



Surrogate Ensemble-Assisted Hyper-Heuristic Algorithm for Expensive Optimization Problems

Rui Zhong¹ · Jun Yu² · Chao Zhang³ · Masaharu Munetomo⁴

Received: 11 July 2023 / Accepted: 6 October 2023
© The Author(s) 2023

Abstract

This paper proposes a novel surrogate ensemble-assisted hyper-heuristic algorithm (SEA-HHA) to solve expensive optimization problems (EOPs). A representative HHA consists of two parts: the low-level and the high-level components. In the low-level component, we regard the surrogate-assisted technique as a type of search strategy and design the four search strategy archives: exploration strategy archive, exploitation strategy archive, surrogate-assisted estimation archive, and mutation strategy archive as low-level heuristics (LLHs), each archive contains one or more search strategies. Once the surrogate-assisted estimation archive is activated to generate the offspring individual, SEA-HHA first selects the dataset for model construction from three principles: *All Data*, *Recent Data*, and *Neighbor*, which correspond to the global and the local surrogate model, respectively. Then, the dataset is randomly divided into training and validation data, and the most accurate model built by polynomial regression (PR), support vector regression (SVR), and Gaussian process regression (GPR) cooperates with the infill sampling criterion is employed for solution estimation. In the high-level component, we design a random selection function based on the pre-defined probabilities to manipulate a set of LLHs. In numerical experiments, we compare SEA-HHA with six optimization techniques on 5-D, 10-D, and 30-D CEC2013 benchmark functions and three engineering optimization problems with only 1000 fitness evaluation times (FEs). The experimental and statistical results show that our proposed SEA-HHA has broad prospects for dealing with EOPs.

Keywords Surrogate ensemble-assisted (SEA) · Hyper-heuristic algorithm (HHA) · Expensive optimization problems (EOPs) · High-level and low-level component design

1 Introduction

Evolutionary computation (EC) and swarm intelligence (SI) have achieved tremendous success in the research field [1–3] and industrial design [4–6] thanks to their superior characteristics such as robustness, flexibility, efficiency, and applicability. However, as a kind of stochastic optimization technique, an enormous number of fitness evaluation times (FEs) are necessary to find acceptable solutions, which severely limits the scalability of these methodologies in computationally expensive optimization problems (EOPs). Therefore, many researchers attempt to combine these algorithms with mathematical methods to further accelerate the convergence of optimization. A common approach is to adopt the surrogate model and the infill sampling criterion to predict potential solutions, and then combine them with conventional evolutionary algorithms (EA) to improve search efficiency. An optimization framework like this that com-

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

Jun Yu
yujun@ie.niigata-u.ac.jp

Chao Zhang
zhang@u-fukui.ac.jp

Masaharu Munetomo
munetomo@iic.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
- ³ Department of Engineering, University of Fukui, Fukui, Japan
- ⁴ Information Initiative Center, Hokkaido University, Sapporo, Japan

bines the two belongs to the surrogate-assisted evolutionary algorithm (SAEA) [7–9].

Up to now, many effective SAEAs have been reported: Dong et al. [10] extended the surrogate-assisted approach to the popular and powerful grey wolf optimization (SAGWO). The radial basis function (RBF) is employed to assist the meta-heuristic exploration and fitness landscape knowledge mining. Nishihara et al. [11] noticed that not only the model settings but also the training data chosen will influence the estimation performance and designed an adaption scheme for training data selection, which includes four criteria: *All Data*, *Current Population*, *Recent Data*, and *Neighbor*. These schemes collaborate with differential evolution (DE) for computationally expensive optimization problems. Wang et al. [12] proposed a global and local surrogate-assisted DE (GL-SADE) to solve high-dimensional EOPs. The global RBF model is trained with all samples to approximate the tendency of the whole fitness landscape, and the local Kriging model trained with the local population prefers to select solutions with well-performed prediction and great uncertainty, which can prevent the search direction from getting trapped into local optima. The unique reward search strategy in GL-SADE encourages the re-utilization of the Kriging model when the solution found by the local Kriging model is the best so far. Cai et al. [13] introduced the surrogate-assisted technique to multi-objective EOPs. Two strategies are proposed to balance the global and local search for multi-objective optimization: (1) Maximum angle-distance sequential sampling based improved surrogate-based multi-objective local search method. (2) Diversity-enhanced expected improvement matrix infill criterion-based pre-screening strategy. In addition, many SAEAs have been employed to deal with real-world applications: Wang et al. [14] proposed a committee-based active learning surrogate-assisted particle swarm optimization (CAL-SAPSO) for an airfoil design problem, where the best and most uncertain solutions found by the surrogate ensemble technique are evaluated using the expensive objective function, and the local surrogate model is built around the best solution obtained so far. Xiang et al. [15] proposed a clustering-based surrogate-assisted multi-objective evolutionary algorithm termed AR-MOEA+SA for the shelter location planning problem. The RBF is adopted to approximately calculate the evacuation distance under the uncertainty of road networks and the clustering strategy is cooperated to estimate the position of communities. Wakjira et al. [16] proposed a data-driven approach to determine the load and flexural capacities of reinforced concrete beams strengthened with fabric-reinforced cementitious matrix composites in flexure. Seven efficient surrogate models including kernel ridge regression, K-nearest neighbors, support vector regression, classification and regression trees, random forest, gradient boosted trees, and extreme gra-

dient boosting are involved to estimate the best predictive model for this problem.

The introduction of surrogate-assisted techniques has promoted the development of the EC community rapidly and endowed the optimizers with a stronger ability to tackle more complex optimization problems at the cost of computational resources. However, No Free Lunch Theory [17] states that any pair of black-box optimization algorithms have identical averaging performance on all possible problems, and if an algorithm performs well on a specific category of problems, then, it must degenerate on the remaining problems, which is the only way for all algorithms to have the same performance on average across all functions. Thus, many researchers try to develop a generic optimization framework, which can dynamically modify the structure of the algorithm to adapt to the characteristics of certain problems.

Hyper-heuristics framework provides a potential opportunity to realize that. From the perspective of the hyper-heuristic algorithm (HHA), an optimization algorithm can be regarded as a combination of search strategies (e.g. the genetic algorithm (GA) repeats the crossover and mutation operators and particle swarm optimization (PSO) iterates the velocity and location update.), and this sequence of heuristics can also be optimized. As an “off-the-peg” technique rather than “made-to-measure” meta-heuristics [18], HHA is a high-level automatic methodology that manipulates a set of low-level heuristics (LLHs) to search for acceptable solutions [19]. Meanwhile, many HHAs have been reported to deal with combinatorial optimization problems [20–23] whilst only a few have dealt with continuous problems [24]. Therefore, the motivation of this research is to develop a hyper-heuristic algorithm for continuous optimization problems.

In this paper, we regard the surrogate-assisted estimation as a novel search operator and propose a novel surrogate ensemble-assisted hyper-heuristic algorithm (SEA-HHA) for continuous and computational EOPs. In the low-level component of SEA-HHA, we design four search strategy archives as the low-level heuristics (LLHs): The exploration strategy archive, exploitation strategy archive, surrogate-assisted estimation archive, and mutation strategy archive, each archive contains several search strategies. In the high-level component design, we apply a probabilistic random selection function to construct the optimization sequence dynamically. Specifically, the main contributions of this paper can be summarized as follows.

- (1) Four flexible and easy-implemented search strategy archives are designed as the LLHs, while the high-level component acts as the “brain” of SEA-HHA which randomly constructs the optimization sequence based on the pre-defined probabilities. This high-level component

design is expected to enhance the diversity of the selected search strategies and avoid premature.

- (2) In the surrogate-assisted estimation archive, we provide three kinds of data selection fashions: *All Data*, *Recent Data*, and *Neighbor*, which corresponds to the global and local concepts. The selected data are randomly separated into the training dataset and testing dataset, and the most accurate model constructed by polynomial regression (PR), support vector regression (SVR), and Gaussian process regression (GPR) is chosen for solution estimation, which is hoped to estimate high-quality solutions and accelerate the convergence of optimization.
- (3) We implement a set of experiments on CEC2013 benchmark functions [25] and three real-world engineering optimization problems to evaluate our proposal. Four meta-heuristics algorithms and two surrogate-assisted optimization methods are applied as comparative algorithms. Numerical experiments show that our proposed SEA-HHA is competitive with these popular and state-of-the-art optimization techniques.

The remainder of this paper is organized as follows: Sect. 2 introduces the related works. Section 3 provides a detailed introduction to our proposal. Section 4 covers the numerical experiments and statistical results. Section 5 analyzes our proposal and lists some open topics. Finally, Sect. 6 concludes this paper.

2 Related Works

2.1 Hyper-Heuristic Algorithm (HHA)

Motivated by solving classes of problems rather than one problem, the appearance of HHA can be traced to the early 1960s [26]. As an advanced methodology, HHA takes the sequence of strategies as the optimization object based on the knowledge, which can be described as “Heuristics to choose heuristics” [19]. Here, a typical structure of the HHA is shown in Fig. 1.

A classic HHA contains two constituents: the low-level component and the high-level component. The low-level component includes problem representation, the objective function(s), initial solutions, and a set of low-level heuristics (LLHs). The high-level component dominates the LLHs and constructs the sequence of heuristics. The move acceptance principle judges whether the generated offspring are accepted or rejected. Feedback is utilized as the reward to dynamically adjust the LLHs selection module. Here, we briefly review some works of literature on HHA approaches: Zhao et al. [29] proposed a cooperative multi-stage hyper-heuristic (CMS-HH) algorithm for combinatorial optimization, the GA is introduced to perturb the initial solution while an

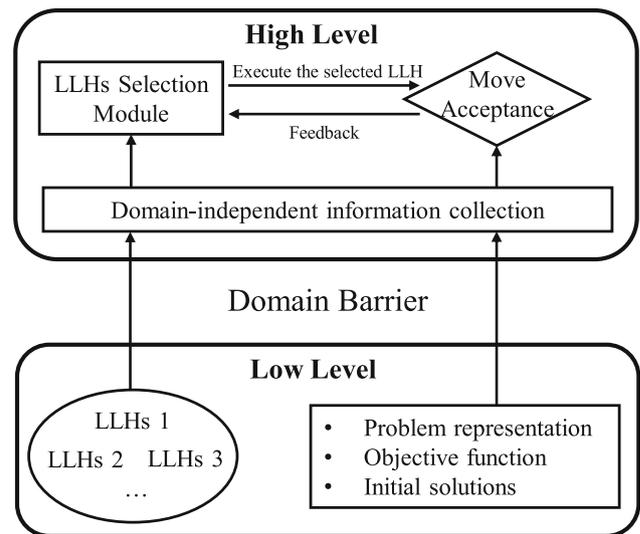


Fig. 1 Representative architecture of the HHA [27, 28]. LLHs i : the i th low-level heuristics

online learning mechanism based on the multi-armed bandits and relay hybridization technology is adopted to improve the quality of solutions. Qin et al. [30] developed a reinforcement learning-based hyper-heuristic algorithm to solve a practical heterogeneous vehicle routing problem. The policy-based reinforcement learning is a high-level selection strategy while several meta-heuristics with different characteristics are employed as low-level heuristics. Zhang et al. [31] proposed a hyper-heuristic algorithm for time-dependent green location routing problems with time windows, the Tabu search is adopted as the high-level selection module, and the greedy scheme is taken as the acceptance criterion. Most research focuses on combinatorial optimization problems.

2.2 Surrogate Models

In the usage of the surrogate-assisted technique, we concentrate on the performance indicators of the surrogate model, such as the robustness, computational complexity, flexibility, approximation ability, and so on. Polynomial regression (PR), support vector regression (SVR), and Gaussian process regression (GPR) are the three most popular and well-studied surrogate models, and the SAEAs can benefit from their easy implementation and excellent regression ability while the computational budget is affordable. Therefore, we adopt these three surrogate models to construct the surrogate-assisted estimation archive, and the detailed introduction is as follows.

2.2.1 Polynomial Regression (PR)

PR technique is an efficient and well-known model to solve the regression task, and the relationship between independent

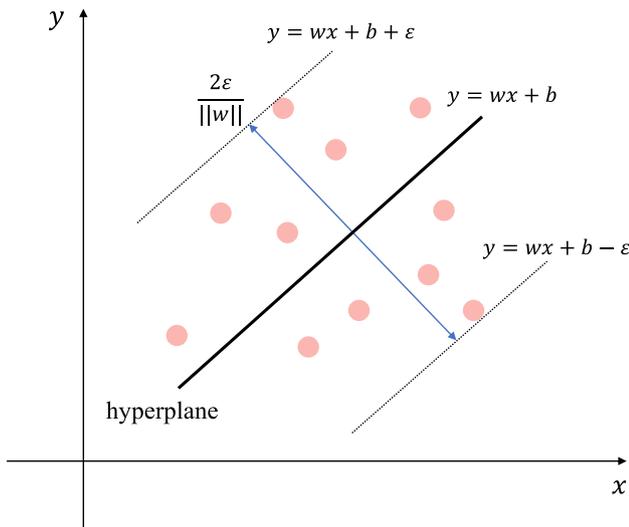


Fig. 2 Demonstration of SVR

decision variables $X = \{x_1, x_2, \dots, x_n\}$ and the dependent variables $Y = \{y_1, y_2, \dots, y_n\}$ is described as an n^{th} polynomial in X [32]:

$$E_i(y|X_i) = w_1X_i + w_2X_i^2 + w_3X_i^3 + \dots + w_nX_i^n + b_i. \tag{1}$$

From the matrix computation form, Eq. (1) can be rewritten to

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ 1 & x_3 & x_3^2 & \dots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}. \tag{2}$$

And this estimation can be approximated by least squares analysis [33]:

$$\min \sum_{i=1}^n (Y_i - E_i(y|X_i))^2, \tag{3}$$

where X_i is the i th sample, Y_i is the true solution of x_i , and $E_i(y|X_i)$ is the predicted solution of X_i by PR model.

2.2.2 Support Vector Regression (SVR)

SVR is a non-parametric machine learning technique that was first identified by Vladimir et al. in 1992 [34], which attempts to find a flat hyperplane that is within the tolerance margin (ϵ). Here, Fig. 2 demonstrates the SVR model in a regression task.

Mathematically, the optimization of SVR can be expressed in Eq. (4):

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m l_\epsilon(E_i(y|X_i), Y_i) \tag{4}$$

where $E_i(y|X_i)$ has similar structure to Eq. (1), C is a constant for regularization. $l_\epsilon(E_i(y|X_i), Y_i)$ is a ϵ -insensitive loss function that

$$l_\epsilon(E_i(y|X_i), Y_i) = \begin{cases} 0, & \text{if } |E_i(y|X_i) - Y_i| < \epsilon \\ |E_i(y|X_i) - Y_i| - \epsilon, & \text{otherwise.} \end{cases} \tag{5}$$

More details can be found in [35].

2.2.3 Gaussian Process Regression (GPR)

A GP is a collection of random variables, any finite set of which have a joint Gaussian distribution and is completely specified by its mean function $m(x)$ and the covariance function $k(x, x')$:

$$f(x) \sim \mathcal{GPR}(m(x), k(x, x')). \tag{6}$$

In the regression problem, the prior distribution of output y can be denoted as

$$y \sim N(0, k(x, x') + \sigma_n^2 I_n), \tag{7}$$

where $N(\cdot)$ is the normal distribution. σ_n^2 denotes the noise term. Assuming the distribution of testing dataset x' and training dataset x are identical, then, the prediction y' would follow a joint prior distribution with the training output y as [36]:

$$\begin{bmatrix} y \\ y' \end{bmatrix} \sim N\left(0, \begin{bmatrix} k(x, x) + \sigma_n^2 I_n & k(x, x') \\ k(x, x')^T & k(x', x') \end{bmatrix}\right), \tag{8}$$

where $k(x, x)$, $k(x, x')$ and $k(x', x')$ represent the covariance matrices among inputs from the training dataset, the training and testing dataset, as well as the testing dataset.

To guarantee the performance of the GPR, some hyper-parameters θ in the covariance function require to be optimized with n samples in the training process. One efficient optimization solution is to minimize the negative log marginal likelihood $L(\theta)$ as [37]:

$$L(\theta) = \frac{1}{2} \log \left[\det \lambda(\theta) + \frac{1}{2} y^T \lambda^{-1}(\theta) y + \frac{n}{2} \log(2\pi) \right] \tag{9}$$

$$\lambda(\theta) = k(\theta) + \sigma_n^2 I_n.$$

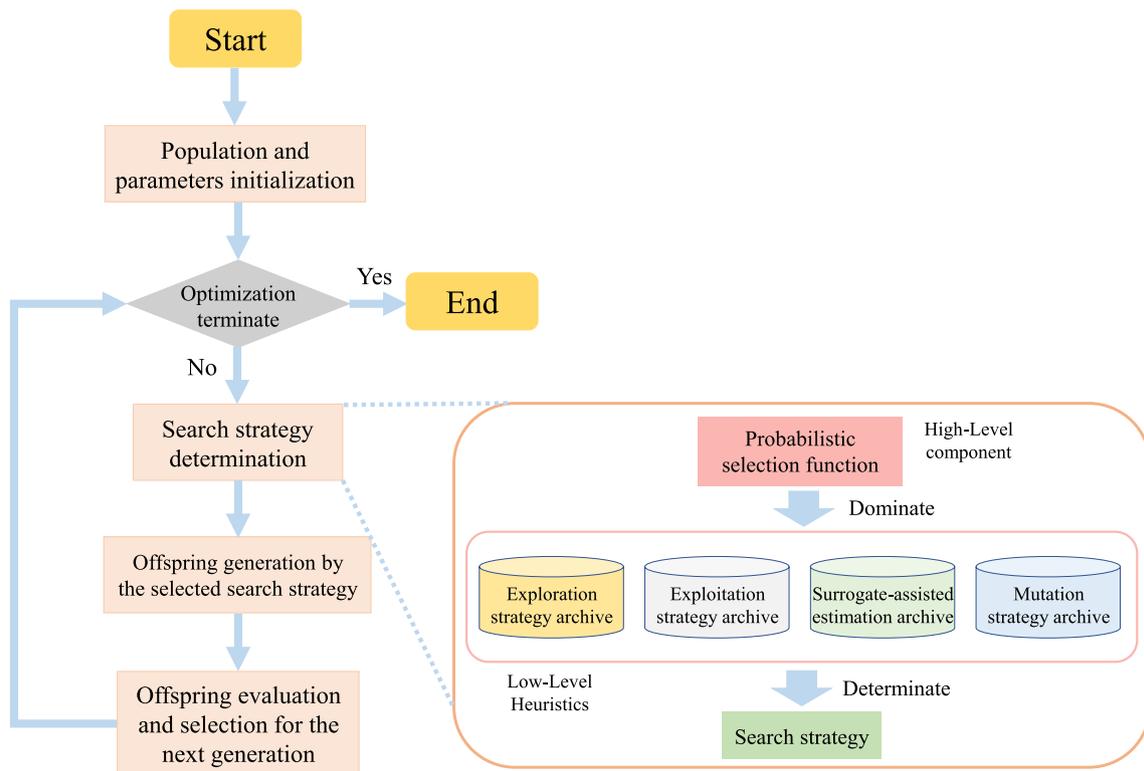


Fig. 3 Main architecture of SEA-HHA, the flowchart is similar to most EAs, and our proposal focuses on the search strategy determination part

After the hyper-parameters optimization of the GPR, the prediction y' can be obtained at data set x' through calculating the corresponding conditional distribution $p(y'|x', x, y)$ as

$$\begin{aligned}
 p(y'|x', x, y) &\sim N(y'|\bar{y}', \text{cov}(y')) \\
 \bar{y}' &= k(x, x')^T [k(x, x) + \sigma_n^2 I_n]^{-1} y \\
 \text{cov}(y') &= k(x', x') - k(x, x')^T \\
 &\quad [k(x, x) + \sigma_n^2 I_n]^{-1} k(x, x'),
 \end{aligned}
 \tag{10}$$

where \bar{y}' stands for values of prediction. $\text{cov}(y')$ denotes a variance matrix to reflect the uncertainty range of these predictions. More details of the GPR model can be found in [38].

3 Our Proposal: SEA-HHA

The overall optimization framework of the proposed SEA-HHA can be summarized in Fig. 3. Specifically, in the low-level component of SEA-HHA, we design four generation archives containing the various LLHs: Exploration strategy archive, exploitation strategy archive, surrogate-assisted estimation archive, and mutation strategy archive. Besides, each archive has one or more search strategies,

and different strategies in the same archive have an unbiased probability to be chosen. In the high-level component of SEA-HHA, a stochastic selection function based on predefined probabilities is employed as the decision function to determine the optimization sequence dynamically.

3.1 Exploration Strategy Archive

The differential-based search strategy is first proposed in DE [39] and has been adopted in many bio-inspired EAs to describe the foraging behaviors of natural organism [40–43]. In this paper, we also use the basic form of the differential-based search strategy in the exploration strategy archive and provide three different ways to select the base individual:

$$X_{i+1} = X_{base} + F \cdot (X_{r2} - X_{r3})
 \tag{11}$$

where X_{base} is randomly selected from $\{X_i, X_{best}, X_{r1}\}$ with equal probability. X_i is the i th individual, X_{best} represents the best solution in the current population, and X_{r1}, X_{r2} and X_{r3} are mutually different solutions which randomly sampled from the current population. F is a scaling vector and each element is randomly sampled from $[-0.8, 0.8]$ [44].

The pseudocode of the exploration operation is shown in Algorithm 1.

Algorithm 1 Exploration operation

Require: Dimension: D , Population: P , Fitness: Fit , Current index: i
Ensure: Offspring: O

- 1: $r \leftarrow \text{rand}()$
- 2: $r_1, r_2, r_3 \leftarrow \text{rand}()$ # r_1, r_2 and r_3 are mutually different
- 3: **if** $r < 1/3$ **then**
- 4: $X_{base} \leftarrow X_i$
- 5: **else if** $r < 2/3$ **then**
- 6: $X_{base} \leftarrow P[\arg \min(Fit)]$ # select the current best solution
- 7: **else**
- 8: $X_{base} \leftarrow X_{r1}$
- 9: **end if**
- 10: Construct a D -dimensional scaling vector F
- 11: $O \leftarrow X_{base} + F \cdot (X_{r2} - X_{r3})$
- 12: Ensure the O is within the search space
- 13: **return** O

3.2 Exploitation Strategy Archive

Rather than using the complex mechanism and parameters to realize the exploitation operation, we only adopt two parameters to determine the exploitation search strategy: the search direction D and the exploitation radius R . In addition, these operators can be described in Eq. (12):

$$X_{i+1} = X_{base} + D \cdot R \quad (12)$$

X_{base} is randomly selected from $\{X_i, X_{best}, X_{r1}\}$ as well. D is a random vector and R is a constant. Once these two parameters are specified, the location of X_{i+1} can be identified. In our experiment settings, each element in D is uniformly sampled from $[-1, 1]$ and $R = 2$ as suggested in [45].

The pseudocode of the exploitation operation is shown in Algorithm 2.

Algorithm 2 Exploitation operation

Require: Dimension: D , Population: P , Fitness: F , Current index: i , Radius: R
Ensure: Offspring: O

- 1: $r \leftarrow \text{rand}()$
- 2: $r_1, r_2, r_3 \leftarrow \text{rand}()$ # r_1, r_2 and r_3 are mutually different
- 3: **if** $r < 1/3$ **then**
- 4: $X_{base} \leftarrow X_i$
- 5: **else if** $r < 2/3$ **then**
- 6: $X_{base} \leftarrow P[\arg \min(F)]$
- 7: **else**
- 8: $X_{base} \leftarrow X_{r1}$
- 9: **end if**
- 10: **for** $i = 0$ to D **do**
- 11: $O^i \leftarrow X_{base}^i + R \cdot \text{rand}(-1, 1)$
- 12: **end for**
- 13: Ensure the O is within the search space
- 14: **return** O

3.3 Surrogate-Assisted Estimation Archive

We regard surrogate-assisted estimation as a kind of search strategy to generate high-quality solutions, and the basic process is as follows. We first choose the dataset selection fashion among *All Data*, *Recent Data*, and *Neighbor* randomly, then, the dataset is randomly divided into the training dataset and the validation dataset with the proportion of 80% and 20% respectively. Three kinds of models described in Sect. 2.2 are employed to construct the approximation model, and we use an extra DE to estimate the best solution in the surrogate model which has the highest accuracy on the validation dataset. This estimated solution will be evaluated by the real objective function and participate in the optimization as an offspring individual. Next, we will introduce the selection training dataset and the surrogate model selection principles in detail.

Inspired by the SADE-ATDSC [11], three different strategies for selecting training datasets are applied in the archive: *All Data*, *Recent Data*, and *Neighbor*. *All Data* utilizes all solutions from optimization beginning to approximate the overview of the fitness landscape. *Recent Data* represents the recent k generated solutions which are selected as the dataset to describe the regularity of solution movements in the optimization. And *Neighbor* denotes the nearest- k solutions of X_{best} determined by the Manhattan distance, which can depict the characteristics of the fitness landscape near the current best solution. In our experimental setting, the k is set to 100, and a general demonstration of dataset selection is shown in Fig. 4.

A subsequent problem is which model can approximate the fitness landscape better with these selected solutions. As we mentioned before, the selected dataset is randomly separated into two parts: The training dataset with the 80% proportion of the original data and the validation dataset with the remaining 20% proportion. Then, three kinds of models are constructed based on the training dataset, and the model that has the lowest mean squared error (MSE) loss on the validation dataset is considered the most accurate in this regression task. The calculation of MSE is in Eq. (13):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (E(y|x_i) - y_i)^2, \quad (13)$$

where n is the size of the dataset, $E(y|x_i)$ is the expectation of the model given the solution x_i , and y_i is the real fitness value of x_i , and the well-performed solution in the surrogate model is considered as a high-quality solution on the real fitness landscape and will be evaluated by the real objective function and participate in the optimization process.

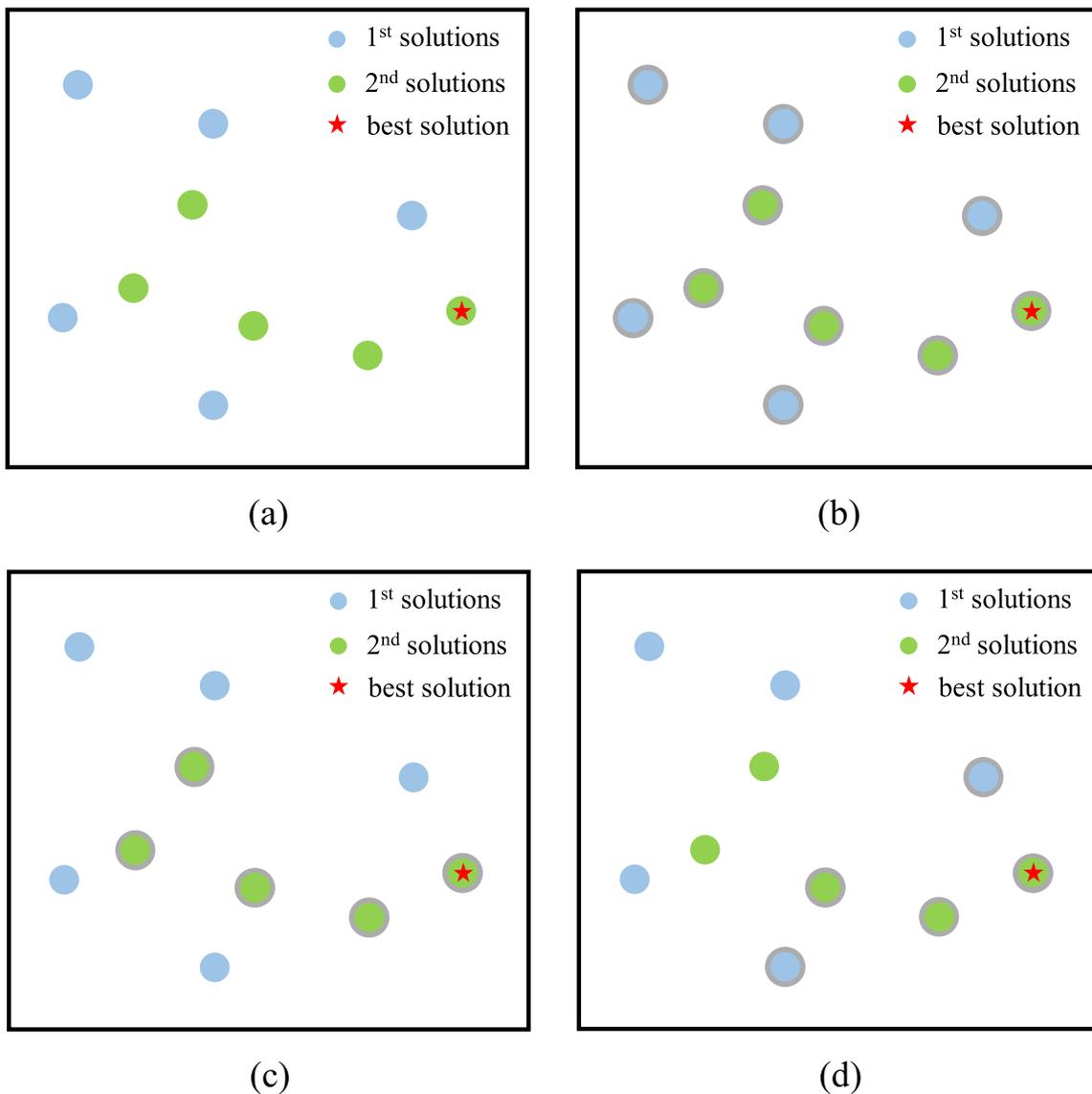


Fig. 4 Demonstration of dataset selection criteria, blue points are solutions in the 1st generation, green points are solutions in the 2nd generation, the red star represents the current best solution, and the selected dataset scale k is five in this example. The grey boundary point

represents the selected data for model construction. **a** The original distribution of solutions. **b** All Data principle. **c** Recent Data principle. **d** Neighbor principle

The pseudocode of the surrogate-assisted estimation strategy is shown in Algorithm 3.

3.4 Mutation Strategy Archive

The mutation strategy archive only contains one strategy that

$$X_{i+1} = X_{lb} + r \cdot (X_{ub} - X_{lb}) \tag{14}$$

r is a uniform random value from $[0, 1]$. X_{lb} and X_{ub} are the lower and upper bound of search space respectively. Simply,

we randomly generate a new solution in the search space to endow an ability to SEA-HHA to get rid of the local optimum.

In summary, the involved search operators are summarized in Table 1, and the pseudocode of SEA-HHA is shown in Algorithm 4.

Algorithm 4 line 7 means to determine a specific search strategy from our designed four archives, and line 8 applies this sampled strategy to generate the offspring. Different from most EAs in which the search strategy is applied to the whole population, the object of the search strategy in our proposed SEA-HHA is applied to the individual. Each individual in the population has a high opportunity to gen-

Table 1 Low-level heuristics (LLHs) in SEA-HHA

Type	Description	Objective
Exploration strategies	Algorithm 1	Strengthen the exploration ability of SEA-HHA
Exploitation strategies	Algorithm 2	Strengthen the exploitation ability of SEA-HHA
Surrogate-assisted estimation	Algorithm 3	Estimate elite solutions to accelerate the optimization
Mutation strategy	Equation (14)	Endow a capacity to get rid of local optima

Algorithm 3 Surrogate-assisted estimation**Require:** Data archive: DA , Model archive: MA **Ensure:** Estimated solution: S

```

1:  $r \leftarrow \text{rand}()$ 
2: if  $r < 1/3$  then
3:    $SD \leftarrow \text{All}(DA)$  # All Data
4: else if  $r < 2/3$  then
5:    $SD \leftarrow \text{Recent}(DA)$  # Recent Data
6: else
7:    $SD \leftarrow \text{Neighbor}(DA)$  # Neighbor
8: end if
9:  $TD, VD \leftarrow \text{split}(SD)$  # 80% Training data and 20% Validation data
10: Train the models in  $MA$  with  $TD$ 
11: Validate the models in  $MA$  with  $VD$ 
12: Estimate the best solution  $S$  on the most accurate model in terms of the DE
13: Ensure the  $S$  is within the search space
14: return  $S$ 

```

Algorithm 4 SEA-HHA**Require:** Dimension: D , Population size: PS , Maximum iteration: M , Pre-defined probability: PR , Search strategy archives: SSA ,**Ensure:** Global Optimum: GO

```

1:  $P \leftarrow \text{LHS}(D, PS)$  # Latin hypercube sampling for population initialization
2:  $GO \leftarrow \text{best}(P)$ 
3: Data archive  $DA \leftarrow P$ 
4:  $g = 0$ 
5: while  $g < M$  do
6:   for  $i = 0$  to  $PS$  do
7:     Determine a certain search strategy from search strategy archives by pre-defined probability
8:     Construct the offspring  $O_i$  with sampled search strategy
9:     Save the  $O_i$  to  $DA$ 
10:   end for
11: Survive the best- $PS$  solutions among population and offspring
12:  $GO \leftarrow \text{best}(P)$ 
13:  $g = g + 1$ 
14: end while
15: return  $GO$ 

```

erate offspring individual with various strategies, which is expected to enhance the diversity of the population and prevent premature convergence.

4 Numerical Experiments

We implement a set of experiments to evaluate the performance of our proposed SEA-HHA. Section 4.1 introduces the experiment settings, and Sect. 4.2 shows the experimental results.

4.1 Experiment Settings

4.1.1 Experiment Environment

The proposed SEA-HHA is programmed with Python 3.11 and implemented in Hokkaido University's high-performance intercloud supercomputer equipped with a CentOS operating system, Intel Xeon Gold 6148 CPU, and 384GB RAM.

4.1.2 Benchmark Functions

We evaluate the performance of SEA-HHA on the 5-D, 10-D, and 30-D of 28 CEC2013 benchmark functions and three complex engineering problems, and the detailed features of the CEC2013 suite are listed in Table 2.

In addition, three famous engineering optimization problems include Cantilever Beam Design [46], Tension/Compression Spring Design [47], and Pressure Vessel Design [48].

Cantilever Beam Design: This problem is a structural engineering optimization problem that is related to the weight optimization of a cantilever beam with a square cross section. Equation (15) shows the mathematical model of this problem:

minimize

$$f(X) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5)$$

subject to

$$g(X) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \quad (15)$$

where

$$0.01 \leq x_i \leq 100, i = 1, 2, \dots, 5$$

Tension/Compression Spring Design: The objective of this problem is to minimize the weight of a tension/compression spring under the constraints of minimum deflection, shear stress, surge frequency, and outside diame-

Table 2 Summary of the CEC2013 suite: *Uni.* unimodal function, *Multi.* multimodal function, *Comp.* composition function

Fun	Description	Feature	Optimum
f_1	Sphere Function	Uni.	-1400
f_2	Rotated High Conditioned Elliptic Function		-1300
f_3	Rotated Bent Cigar Function		-1200
f_4	Rotated Discus Function		-1100
f_5	Different Powers Function		-1000
f_6	Rotated Rosenbrock's Function	Multi.	-900
f_7	Rotated Schaffers F7 Function		-800
f_8	Rotated Ackley's Function		-700
f_9	Rotated Weierstrass Function		-600
f_{10}	Rotated Griewank's Function		-500
f_{11}	Rastrigin's Function		-400
f_{12}	Rotated Rastrigin's Function		-300
f_{13}	Non-Continuous Rotated Rastrigin's Function		-200
f_{14}	Schwefel's Function		-100
f_{15}	Rotated Schwefel's Function		100
f_{16}	Rotated Katsuura Function		200
f_{17}	Lunacek Bi-Rastrigin Function		300
f_{18}	Rotated Lunacek Bi-Rastrigin Function		400
f_{19}	Expanded Griewank's plus Rosenbrock's Function		500
f_{20}	Expanded Scaffer's F6 Function		600
f_{21}	Composition Function 1 ($n = 5$, Rotated)	Comp.	700
f_{22}	Composition Function 2 ($n = 3$, Unrotated)		800
f_{23}	Composition Function 3 ($n = 3$, Rotated)		900
f_{24}	Composition Function 4 ($n = 3$, Rotated)		1000
f_{25}	Composition Function 5 ($n = 3$, Rotated)		1100
f_{26}	Composition Function 6 ($n = 5$, Rotated)		1200
f_{27}	Composition Function 7 ($n = 5$, Rotated)		1300
f_{28}	Composition Function 8 ($n = 5$, Rotated)		1400

ter limitation. The formulation is presented in Eq. (16):

minimize

$$f(X) = (x_3 + 2)x_2x_1^2$$

subject to

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

$$g_2(X) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0$$

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

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

where

$$0.05 \leq x_1 \leq 2$$

$$0.25 \leq x_2 \leq 1.3$$

$$2 \leq x_3 \leq 15$$

(16)

Pressure Vessel Design: This problem attempts to minimize the cost of the pressure vessel including the cost of forming, material, and welding. And this optimization problem can be expressed in Eq. (17):

minimize

$$f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

subject to

$$g_1(X) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(X) = -x_2 + 0.00954x_3 \leq 0$$

$$g_3(X) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0$$

$$g_4(X) = x_4 - 240 \leq 0$$

where

$$0 \leq x_1 \leq 99$$

Table 3 Compared optimization techniques and their parameter configuration

Method	Parameters	Value
SEA-HHA	Radius R	2
	Sample size for model construction k	100
	Exploration probability	0.33
	Exploitation probability	0.33
	Surrogate-assisted estimation probability	0.33
	Mutation probability	0.01
SFO (2019) [50]	Coefficients of power attack A and ε	4 and 0.001
DSIDE (2020) [51]	Mutation strategy	DE/rand/1/bin
POA (2022) [52]	Parameter-free	
SCSO (2022) [53]	Sensitivity range r_G	[0, 2]
	Phases control range R	$[-2r_G, 2r_G]$
aRBF-NFO (2022) [54]	Surrogate model	RBF
	basic optimizer	NFO [55]
SHEALED (2023) [56]	Surrogate model	RBFnmv
	Scaling factor F	0.5
	Crossover rate Cr	0.8
	Probability of mutation Pm	0.3
	Probability of crossover Pc	0.8

$$\begin{aligned}
 0 &\leq x_2 \leq 99 \\
 10 &\leq x_3 \leq 200 \\
 10 &\leq x_4 \leq 200
 \end{aligned} \tag{17}$$

More detailed explanations and visual demonstrations of these engineering optimization problems can be found in [49].

4.1.3 Compared Methods and Parameters

We compare our proposal SEA-HHA with four EAs and two SAEAs, which are listed in Table 3. The selected probability for each search strategy archive in SEA-HHA plays an important role in guiding the optimization sequence construction. However, the determination of these parameters is also a difficult task. In this research, we fix the exploration probability, exploitation probability, surrogate-assisted estimation probability, and mutation probability with 0.33, 0.33, 0.33, and 0.01 respectively, which are also corresponding to the intuition of optimization algorithm design.

For all compared algorithms, the population size is 100, the maximum FEs by the real objective function in both the CEC2013 suite and engineering optimization problems are 1000, the sample size of the random search for promising solutions in surrogate models is 1000, which follows the recommend parameter setting in [11], and the independent trial run for each method is 30.

4.2 Experimental Results

This section shows the experimental and statistical results among seven compared optimization methods on CEC2013 benchmark functions and engineering optimization problems. Here, we collect the optimal fitness values in 30 trial runs of each optimization algorithm, and the Friedman test is applied to determine the significance. If significance exists, the Mann–Whitney U test is used to estimate the p value of every pair of algorithms, and the Holm multiple comparison test [57] corrects the p value obtained from the Mann–Whitney U test and further identifies the statistical significance.

$+$, \approx , and $-$ are applied to represent that our proposed SEA-HHA is significantly better, with no significance, and significantly worse with the compared method, and the best fitness value is in bold. In addition, the convergence curve of representative functions (i.e., unimodal functions: f_2 and f_4 ; multimodal functions: $f_6, f_9, f_{11}, f_{12}, f_{13}, f_{14}$, and f_{15} ; composite functions: f_{25}, f_{26} , and f_{28}) of in 5-D and 30-D are provided in Figs. 5 and 6.

4.2.1 Optimization on CEC2013 Suite

Tables 4, 5, and 6 provide the experimental and statistical results on CEC2013 benchmark functions. The mean and standard deviation (std) are calculated at the end of the optimization within 30 trial runs.

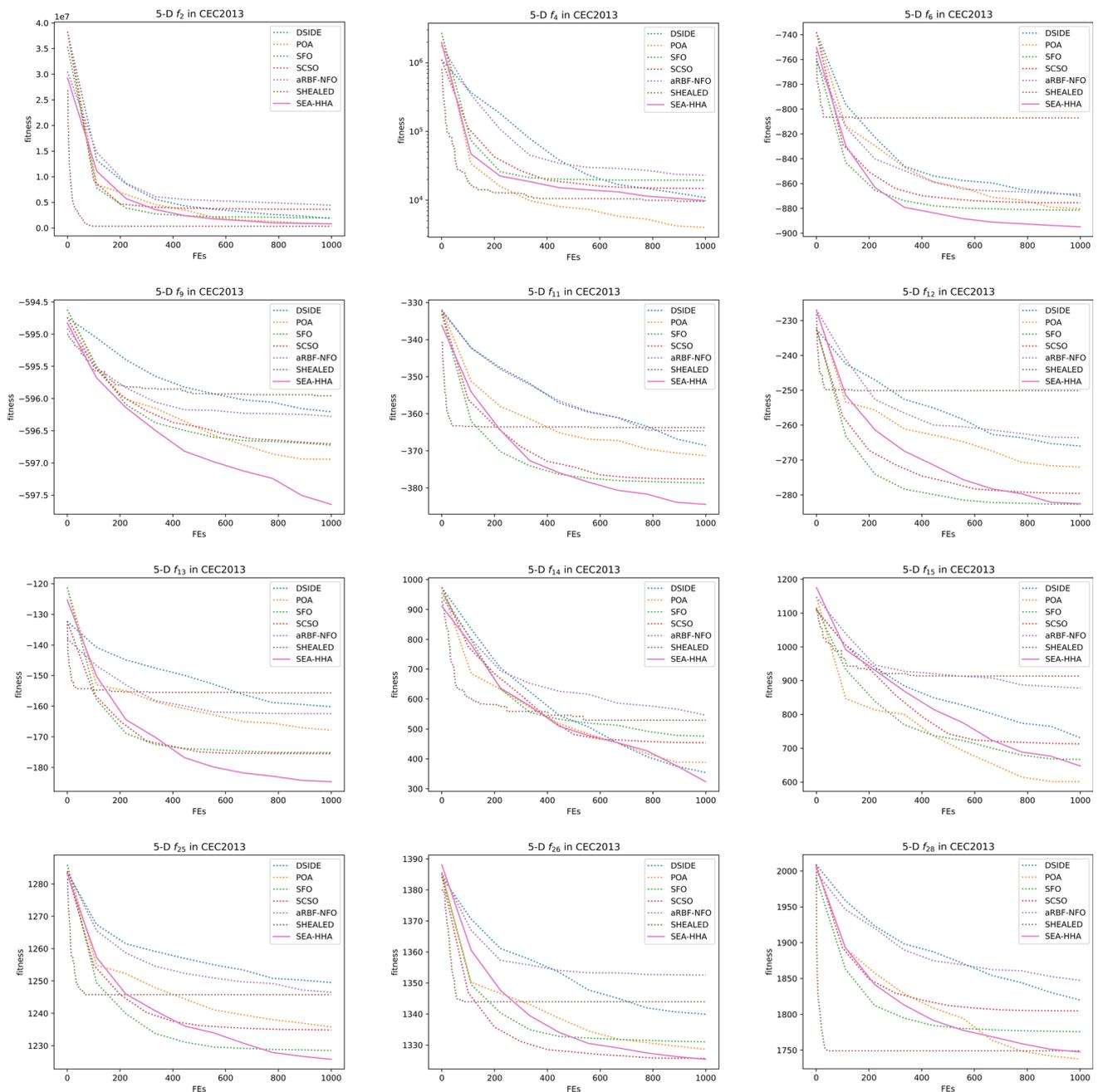


Fig. 5 Convergence graphs of 5-D representative functions in the CEC2013 suite

4.2.2 Optimization on Engineering Optimization Problems

The original SEA-HHA cannot solve the constrained optimization problems while the real-world engineering problems presented in Sect. 4.1.2 contain constraints. Therefore, we need to introduce a constraint-handling technique to SEA-HHA. Coello et al. [58] summarized the various penalty functions including static, dynamic, simulated annealing, adaptive, and death penalty. As one of the simplest methods, the death penalty assigns an enormous fitness value

to the individual which violates the constraint in the minimization optimization. For the sake of simplicity, we equip the SEA-HHA and all compared algorithms with a death penalty function to deal with constrained optimization problems. Tables 7 and 8 show the comparative results on the Cantilever Beam Design problem, Tables 9 and 10 show the optimization results on the Tension/Compression Spring Design problem, and Tables 11 and 12 show the results on the Pressure Vessel Design problem.

Table 4 Experimental and statistical results on 5-D CEC2013 Suite

Func	SFO		DSIDE		POA		SCSO		aRBF-NFO		SHEALED		SEA-HHA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
f_1	-1.1e+03 +	2.4e+02	-8.7e+02 +	2.4e+02	-4.3e+02 +	7.2e+02	-1.1e+03 +	3.3e+02	-1.4e+03 +	1.7e+01	-1.4e+03 -	7.6e-04	-1.4e+03	2.2e-01
f_2	2.0e+06 ≈	2.7e+06	1.9e+06 +	1.4e+06	7.9e+05 ≈	5.9e+05	3.7e+06 +	3.6e+06	4.4e+06 +	3.2e+06	3.4e+05 ≈	4.1e+05	8.1e+05	9.4e+05
f_3	1.0e+08 +	2.3e+08	4.1e+08 +	2.5e+08	7.9e+08 +	7.2e+08	2.0e+08 +	3.5e+08	3.9e+08 +	3.2e+08	1.3e+09 +	1.3e+09	2.3e+07	5.6e+07
f_4	2.0e+04 +	1.2e+04	1.1e+04 ≈	6.7e+03	4.0e+03 -	2.8e+03	1.5e+04 ≈	9.6e+03	2.3e+04 +	1.4e+04	9.7e+03 ≈	6.0e+03	9.8e+03	5.9e+03
f_5	-7.6e+02 +	2.2e+02	-7.2e+02 +	1.5e+02	-7.7e+02 +	1.6e+02	-7.2e+02 +	2.0e+02	-7.4e+02 +	1.7e+02	-8.7e+02 +	1.4e+02	-9.7e+02	2.1e+01
f_6	-8.8e+02 +	2.6e+01	-8.7e+02 +	1.5e+01	-8.8e+02 +	1.9e+01	-8.8e+02 +	2.7e+01	-8.7e+02 +	1.6e+01	-8.1e+02 +	5.7e+01	-8.9e+02	1.1e+01
f_7	-7.8e+02 ≈	1.4e+01	-7.6e+02 +	1.3e+01	6.1e+01 +	3.1e+02	-7.8e+02 ≈	1.6e+01	-7.5e+02 +	2.1e+01	-7.2e+02 +	2.0e+01	-7.8e+02	1.2e+01
f_8	-6.8e+02 ≈	1.1e-01	-6.8e+02 ≈	1.1e-01	-6.8e+02 ≈	6.1e-02	-6.8e+02 ≈	3.3e+00	-6.8e+02 ≈	7.5e-01	-6.8e+02 ≈	1.7e-01	-6.8e+02	9.2e-02
f_9	-6.0e+02 +	9.1e-01	-6.0e+02 +	6.1e-01	-6.0e+02 +	6.8e-01	-6.0e+02 +	1.0e+00	-6.0e+02 +	5.8e-01	-6.0e+02 +	8.0e-01	-6.0e+02	8.1e-01
f_{10}	-4.7e+02 +	2.6e+01	-4.4e+02 +	2.9e+01	-4.4e+02 +	3.4e+01	-4.4e+02 +	5.2e+01	-4.3e+02 +	3.3e+01	-5.0e+02 -	1.9e-01	-5.0e+02	1.7e-01
f_{11}	-3.8e+02 ≈	9.4e+00	-3.7e+02 +	7.7e+00	-3.7e+02 +	7.8e+00	-3.8e+02 ≈	1.1e+01	-3.6e+02 +	9.3e+00	-3.6e+02 +	9.0e+00	-3.8e+02	7.3e+00
f_{12}	-2.8e+02 ≈	7.9e+00	-2.7e+02 +	8.5e+00	-2.7e+02 +	8.8e+00	-2.8e+02 ≈	9.0e+00	-2.6e+02 +	9.1e+00	-2.5e+02 +	9.4e+00	-2.8e+02	6.5e+00
f_{13}	-1.8e+02 +	9.0e+00	-1.6e+02 +	6.9e+00	-1.7e+02 +	8.7e+00	-1.8e+02 +	1.4e+01	-1.6e+02 +	8.4e+00	-1.6e+02 +	1.1e+01	-1.8e+02	6.9e+00
f_{14}	4.8e+02 +	2.0e+02	3.5e+02 ≈	1.5e+02	3.9e+02 +	1.2e+02	4.5e+02 ≈	1.5e+02	5.5e+02 +	1.8e+02	5.3e+02 +	1.8e+02	3.2e+02	1.8e+02
f_{15}	6.7e+02 ≈	2.2e+02	7.3e+02 ≈	1.2e+02	6.0e+02 ≈	1.2e+02	7.1e+02 ≈	2.0e+02	8.8e+02 +	1.6e+02	9.1e+02 +	1.9e+02	6.5e+02	1.4e+02
f_{16}	2.0e+02 ≈	5.2e-01	2.0e+02 ≈	4.1e-01	2.0e+02 ≈	4.5e-01	2.0e+02 -	5.0e-01	2.0e+02 ≈	7.1e-01	2.0e+02 ≈	5.3e-01	2.0e+02	4.9e-01
f_{17}	3.2e+02 ≈	7.7e+00	3.3e+02 +	6.8e+00	3.3e+02 +	6.9e+00	3.3e+02 +	9.1e+00	3.3e+02 +	7.5e+00	3.4e+02 +	9.6e+00	3.2e+02	5.1e+00
f_{18}	4.2e+02 ≈	9.1e+00	4.3e+02 +	7.4e+00	4.7e+02 +	2.6e+01	4.3e+02 +	1.0e+01	4.3e+02 +	6.8e+00	4.4e+02 +	9.1e+00	4.2e+02	3.3e+00
f_{19}	5.0e+02 ≈	5.0e+00	5.1e+02 +	7.5e+00	5.6e+02 +	1.5e+02	5.0e+02 ≈	4.1e+00	5.4e+02 +	5.2e+01	5.5e+02 +	8.1e+01	5.0e+02	1.0e+00
f_{20}	6.0e+02 ≈	4.3e-01	6.0e+02 +	2.2e-01	6.0e+02 ≈	1.2e-01	6.0e+02 ≈	3.9e-01	6.0e+02 +	2.6e-01	6.0e+02 +	2.4e-01	6.0e+02	2.3e-01
f_{21}	1.1e+03 +	9.5e+01	1.1e+03 +	6.1e+01	1.1e+03 +	1.3e+02	1.1e+03 +	7.8e+01	1.2e+03 +	8.7e+01	1.0e+03 ≈	7.0e+01	1.0e+03	8.0e+01
f_{22}	1.6e+03 ≈	2.0e+02	1.6e+03 ≈	1.1e+02	1.4e+03 ≈	1.4e+02	1.6e+03 ≈	1.9e+02	1.7e+03 +	1.3e+02	1.6e+03 ≈	2.7e+02	1.5e+03	1.8e+02
f_{23}	1.7e+03 ≈	2.0e+02	1.8e+03 +	1.3e+02	1.5e+03 ≈	1.4e+02	1.7e+03 +	1.8e+02	1.9e+03 +	1.1e+02	1.8e+03 +	2.5e+02	1.6e+03	1.6e+02
f_{24}	1.2e+03 ≈	3.4e+01	1.2e+03 +	1.7e+01	1.1e+03 ≈	1.3e+01	1.2e+03 +	3.1e+01	1.2e+03 +	1.7e+01	1.2e+03 +	2.7e+01	1.1e+03	3.3e+01
f_{25}	1.2e+03 ≈	1.4e+01	1.2e+03 +	1.1e+01	1.2e+03 +	1.3e+01	1.2e+03 +	1.5e+01	1.2e+03 +	1.0e+01	1.2e+03 +	2.0e+01	1.2e+03	1.6e+01
f_{26}	1.3e+03 ≈	2.9e+01	1.3e+03 +	1.0e+01	1.3e+03 ≈	1.1e+01	1.3e+03 ≈	1.6e+01	1.4e+03 +	2.0e+01	1.3e+03 +	1.7e+01	1.3e+03	2.6e+01
f_{27}	1.7e+03 ≈	2.3e+01	1.7e+03 +	2.0e+01	1.7e+03 ≈	2.9e+01	1.7e+03 ≈	2.3e+01	1.7e+03 +	2.2e+01	1.7e+03 +	2.0e+01	1.7e+03	2.7e+01
f_{28}	1.8e+03 +	4.2e+01	1.8e+03 +	3.2e+01	1.7e+03 ≈	1.0e+02	1.8e+03 +	4.4e+01	1.8e+03 +	5.0e+01	1.7e+03 ≈	2.6e+01	1.7e+03	4.1e+01
	+ / ≈ / - ; 11/17/0		+ / ≈ / - ; 22/6/0		+ / ≈ / - ; 16/11/1		+ / ≈ / - ; 15/12/1		+ / ≈ / - ; 26/2/0		+ / ≈ / - ; 19/7/2			

Table 5 Experimental and statistical results on 10-D CEC2013 Suite

Func	SFO		DSIDE		POA		SCSO		aRBF-NFO		SHEALED		SEA-HHA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
f_1	2.5e+03 +	1.8e+03	2.6e+03 +	1.1e+03	6.2e+03 +	2.2e+03	2.8e+03 +	2.4e+03	4.7e+02 +	1.5e+03	-1.4e+03 -	1.3e+01	-9.3e+02	3.6e+02
f_2	2.8e+07 ≈	2.0e+07	3.2e+07 +	1.2e+07	2.5e+07 +	1.6e+07	2.1e+07 ≈	1.3e+07	6.0e+07 +	3.2e+07	1.6e+07 ≈	1.0e+07	1.6e+07	9.9e+06
f_3	1.9e+10 +	1.8e+10	2.0e+10 +	9.6e+09	1.6e+10 +	9.7e+09	3.5e+10 +	1.3e+11	1.0e+10 +	5.5e+09	2.8e+10 +	2.5e+10	6.2e+09	5.3e+09
f_4	2.9e+04 ≈	1.3e+04	3.4e+04 ≈	1.4e+04	1.6e+04 -	4.2e+03	2.4e+04 ≈	8.8e+03	5.3e+04 +	2.4e+04	4.9e+04 +	2.2e+04	3.3e+04	1.5e+04
f_5	5.2e+01 +	5.8e+02	4.2e+02 +	6.8e+02	3.3e+03 +	2.4e+03	4.9e+02 +	8.6e+02	7.7e+02 +	1.1e+03	-7.1e+02 ≈	9.9e+01	-5.8e+02	3.4e+02
f_6	-6.1e+02 +	1.4e+02	-5.7e+02 +	1.2e+02	-3.8e+02 +	2.3e+02	-6.4e+02 +	1.2e+02	-6.6e+02 +	8.7e+01	-5.9e+02 +	1.1e+02	-8.2e+02	3.0e+01
f_7	-5.9e+02 ≈	2.9e+02	-5.6e+02 +	7.1e+01	1.2e+04 +	5.0e+03	-6.5e+02 ≈	1.6e+02	-6.3e+02 +	5.2e+01	-5.4e+02 +	1.1e+02	-6.7e+02	4.8e+01
f_8	-6.8e+02 ≈	9.5e-02	-6.8e+02 ≈	1.1e-01	-6.8e+02 ≈	1.3e-01	-6.8e+02 ≈	8.3e-02	-6.8e+02 ≈	1.0e-01	-6.8e+02 ≈	1.3e-01	-6.8e+02	1.2e-01
f_9	-5.9e+02 ≈	1.8e+00	-5.9e+02 +	9.4e-01	-5.9e+02 ≈	8.7e-01	-5.9e+02 +	1.2e+00	-5.9e+02 +	1.0e+00	-5.9e+02 +	9.3e-01	-5.9e+02	1.5e+00
f_{10}	-5.0e+01 +	2.1e+02	5.7e+01 +	1.9e+02	2.6e+02 +	2.7e+02	-6.8e+01 +	2.1e+02	1.3e+02 +	3.0e+02	-4.9e+02 -	5.3e+00	-3.4e+02	1.1e+02
f_{11}	-3.0e+02 +	2.9e+01	-2.6e+02 +	2.2e+01	-2.6e+02 +	2.8e+01	-2.9e+02 +	3.7e+01	-2.9e+02 +	3.1e+01	-2.9e+02 +	1.6e+01	-3.2e+02	1.6e+01
f_{12}	-2.0e+02 +	2.6e+01	-1.6e+02 +	2.1e+01	-1.7e+02 +	2.6e+01	-1.8e+02 +	2.8e+01	-1.7e+02 +	3.2e+01	-1.6e+02 +	2.3e+01	-2.3e+02	1.6e+01
f_{13}	-9.6e+01 +	2.9e+01	-7.3e+01 +	2.2e+01	-8.0e+01 +	2.8e+01	-8.2e+01 +	3.2e+01	-7.2e+01 +	2.3e+01	-7.8e+01 +	1.7e+01	-1.2e+02	1.4e+01
f_{14}	1.8e+03 ≈	3.7e+02	1.7e+03 ≈	2.0e+02	1.7e+03 ≈	1.9e+02	1.7e+03 ≈	2.8e+02	1.8e+03 ≈	2.7e+02	2.1e+03 +	2.6e+02	1.7e+03	2.8e+02
f_{15}	2.0e+03 ≈	3.2e+02	2.1e+03 ≈	2.1e+02	1.7e+03 ≈	1.9e+02	1.8e+03 ≈	2.6e+02	2.0e+03 ≈	3.1e+02	2.4e+03 +	2.1e+02	1.9e+03	2.7e+02
f_{16}	2.0e+02 ≈	4.1e-01	2.0e+02 ≈	4.2e-01	2.0e+02 ≈	5.2e-01	2.0e+02 ≈	6.2e-01	2.0e+02 ≈	6.3e-01	2.0e+02 ≈	5.1e-01	2.0e+02	5.6e-01
f_{17}	4.2e+02 +	2.8e+01	4.4e+02 +	2.5e+01	4.4e+02 +	2.5e+01	4.1e+02 ≈	2.6e+01	4.0e+02 ≈	4.2e+01	4.0e+02 ≈	1.3e+01	3.9e+02	2.2e+01
f_{18}	5.2e+02 +	2.6e+01	5.4e+02 +	1.9e+01	7.9e+02 +	9.4e+01	5.3e+02 +	2.9e+01	4.9e+02 ≈	2.9e+01	4.9e+02 ≈	9.6e+00	5.0e+02	2.4e+01
f_{19}	1.9e+03 +	3.1e+03	2.9e+03 +	2.4e+03	1.2e+04 +	1.7e+04	1.9e+03 +	2.6e+03	2.4e+03 +	2.6e+03	6.2e+03 +	5.6e+03	5.3e+02	4.7e+01
f_{20}	6.0e+02 ≈	4.2e-01	6.0e+02 +	2.0e-01	6.0e+02 ≈	2.7e-01	6.0e+02 ≈	3.7e-01	6.0e+02 +	2.1e-01	6.0e+02 +	1.9e-01	6.0e+02	2.5e-01
f_{21}	1.3e+03 +	7.3e+01	1.3e+03 +	5.0e+01	1.5e+03 +	6.6e+01	1.3e+03 +	6.1e+01	1.4e+03 +	1.1e+02	1.2e+03 ≈	4.1e+01	1.2e+03	4.2e+01
f_{22}	2.9e+03 ≈	2.9e+02	2.8e+03 ≈	1.8e+02	2.9e+03 ≈	2.3e+02	2.9e+03 ≈	2.9e+02	2.8e+03 ≈	2.8e+02	3.1e+03 +	2.6e+02	2.8e+03	2.9e+02
f_{23}	3.1e+03 ≈	3.3e+02	3.3e+03 +	2.3e+02	2.8e+03 -	2.0e+02	3.2e+03 ≈	3.1e+02	3.2e+03 ≈	3.2e+02	3.4e+03 +	2.4e+02	3.0e+03	2.6e+02
f_{24}	1.2e+03 +	5.2e+00	1.2e+03 +	8.3e+00	1.2e+03 ≈	1.8e+01	1.2e+03 +	4.6e+00	1.2e+03 +	2.9e+00	1.2e+03 +	2.9e+00	1.2e+03	4.3e+00
f_{25}	1.3e+03 +	5.2e+00	1.3e+03 +	5.8e+00	1.3e+03 +	1.2e+01	1.3e+03 +	4.2e+00	1.3e+03 +	3.1e+00	1.3e+03 +	3.3e+00	1.3e+03	4.6e+00
f_{26}	1.4e+03 +	5.6e+01	1.4e+03 +	1.3e+01	1.4e+03 +	1.1e+01	1.4e+03 +	5.3e+01	1.4e+03 +	3.3e+01	1.5e+03 +	4.3e+01	1.4e+03	5.0e+01
f_{27}	2.0e+03 +	7.5e+01	1.9e+03 ≈	7.3e+01	1.9e+03 ≈	4.8e+01	2.0e+03 +	6.7e+01	2.0e+03 +	4.5e+01	2.0e+03 +	5.4e+01	1.9e+03	7.4e+01
f_{28}	2.4e+03 +	1.1e+02	2.5e+03 +	5.2e+01	2.3e+03 ≈	1.2e+02	2.4e+03 +	1.0e+02	2.5e+03 +	1.7e+02	2.4e+03 +	9.8e+01	2.2e+03	1.6e+02
	+/≈/-; 17/11/0		+/≈/-; 21/7/0		+/≈/-; 16/10/2		+/≈/-; 17/11/0		+/≈/-; 20/8/0		+/≈/-; 19/7/2			

Table 6 Experimental and statistical results on 30-D CEC2013 Suite

Func	SFO		DSIDE		POA		SCSO		aRBF-NFO		SHEALED		SEA-HHA	
	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
f_1	3.9e+04 +	5.2e+03	4.3e+04 +	7.1e+03	5.2e+04 +	5.3e+03	4.5e+04 +	6.6e+03	2.4e+04 +	6.6e+03	8.0e+03 -	5.9e+03	2.0e+04	6.3e+03
f_2	6.5e+08 +	2.1e+08	7.8e+08 +	1.6e+08	1.2e+09 +	5.2e+08	9.3e+08 +	4.0e+08	7.7e+08 +	4.0e+08	1.1e+09 +	2.3e+08	3.7e+08	1.7e+08
f_3	4.5e+16 +	1.3e+17	5.1e+17 +	1.4e+18	9.3e+17 +	2.7e+18	5.8e+17 +	2.0e+18	7.2e+14 +	2.0e+18	9.9e+16 +	3.1e+15	1.8e+14	9.4e+14
f_4	6.7e+04 -	5.0e+03	1.1e+05 ≈	3.0e+04	6.7e+04 -	6.7e+03	6.9e+04 -	3.1e+03	1.4e+05 ≈	3.1e+03	1.8e+05 +	3.0e+04	1.3e+05	2.9e+04
f_5	1.4e+04 +	4.8e+03	1.7e+04 +	5.2e+03	5.7e+04 +	2.0e+04	1.6e+04 +	7.4e+03	1.7e+04 +	7.4e+03	1.0e+04 +	7.3e+03	7.3e+03	4.4e+03
f_6	5.9e+03 +	2.2e+03	5.9e+03 +	1.5e+03	1.1e+04 +	3.0e+03	7.1e+03 +	2.4e+03	2.5e+03 +	2.4e+03	2.6e+03 +	1.7e+03	9.1e+02	8.4e+02
f_7	2.1e+05 +	4.6e+05	3.2e+05 +	4.1e+05	1.9e+08 +	2.0e+08	4.2e+05 +	8.8e+05	2.3e+04 +	8.8e+05	2.1e+05 +	8.4e+04	4.0e+03	1.5e+04
f_8	-6.8e+02 ≈	6.0e-02	-6.8e+02 ≈	6.2e-02	-6.8e+02 ≈	5.2e-02	-6.8e+02 ≈	6.3e-02	-6.8e+02 ≈	-6.8e+02 ≈	-6.8e+02 ≈	6.8e-02	-6.8e+02	7.0e-02
f_9	-5.6e+02 ≈	2.7e+00	-5.6e+02 +	1.5e+00	-5.6e+02 ≈	1.7e+00	-5.6e+02 ≈	3.5e+00	-5.6e+02 +	3.5e+00	-5.5e+02 +	1.9e+00	-5.6e+02	3.4e+00
f_{10}	5.3e+03 +	1.1e+03	5.4e+03 +	1.0e+03	8.4e+03 +	1.5e+03	6.4e+03 +	1.7e+03	5.6e+03 +	1.7e+03	5.3e+03 +	1.4e+03	2.4e+03	9.5e+02
f_{11}	3.2e+02 +	1.1e+02	3.6e+02 +	6.8e+01	4.7e+02 +	1.2e+02	3.8e+02 +	1.2e+02	3.5e+02 +	1.2e+02	3.5e+02 +	8.9e+01	1.8e+02	1.0e+02
f_{12}	4.1e+02 +	1.1e+02	4.5e+02 +	8.2e+01	5.6e+02 +	8.1e+01	4.8e+02 +	1.1e+02	4.6e+02 +	1.1e+02	4.8e+02 +	1.1e+02	2.9e+02	1.1e+02
f_{13}	5.2e+02 +	9.0e+01	5.9e+02 +	6.0e+01	6.1e+02 +	8.9e+01	6.3e+02 +	9.7e+01	6.0e+02 +	9.7e+01	5.5e+02 +	1.4e+02	4.0e+02	9.1e+01
f_{14}	8.3e+03 ≈	4.3e+02	8.3e+03 ≈	3.5e+02	8.1e+03 ≈	3.3e+02	8.2e+03 ≈	5.9e+02	7.5e+03 -	5.9e+02	8.7e+03 +	7.1e+02	8.3e+03	4.4e+02
f_{15}	8.3e+03 ≈	7.0e+02	8.8e+03 ≈	4.3e+02	8.2e+03 ≈	4.0e+02	8.5e+03 ≈	4.2e+02	8.4e+03 ≈	4.2e+02	9.0e+03 +	7.8e+02	8.6e+03	3.7e+02
f_{16}	2.0e+02 ≈	7.2e-01	2.0e+02 ≈	5.8e-01	2.0e+02 ≈	5.3e-01	2.0e+02 -	6.7e-01	2.0e+02 ≈	6.7e-01	2.0e+02 ≈	7.5e-01	2.0e+02	5.7e-01
f_{17}	1.1e+03 ≈	1.0e+02	1.3e+03 +	1.0e+02	1.3e+03 +	7.4e+01	1.2e+03 +	8.7e+01	1.5e+03 +	8.7e+01	7.7e+02 -	2.9e+02	1.1e+03	1.6e+02
f_{18}	1.2e+03 ≈	7.3e+01	1.4e+03 +	9.1e+01	2.9e+03 +	2.2e+02	1.3e+03 +	7.4e+01	1.7e+03 +	7.4e+01	8.7e+02 -	2.4e+02	1.2e+03	1.9e+02
f_{19}	4.2e+05 +	2.8e+05	8.4e+05 +	3.3e+05	2.2e+06 +	1.3e+06	2.8e+05 +	1.2e+05	7.9e+05 +	1.2e+05	8.6e+05 +	7.5e+05	1.8e+05	2.0e+05
f_{20}	6.1e+02 +	1.5e-04	6.1e+02 ≈	2.8e-08	6.1e+02 ≈	3.0e-09	6.1e+02 ≈	1.5e-02	6.1e+02 ≈	1.5e-02	6.1e+02 +	2.3e-05	1.7e-09	1.6e-02
f_{21}	3.2e+03 -	1.0e+02	3.5e+03 ≈	1.6e+02	3.7e+03 +	2.8e+02	3.2e+03 -	7.4e+01	3.9e+03 +	7.4e+01	4.8e+03 +	3.6e+02	3.5e+03	2.5e+02
f_{22}	9.8e+03 +	5.7e+02	9.8e+03 +	4.1e+02	9.7e+03 +	4.0e+02	9.6e+03 ≈	6.1e+02	9.3e+03 ≈	6.1e+02	1.0e+04 +	2.7e+02	9.4e+03	4.8e+02
f_{23}	9.6e+03 ≈	5.8e+02	1.0e+04 ≈	3.1e+02	9.6e+03 ≈	4.4e+02	1.0e+04 ≈	4.0e+02	9.5e+03 ≈	4.0e+02	1.0e+04 +	8.1e+02	9.7e+03	5.2e+02
f_{24}	1.4e+03 +	2.3e+01	1.4e+03 +	3.5e+01	1.4e+03 +	2.8e+01	1.4e+03 +	3.7e+01	1.3e+03 +	3.7e+01	1.3e+03 +	1.1e+01	1.3e+03	9.5e+00
f_{25}	1.5e+03 +	1.7e+01	1.5e+03 +	1.8e+01	1.5e+03 +	1.1e+01	1.5e+03 +	2.1e+01	1.4e+03 +	2.1e+01	1.5e+03 +	6.6e+00	1.4e+03	1.1e+01
f_{26}	1.6e+03 +	6.5e+01	1.5e+03 -	3.8e+01	1.6e+03 ≈	3.2e+01	1.6e+03 ≈	7.6e+01	1.6e+03 ≈	7.6e+01	1.6e+03 ≈	5.9e+01	1.6e+03	7.3e+01
f_{27}	2.8e+03 +	8.3e+01	2.9e+03 +	5.9e+01	3.2e+03 +	1.6e+02	2.8e+03 +	1.1e+02	2.8e+03 +	1.1e+02	2.8e+03 +	4.6e+01	2.7e+03	7.7e+01
f_{28}	6.8e+03 +	6.8e+02	8.6e+03 +	7.0e+02	6.6e+03 ≈	4.6e+02	7.2e+03 +	7.0e+02	6.8e+03 +	7.0e+02	8.1e+03 +	7.3e+02	6.4e+03	5.2e+02
	+/≈/-; 18/8/2		+/≈/-; 18/9/1		+/≈/-; 18/9/1		+/≈/-; 17/8/3		+/≈/-; 19/8/1		+/≈/-; 22/3/3			

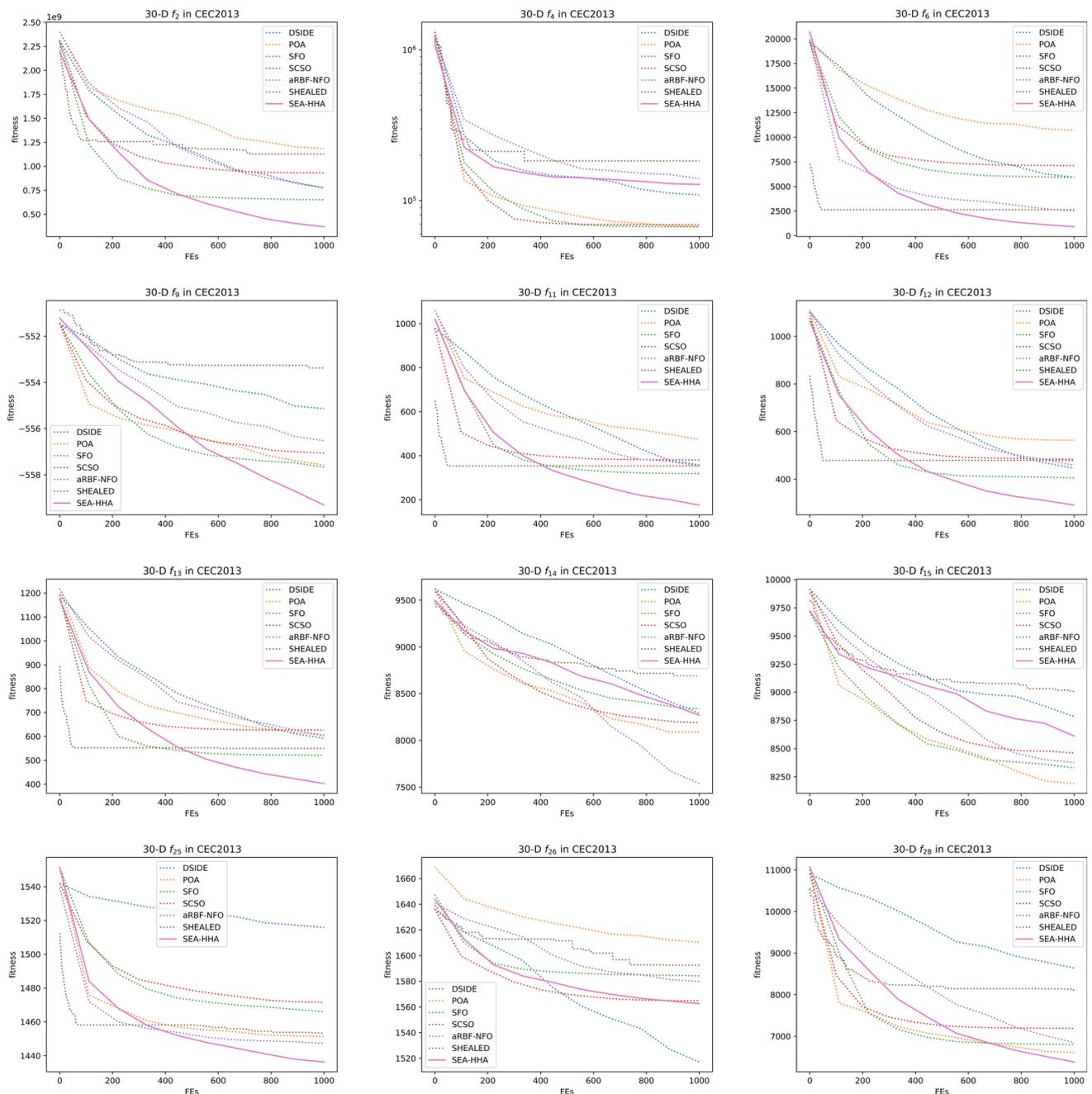


Fig. 6 Convergence graphs of 30-D representative functions in the CEC2013 suite

5 Discussion

5.1 Computational Complexity Analysis of SEA-HHA

In this section, we analyze the computational complexity of SEA-HHA. Supposing the population size is N , the dimension of the problem is D , the maximum iteration is T , and the computational complexity for surrogate-assisted estimation is C . For the sake of simplicity, we analyze each process independently.

- population initialization: $O(N \cdot D)$.
- exploitative search operator: $O(N \cdot D)$.
- explorative search operator: $O(N \cdot D)$.
- surrogate-assisted estimation: $O(C)$.
- mutation search operator: $O(N \cdot D)$.
- selection operator: $O(N)$.

Table 7 Optimization results on the Cantilever Beam Design problem

Alg	worst	mean	best	std
SFO	2.6355	1.6819 \approx	1.3817	0.3057
DSIDE	2.2361	1.9216 +	1.5831	0.2121
POA	4.0492	2.1715 +	1.5684	0.5524
SCSO	2.3415	1.6918 \approx	1.4204	0.2177
aRBF-NFO	5.9947	4.1311 +	2.2430	0.9052
SHEALED	2.1880	1.8121 +	1.3539	0.2142
SEA-HHA	2.1469	1.6504	1.4036	0.2175

Therefore, the total computational complexity of SEA-HHA can be summarized by Eq. (18):

$$O(N \cdot D) + T \cdot (O(\max(N \cdot D, C)) + O(N)) \\ := O(T \cdot \max(N \cdot D, C)). \quad (18)$$

In numerical experiments, the real CPU time of C is larger than $N \cdot D$ since the surrogate-assisted estimation involves the construction of the mathematical model and the sampling process based on the model.

5.2 Performance Analysis of Optimization on CEC2013

From the overview performance on CEC2013 benchmark functions among seven optimization techniques, our proposed SEA-HHA is competitive with these advanced algorithms, and we will analyze the performance of SEA-HHA from two perspectives: exploitation ability and exploration ability.

5.2.1 Exploitation Ability of SEA-HHA

In the CEC2013 suite, functions f_1 through f_5 are unimodal so that they are allowed to evaluate the exploitation ability of optimization algorithms. It's worth noting that in f_1 , SHEALED outperforms our proposed SEA-HHA across three scales, which proves the efficiency and effectiveness of SHEALED in addressing such optimization problems. However, excluding f_1 , SEA-HHA consistently matches or

even outperforms SHEALED, and the superior exploitation ability of SEA-HHA can be observed from these functions. When compared to other optimization algorithms on unimodal functions, our proposal outperforms them in most scenarios. Thus, experimental and statistical results provide adequate support for the excellent exploitation capacity of SEA-HHA.

However, the deterioration of SEA-HHA on f_4 can be observed in Tables 4, 5, 6 and Figs. 5, 6, and this degeneration can be explained by the No Free Lunch Theory [17]. No Free Lunch Theory states that all stochastic optimization algorithms have identical average performance on all possible problems, and if an algorithm is well-performed on a category of the problem, it must compensate for the rest problems. Therefore, we can reasonably infer that the designed LLHs in SEA-HHA may not be good at dealing with this specific problem. Furthermore, as the dimension of the problem increases, the deterioration has been amplified, and we speculate that one reason is due to the curse of dimensionality [59]. As the dimension of the problem increases, the search space will increase exponentially, and the presence of this phenomenon can degenerate the accuracy of the surrogate model rapidly and further affect the quality of estimated solutions.

5.2.2 Exploration Ability of SEA-HHA

Considering that functions f_6 through f_{20} are multimodal, and f_{21} through f_{28} are composition functions, these functions exhibit complex fitness landscapes and many local optima. Thus, they are allowed to evaluate the exploration capacity of optimization techniques. Through the experimental and statistical results in Tables 4, 5, and 6, the superior performance of SEA-HHA can be observed, and we owe this excellent performance to the diverse search strategy and effective surrogate-assisted estimation.

However, we also notice that slight degeneration exists in some benchmark functions such as f_{17} , f_{18} , and f_{21} . These types of degeneration happen when the dimension of the problem increases, and we reasonably believe that this degeneration is also caused by the curse of dimensionality,

Table 8 Optimum found by optimization techniques on the Cantilever Beam Design problem

Alg	x_1	x_2	x_3	x_4	x_5	cons.	obj
SFO	6.534790	4.653559	4.241771	4.377597	2.335527	-0.0033	1.3817
DSIDE	9.409405	4.737502	5.103161	3.361869	2.758319	-0.2039	1.5831
POA	6.426340	4.766779	7.823874	3.184123	2.93497	-0.1324	1.5684
SCSO	5.745827	5.346359	4.685947	3.114073	3.872125	-0.0026	1.4204
aRBF-NFO	9.324028	7.978926	4.897215	8.857559	4.889085	-0.6715	2.2430
SHEALED	5.752486	5.717425	4.170125	3.936549	2.121631	-0.0001	1.3539
SEA-HHA	5.3868787	5.417096	5.144122	3.389277	3.157643	-0.0258	1.4036

Table 9 Optimization results on the Tension/Compression Spring Design problem. If more than 1 of 30 trial runs does not find a feasible solution, the worst, mean, and std cannot be calculated and we manually fill them with *Nan*. Statistical analysis is also meaningless

Alg	worst	mean	best	std
SFO	<i>Nan</i>	<i>Nan</i>	0.0132	<i>Nan</i>
DSIDE	0.1551	0.0433 \approx	0.0138	0.0286
POA	0.0603	0.0285 \approx	0.0133	0.0123
SCSO	<i>Nan</i>	<i>Nan</i>	0.0131	<i>Nan</i>
aRBF-NFO	<i>Nan</i>	<i>Nan</i>	0.0204	<i>Nan</i>
SHEALED	0.0420	0.0230 \approx	0.0130	0.0085
SEA-HHA	0.0548	0.0314	0.0135	0.0100

which further affects the quality of solutions estimated by the approximation model, and how to overcome this issue will be considered in our future research.

5.3 Performance Analysis of Optimization on Three Engineering Problems

These engineering optimization problems contain multiple constraints and complex fitness landscapes, the optimization performance on these problems can reflect the ability of the algorithm to deal with the real-world tasks. Besides, this research focuses on solving EOPs, and only 1000 FEs are assigned for each task optimization, which is a severe challenge for optimization techniques.

Statistical results in Tables 7, 9, and 11 show that SEA-HHA at least is not inferior to any optimization method for any problem and can outperform in some problems (e.g. compared with DSIDE, aRBF-NFO, and SHEALED in Cantilever Beam Design). Another advantage of SEA-HHA is that the optimization process is stable even under the FEs limitation. In the Tension/Compression Spring Design problem, SEA-HHA can find a feasible solution in any independent trial run while SFO, SCSO, and aRBF-NFO can not at least once. In the Pressure Vessel Design, the worst solution found by SEA-HHA is apparently better than the compared methods and the standard deviation is also small. These experimental results reveal the excellent exploration and exploitation abilities of SEA-HHA in engineering optimization problems, which has great potential to deal with real-world applications.

5.4 Potential and Future Topics

Through the above analyses, we have known that our proposed SEA-HHA has broad prospects for dealing with EOPs. However, as a new optimization technique, there are still many aspects for further improvement. Here, we list some open topics.

5.4.1 More Powerful and Efficient Operators

Three exploration and exploitation strategies and one mutation strategy are employed as our basic search strategy archive. Without complex parameter tuning, our designed search strategies are the most common and easy-implemented. Meanwhile, Cruz et al. [24] summarizes ten search operations from well-known meta-heuristics such as Random Sample, Random Walk, Firefly Dynamic, Gravitational Search, and so on, which can also be absorbed into our proposed SEA-HHA to strengthen the diversity of the search strategy.

5.4.2 Dealing with High-Dimensional and Large-Scale EOPs

We implement the optimization experiments of SEA-HHA on relatively low-dimensional problems and have achieved satisfactory performance. However, we also observed the deterioration of SEA-HHA as the dimension of the problem increases, and how to alleviate the negative effect of the curse of dimensionality is a challenging topic. Inspired by the divide-and-conquer, cooperative coevolution (CC) [60] framework is a mature approach to solving high-dimensional and large-scale optimization problems, which divides the original problems into several sub-components and optimizes them separately. The remaining problem is how to decompose the original problems properly. To the best of our knowledge, merged differential grouping (MDG) [61] is the lightest decomposition method that only consumes $6.41e3$ for CEC2013 large-scale benchmark functions on average with high accuracy. Therefore, the collaboration of MDG and our proposed optimizer SEA-HHA is promising to deal with high-dimensional and large-scale EOPs.

5.4.3 Determining the Optimization Sequence More Intelligently

As our first attempt to introduce the surrogate-assisted technique to the hyper-heuristic algorithm, we simply determine the sequence of heuristics by probabilistic selection function with pre-defined probabilities in this paper. Actually, many effective methodologies can contribute to the optimization sequence construction, such as Genetic Algorithm (GA) [62, 63], Reinforcement Learning techniques [28, 64], improvement-based choice function [65, 66], and so on. In our future research, we want to design a more flexible and intelligent method to determine the construction of the optimization sequence. A primary idea is to evaluate the generated solutions of different archives by the surrogate model and dynamically adjust the selected probability, which can fully utilize the surrogate model and is computationally cheap for EOPs.

Table 10 Optimum found by optimization techniques on the Tension/Compression Spring Design problem

Alg	x_1	x_2	x_3	cons.1	cons.2	cons.3	cons.4	obj.
SFO	0.053517	0.393260	9.745187	-0.0065	-0.0184	-3.9872	-0.7021	0.0132
DSIDE	0.056095	0.466966	7.392016	-0.0589	-0.0094	-3.8877	-0.6512	0.0138
POA	0.050000	0.313423	15.000000	-0.0293	-0.0099	-3.7658	-0.7577	0.0133
SCSO	0.053946	0.412892	8.965602	-0.0380	-0.0011	-3.9571	-0.6887	0.0131
aRBF-NFO	0.066801	0.812510	3.645216	-0.3678	-0.0301	-2.8987	-0.4137	0.0204
SHEALED	0.050000	0.315625	14.575956	-0.0214	-0.0044	-3.8362	-0.7562	0.0130
SEA-HHA	0.058430	0.540130	5.358699	-0.0091	-0.0023	-4.2493	-0.6009	0.0135

Table 11 Optimization results on the Pressure Vessel Design problem

Alg	worst	mean	best	std
SFO	182308.88	37182.20 \approx	6811.07	53,171.32
DSIDE	286301.67	71823.97 +	16,446.54	49,198.51
POA	122179.92	60392.19 +	16,118.21	31,410.21
SCSO	240564.27	85641.49 +	21,920.95	50,317.78
aRBF-NFO	19,2761.70	25,873.81 \approx	7117.16	38,378.12
SHEALED	155,058.38	42,531.62 +	6811.27	31,670.85
SEA-HHA	29877.51	13509.01	6648.46	4690.01

Table 12 Optimum found by optimization techniques on the Pressure Vessel Design problem

Alg	x_1	x_2	x_3	x_3	cons.1	cons.2	cons.3	cons.4	obj.
SFO	0.97065	0.56005	48.77169	108.81143	-0.02	-0.09	-3080.51	-131.18	6811.07
DSIDE	1.57724	1.31905	43.38856	195.95323	-0.73	-0.90	-205066.52	-44.04	16446.54
POA	1.78198	0.93542	56.50870	99.64362	-0.69	-0.39	-459457.48	-140.35	16118.21
SCSO	1.42960	0.85228	72.49971	155.19587	-0.03	-0.16	-2862965.50	-84.80	21920.95
aRBF-NFO	1.14105	0.58046	56.86348	51.92381	-0.04	-0.03	-1626.53	-188.07	7117.10
SHEALED	1.00300	0.46802	47.03495	123.86394	-0.09	-0.01	-730.81	-116.13	6811.27
SEA-HHA	1.00388	0.53724	51.45276	87.45941	-0.01	-0.04	-1978.28	-152.54	6648.46

6 Conclusion

In this paper, we propose a novel surrogate ensemble-assisted hyper-heuristic algorithm (SEA-HHA) to solve EOPs. In the high-level component design, the random selection function based on the pre-defined probabilities is adopted to dominate the LLHs. In the low-level component, we design four search strategy archives as LLHs: exploration strategy archive, exploitation strategy archive, surrogate-assisted estimation archive, and mutation strategy archive, each search strategy is easy-implemented. Besides, in the surrogate-assisted estimation archive, three different data selection principles are applied for model construction: *All Data*, *Recent Data*, and *Neighbor*, which correspond to the global and local concepts, and the most accurate model constructed by PR, SVR, and GPR is utilized to estimate the promising solutions.

In the numerical experiments, we compare our proposed SEA-HHA with six advanced optimization techniques on

the CEC2013 benchmark functions and three popular engineering optimization problems. Experimental and statistical results show that SEA-HHA has broad prospects for solving EOPs.

At the end of this paper, we list some open topics to further develop the SEA-HHA. In the future, we will focus on combining the learning-based methods to determine the optimization sequence more intelligently and extend SEA-HHA to solve high-dimensional EOPs.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number JP20K11967, JST SPRING Grant Number JPMJSP21119, and Interdisciplinary large-scale computer system (Supercomputing system), Information Initiative Center, Hokkaido University.

Author Contributions RZ: Conceptualization, methodology, investigation, writing-original draft, writing-review & editing, and funding acquisition. JY: formal analysis, investigation, methodology, and writing-review & editing. CZ: formal analysis, resources, and writing-review & editing. MM: methodology, writing-review & editing, and

project administration. All authors have read and agreed to the published version of the manuscript.

Funding This work was supported by JSPS KAKENHI Grant number JP20K11967 and JST SPRING Grant number JPMJSP2119.

Data Availability The source code of this research can be downloaded from <https://github.com/RuiZhong961230/SEA-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.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Al-Sahaf, H., Bi, Y., Chen, Q., Lensen, A., Mei, Y., Sun, Y., Tran, B., Xue, B., Zhang, M.: A survey on evolutionary machine learning. *J. R. Soc. N. Z.* **49**(2), 205–228 (2019). <https://doi.org/10.1080/03036758.2019.1609052>
- Wang, Z., Sobey, A.: A comparative review between genetic algorithm use in composite optimisation and the state-of-the-art in evolutionary computation. *Compos. Struct.* **233**, 111739 (2020). <https://doi.org/10.1016/j.compstruct.2019.111739>
- Tan, K.C., Feng, L., Jiang, M.: Evolutionary transfer optimization - a new frontier in evolutionary computation research. *IEEE Comput. Intell. Mag.* **16**(1), 22–33 (2021). <https://doi.org/10.1109/MCI.2020.3039066>
- Fernandes Junior, F.E., Yen, G.G.: Particle swarm optimization of deep neural networks architectures for image classification. *Swarm Evol. Comput.* **49**, 62–74 (2019). <https://doi.org/10.1016/j.swevo.2019.05.010>
- Telikani, A., Gandomi, A.H., Shahbahrami, A.: A survey of evolutionary computation for association rule mining. *Inf. Sci.* **524**, 318–352 (2020). <https://doi.org/10.1016/j.ins.2020.02.073>
- Zhao, F., He, X., Wang, L.: A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of no-wait flow-shop problem. *IEEE Trans. Cybern.* **51**(11), 5291–5303 (2021). <https://doi.org/10.1109/TCYB.2020.3025662>
- Chatterjee, T., Chakraborty, S., Chowdhury, R.: A critical review of surrogate assisted robust design optimization. *Arch. Comput. Methods Eng.* **26**, 245–274 (2019). <https://doi.org/10.1007/s11831-017-9240-5>
- Gu, H., Wang, H., Jin, Y.: Surrogate-assisted differential evolution with adaptive multi-subspace search for large-scale expensive optimization. *IEEE Trans. Evol. Comput.* (2022). <https://doi.org/10.1109/TEVC.2022.3226837>
- Wang, Y., Lin, J., Liu, J., Sun, G., Pang, T.: Surrogate-assisted differential evolution with region division for expensive optimization problems with discontinuous responses. *IEEE Trans. Evol. Comput.* **26**(4), 780–792 (2022). <https://doi.org/10.1109/TEVC.2021.3117990>
- Dong, H., Dong, Z.: Surrogate-assisted grey wolf optimization for high-dimensional, computationally expensive black-box problems. *Swarm Evol. Comput.* **57**, 100713 (2020). <https://doi.org/10.1016/j.swevo.2020.100713>
- Nishihara, K., Nakata, M.: Surrogate-assisted differential evolution with adaptation of training data selection criterion. In: 2022 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1675–1682 (2022). <https://doi.org/10.1109/SSCI51031.2022.10022105>
- Wang, W., Liu, H.-L., Tan, K.C.: A surrogate-assisted differential evolution algorithm for high-dimensional expensive optimization problems. *IEEE Trans. Cybern.* **53**(4), 2685–2697 (2023). <https://doi.org/10.1109/TCYB.2022.3175533>
- Cai, X., Ruan, G., Yuan, B., Gao, L.: Complementary surrogate-assisted differential evolution algorithm for expensive multi-objective problems under a limited computational budget. *Inf. Sci.* **632**, 791–814 (2023). <https://doi.org/10.1016/j.ins.2023.03.005>
- Wang, H., Jin, Y., Doherty, J.: Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Trans. Cybern.* **47**(9), 2664–2677 (2017). <https://doi.org/10.1109/TCYB.2017.2710978>
- Xiang, X., Tian, Y., Xiao, J., Zhang, X.: A clustering-based surrogate-assisted multiobjective evolutionary algorithm for shelter location problem under uncertainty of road networks. *IEEE Trans. Ind. Inf.* **16**(12), 7544–7555 (2020). <https://doi.org/10.1109/TII.2019.2962137>
- Wakjira, T.G., Ibrahim, M., Ebead, U., Alam, M.S.: Explainable machine learning model and reliability analysis for flexural capacity prediction of rc beams strengthened in flexure with frcm. *Eng. Struct.* **255**, 113903 (2022). <https://doi.org/10.1016/j.engstruct.2022.113903>
- 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>
- Dowland, 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>
- 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
- 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
- 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
- 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>
- Kheiri, A., Keedwell, E.: Selection hyper-heuristics. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion. GECCO '22*, pp. 983–996. Association for Computing Machinery, New York, NY, USA (2022). <https://doi.org/10.1145/3520304.3533655>

24. 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>
25. Liang, J., Qu, B., Suganthan, P., Hernández-Díaz, A.: Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization. Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China (2013)
26. Fisher, H.: Probabilistic learning combinations of local job-shop scheduling rules. *Ind. Sched.*, 225–251 (1963)
27. 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>
28. 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>
29. Zhao, F., Di, S., Cao, J., Tang, J.: Jonrinaldi: a novel cooperative multi-stage hyper-heuristic for combination optimization problems. *Complex Syst. Model. Simul.* **1**(2), 91–108 (2021). <https://doi.org/10.23919/CSMS.2021.0010>
30. Qin, W., Zhuang, Z., Huang, Z., Huang, H.: A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem. *Comput. Ind. Eng.* **156**, 107252 (2021). <https://doi.org/10.1016/j.cie.2021.107252>
31. Zhang, C., Zhao, Y., Leng, L.: A hyper-heuristic algorithm for time-dependent green location routing problem with time windows. *IEEE Access* **8**, 83092–83104 (2020). <https://doi.org/10.1109/ACCESS.2020.2991411>
32. Ostertagová, E.: Modelling using polynomial regression. *Proc. Eng.* **48**, 500–506 (2012). <https://doi.org/10.1016/j.proeng.2012.09.545>. (Modelling of Mechanical and Mechatronics Systems)
33. Åke Björck: Least squares methods. *Handbook of Numerical Analysis*, vol. 1, pp. 465–652. Elsevier (1990). [https://doi.org/10.1016/S1570-8659\(05\)80036-5](https://doi.org/10.1016/S1570-8659(05)80036-5). <https://www.sciencedirect.com/science/article/pii/S1570865905800365>
34. Vapnik, V.: The nature of statistical learning theory (1995). <https://doi.org/10.1007/978-1-4757-2440-0>
35. Awad, M., Khanna, R.: Support Vector Regression, pp. 67–80. Apress, Berkeley (2015). https://doi.org/10.1007/978-1-4302-5990-9_4
36. Yang, D., Zhang, X., Pan, R., Wang, Y., Chen, Z.: A novel gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve. *J. Power Sources* **384**, 387–395 (2018). <https://doi.org/10.1016/j.jpowsour.2018.03.015>
37. Liu, D., Pang, J., Zhou, J., Peng, Y., Pecht, M.: Prognostics for state of health estimation of lithium-ion batteries based on combination gaussian process functional regression. *Microelectron. Reliab.* **53**(6), 832–839 (2013). <https://doi.org/10.1016/j.microrel.2013.03.010>
38. Rasmussen, C.E., Nickisch, H.: Gaussian processes for machine learning (gpml) toolbox. *J. Mach. Learn. Res.* **11**, 3011–3015 (2010)
39. Storn, R.: On the usage of differential evolution for function optimization. In: Proceedings of North American Fuzzy Information Processing, pp. 519–523 (1996). <https://doi.org/10.1109/NAFIPS.1996.534789>
40. Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: algorithm and applications. *Futur. Gener. Comput. Syst.* **97**, 849–872 (2019). <https://doi.org/10.1016/j.future.2019.02.028>
41. Braik, M.S.: Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems. *Expert Syst. Appl.* **174**, 114685 (2021). <https://doi.org/10.1016/j.eswa.2021.114685>
42. Abdollahzadeh, B., Soleimani Gharehchopogh, F., Mirjalili, S.: Artificial gorilla troops optimizer: a new nature-inspired meta-heuristic algorithm for global optimization problems. *Int. J. Intell. Syst.* (2021). <https://doi.org/10.1002/int.22535>
43. Trojovská, E., Dehghani, M., Trojovský, P.: Zebra optimization algorithm: a new bio-inspired optimization algorithm for solving optimization algorithm. *IEEE Access* **10**, 49445–49473 (2022). <https://doi.org/10.1109/ACCESS.2022.3172789>
44. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* **13**(2), 398–417 (2009). <https://doi.org/10.1109/TEVC.2008.927706>
45. 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>
46. Chickermane, H., Gea, H.C.: Structural optimization using a new local approximation method. *Int. J. Numer. Methods Eng.* **39**(5), 829–846 (1996). [https://doi.org/10.1002/\(SICI\)1097-0207\(19960315\)39:5<829::AID-NME884>3.0.CO;2-U](https://doi.org/10.1002/(SICI)1097-0207(19960315)39:5<829::AID-NME884>3.0.CO;2-U)
47. Arora, J.S.: Copyright. In: Arora, J.S. (ed.) *Introduction to Optimum Design (Fourth Edition)*, 4th edn. Academic Press, Boston (2017). <https://doi.org/10.1016/B978-0-12-800806-5.00025-1>
48. 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>
49. Bayzidi, H., Talatahari, S., Saraee, M., Lamarche, C.-P.: Social network search for solving engineering optimization problems. *Comput. Intell. Neurosci.* **2021**, 1–32 (2021). <https://doi.org/10.1155/2021/8548639>
50. Shadravan, S., Naji, H.R., Bardsiri, V.K.: The sailfish optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Eng. Appl. Artif. Intell.* **80**, 20–34 (2019). <https://doi.org/10.1016/j.engappai.2019.01.001>
51. Zhong, X., Cheng, P.: An improved differential evolution algorithm based on dual-strategy. *Math. Probl. Eng.* **2020**, 1–14 (2020). <https://doi.org/10.1155/2020/9767282>
52. Trojovský, P., Dehghani, M.: Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications. *Sensors* (2022). <https://doi.org/10.3390/s22030855>
53. 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>
54. Yu, M., Liang, J., Zhao, K., Wu, Z.: An arbf surrogate-assisted neighborhood field optimizer for expensive problems. *Swarm Evol. Comput.* **68**, 100972 (2022). <https://doi.org/10.1016/j.swevo.2021.100972>
55. Wu, Z., Chow, T.W.: Neighborhood field for cooperative optimization. *Soft. Comput.* **17**(5), 819–834 (2013). <https://doi.org/10.1007/s00500-012-0955-9>
56. Liu, Y., Wang, H.: Surrogate-assisted hybrid evolutionary algorithm with local estimation of distribution for expensive mixed-variable optimization problems. *Appl. Soft Comput.* **133**, 109957 (2023). <https://doi.org/10.1016/j.asoc.2022.109957>
57. Holm, S.: A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **6**(2), 65–70 (1979)
58. 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](https://doi.org/10.1016/S0045-7825(01)00323-1)
59. Köppen, M.: The curse of dimensionality. In: 5th Online World Conference on Soft Computing in Industrial Applications (WSC5), vol. 1, pp. 4–8 (2000)
60. Potter, M.A., De Jong, K.A.: A cooperative coevolutionary approach to function optimization. In: *Lecture Notes in Computer*

- Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 866 LNCS, 249–257 (1994)
61. Ma, X., Huang, Z., Li, X., Wang, L., Qi, Y., Zhu, Z.: Merged differential grouping for large-scale global optimization. *IEEE Trans. Evol. Comput.* **26**(6), 1439–1451 (2022). <https://doi.org/10.1109/TEVC.2022.3144684>
 62. FANG, H.: A promising genetic algorithm approach to job-shop scheduling, rescheduling, and open-shop scheduling problems. *Proc. the 5th International Conference on Genetic Algorithms*, 375–382 (1993)
 63. Hart, E., Ross, P., Nelson, J.: Solving a real-world problem using an evolving heuristically driven schedule builder. *Evol. Comput.* **6**(1), 61–80 (1998). <https://doi.org/10.1162/evco.1998.6.1.61>
 64. 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>
 65. Choong, S.S., Wong, L.-P., Lim, C.P.: An artificial bee colony algorithm with a modified choice function for the traveling salesman problem. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 357–362 (2017). <https://doi.org/10.1109/SMC.2017.8122629>
 66. Choong, S.S., Wong, L.-P., Lim, C.P.: An artificial bee colony algorithm with a modified choice function for the traveling salesman problem. *Swarm Evol. Comput.* **44**, 622–635 (2019). <https://doi.org/10.1016/j.swevo.2018.08.004>

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