18e+08 +	2.45e+08 +	3.44e+08 +	1.87e+13 +	1.29e+09 +	1.54e+09 +	2.99e+09 +	8.19e+14 +	2.22e+09 +	4.55e+09 +	1.05e+09 +	**2.80e+05**
+/≈/−	28/2/0	28/2/0	27/2/1	25/5/0	25/3/2	11/5/14	27/2/1	24/2/4	20/5/5	26/3/1	26/4/0	21/4/5	22/4/4	17/9/4	–

determine the optimization sequence automatically and intelligently.

4.2.3 Sensitivity experiments on CEC2014 benchmark functions

Sensitivity experiments involving the weight (w_1) in the evolutionary status guided probabilistic selection function between the FDC and the PD are implemented. Tables 14 and 15 summarize the results on CEC2014 benchmark functions.

Through the statistical analysis of sensitivity experiments summarized in Tables 14 and 15, we fix the hyper-parameter w_1 in evolutionary status guided probabilistic selection function at $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ to investigate the sensitivity of our proposed ES-HHA to the control parameter. Experimental results and statistical analysis demonstrate that our proposed ES-HHA is not sensitive to the parameter w_1 in most instances, and ES-HHA is expected to exhibit strong robustness in unknown optimization problems.

5 Discussion

5.1 Computational complexity analysis

This section analyzes the computational complexity of our proposed ES-HHA. Assuming the population size is N, the dimension of the problem is D, and the maximum iteration time is T. First, the computational complexity of the population initialization is $O(N \cdot D)$, then, ES-HHA enters the iteration. For the sake of simplicity, we analyze the computational complexity at one iteration.

FDC and PD are calculated at first. For the FDC, the Euclidean distances between every individual to the optimum are necessary, and the complexity is $O(N \cdot D)$. According to Eq. (2), the total computational complexity of FDC calculation is $O(N \cdot D + D) := O(N \cdot D)$. For the PD estimation, the average value of each dimension corresponding to $\bar{X^j}$ in Eq. (3) is computed, where the complexity is $O(N \cdot D)$. Thus, the total complexity of PD computation is $O(N \cdot D + N \cdot D) := O(N \cdot D)$.

Next, ES-HHA selects the subsequent search operators for individuals based on the probability calculated by FDC and PD. Given the number of search operators is constant, and the complexity of the search operator selection and offspring generation is $O(N)$ and $O(N \cdot D)$, respectively. Finally, we apply the one-to-one selection scheme for population update, and the complexity is $O(N)$.

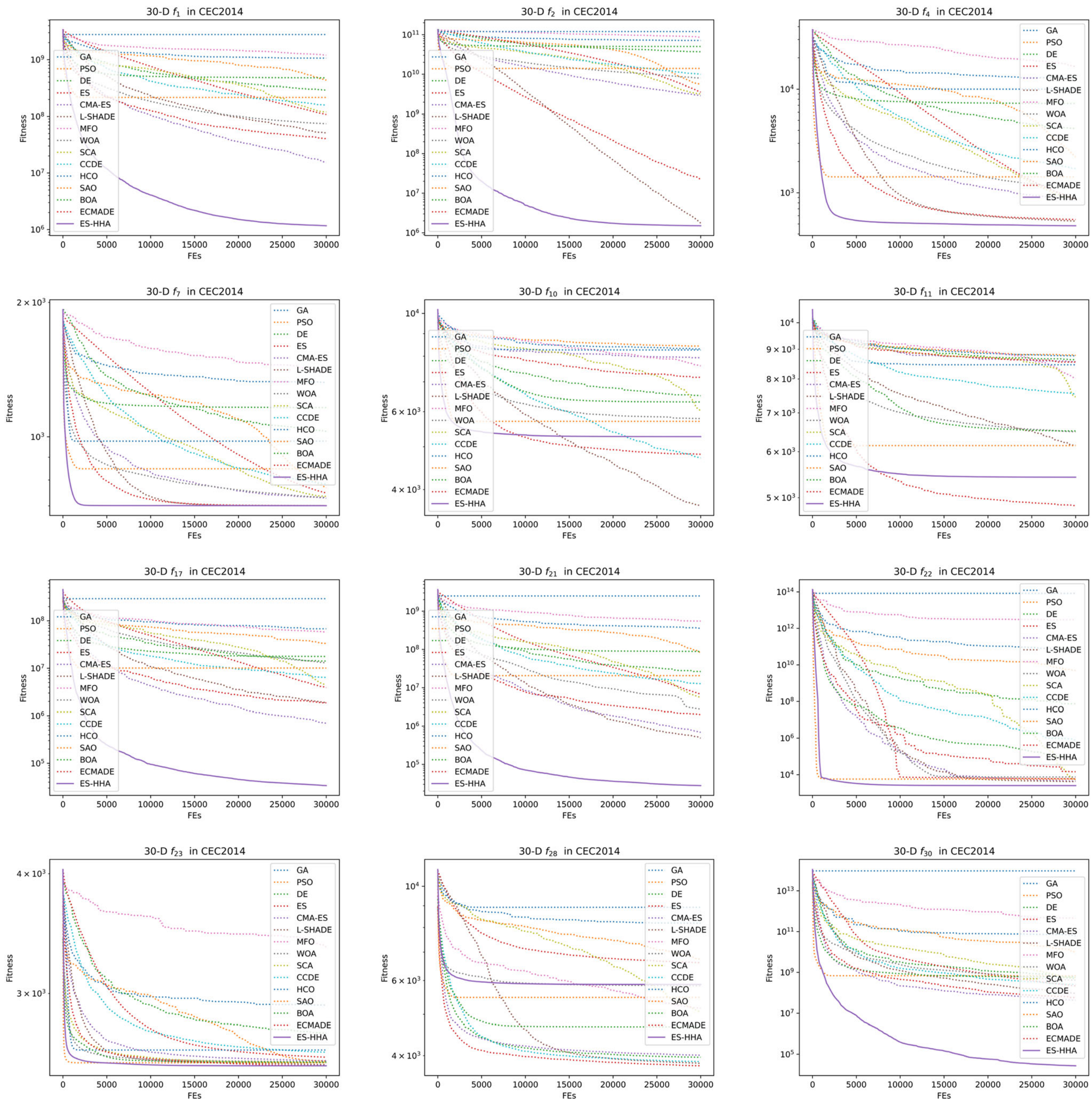


Fig. 5 Convergence curves on representative 30-D CEC2014 benchmark functions

From the above analysis, the computational complexity of one iteration optimization is $O(N \cdot D + N \cdot D + N + N \cdot D + N) := O(N \cdot D)$, and the total complexity of ES-HHA is $O(N \cdot D) + O(T \cdot N \cdot D) := O(T \cdot N \cdot D)$. Since the sort operator is not involved in ES-HHA, it has cheaper computational complexity than the intricate meta-heuristics such as the grey wolf optimizer (GWO) [73] and the vegetation evolution (VEGE) [34].

5.2 Potential and open topics

The above experiments and analysis show the satisfactory performance of our proposal in continuous optimization.

Table 9 The experimental results and statistical analyses on 10-D CEC2022 benchmark functions

Func.		GA	PSO	DE	ES	CMA-ES	L-SHADE	MFO	WOA	SCA	CCDE	HCO	SAO	BOA	ECMADE	ES-HHA
f_1	Mean	8.17e+03 +	1.66e+03 +	7.14e+03 +	2.80e+03 +	4.22e+02 +	1.32e+03 +	1.78e+04 +	3.50e+03 +	6.02e+02 +	4.60e+03 +	2.15e+04 +	9.97e+02 +	1.98e+03 +	3.50e+02 +	**3.00e+02**
	Std	1.72e+03	1.56e+03	1.73e+03	1.65e+03	4.01e+01	3.35e+02	5.27e+03	1.59e+03	1.21e+02	1.25e+03	6.18e+03	1.10e+03	1.28e+03	2.71e+01	8.20e−03
f_2	Mean	7.81e+02 +	5.16e+02 +	4.60e+02 +	5.02e+02 +	4.17e+02 +	4.21e+02 +	6.40e+02 +	4.58e+02 +	4.37e+02 +	4.43e+02 +	7.70e+02 +	4.38e+02 +	5.24e+02 +	4.07e+02 +	**4.06e+02**
	Std	1.28e+02	4.47e+01	1.43e+01	9.38e+01	3.04e+00	8.91e+00	1.26e+02	6.12e+01	1.40e+01	1.46e+01	1.96e+02	2.77e+01	9.79e+01	1.74e+00	3.36e+00
f_3	Mean	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	**6.00e+02**
	Std	7.23e−02	6.02e−03	1.40e−02	3.69e−02	6.56e−04	2.28e−04	9.21e−02	1.33e−02	2.29e−03	4.47e−03	4.57e−02	1.20e−02	6.82e−02	7.87e−07	2.74e−07
f_4	Mean	8.01e+02 +	8.01e+02 +	8.01e+02 +	**8.00e+02** ≈	8.01e+02 +	8.00e+02 +	8.01e+02 +	8.00e+02 +	8.00e+02 +	8.01e+02 +	8.01e+02 +	8.01e+02 +	8.00e+02 ≈	8.01e+02 +	8.00e+02
	Std	1.81e−01	1.92e−01	1.96e−01	1.52e−01	1.57e−01	1.05e−01	3.23e−01	2.69e−01	1.41e−01	1.49e−01	3.06e−01	3.13e−01	1.33e−01	2.21e−01	2.49e−01
f_5	Mean	9.03e+02 +	9.01e+02 ≈	9.03e+02 +	9.02e+02 +	9.00e+02 −	9.00e+02 −	9.05e+02 +	9.02e+02 +	9.00e+02 −	9.02e+02 +	9.01e+02 ≈	9.01e+02 ≈	9.00e+02 −	**9.00e+02** −	9.01e+02
	Std	6.18e−01	6.04e−01	7.56e−01	1.15e+00	6.38e−02	1.13e−01	1.16e+00	1.83e+00	2.95e−02	4.95e−01	2.54e−01	5.26e−01	3.09e−01	1.33e−03	7.14e−01
f_6	Mean	2.53e+07 +	2.02e+06 +	1.00e+05 +	5.08e+04 +	4.10e+03 +	4.56e+04 +	3.26e+07 +	3.57e+04 +	2.05e+05 +	1.17e+05 +	3.51e+08 +	3.91e+04 +	2.10e+04 +	7.93e+03 +	**1.91e+03**
	Std	1.69e+07	4.47e+06	2.88e+04	1.80e+04	6.09e+02	2.31e+04	2.91e+07	1.52e+04	1.01e+05	4.96e+04	2.45e+08	1.80e+04	1.30e+04	2.65e+03	6.07e+01
f_7	Mean	2.39e+03 +	2.12e+03 +	2.07e+03 +	2.14e+03 +	2.08e+03 +	2.04e+03 +	2.52e+03 +	2.16e+03 +	2.07e+03 +	2.07e+03 +	2.18e+03 +	2.09e+03 +	2.04e+03 +	2.05e+03 +	**2.04e+03**
	Std	1.43e+02	5.24e+01	1.02e+01	9.73e+01	1.48e+01	5.07e+00	2.11e+02	9.38e+01	1.36e+01	1.63e+01	1.19e+02	1.88e+02	7.92e+00	1.86e+01	1.99e+01
f_8	Mean	2.99e+04 +	2.50e+03 +	2.26e+03 +	2.98e+03 +	2.23e+03 +	2.24e+03 +	8.65e+06 +	3.31e+03 +	2.31e+03 +	2.27e+03 +	1.04e+11 +	2.75e+03 +	2.23e+03 ≈	2.25e+03 +	**2.22e+03**
	Std	6.25e+04	4.84e+02	1.03e+01	7.46e+02	1.94e+00	9.09e+00	3.45e+07	9.56e+02	5.04e+01	2.22e+01	2.43e+11	5.03e+02	3.46e+00	1.02e+02	5.81e+00
f_9	Mean	2.88e+03 +	2.63e+03 +	2.67e+03 +	2.74e+03 +	2.60e+03 +	2.36e+03 ≈	2.85e+03 +	2.66e+03 +	**2.34e+03** ≈	2.63e+03 +	2.84e+03 +	2.64e+03 +	2.54e+03 ≈	2.62e+03 +	2.52e+03
	Std	6.67e+01	1.48e+02	2.67e+01	1.55e+02	1.11e+02	3.71e+01	6.96e+01	2.02e+02	1.27e+01	7.57e+01	1.99e+02	1.93e+02	1.75e+02	1.07e+02	1.76e+02
f_{10}	Mean	2.68e+03 ≈	2.62e+03 −	2.62e+03 −	2.67e+03 ≈	**2.61e+03** −	2.61e+03 −	2.79e+03 +	2.69e+03 ≈	2.61e+03 −	2.63e+03 ≈	2.63e+03 ≈	2.66e+03 ≈	2.62e+03 −	2.62e+03 −	2.67e+03
	Std	3.92e+01	8.24e+00	5.49e+00	8.96e+01	1.07e+00	3.63e+00	9.80e+01	1.02e+02	3.17e+00	5.89e+00	4.91e+01	6.62e+01	4.06e+01	4.73e+01	8.50e+01
f_{11}	Mean	2.77e+03 +	2.63e+03 −	2.66e+03 ≈	2.68e+03 +	2.74e+03 +	**2.61e+03** −	2.97e+03 +	2.62e+03 −	2.61e+03 −	2.63e+03 −	2.68e+03 +	2.65e+03 ≈	2.64e+03 −	2.63e+03 −	2.66e+03
	Std	7.01e+01	2.07e+01	1.52e+02	1.70e+02	3.01e+02	2.38e+00	2.40e+02	2.86e+01	2.74e+00	3.47e+00	4.06e+01	1.65e+02	3.05e+01	1.57e+02	2.18e+02
f_{12}	Mean	2.98e+03 +	2.89e+03 +	2.87e+03 ≈	2.97e+03 +	2.87e+03 ≈	2.87e+03 ≈	2.88e+03 +	2.91e+03 +	2.89e+03 +	2.87e+03 ≈	3.08e+03 +	2.88e+03 +	2.88e+03 +	**2.87e+03** −	2.87e+03
	Std	3.59e+01	1.03e+01	6.64e−01	5.70e+01	5.35e−01	1.04e+00	8.92e+00	5.12e+01	4.23e+00	6.69e−01	6.86e+01	1.51e+01	1.85e+01	1.48e+00	4.91e+00
+/≈/−		11/0/1	9/1/2	9/2/1	10/2/0	9/1/2	7/2/3	12/0/0	10/1/1	8/1/3	9/2/1	10/2/0	9/3/0	6/3/3	8/0/4	–

f_1: unimodal function; $f_2 - f_5$: basic functions; $f_6 - f_8$: hybrid functions; $f_9 - f_{12}$: composition functions

Table 10 The experimental results and statistical analyses on 20-D CEC2022 benchmark functions

Func.		GA	PSO	DE	ES	CMA-ES	L-SHADE	MFO	WOA	SCA	CCDE	HCO	SAO	BOA	ECMADE	ES-HHA
f_1	Mean	3.23e+04 +	1.36e+04 +	3.38e+04 +	6.92e+03 +	2.55e+03 +	2.55e+03 +	6.15e+04 +	6.83e+03 +	4.51e+03 +	2.32e+04 +	6.76e+04 +	9.05e+03 +	1.43e+04 +	1.70e+03 +	**3.00e+02**
	Std	5.14e+03	5.98e+03	5.79e+03	2.92e+03	4.33e+02	5.75e+02	1.06e+04	2.91e+03	1.14e+03	3.44e+03	1.23e+04	3.79e+03	3.72e+03	3.93e+02	1.02e−01
f_2	Mean	2.28e+03 +	7.14e+02 +	1.05e+03 +	5.95e+02 +	5.10e+02 +	4.59e+02 +	2.96e+03 +	6.03e+02 +	5.51e+02 +	6.66e+02 +	1.54e+03 +	6.18e+02 +	1.37e+03 +	4.48e+02 +	**4.40e+02**
	Std	4.26e+02	1.99e+02	1.49e+02	7.65e+01	1.54e+01	4.64e+00	9.74e+02	7.70e+01	2.68e+01	5.53e+01	3.81e+02	1.11e+02	3.28e+02	1.63e+00	2.37e+01
f_3	Mean	6.01e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.01e+02 +	6.00e+02 +	6.00e+02 +	6.00e+02 +	6.01e+02 +	6.00e+02 +	6.01e+02 +	6.00e+02 +	**6.00e+02**
	Std	1.65e−01	6.54e−02	5.53e−02	8.69e−02	7.24e−03	4.64e−05	2.73e−01	3.42e−02	1.05e−02	1.59e−02	1.55e−01	1.29e−01	2.62e−01	2.16e−05	3.88e−06
f_4	Mean	8.04e+02 +	8.04e+02 +	8.04e+02 +	8.01e+02 ≈	8.04e+02 +	8.02e+02 +	8.04e+02 +	8.01e+02 +	8.02e+02 +	8.03e+02 +	8.03e+02 +	8.01e+02 +	8.02e+02 +	8.04e+02 +	**8.01e+02**
	Std	5.02e−01	4.04e−01	4.06e−01	3.92e−01	3.57e−01	2.40e−01	6.76e−01	6.23e−01	3.09e−01	3.17e−01	8.06e−01	4.75e−01	3.44e−01	3.83e−01	3.14e−01
f_5	Mean	9.14e+02 +	9.05e+02 ≈	9.16e+02 +	9.08e+02 +	9.03e+02 −	9.02e+02 −	9.23e+02 +	9.08e+02 +	9.01e+02 −	9.09e+02 +	9.03e+02 −	9.02e+02 −	9.03e+02 −	**9.00e+02** −	9.06e+02
	Std	2.86e+00	3.33e+00	2.30e+00	2.31e+00	4.79e−01	6.32e−01	4.54e+00	2.87e+00	2.61e−01	2.03e+00	1.40e+00	7.73e−01	8.26e−01	1.11e−01	2.79e+00
f_6	Mean	1.55e+09 +	2.71e+08 +	2.22e+08 +	5.22e+05 +	5.28e+06 +	3.20e+06 +	7.55e+08 +	3.08e+05 +	1.01e+07 +	1.06e+08 +	6.45e+09 +	3.26e+06 +	3.09e+08 +	1.03e+07 +	**2.62e+04**
	Std	5.38e+08	2.11e+08	8.54e+07	1.49e+05	2.08e+06	1.50e+06	8.11e+08	8.82e+05	5.35e+06	3.40e+07	2.29e+09	1.36e+07	3.07e+08	6.79e+06	1.22e+04
f_7	Mean	4.33e+03 +	2.96e+03 +	2.82e+03 +	2.72e+03 +	2.29e+03 +	2.14e+03 +	4.39e+03 +	3.30e+03 +	2.23e+03 +	2.51e+03 +	3.17e+03 +	2.28e+03 +	2.60e+03 +	2.19e+03 +	**2.10e+03**
	Std	6.83e+02	4.13e+02	2.06e+02	3.73e+02	7.37e+01	3.53e+01	8.57e+02	8.20e+02	5.89e+01	1.29e+02	4.99e+02	2.09e+02	3.76e+02	8.65e+01	1.39e+02
f_8	Mean	8.54e+08 +	4.65e+07 +	1.77e+05 +	4.99e+03 +	2.41e+03 +	2.85e+03 +	1.32e+10 +	5.68e+03 +	3.81e+03 +	9.62e+03 +	1.60e+13 +	4.73e+03 +	3.36e+03 +	4.44e+03 +	**2.23e+03**
	Std	1.40e+09	1.94e+08	3.47e+05	1.62e+03	5.48e+01	2.35e+02	2.58e+10	1.90e+03	8.06e+02	3.88e+03	3.16e+13	1.31e+03	7.12e+02	9.32e+02	1.03e+00
f_9	Mean	4.09e+03 +	3.18e+03 +	2.81e+03 +	2.98e+03 +	2.65e+03 +	2.67e+03 +	3.42e+03 +	2.87e+03 +	2.84e+03 +	2.75e+03 +	5.37e+03 +	2.84e+03 +	3.55e+03 +	2.64e+03 +	**2.62e+03**
	Std	3.50e+02	2.21e+02	4.12e+01	1.91e+02	3.92e+00	9.14e+00	3.53e+02	1.88e+02	4.47e+01	2.33e+01	8.40e+02	1.12e+02	4.44e+02	6.20e−01	6.02e+01
f_{10}	Mean	3.72e+03 −	2.84e+03 −	4.02e+03 ≈	4.20e+03 ≈	3.14e+03 −	**2.82e+03** −	5.12e+03 +	4.67e+03 ≈	2.83e+03 −	3.32e+03 −	4.92e+03 +	4.41e+03 ≈	3.08e+03 −	3.47e+03 −	4.39e+03
	Std	7.69e+02	3.90e+01	1.65e+03	7.82e+02	9.33e+02	3.01e+01	1.36e+03	1.16e+03	7.99e+01	9.72e+02	1.76e+03	1.28e+03	5.46e+02	1.45e+03	1.06e+03
f_{11}	Mean	4.86e+03 +	2.73e+03 +	2.81e+03 +	2.92e+03 +	2.62e+03 +	2.60e+03 +	4.67e+03 +	2.94e+03 +	2.63e+03 +	2.66e+03 +	6.40e+03 +	3.00e+03 +	4.03e+03 +	2.61e+03 +	**2.60e+03**
	Std	9.30e+02	1.55e+02	3.13e+02	5.32e+02	3.11e+00	1.40e+00	4.06e+02	5.39e+02	4.93e+00	1.14e+01	2.10e+03	6.21e+02	8.45e+02	3.24e+01	2.48e+00
f_{12}	Mean	3.49e+03 +	3.23e+03 +	2.96e+03 +	3.35e+03 +	2.96e+03 +	2.96e+03 +	3.06e+03 +	3.21e+03 +	3.13e+03 +	2.96e+03 +	3.75e+03 +	3.06e+03 ≈	3.08e+03 +	**2.95e+03** −	3.03e+03
	Std	8.72e+01	6.97e+01	4.04e+00	1.34e+02	7.64e+00	2.95e+00	5.32e+01	1.30e+02	2.70e+01	4.63e+00	1.72e+02	8.40e+01	6.03e+01	2.07e+01	4.13e+01
+/≈/−		11/0/1	10/1/1	11/1/0	10/2/0	10/0/2	10/0/2	12/0/0	11/1/0	10/0/2	11/0/1	11/0/1	9/2/1	10/0/2	9/0/3	–

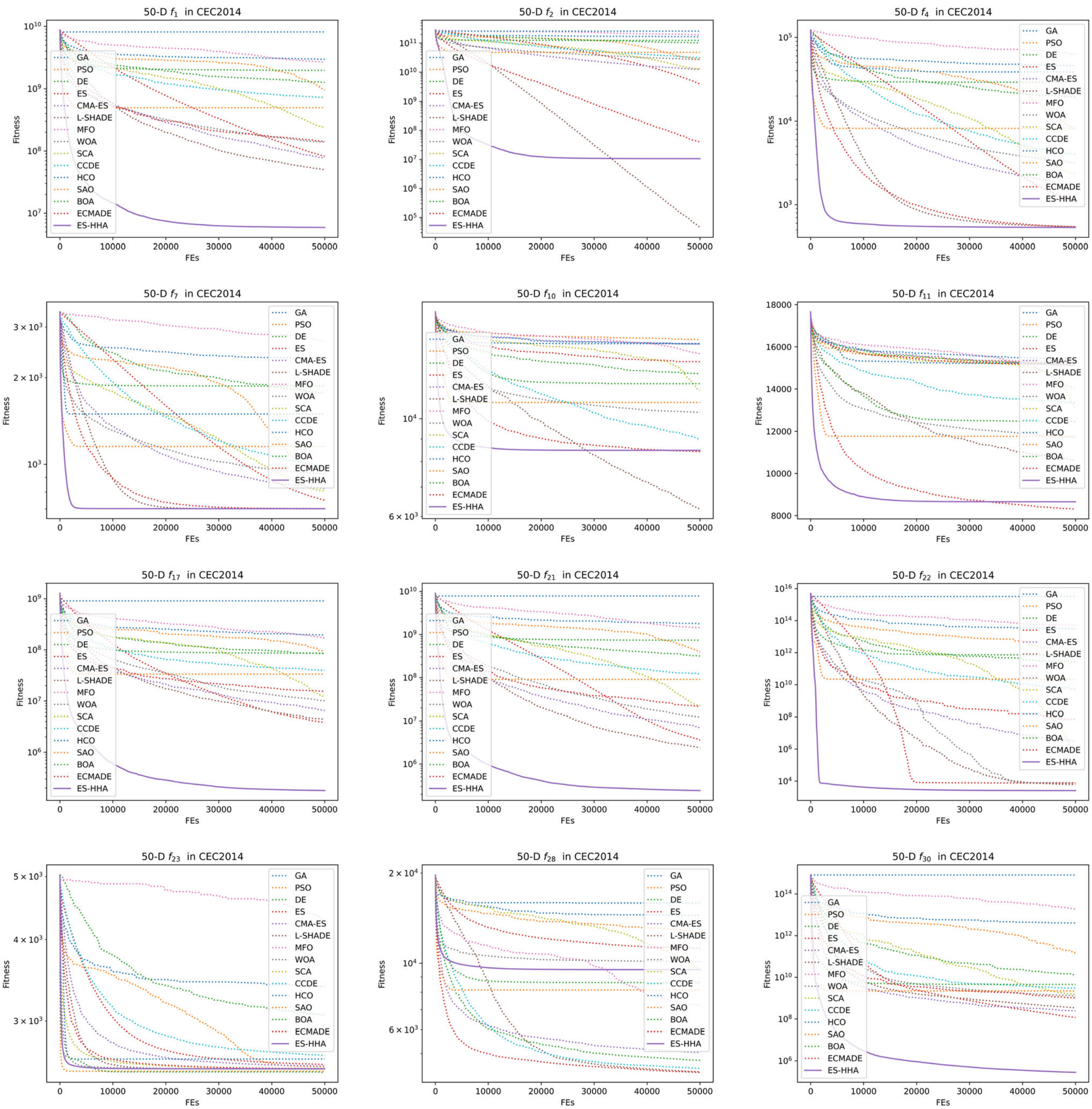


Fig. 6 Convergence curves on representative 50-D CEC2014 benchmark functions

However, there are still many aspects that can improve the ES-HHA. Here, we list some open topics for future research.

5.2.1 Various search operator ensemble

In the LLHs design of ES-HHA, we employ the most common search operators as LLHs: variants of mutation strategies in DE and local search operators, and the

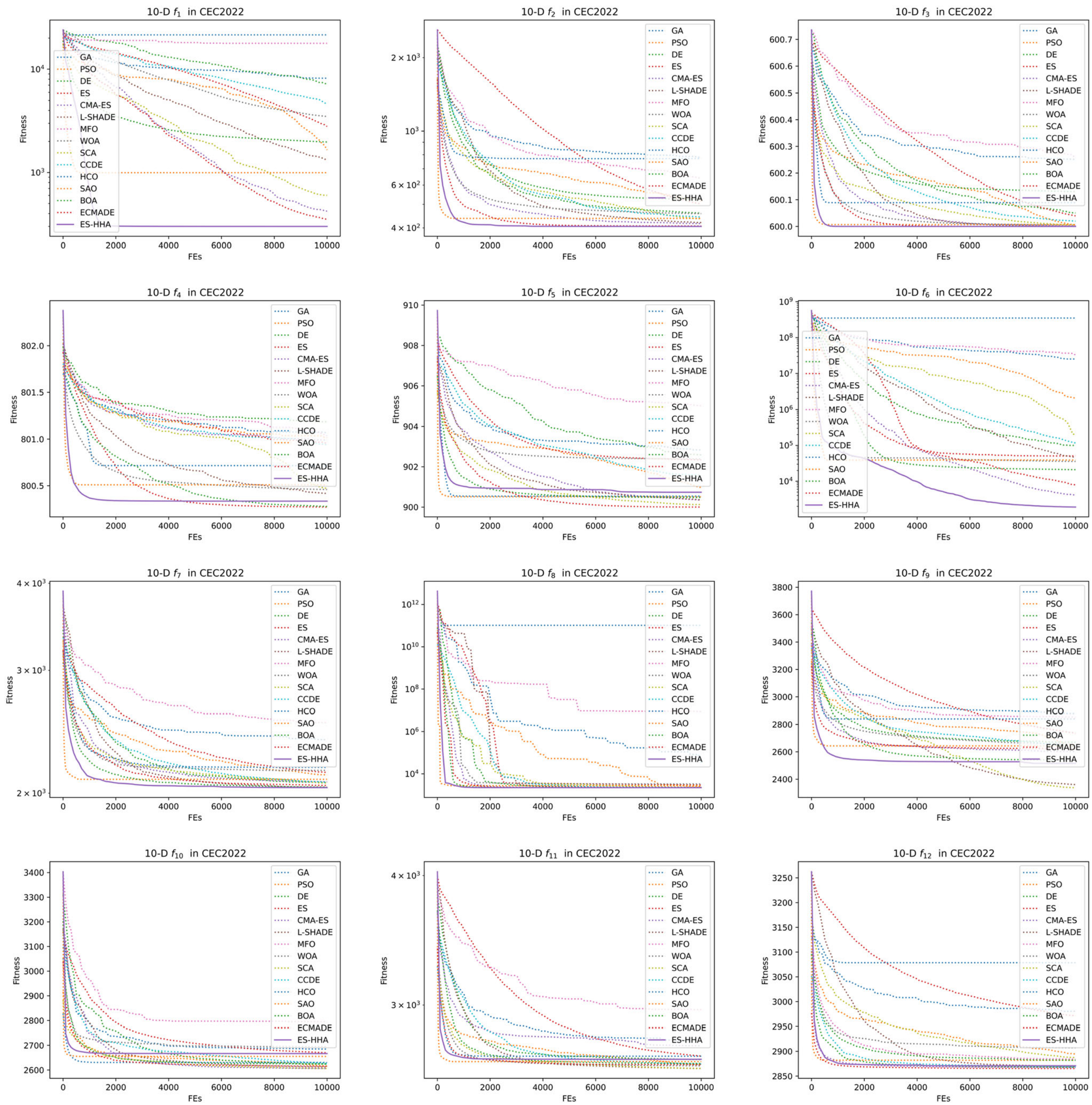


Fig. 7 Convergence curves on 10-D CEC2022 benchmark functions

combination of these simple search operators has achieved competitive performance in our numerical experiments. However, the NFLT similarly limits the robustness and scalability of these strategies, and it is necessary to introduce various search operators as LLHs. Jorge et al. [16] preliminarily concluded some efficient search operators from ten well-known meta-heuristics in the literature such as firefly dynamic, spiral dynamic, gravitational search, and so on. In our future research, we will introduce these unique strategies to our proposal, which enhance the diversity of search operators and further strengthen the robustness of our proposal.

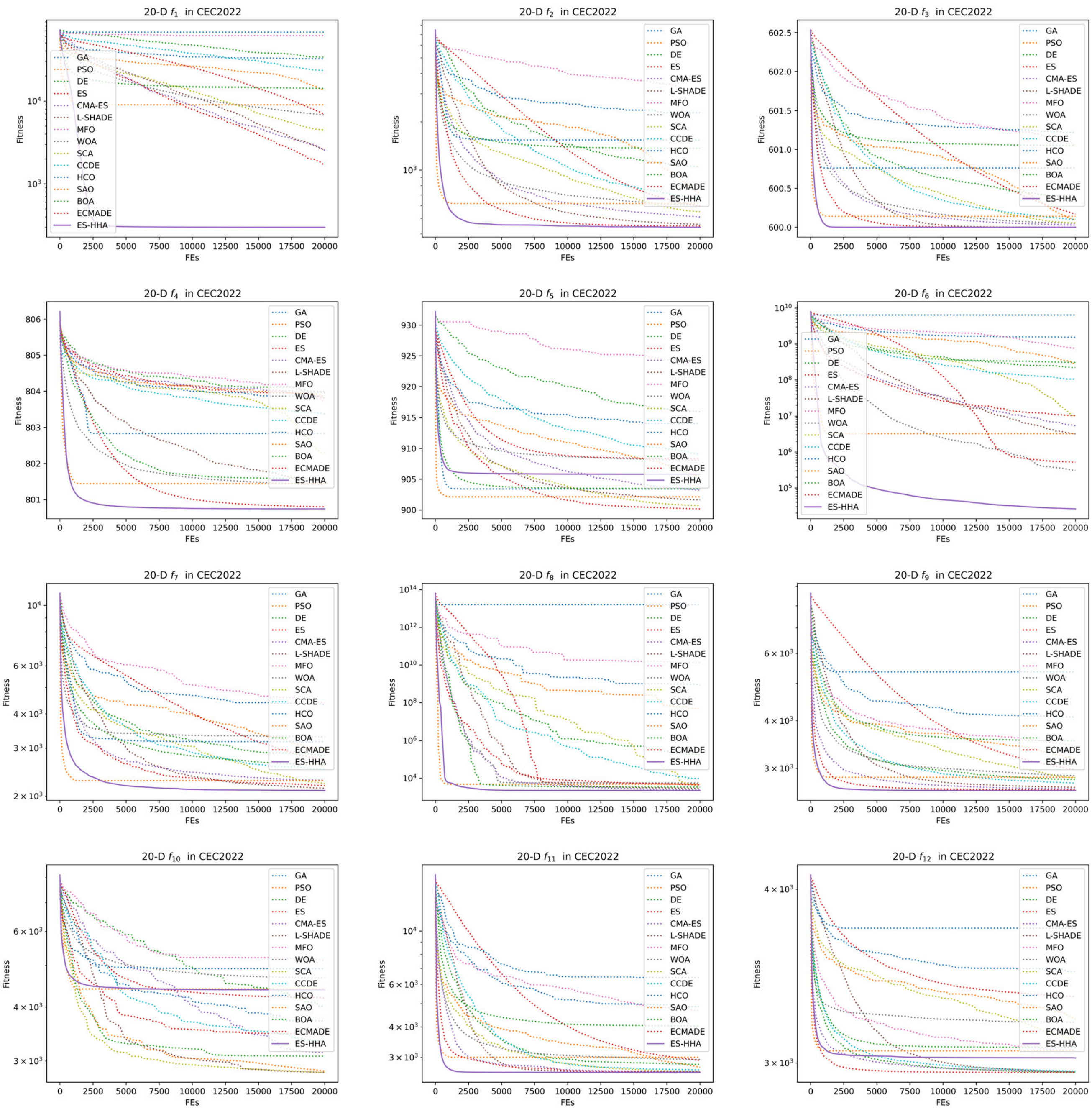


Fig. 8 Convergence curves on 20-D CEC2022 benchmark functions

5.2.2 Machine learning for the high-level component design

We design the high-level component of ES-HHA based on the evolutionary status involving two metrics: the fitness distance correlation (FDC) and the population diversity (PD). The ablation experiments with three baseline hyper-heuristic frameworks practically prove the effectiveness of the cooperation of these two metrics. Furthermore, many machine learning techniques are promising to be introduced to act as the high-level component of hyper-heuristics, and one representative methodology is reinforcement learning. Reinforcement learning is a machine learning method that attempts to find the optimal action sequence by

Table 11 The experimental results and statistical analyses on engineering optimization problems

Prob.		GA	PSO	DE	ES	CMA-ES	L-SHADE	MFO	WOA	SCA	CCDE	HCO	SAO	BOA	ECMADE	ES-HHA
WBP	Mean	2.437e+00	1.978e+00	1.691e+00	2.173e+00	1.683e+00	1.913e+00	3.329e+00	3.102e+00	1.730e+00	1.741e+00	3.847e+00	1.948e+00	1.683e+00	**1.683e+00**	1.683e+00
	Std	4.344e−01	1.023e−01	2.850e−03	1.874e−01	2.735e−04	9.804e−02	5.498e−01	7.688e−01	1.933e−02	2.461e−02	6.909e−01	3.299e−01	1.537e−04	2.410e−05	3.406e−05
	Best	1.782e+00	1.960e+00	1.691e+00	2.225e+00	1.683e+00	1.975e+00	3.034e+00	2.544e+00	1.729e+00	1.755e+00	4.480e+00	2.060e+00	1.683e+00	1.683e+00	1.683e+00
	Worst	3.380e+00	4.047e+00	4.232e+00	4.480e+00	4.480e+00	4.422e+00	4.480e+00	3.056e+00	4.096e+00	4.480e+00	4.480e+00	5.174e+00	3.585e+00	4.480e+00	2.121e+00
SRD	Mean	2.991e+03	3.138e+03	2.987e+03	3.081e+03	2.987e+03	2.987e+03	3.095e+03	4.152e+03	3.196e+03	2.987e+03	1.866e+06	2.999e+03	2.989e+03	2.987e+03	**2.987e+03**
	Std	2.911e+00	5.258e+01	1.476e−01	2.483e+01	8.055e−02	5.929e−04	3.361e+01	9.313e+02	6.322e+01	1.479e−01	1.468e+06	9.112e+00	3.160e+00	5.744e−05	2.317e−02
	Best	2.989e+03	3.107e+03	2.987e+03	3.054e+03	2.987e+03	2.987e+03	3.159e+03	3.151e+03	3.179e+03	2.987e+03	1.043e+06	3.005e+03	2.987e+03	2.987e+03	2.987e+03
	Worst	4.822e+03	1.694e+06	4.135e+03	7.018e+05	1.924e+05	1.776e+06	1.350e+06	5.167e+03	1.274e+06	9.268e+05	1.043e+06	2.261e+06	2.045e+06	5.196e+03	3.107e+06
TBTD	Mean	2.648e+02	2.641e+02	2.639e+02	2.643e+02	2.639e+02	2.639e+02	2.653e+02	2.669e+02	2.639e+02	2.639e+02	2.685e+02	2.642e+02	2.639e+02	2.639e+02	**2.639e+02**
	Std	1.057e+00	1.288e−01	4.073e−11	3.781e−01	1.524e−12	7.192e−07	1.175e+00	3.312e+00	2.332e−02	8.343e−06	3.542e+00	1.108e+00	4.524e−14	2.076e−14	3.282e−14
	Best	2.639e+02	2.640e+02	2.639e+02	2.642e+02	2.639e+02	2.639e+02	2.644e+02	2.732e+02	2.640e+02	2.639e+02	2.709e+02	2.639e+02	2.639e+02	2.639e+02	2.639e+02
	Worst	2.722e+02	2.684e+02	2.757e+02	2.663e+02	2.721e+02	2.682e+02	2.669e+02	2.743e+02	2.689e+02	2.757e+02	2.709e+02	2.674e+02	2.657e+02	2.699e+02	2.757e+02
GTD	Mean	5.837e−10	3.690e−09	1.718e−11	1.157e−12	1.943e−11	7.687e−13	3.697e−10	**0.000e+00**	4.815e−11	9.940e−11	1.242e−05	**0.000e+00**	1.095e−18	1.619e−12	5.971e−14
	Std	1.200e−09	8.083e−09	2.288e−11	1.653e−12	6.041e−11	1.643e−12	1.073e−09	0.000e+00	1.110e−10	1.387e−10	6.679e−05	0.000e+00	2.199e−18	5.507e−12	1.902e−13
	Best	1.108e−11	1.263e−10	6.477e−12	1.359e−12	5.389e−16	1.856e−13	5.431e−09	0.000e+00	2.452e−10	1.319e−11	0.000e+00	0.000e+00	5.006e−18	1.193e−19	9.418e−16
	Worst	1.266e−06	5.622e−09	6.518e−06	9.256e−05	1.645e−04	9.385e−06	1.616e−04	2.797e−07	4.431e−06	7.623e−05	1.745e−05	7.935e−07	7.783e−05	1.330e−07	9.955e−08
CBD	Mean	1.466e+00	2.103e+00	1.382e+00	1.351e+00	1.342e+00	1.340e+00	6.099e+00	4.070e+00	1.345e+00	1.365e+00	5.682e+00	1.342e+00	1.341e+00	1.340e+00	**1.340e+00**
	Std	8.818e−02	2.892e−01	1.790e−02	3.621e−03	1.057e−03	1.591e−04	1.371e+00	1.497e+00	2.185e−03	1.104e−02	1.353e+00	1.786e−03	2.488e−03	1.614e−06	1.461e−07
	Best	1.562e+00	1.635e+00	1.354e+00	1.348e+00	1.342e+00	1.340e+00	6.279e+00	4.839e+00	1.349e+00	1.361e+00	6.563e+00	1.342e+00	1.340e+00	1.340e+00	1.340e+00
	Worst	5.102e+00	5.271e+00	7.932e+00	5.271e+00	7.882e+00	6.747e+00	7.932e+00	5.414e+00	5.069e+00	7.932e+00	6.563e+00	6.452e+00	7.652e+00	6.291e+00	3.290e+00
TCD	Mean	3.078e+01	3.029e+01	3.015e+01	3.038e+01	3.015e+01	3.015e+01	3.111e+01	3.124e+01	3.018e+01	3.015e+01	3.211e+01	3.015e+01	**3.015e+01**	3.015e+01	3.015e+01
	Std	5.768e−01	7.617e−02	7.147e−09	1.169e−01	2.079e−09	3.273e−06	4.985e−01	1.056e+00	1.812e−02	2.015e−05	1.355e+00	1.056e−02	1.066e−14	1.157e−14	4.889e−11
	Best	3.111e+01	3.024e+01	3.015e+01	3.033e+01	3.015e+01	3.015e+01	3.117e+01	3.019e+01	3.021e+01	3.015e+01	3.204e+01	3.015e+01	3.015e+01	3.015e+01	3.015e+01
	Worst	3.216e+01	3.249e+01	3.585e+01	3.179e+01	3.189e+01	3.108e+01	3.360e+01	3.116e+01	3.347e+01	3.381e+01	3.216e+01	3.197e+01	3.562e+01	3.255e+01	3.424e+01
CBHD	Mean	8.467e+00	7.360e+00	6.848e+00	7.080e+00	6.844e+00	6.852e+00	9.335e+00	7.289e+00	6.922e+00	6.865e+00	1.014e+01	6.939e+00	6.844e+00	**6.843e+00**	6.843e+00
	Std	1.027e+00	2.349e−01	2.264e−03	1.870e−01	3.344e−04	6.397e−03	7.434e−01	3.598e−01	2.733e−02	6.419e−03	1.033e+00	1.530e−01	1.700e−03	4.717e−06	2.450e−05
	Best	1.015e+01	7.478e+00	6.845e+00	7.004e+00	6.844e+00	6.854e+00	8.852e+00	7.043e+00	6.926e+00	6.867e+00	1.097e+01	7.027e+00	6.843e+00	6.843e+00	6.843e+00
	Worst	1.046e+01	8.590e+00	1.097e+01	1.115e+01	1.097e+01	8.373e+00	8.852e+00	8.809e+00	9.631e+00	1.097e+01	1.097e+01	1.543e+01	1.059e+01	1.097e+01	1.050e+01
RCB	Mean	1.641e+02	1.617e+02	1.666e+02	**1.595e+02**	1.666e+02	1.668e+02	1.684e+02	1.683e+02	1.599e+02	1.666e+02	1.743e+02	1.673e+02	1.666e+02	1.664e+02	**1.601e+02**
	Std	2.081e+00	2.090e+00	6.055e−01	1.767e−01	6.055e−01	2.588e−08	2.189e+00	2.654e+00	4.878e−01	6.457e−01	6.849e+00	1.928e+00	6.055e−01	1.009e+00	5.942e−01
	Best	1.640e+02	1.616e+02	1.668e+02	1.594e+02	1.668e+02	1.668e+02	1.683e+02	1.668e+02	1.594e+02	1.668e+02	1.809e+02	1.668e+02	1.668e+02	1.667e+02	1.600e+02
	Worst	1.811e+02	1.712e+02	1.845e+02	1.780e+02	1.845e+02	1.711e+02	1.694e+02	1.705e+02	1.760e+02	1.814e+02	1.809e+02	1.830e+02	1.698e+02	1.817e+02	1.794e+02

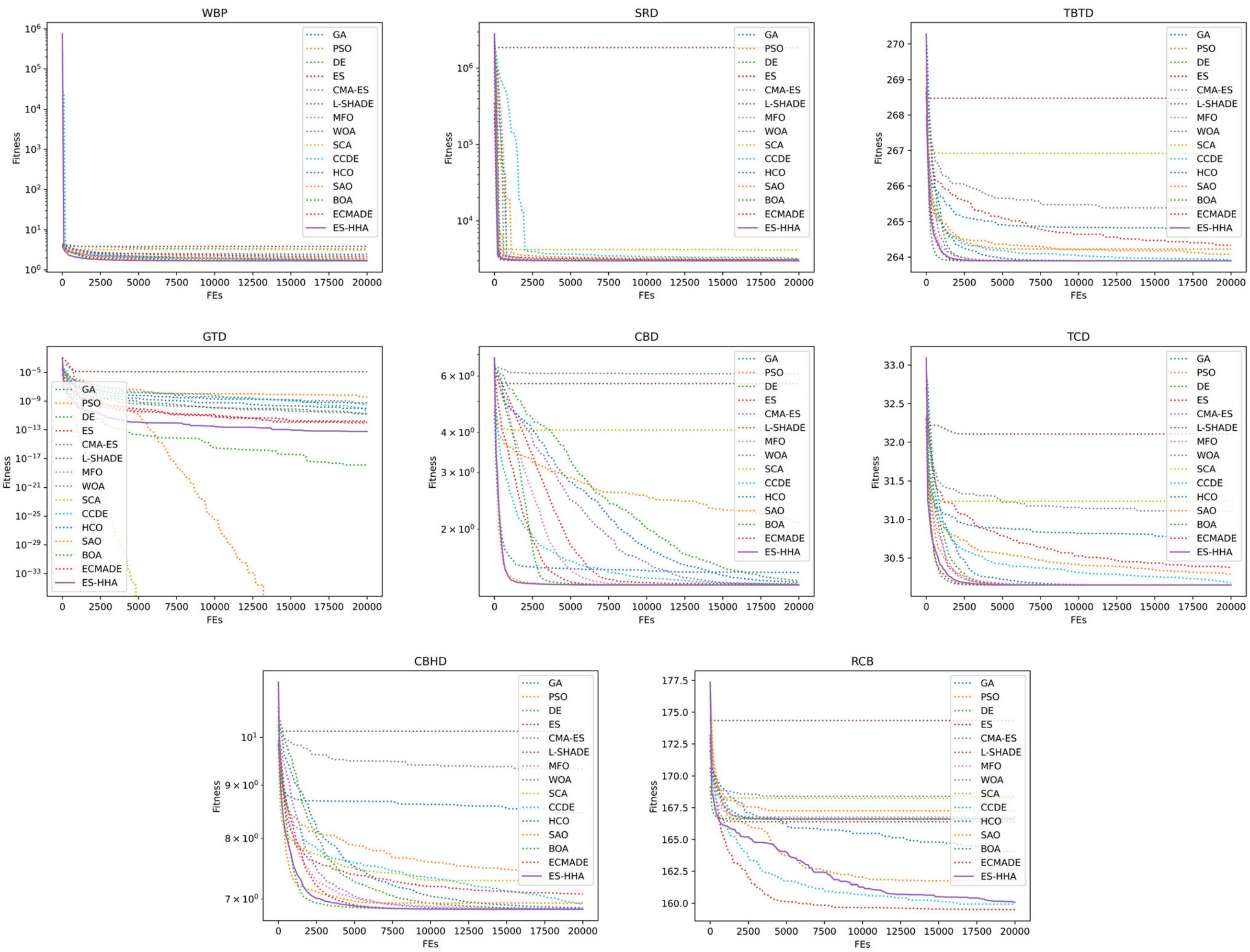


Fig. 9 Convergence curves on engineering optimization problems

maximizing the cumulative reward, and the success of the implementation of reinforcement learning has been reported by some published works [9, 14, 74]. The key to employing reinforcement learning in hyper-heuristic algorithms is the design of state space and reward function in optimization space, and in the future, we will try to develop a proper reward function for reinforcement learning in hyper-heuristic algorithms.

5.2.3 Extending to other applications

The application of our proposed ES-HHA is also a potential topic of future research. The first branch is applying our proposal to solve complex real-world optimization problems such as dispatch optimization of cascade hydro-power stations [75], heavy-duty gas turbine operation optimization [76], wireless sensor network deployment optimization

Table 12 The experimental results and statistical analyses on 30-D CEC2014 benchmark functions with three baseline hyper-heuristic frameworks

Func	SR		RD		RP		ES-HHA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	1.84e+06 +	1.10e+06	1.25e+06 ≈	8.74e+05	2.30e+06 +	1.22e+06	**1.17e+06**	6.10e+05
f_2	2.95e+06 +	1.16e+06	1.72e+06 ≈	6.69e+05	2.27e+06 +	8.61e+05	**1.51e+06**	6.27e+05
f_3	5.04e+03 ≈	2.84e+03	**3.35e+03** ≈	2.08e+03	3.71e+03 ≈	1.71e+03	3.79e+03	1.75e+03
f_4	5.40e+02 +	3.67e+01	5.15e+02 +	2.82e+01	5.27e+02 +	2.70e+01	**4.80e+02**	3.15e+01
f_5	**5.21e+02** ≈	5.18e−02	5.21e+02 ≈	5.34e−02	5.21e+02 ≈	5.67e−02	5.21e+02	4.40e−02
f_6	6.30e+02 +	4.36e+00	6.32e+02 +	3.58e+00	6.30e+02 ≈	4.33e+00	**6.27e+02**	4.03e+00
f_7	7.01e+02 +	9.59e−02	7.01e+02 ≈	1.29e−01	7.01e+02 +	7.19e−02	**7.01e+02**	1.51e−01
f_8	**9.73e+02** ≈	3.78e+01	9.99e+02 ≈	4.17e+01	9.74e+02 ≈	4.69e+01	9.73e+02	3.29e+01
f_9	1.16e+03 ≈	6.75e+01	1.18e+03 +	5.48e+01	1.15e+03 ≈	6.05e+01	**1.14e+03**	5.39e+01
f_{10}	5.02e+03 ≈	5.97e+02	5.58e+03 ≈	5.58e+02	**4.93e+03** ≈	9.08e+02	5.27e+03	7.10e+02
f_{11}	5.55e+03 ≈	6.36e+02	5.62e+03 ≈	6.18e+02	**5.38e+03** ≈	5.62e+02	5.42e+03	6.45e+02
f_{12}	1.20e+03 +	6.22e−02	1.20e+03 +	5.56e−02	1.20e+03 +	5.61e−02	**1.20e+03**	4.24e−02
f_{13}	1.30e+03 ≈	1.49e−01	1.30e+03 ≈	1.12e−01	1.30e+03 ≈	1.58e−01	**1.30e+03**	1.36e−01
f_{14}	**1.40e+03** ≈	1.82e−01	1.40e+03 ≈	1.90e−01	1.40e+03 ≈	1.85e−01	1.40e+03	1.97e−01
f_{15}	1.52e+03 +	3.09e+00	1.52e+03 +	2.66e+00	1.52e+03 +	3.82e+00	**1.52e+03**	2.01e+00
f_{16}	1.61e+03 +	4.33e−01	1.61e+03 ≈	5.34e−01	1.61e+03 +	5.56e−01	**1.61e+03**	5.23e−01
f_{17}	4.61e+04 ≈	2.39e+04	4.03e+04 ≈	1.97e+04	4.26e+04 ≈	1.96e+04	**3.39e+04**	1.09e+04
f_{18}	6.87e+04 ≈	2.29e+04	**6.13e+04** ≈	1.97e+04	7.52e+04 ≈	2.07e+04	6.72e+04	2.19e+04
f_{19}	1.97e+03 ≈	1.52e+02	**1.93e+03** ≈	1.20e+01	1.94e+03 ≈	2.95e+01	1.93e+03	2.22e+01
f_{20}	3.78e+04 ≈	1.73e+04	3.83e+04 ≈	2.05e+04	4.28e+04 ≈	1.87e+04	**3.24e+04**	1.39e+04
f_{21}	3.06e+04 ≈	1.55e+04	3.61e+04 ≈	1.87e+04	3.23e+04 ≈	1.73e+04	**2.78e+04**	1.82e+04
f_{22}	2.60e+03 ≈	2.51e+02	2.56e+03 ≈	2.11e+02	2.60e+03 ≈	2.44e+02	**2.53e+03**	2.15e+02
f_{23}	2.53e+03 +	5.56e+00	2.53e+03 +	2.70e+00	2.53e+03 +	5.69e+00	**2.52e+03**	1.50e−01
f_{24}	2.65e+03 +	1.17e+01	2.66e+03 +	1.56e+01	2.65e+03 +	1.03e+01	**2.64e+03**	1.07e+01
f_{25}	2.73e+03 ≈	1.59e+01	2.73e+03 ≈	1.58e+01	**2.72e+03** ≈	1.25e+01	2.73e+03	1.28e+01
f_{26}	2.73e+03 ≈	4.68e+01	2.74e+03 ≈	4.86e+01	**2.73e+03** ≈	4.38e+01	2.73e+03	4.68e+01
f_{27}	**3.67e+03** ≈	2.99e+02	3.79e+03 +	2.84e+02	3.73e+03 ≈	2.36e+02	3.71e+03	2.18e+02
f_{28}	5.79e+03 ≈	9.73e+02	6.44e+03 ≈	9.55e+02	**5.57e+03** ≈	8.43e+02	5.88e+03	1.01e+03
f_{29}	3.28e+05 +	1.13e+06	1.12e+06 +	3.40e+06	1.12e+06 +	4.13e+06	**1.05e+04**	7.07e+03
f_{30}	2.54e+06 +	6.30e+06	1.05e+06 +	2.79e+06	1.60e+06 +	1.97e+06	**2.73e+04**	8.09e+03
+/≈/−	12/18/0		10/20/0		11/19/0			

[77], and so on. The second branch is that we want to develop a binary ES-HHA to solve combinatorial optimization problems such as the feature selection task and the evolutionary neural architecture search problem. We will focus on extending our proposal to various optimization problems.

6 Conclusion

In this paper, we propose a novel evolutionary status guided hyper-heuristic algorithm named ES-HHA. In the low-level component, we design an exploitative operator pool

Table 13 The experimental results and statistical analyses on 50-D CEC2014 benchmark functions with three baseline hyper-heuristic frameworks

Func	SR		RD		RP		ES-HHA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	9.93e+06 +	5.35e+06	6.22e+06 ≈	1.59e+06	9.72e+06 +	3.92e+06	**5.90e+06**	1.63e+06
f_2	1.89e+07 +	5.44e+06	**1.00e+07** ≈	2.73e+06	2.54e+07 +	1.38e+07	1.07e+07	2.66e+06
f_3	1.23e+04 +	3.94e+03	1.44e+04 +	5.57e+03	1.17e+04 +	3.98e+03	4.64e+03	1.63e+03
f_4	6.46e+02 +	7.52e+01	5.72e+02 +	4.46e+01	6.31e+02 +	7.35e+01	**5.34e+02**	4.53e+01
f_5	5.21e+02 ≈	3.64e−02	5.21e+02 ≈	3.82e−02	**5.21e+02** ≈	4.01e−02	5.21e+02	3.14e−02
f_6	**6.56e+02** ≈	4.62e+00	6.62e+02 +	3.57e+00	6.57e+02 ≈	5.02e+00	6.57e+02	5.39e+00
f_7	7.01e+02 +	4.85e−02	**7.01e+02** ≈	2.10e−02	7.01e+02 +	9.60e−02	7.01e+02	2.90e−02
f_8	**1.16e+03** ≈	7.77e+01	1.24e+03 ≈	8.03e+01	1.18e+03 ≈	8.23e+01	1.19e+03	6.56e+01
f_9	**1.37e+03** ≈	1.18e+02	1.48e+03 +	1.15e+02	1.44e+03 ≈	1.71e+02	1.39e+03	8.10e+01
f_{10}	8.49e+03 ≈	9.75e+02	8.91e+03 ≈	9.49e+02	**8.42e+03** ≈	1.09e+03	8.48e+03	9.41e+02
f_{11}	8.79e+03 ≈	7.27e+02	9.14e+03 ≈	7.92e+02	8.68e+03 ≈	9.31e+02	**8.66e+03**	9.25e+02
f_{12}	1.20e+03 +	6.97e−02	1.20e+03 ≈	8.07e−02	1.20e+03 +	9.93e−02	**1.20e+03**	5.93e−02
f_{13}	1.30e+03 ≈	1.10e−01	**1.30e+03** ≈	1.18e−01	1.30e+03 ≈	1.37e−01	1.30e+03	1.21e−01
f_{14}	1.40e+03 ≈	2.58e−01	1.40e+03 ≈	2.63e−01	1.40e+03 ≈	2.40e−01	**1.40e+03**	2.53e−01
f_{15}	1.55e+03 +	7.05e+00	1.55e+03 +	7.36e+00	1.55e+03 +	6.41e+00	**1.54e+03**	3.08e+00
f_{16}	1.62e+03 +	5.88e−01	1.62e+03 +	6.61e−01	1.62e+03 +	3.59e−01	**1.62e+03**	4.86e−01
f_{17}	2.28e+05 ≈	9.32e+04	1.99e+05 ≈	7.64e+04	2.27e+05 ≈	7.77e+04	**1.78e+05**	6.66e+04
f_{18}	1.82e+05 +	3.05e+04	1.78e+05 ≈	4.04e+04	1.90e+05 +	3.48e+04	**1.57e+05**	3.44e+04
f_{19}	2.69e+03 +	1.71e+03	2.31e+03 ≈	1.41e+03	**2.09e+03** ≈	3.29e+02	2.11e+03	6.47e+02
f_{20}	1.55e+05 +	4.79e+04	1.49e+05 +	3.93e+04	1.65e+05 +	3.54e+04	**1.16e+05**	2.21e+04
f_{21}	2.37e+05 ≈	1.08e+05	**2.35e+05** ≈	1.08e+05	3.21e+05 ≈	2.06e+05	2.43e+05	1.26e+05
f_{22}	2.82e+03 +	3.23e+02	2.64e+03 ≈	2.88e+02	2.94e+03 +	3.25e+02	**2.58e+03**	2.67e+02
f_{23}	2.54e+03 +	3.51e+00	2.54e+03 +	8.19e−01	2.54e+03 +	2.17e+00	**2.54e+03**	1.37e−01
f_{24}	2.73e+03 +	2.05e+01	2.76e+03 +	2.78e+01	2.73e+03 +	1.55e+01	**2.72e+03**	2.29e+01
f_{25}	2.75e+03 ≈	2.89e+01	2.77e+03 ≈	4.18e+01	2.76e+03 ≈	3.22e+01	**2.75e+03**	3.54e+01
f_{26}	2.81e+03 ≈	1.32e+02	2.81e+03 +	1.02e+02	2.83e+03 ≈	1.39e+02	**2.77e+03**	7.29e+01
f_{27}	4.56e+03 ≈	1.76e+02	4.74e+03 +	1.29e+02	4.49e+03 ≈	1.65e+02	**4.49e+03**	1.18e+02
f_{28}	**8.95e+03** ≈	1.22e+03	1.06e+04 ≈	1.49e+03	9.38e+03 ≈	1.47e+03	9.52e+03	1.61e+03
f_{29}	1.16e+07 +	2.30e+07	2.93e+06 +	1.29e+07	1.12e+07 +	2.39e+07	**1.27e+05**	9.44e+04
f_{30}	7.79e+06 +	7.76e+06	2.43e+06 +	2.87e+06	7.25e+06 +	8.40e+06	**2.80e+05**	9.12e+04
+/≈/−	16/14/0		13/17/0		15/15/0			

and an explorative operator pool to save LLHs. In the low-level component, we develop an evolutionary status guided probabilistic selection function involving the fitness distance correlation (FDC) and the population diversity (PD) to play a decision-maker role in determining the optimization sequence. To evaluate the performance of our proposed ES-HHA, we conduct comprehensive numerical experiments with 14 famous meta-heuristic algorithms on

Table 14 Results of sensitivity experiments on 30-D CEC2014 benchmark functions (ES-HHA$_1$: $w_1 = 0.1$; ES-HHA$_3$: $w_1 = 0.3$; ES-HHA$_5$: $w_1 = 0.5$; ES-HHA$_7$: $w_1 = 0.7$; ES-HHA$_9$: $w_1 = 0.9$)

Func	ES-HHA$_1$		ES-HHA$_3$		ES-HHA$_5$		ES-HHA$_7$		ES-HHA$_9$	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	**1.10e+06** ≈	5.41e+05	1.18e+06 ≈	6.79e+05	1.17e+06	6.10e+05	1.31e+06 ≈	9.25e+05	1.46e+06 ≈	8.60e+05
f_2	1.38e+06 ≈	5.65e+05	**1.32e+06** ≈	5.72e+05	1.51e+06	6.27e+05	2.25e+06 +	1.07e+06	3.48e+06 +	2.13e+06
f_3	**2.15e+03** −	1.05e+03	2.42e+03 −	1.45e+03	3.79e+03	1.75e+03	4.80e+03 ≈	3.37e+03	1.02e+04 +	5.68e+03
f_4	4.88e+02 ≈	3.89e+01	4.93e+02 ≈	3.80e+01	**4.80e+02**	3.15e+01	5.00e+02 ≈	3.80e+01	4.86e+02 ≈	4.29e+01
f_5	5.21e+02 ≈	5.76e−02	5.21e+02 ≈	3.66e−02	**5.21e+02**	4.40e−02	5.21e+02 ≈	5.67e−02	5.21e+02 ≈	5.81e−02
f_6	6.30e+02 +	3.24e+00	6.28e+02 ≈	4.75e+00	6.27e+02	4.03e+00	6.28e+02 ≈	4.77e+00	**6.25e+02** ≈	3.96e+00
f_7	**7.01e+02** ≈	1.27e−01	7.01e+02 ≈	1.56e−01	7.01e+02	1.51e−01	7.01e+02 ≈	1.75e−01	7.01e+02 +	1.17e−01
f_8	9.96e+02 ≈	3.65e+01	9.89e+02 ≈	3.23e+01	9.73e+02	3.29e+01	9.81e+02 ≈	4.30e+01	**9.72e+02** ≈	3.42e+01
f_9	1.16e+03 ≈	5.57e+01	1.14e+03 ≈	5.34e+01	1.14e+03	5.39e+01	1.12e+03 ≈	5.93e+01	**1.08e+03** −	5.50e+01
f_{10}	5.14e+03 ≈	6.84e+02	**5.05e+03** ≈	5.50e+02	5.27e+03	7.10e+02	5.24e+03 ≈	7.64e+02	5.28e+03 ≈	7.35e+02
f_{11}	5.53e+03 ≈	6.06e+02	5.34e+03 ≈	5.90e+02	5.42e+03	6.45e+02	5.42e+03 ≈	6.32e+02	**5.27e+03** ≈	5.36e+02
f_{12}	1.20e+03 ≈	4.95e−02	1.20e+03 ≈	1.47e−01	**1.20e+03**	4.24e−02	1.20e+03 ≈	4.36e−02	1.20e+03 ≈	5.02e−02
f_{13}	1.30e+03 ≈	1.39e−01	1.30e+03 ≈	1.32e−01	**1.30e+03**	1.36e−01	1.30e+03 ≈	1.58e−01	1.30e+03 ≈	1.52e−01
f_{14}	1.40e+03 ≈	1.97e−01	1.40e+03 ≈	2.19e−01	1.40e+03	1.97e−01	**1.40e+03** ≈	1.87e−01	1.40e+03 ≈	2.53e−01
f_{15}	1.52e+03 ≈	2.18e+00	**1.52e+03** ≈	2.74e+00	1.52e+03	2.01e+00	1.52e+03 ≈	2.25e+00	1.52e+03 ≈	2.65e+00
f_{16}	1.61e+03 ≈	6.37e−01	1.61e+03 ≈	5.23e−01	**1.61e+03**	5.23e−01	1.61e+03 ≈	5.03e−01	1.61e+03 ≈	3.85e−01
f_{17}	3.13e+04 ≈	1.57e+04	3.47e+04 ≈	1.94e+04	3.39e+04	1.09e+04	3.41e+04 ≈	1.79e+04	**2.68e+04** ≈	1.73e+04
f_{18}	7.14e+04 ≈	2.02e+04	7.19e+04 ≈	2.37e+04	6.72e+04	2.19e+04	7.50e+04 ≈	2.59e+04	**5.98e+04** ≈	2.29e+04
f_{19}	**1.93e+03** ≈	9.74e+00	1.95e+03 ≈	6.73e+01	1.93e+03	2.22e+01	1.93e+03 ≈	9.73e+00	1.93e+03 ≈	2.68e+01
f_{20}	**2.73e+04** ≈	9.26e+03	3.22e+04 ≈	1.71e+04	3.24e+04	1.39e+04	4.31e+04 ≈	1.79e+04	4.54e+04 ≈	1.51e+04
f_{21}	3.07e+04 ≈	1.80e+04	**2.56e+04** ≈	1.67e+04	2.78e+04	1.82e+04	2.79e+04 ≈	1.96e+04	2.70e+04 ≈	1.41e+04
f_{22}	2.58e+03 ≈	1.89e+02	2.54e+03 ≈	1.78e+02	**2.53e+03**	2.15e+02	2.55e+03 ≈	2.15e+02	2.55e+03 ≈	1.76e+02
f_{23}	**2.52e+03** −	1.36e−01	2.52e+03 ≈	1.34e−01	2.52e+03	1.50e−01	2.52e+03 ≈	2.29e−01	2.52e+03 +	4.65e−01
f_{24}	2.64e+03 ≈	1.00e+01	2.64e+03 ≈	1.13e+01	2.64e+03	1.07e+01	2.64e+03 ≈	8.56e+00	**2.64e+03** ≈	1.02e+01
f_{25}	2.73e+03 ≈	1.11e+01	2.73e+03 ≈	1.12e+01	2.73e+03	1.28e+01	2.72e+03 ≈	7.89e+00	**2.72e+03** ≈	1.03e+01
f_{26}	**2.72e+03** ≈	3.97e+01	2.72e+03 ≈	3.97e+01	2.73e+03	4.68e+01	2.73e+03 ≈	4.55e+01	2.73e+03 ≈	4.68e+01
f_{27}	3.72e+03 ≈	2.62e+02	3.66e+03 ≈	2.70e+02	3.71e+03	2.18e+02	3.65e+03 ≈	2.14e+02	**3.58e+03** −	2.17e+02
f_{28}	5.82e+03 ≈	6.95e+02	5.52e+03 ≈	8.79e+02	5.88e+03	1.01e+03	5.90e+03 ≈	8.05e+02	**5.16e+03** ≈	7.71e+02
f_{29}	**9.67e+03** ≈	3.93e+03	1.10e+04 ≈	5.88e+03	1.05e+04	7.07e+03	9.70e+03 ≈	3.34e+03	5.77e+05 ≈	3.05e+06
f_{30}	3.60e+04 ≈	2.47e+04	3.76e+04 ≈	3.69e+04	**2.73e+04**	8.09e+03	3.84e+04 ≈	2.54e+04	1.85e+05 −	6.59e+05
+/≈/−	1/27/2		0/29/1				1/29/0		4/23/3	

the CEC2014 suite, the CEC2022 suite, and eight engineering optimization problems. The ablation experiments competing with three baseline hyper-heuristic frameworks are implemented to confirm the effectiveness of the proposed evolutionary status guided probabilistic selection function, and the sensitivity experiments are executed to investigate the sensitivity of ES-HHA to the control parameter w_1 in the high-level component. The experimental results and statistical analysis demonstrate that our proposed ES-HHA is a promising approach to dealing with complex optimization problems.

Finally, we list some open topics for the further improvement of our proposal. In the future, we will focus on extending the ES-HHA to other application scenarios.

Table 15 Results of sensitivity experiments on 50-D CEC2014 benchmark functions

Func	ES-HHA_1		ES-HHA_3		ES-HHA_5		ES-HHA_7		ES-HHA_9	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	**5.79e+06** ≈	2.01e+06	6.34e+06 ≈	2.33e+06	5.90e+06	1.63e+06	6.87e+06 ≈	1.94e+06	9.65e+06 +	3.55e+06
f_2	**9.36e+06** ≈	2.62e+06	1.25e+07 ≈	4.18e+06	1.07e+07	2.66e+06	1.74e+07 +	3.84e+06	2.45e+07 +	7.64e+06
f_3	**2.61e+03** −	8.17e+02	6.91e+03 +	2.51e+03	4.64e+03	1.63e+03	1.43e+04 +	5.36e+03	2.65e+04 +	7.64e+03
f_4	**5.26e+02** ≈	3.60e+01	5.42e+02 ≈	4.25e+01	5.34e+02	4.53e+01	5.33e+02 ≈	4.31e+01	5.44e+02 ≈	4.87e+01
f_5	5.21e+02 ≈	4.87e−02	5.21e+02 ≈	3.36e−02	**5.21e+02**	3.14e−02	5.21e+02 ≈	3.80e−02	5.21e+02 ≈	3.84e−02
f_6	6.58e+02 ≈	3.92e+00	6.55e+02 ≈	5.65e+00	6.57e+02	5.39e+00	6.53e+02 ≈	4.95e+00	**6.52e+02** −	4.09e+00
f_7	**7.01e+02** ≈	2.86e−02	7.01e+02 ≈	4.34e−02	7.01e+02	2.90e−02	7.01e+02 +	4.81e−02	7.01e+02 +	6.05e−02
f_8	1.21e+03 ≈	6.39e+01	1.21e+03 ≈	5.47e+01	1.19e+03	6.56e+01	1.18e+03 ≈	6.73e+01	**1.15e+03** ≈	6.12e+01
f_9	1.40e+03 ≈	9.81e+01	1.37e+03 ≈	8.74e+01	1.39e+03	8.10e+01	1.33e+03 −	9.63e+01	**1.29e+03** −	8.20e+01
f_{10}	8.46e+03 ≈	8.82e+02	8.58e+03 ≈	9.71e+02	8.48e+03	9.41e+02	8.42e+03 ≈	8.71e+02	**8.41e+03** ≈	7.62e+02
f_{11}	8.90e+03 ≈	7.16e+02	9.05e+03 ≈	7.97e+02	**8.66e+03**	9.25e+02	9.09e+03 ≈	7.94e+02	9.19e+03 ≈	9.69e+02
f_{12}	1.20e+03 ≈	5.63e−02	**1.20e+03** ≈	5.11e−02	1.20e+03	5.93e−02	1.20e+03 ≈	6.35e−02	1.20e+03 ≈	8.89e−02
f_{13}	**1.30e+03** ≈	1.27e−01	1.30e+03 ≈	1.72e−01	1.30e+03	1.21e−01	1.30e+03 ≈	9.88e−02	1.30e+03 ≈	1.22e−01
f_{14}	**1.40e+03** ≈	2.54e−01	1.40e+03 ≈	2.59e−01	1.40e+03	2.53e−01	1.40e+03 ≈	3.04e−01	1.40e+03 ≈	2.89e−01
f_{15}	**1.54e+03** ≈	4.02e+00	1.54e+03 ≈	3.91e+00	1.54e+03	3.08e+00	1.54e+03 ≈	4.63e+00	1.54e+03 ≈	4.61e+00
f_{16}	1.62e+03 ≈	5.10e−01	1.62e+03 ≈	6.14e−01	1.62e+03	4.86e−01	**1.62e+03** ≈	7.08e−01	1.62e+03 ≈	7.19e−01
f_{17}	2.24e+05 ≈	1.06e+05	1.84e+05 ≈	7.56e+04	**1.78e+05**	6.66e+04	2.07e+05 ≈	9.61e+04	2.26e+05 ≈	1.33e+05
f_{18}	1.74e+05 ≈	5.28e+04	1.61e+05 ≈	3.55e+04	1.57e+05	3.44e+04	1.60e+05 ≈	3.65e+04	**1.55e+05** ≈	3.71e+04
f_{19}	2.11e+03 ≈	5.99e+02	1.96e+03 ≈	2.52e+01	2.11e+03	6.47e+02	2.12e+03 ≈	6.26e+02	**1.96e+03** ≈	2.73e+01
f_{20}	**9.95e+04** −	3.23e+04	1.34e+05 ≈	3.44e+04	1.16e+05	2.21e+04	1.64e+05 +	5.26e+04	1.88e+05 +	4.17e+04
f_{21}	**1.87e+05** ≈	1.11e+05	2.48e+05 ≈	1.46e+05	2.43e+05	1.26e+05	2.49e+05 ≈	1.27e+05	3.01e+05 ≈	1.32e+05
f_{22}	**2.52e+03** ≈	2.29e+02	2.60e+03 ≈	2.13e+02	2.58e+03	2.67e+02	2.75e+03 ≈	2.83e+02	2.79e+03 +	2.49e+02
f_{23}	**2.54e+03** ≈	1.43e−01	2.54e+03 +	2.05e−01	2.54e+03	1.37e−01	2.54e+03 +	6.22e−01	2.54e+03 +	8.40e−01
f_{24}	2.72e+03 ≈	1.64e+01	2.73e+03 ≈	1.46e+01	2.72e+03	2.29e+01	2.72e+03 ≈	1.54e+01	**2.71e+03** ≈	1.40e+01
f_{25}	2.75e+03 ≈	3.23e+01	2.76e+03 ≈	2.01e+01	2.75e+03	3.54e+01	2.75e+03 ≈	2.05e+01	**2.75e+03** ≈	1.99e+01
f_{26}	2.80e+03 ≈	9.22e+01	2.79e+03 ≈	6.23e+01	**2.77e+03**	7.29e+01	2.79e+03 ≈	6.97e+01	2.80e+03 ≈	7.83e+01
f_{27}	4.51e+03 ≈	1.79e+02	4.50e+03 ≈	1.49e+02	4.49e+03	1.18e+02	4.39e+03 ≈	1.62e+02	**4.29e+03** −	1.53e+02
f_{28}	9.45e+03 ≈	1.26e+03	9.44e+03 ≈	1.50e+03	9.52e+03	1.61e+03	8.66e+03 ≈	1.70e+03	**7.82e+03** −	1.41e+03
f_{29}	1.33e+05 ≈	9.10e+04	1.78e+05 ≈	1.63e+05	**1.27e+05**	9.44e+04	6.81e+05 ≈	2.76e+06	1.59e+05 ≈	1.14e+05
f_{30}	3.08e+05 ≈	1.62e+05	3.20e+05 ≈	1.77e+05	**2.80e+05**	9.12e+04	3.70e+05 ≈	1.61e+05	6.69e+05 +	5.72e+05
+/≈/−	0/28/2		2/28/0				5/24/1		8/18/4	

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

Author Contributions Rui Zhong conceptualization, methodology, investigation, writing—original draft, writing—review & editing, and funding acquisition. Jun Yu investigation, methodology, formal analysis, writing—review & editing, and project administration.

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

Declarations

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

References

1. FISHER, H.: Probabilistic learning combinations of local job-shop scheduling rules. Industrial Scheduling (1963)
2. Cowling, P.I., Kendall, G., Soubeiga, E.: A hyperheuristic approach to scheduling a sales summit. In: International Conference on the Practice and Theory of Automated Timetabling (2000)
3. Dowsland, K.A.: Off-the-peg or made-to-measure? Timetabling and scheduling with sa and ts. In: Burke, E., Carter, M. (eds.) Practice and Theory of Automated Timetabling II, pp. 37–52. Springer, Berlin, Heidelberg (1998). https://doi.org/10.1007/BFb0055880
4. Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Qu, R.: Hyper-heuristics: a survey of the state of the art. J. Oper. Res. Soc. **64**(12), 1695–1724 (2013). https://doi.org/10.1057/jors.2013.71

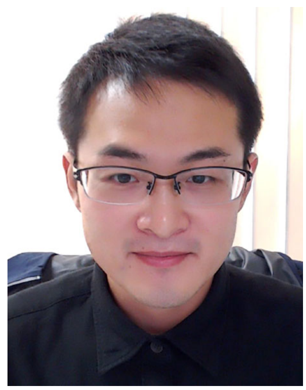
5. Choong, S.S., Wong, L.-P., Lim, C.P.: Automatic design of hyper-heuristic based on reinforcement learning. Inf. Sci. **436–437**, 89–107 (2018). https://doi.org/10.1016/j.ins.2018.01.005
6. Bai, R., Kendall, G.: An Investigation of Automated Planograms Using a Simulated Annealing Based Hyper-Heuristic, pp. 87–108. Springer, Boston (2005). https://doi.org/10.1007/0-387-25383-1_4
7. Koulinas, G., Kotsikas, L., Anagnostopoulos, K.: A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem. Inf. Sci. **277**, 680–693 (2014). https://doi.org/10.1016/j.ins.2014.02.155
8. Tsai, C.-W., Huang, W.-C., Chiang, M.-H., Chiang, M.-C., Yang, C.-S.: A hyper-heuristic scheduling algorithm for cloud. IEEE Trans. Cloud Comput. **2**(2), 236–250 (2014). https://doi.org/10.1109/TCC.2014.2315797
9. Zhang, Y., Bai, R., Qu, R., Tu, C., Jin, J.: A deep reinforcement learning based hyper-heuristic for combinatorial optimisation with uncertainties. Eur. J. Oper. Res. **300**(2), 418–427 (2022). https://doi.org/10.1016/j.ejor.2021.10.032
10. Chen, J., Bai, R., Dong, H., Qu, R., Kendall, G.: A dynamic truck dispatching problem in marine container terminal. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8 (2016). https://doi.org/10.1109/SSCI.2016.7850081
11. Burke, E.K., Silva, J.D.L., Soubeiga, E.: Multi-objective hyper-heuristic approaches for space allocation and timetabling, pp. 129–158. Springer, Boston (2005). https://doi.org/10.1007/0-387-25383-1_6
12. Kumari, A.C., Srinivas, K., Gupta, M.P.: Software module clustering using a hyper-heuristic based multi-objective genetic algorithm. In: 2013 3rd IEEE International Advance Computing Conference (IACC), pp. 813–818 (2013). https://doi.org/10.1109/IAdCC.2013.6514331
13. Mahmud, S., Abbasi, A., Chakrabortty, R.K., Ryan, M.J.: A self-adaptive hyper-heuristic based multi-objective optimisation approach for integrated supply chain scheduling problems. Knowl.-Based Syst. **251**, 109190 (2022). https://doi.org/10.1016/j.knosys.2022.109190
14. Cao, P., Zhang, Y., Zhou, K., Tang, J.: A reinforcement learning hyper-heuristic in multi-objective optimization with application to structural damage identification. Struct. Multidiscip. Optim. (2022). https://doi.org/10.1007/s00158-022-03432-5
15. Rui, Z., Jun, Y., Chao, Z., Masaharu, M.: Surrogate ensemble-assisted hyper-heuristic algorithm for expensive optimization problems. Int. J. Comput. Intell. Syst. (2023). https://doi.org/10.1007/s44196-023-00346-y
16. Cruz-Duarte, J.M., Amaya, I., Ortiz-Bayliss, J.C., Conant-Pablos, S.E., Terashima-Marín, H.: A primary study on hyper-heuristics to customise metaheuristics for continuous optimisation. In: 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2020). https://doi.org/10.1109/CEC48606.2020.9185591
17. Tapia-Avitia, J.M., Cruz-Duarte, J.M., Amaya, I., Ortiz-Bayliss, J.C., Terashima-Marin, H., Pillay, N.: A primary study on hyper-heuristics powered by artificial neural networks for customising population-based metaheuristics in continuous optimisation problems. In: 2022 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2022). https://doi.org/10.1109/CEC55065.2022.9870275
18. Jones, T., Forrest, S.: Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: Proceedings of the 6th International Conference on Genetic Algorithms, pp. 184–192. Morgan Kaufmann Publishers Inc., San Francisco (1995)
19. Ursem, R.K.: Diversity-guided evolutionary algorithms. In: Parallel Problem Solving from Nature—PPSN VII, pp. 462–471. Springer, Berlin (2002)
20. Qiao, Y., Luo, W., Lin, X., Xu, P., Preuss, M.: Dbcc2: an improved difficulty-based cooperative co-evolution for many-modal optimization. Complex Intell. Syst. (2023). https://doi.org/10.1007/s40747-022-00937-w
21. Xu, P., Luo, W., Lin, X., Chang, Y., Tang, K.: Difficulty and contribution based cooperative coevolution for large-scale optimization. IEEE Trans. Evolut. Comput. (2022). https://doi.org/10.1109/TEVC.2022.3201691
22. Li, W., Meng, X., Huang, Y.: Fitness distance correlation and mixed search strategy for differential evolution. Neurocomputing **458**, 514–525 (2021). https://doi.org/10.1016/j.neucom.2019.12.141
23. Tan, Z., Li, K., Wang, Y.: Differential evolution with adaptive mutation strategy based on fitness landscape analysis. Inf. Sci. **549**, 142–163 (2021). https://doi.org/10.1016/j.ins.2020.11.023
24. Luo, W., Qiao, Y., Lin, X., Xu, P., Preuss, M.: Many-modal optimization by difficulty-based cooperative co-evolution. In: 2019 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1907–1914 (2019). https://doi.org/10.1109/SSCI44817.2019.9003005
25. Pant, M., Radha, T., Singh, V.P.: A simple diversity guided particle swarm optimization. In: 2007 IEEE Congress on Evolutionary Computation, pp. 3294–3299 (2007). https://doi.org/10.1109/CEC.2007.4424896
26. Ghosh, A., Das, S., Mallipeddi, R., Das, A.K., Dash, S.S.: A modified differential evolution with distance-based selection for continuous optimization in presence of noise. IEEE Access **5**, 26944–26964 (2017). https://doi.org/10.1109/ACCESS.2017.2773825
27. Sudholt, D.: The Benefits of Population Diversity in Evolutionary Algorithms: A Survey of Rigorous Runtime Analyses, pp. 359–404. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-29414-4_8
28. Poláková, R., Tvrdík, J., Bujok, P.: Differential evolution with adaptive mechanism of population size according to current population diversity. Swarm Evol. Comput. **50**, 100519 (2019). https://doi.org/10.1016/j.swevo.2019.03.014
29. Yu, Y., Wang, K., Zhang, T., Wang, Y., Peng, C., Gao, S.: A population diversity-controlled differential evolution for parameter estimation of solar photovoltaic models. Sustain. Energy Technol. Assess. **51**, 101938 (2022). https://doi.org/10.1016/j.seta.2021.101938
30. Li, C., Sun, G.-J., Deng, L., Li-yan, Q., Yang, G.: A population state evaluation-based improvement framework for differential evolution. Inf. Sci. (2023). https://doi.org/10.1016/j.ins.2023.01.120
31. Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., Schulenburg, S.: Hyper-Heuristics: An Emerging Direction in Modern Search Technology, pp. 457–474. Springer (2003). https://doi.org/10.1007/0-306-48056-5_16
32. Dechter, R.: Chapter 7—stochastic greedy local search. In: Dechter, R. (ed.) Constraint Processing. The Morgan Kaufmann Series in Artificial Intelligence, pp. 191–208. Morgan Kaufmann, San Francisco (2003). https://doi.org/10.1016/B978-155860890-0/50008-6
33. Locatelli, M., Maischberger, M., Schoen, F.: Differential evolution methods based on local searches. Comput. Oper. Res. **43**, 169–180 (2014). https://doi.org/10.1016/j.cor.2013.09.010
34. Yu, J.: Vegetation evolution: an optimization algorithm inspired by the life cycle of plants. Int. J. Comput. Intell. Appl. (2022). https://doi.org/10.1142/S1469026822500109
35. Yaguchi, K., Tamura, K., Yasuda, K., Ishigame, A.: Basic study of proximate optimality principle based combinatorial optimization method. In: 2011 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1753–1758 (2011). https://doi.org/10.1109/ICSMC.2011.6083925

36. Houssein, E.H., Saad, M.R., Hashim, F.A., Shaban, H., Hassaballah, M.: Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems. Eng. Appl. Artif. Intell. **94**, 103731 (2020). https://doi.org/10.1016/j.engappai.2020.103731
37. Humphries, N., Queiroz, N., Dyer, J., Pade, N., Musyl, M., Schaefer, K., Fuller, D., Brunnschweiler, J., Doyle, T., Houghton, J., Hays, G., Jones, C., Noble, L., Wearmouth, V., Southall, E., Sims, D.: Environmental context explains lévy and brownian movement patterns of marine predators. Nature **465**, 1066–9 (2010). https://doi.org/10.1038/nature09116
38. Seyyedabbasi, A., Kiani, F.: Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems. Eng. Comput. (2022). https://doi.org/10.1007/s00366-022-01604-x
39. Zhang, Y.-J., Wang, Y.-F., Yan, Y.-X., Zhao, J., Gao, Z.-M.: 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**, 12367–12403 (2022). https://doi.org/10.1016/j.aej.2022.06.017
40. Hashim, F.A., Hussain, K., Houssein, E., Mabrouk, M., Al-Atabany, W.: Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl. Intell. **51**, 1–21 (2021). https://doi.org/10.1007/s10489-020-01893-z
41. Chen, Y., Gou, L., Li, H.: A differential evolution based henry gas solubility optimizer for dynamic performance optimization problems of pro system. Appl. Soft Comput. **125**, 109097 (2022). https://doi.org/10.1016/j.asoc.2022.109097
42. Zhang, Y., Li, S., Wang, Y., Yan, Y., Zhao, J., Gao, Z.: Self-adaptive enhanced learning differential evolution with surprisingly efficient decomposition approach for parameter identification of photovoltaic models. Energy Convers. Manag. **308**, 118387 (2024). https://doi.org/10.1016/j.enconman.2024.118387
43. Azizi, M., Aickelin, U., Khorshidi, H., Baghalzadeh Shishehgarkhaneh, M.: Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. Sci. Rep. **13**, 226 (2023). https://doi.org/10.1038/s41598-022-27344-y
44. Zhang, Y.-J., Wang, Y.-F., Yan, Y.-X., Zhao, J., Gao, Z.-M.: Self-adaptive hybrid mutation slime mould algorithm: case studies on uav path planning, engineering problems, photovoltaic models and infinite impulse response. Alex. Eng. J. **98**, 364–389 (2024). https://doi.org/10.1016/j.aej.2024.04.075
45. Opara, K., Arabas, J.: Comparison of mutation strategies in differential evolution—a probabilistic perspective. Swarm Evol. Comput. **39**, 53–69 (2018). https://doi.org/10.1016/j.swevo.2017.12.007
46. Zhong, R., Zhang, E., Munetomo, M.: Cooperative coevolutionary surrogate ensemble-assisted differential evolution with efficient dual differential grouping for large-scale expensive optimization problems. Complex Intell. Syst. (2023). https://doi.org/10.1007/s40747-023-01262-6
47. Zhong, R., Peng, F., Yu, J., Munetomo, M.: Q-learning based vegetation evolution for numerical optimization and wireless sensor network coverage optimization. Alex. Eng. J. **87**, 148–163 (2024). https://doi.org/10.1016/j.aej.2023.12.028
48. Van Thieu, N., Mirjalili, S.: Mealpy: an open-source library for latest meta-heuristic algorithms in python. J. Syst. Architect. **139**, 102871 (2023). https://doi.org/10.1016/j.sysarc.2023.102871
49. Nguyen, T.: A framework of optimization functions using Numpy (OpFuNu) for optimization problems. Zenodo (2020). https://doi.org/10.5281/zenodo.3620960
50. Thieu, N.V.: ENOPPY: a python library for engineering optimization problems. Zenodo (2023). https://doi.org/10.5281/zenodo.7953206
51. Ezugwu, A., Agushaka, O., Abualigah, L., Mirjalili, S., Gandomi, A.: Prairie dog optimization algorithm. Neural Comput. Appl. **34**, 20017–20065 (2022). https://doi.org/10.1007/s00521-022-07530-9
52. Coello Coello, C.A.: Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput. Methods Appl. Mech. Eng. **191**(11), 1245–1287 (2002). https://doi.org/10.1016/S0045-7825(01)00323-1
53. Holland, J.H.: Genetic algorithms. Sci. Am. **267**(1), 66–73 (1992)
54. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, pp. 1942–19484 (1995). https://doi.org/10.1109/ICNN.1995.488968
55. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**, 341–359 (1997). https://doi.org/10.1023/A:1008202821328
56. Beyer, H.-G., Schwefel, H.-P.: Evolution strategies—a comprehensive introduction. Nat. Comput. **1**, 3–52 (2002). https://doi.org/10.1023/A:1015059928466
57. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evol. Comput. **9**(2), 159–195 (2001). https://doi.org/10.1162/106365601750190398
58. Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl.-Based Syst. **89**, 228–249 (2015). https://doi.org/10.1016/j.knosys.2015.07.006
59. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Adv. Eng. Softw. **95**, 51–67 (2016). https://doi.org/10.1016/j.advengsoft.2016.01.008
60. Mirjalili, S.: Sca: a sine cosine algorithm for solving optimization problems. Knowl.-Based Syst. **96**, 120–133 (2016). https://doi.org/10.1016/j.knosys.2015.12.022
61. Acharya, D., Das, D.: A novel human conception optimizer for solving optimization problems. Sci. Rep. (2022). https://doi.org/10.1038/s41598-022-25031-6
62. Deng, L., Liu, S.: Snow ablation optimizer: a novel metaheuristic technique for numerical optimization and engineering design. Expert Syst. Appl. **225**, 120069 (2023). https://doi.org/10.1016/j.eswa.2023.120069
63. Prakash, T., Singh, P.P., Singh, V.P., Singh, S.N.: A novel brown-bear optimization algorithm for solving economic dispatch problem. In: Advanced Control & Optimization Paradigms for Energy System Operation and Management, pp. 137–164. River Publishers (2023)
64. Tanabe, R., Fukunaga, A.S.: Improving the search performance of shade using linear population size reduction. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 1658–1665 (2014). https://doi.org/10.1109/CEC.2014.6900380
65. Li, R., Zhang, H., Zhuang, Q., Li, R., Chen, Y.: Bp neural network and improved differential evolution for transient electromagnetic inversion. Comput. Geosci. **137**, 104434 (2020). https://doi.org/10.1016/j.cageo.2020.104434
66. Song, Y., Zhao, G., Zhang, B., Chen, H., Deng, W., Deng, W.: An enhanced distributed differential evolution algorithm for portfolio optimization problems. Eng. Appl. Artif. Intell. **121**, 106004 (2023). https://doi.org/10.1016/j.engappai.2023.106004
67. Cowling, P., Kendall, G., Soubeiga, E.: A hyperheuristic approach to scheduling a sales summit. In: Burke, E., Erben, W. (eds.) Practice and Theory of Automated Timetabling III, pp. 176–190. Springer, Berlin (2001). https://doi.org/10.1007/3-540-44629-X_11
68. Jackson, W.G., Özcan, E., Drake, J.H.: Late acceptance-based selection hyper-heuristics for cross-domain heuristic search. In: 2013 13th UK Workshop on Computational Intelligence (UKCI), pp. 228–235 (2013). https://doi.org/10.1109/UKCI.2013.6651310

69. Özcan, E., Kheiri, A.: A hyper-heuristic based on random gradient, greedy and dominance. In: Computer and Information Sciences II, pp. 557–563. Springer, London (2012). https://doi.org/10.1007/978-1-4471-2155-8_71
70. Cowling, P., Kendall, G., Soubeiga, E.: Hyperheuristics: a tool for rapid prototyping in scheduling and optimisation. In: Applications of Evolutionary Computing, pp. 1–10. Springer, Berlin (2002). https://doi.org/10.1007/3-540-46004-7_1
71. Holm, S.: A simple sequentially rejective multiple test procedure. Scand. J. Stat. **6**(2), 65–70 (1979)
72. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997). https://doi.org/10.1109/4235.585893
73. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014). https://doi.org/10.1016/j.advengsoft.2013.12.007
74. de Santiago Júnior, V.A., Özcan, E., de Carvalho, V.R.: Hyper-heuristics based on reinforcement learning, balanced heuristic selection and group decision acceptance. Appl. Soft Comput. **97**, 106760 (2020). https://doi.org/10.1016/j.asoc.2020.106760
75. Liu, J., Li, D., Wu, Y., Liu, D.: Lion swarm optimization algorithm for comparative study with application to optimal dispatch of cascade hydropower stations. Appl. Soft Comput. **87**, 105974 (2020). https://doi.org/10.1016/j.asoc.2019.105974
76. Pierezan, J., Maidl, G., Massashi Yamao, E., dos Santos Coelho, L., Cocco Mariani, V.: Cultural coyote optimization algorithm applied to a heavy duty gas turbine operation. Energy Convers. Manag. **199**, 111932 (2019). https://doi.org/10.1016/j.enconman.2019.111932
77. Zhong, R., Fan, Q., Zhang, C., Yu, J.: Hybrid remora crayfish optimization for engineering and wireless sensor network coverage optimization. Cluster Comput. (2024). https://doi.org/10.1007/s10586-024-04508-1

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

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



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