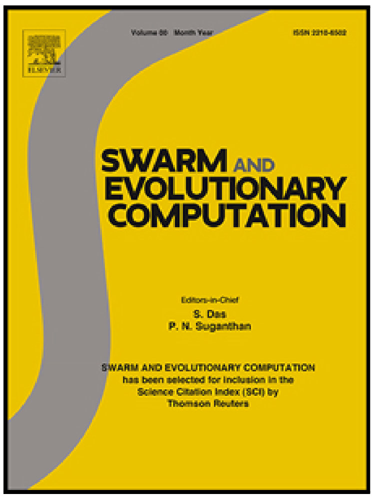
Volume 00 Month Year
ISSN 2210-6502
SWARM AND EVOLUTIONARY COMPUTATION
Editors-in-Chief
S. Das
P. N. Suganthan
SWARM AND EVOLUTIONARY COMPUTATION has been selected for inclusion in the Science Citation Index (SCI) by Thomson Reuters

Differential evolution with evolutionary scale adaptation

Sheng Xin Zhang , Xin Rou Hu , Shao Yong Zheng

PII: S2210-6502(24)00014-2
DOI: https://doi.org/10.1016/j.swevo.2024.101481
Reference: SWEVO 101481

To appear in: *Swarm and Evolutionary Computation*

Received date: 9 September 2023
Revised date: 13 January 2024
Accepted date: 15 January 2024

Differential evolution with evolutionary scale adaptation

Sheng Xin Zhang [a], Xin Rou Hu [a], Shao Yong Zheng [b]

[a] College of Information Science and Technology, Jinan University, Guangzhou, China
[b] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China

Corresponding author: Sheng Xin Zhang (zhangsx@jnu.edu.cn) and
Shao Yong Zheng (zhengshaoy@mail.sysu.edu.cn)

Abstract —The performance of differential evolution (DE) algorithm heavily depends on the evolutionary scale, which is controlled by the generation operations including mutation, crossover and the control parameters including mutation factor and crossover rate. Adjusting the evolutionary scale to suit different types of problems is a critical yet challenging open question in DE research. To efficiently address this issue, this paper proposes a novel DE based on evolutionary scale adaptation, termed as ESADE. First, a successful scale estimation mechanism is proposed to measure the appropriate evolutionary scale by utilizing the successful evolutionary feedback from the solution space provided by the trial vectors and the target vectors. With the appropriate evolutionary scale, an evolutionary scale adaptation mechanism pre-selects the closest or the farthest trial vector from each target vector, which corresponds to a small or large evolutionary scale respectively to match the search requirements of different evolutionary stages. The effectiveness of the ESA method is demonstrated by performance comparisons with each single baseline strategy, state-of-the-art DEs and state-of-the-art multi-strategy methods on 29 benchmark functions with three dimensionalities. Further applications on several real-world optimization problems also reveal the competitive performance of ESADE.

Keywords —Evolutionary scale adaptation, multi-strategy, mutation, differential evolution, global optimization

1. Introduction

Differential evolution (DE) proposed by Storn and Price [1] has become one of the most efficient evolutionary algorithms for solving continuous global optimization problems [2][3][4][5][6]. With its simplicity, straightforward implementation, and impressive computational efficiency, DE has garnered widespread utilization across a multitude of practical domains [2][3] from engineering and finance to data science and artificial intelligence. Inspired by biological evolution, DE predominantly comprises three genetic operations: mutation, crossover, and selection, which are based on a random population. Mutation first perturbs a base vector with one or more differential vectors to generate a mutant vector, introducing new genetic materials. Crossover then forces gene exchanges between the mutant vector and the current vector to generate a trial vector, which controls the percentage of the utilization of the current information from the current vector and the new information from the mutant vector. Selection finally determines the fitter one between the trial vector and the current vector for the next generation. Among them, mutation and crossover determine the generation of the trial vector from the current vector and significantly influence the performance.

In the past two decades, DE has been enhanced by developing new mutation [7][8][9] and crossover [10][11] operations and better adjusting the control parameters [9][12][13][14] including mutation factor, crossover rate and population size. Initially, extensive efforts were dedicated to fine-tuning the control parameter configurations to achieve more desirable results. However, researchers soon realized that a fixed parameter setting could not yield optimal solutions for all types of problems [9]. Moreover, the exhaustive trial-and-error process required to determine the best configuration was time-consuming and laborious. To overcome these limitations, various adaptive parameter methods have been developed [12]. These methods offer a flexible and runtime-dependent approach to parameter tuning. Instead of relying on a predetermined set of values, they enable an algorithm to dynamically adjust its own parameters based on the evolution of the population.

As optimization problems become increasingly diverse and with the emergence of challenges like multimodal problems, it has become evident that focusing solely on adapting control parameters is insufficient. In response, researchers have also paid attention to improving the efficiency of the operators [7]. The mutation and crossover operators play crucial roles in exploring and exploiting the search space to navigate towards optimal solutions. Various operators [8][9][10][11] have been proposed to enhance exploration and exploitation capabilities. Moreover, when dealing with complex optimization problems, particularly those with varying characteristics at different stages, an algorithm needs to meet diverse requirements. In the face of these challenges, it is crucial to have the flexibility to adapt search strategies. For example, in the initial exploration phase, when the search space is vast and relatively unexplored, an algorithm may prioritize global exploration to identify potential promising regions. Conversely, in the later stages, when localized regions of interest are identified or when convergence towards the optimal solution is desired, an algorithm might shift towards a more exploitative strategy focusing on intensifying the search within these regions. Additionally, certain problems may require a fine balance between exploration and exploitation throughout the entire optimization process to avoid getting stuck in local optima which may result in poor solutions.

Evolutionary experiments in DE generate trial vectors that not only propagate better individuals to the next generation but also provide valuable information for further adjusting search strategies to enhance efficiency. In general, we summarize the existing strategy adaptation mechanisms into three categories. The first category is based on strategy competition and the derived successful experiences [15][16][17]. Various competition mechanisms have been developed, such as memory-based [15], multi-population ensemble-based [16] and adaptive parameter control method-based [17] ones. The basic idea is that a strategy that generates successful trial vectors (i.e., offspring) is promising and should gain more opportunities for further generating offspring. The second category involves adaptation methods based on the observation of certain characteristics, such as fitness values [18] or population states [19]. With the observation, convergence and diversity requirements are identified, and strategies could be assigned accordingly. The third category adapts strategies based on pre-selection rule. Multiple candidates are generated by multiple strategies first, and then one of the candidates is filtered as the trial vector by rules such as fitness evaluation [20], surrogate model [21] and similarity selection rule [22].

Although significant advances have been achieved by these works, the search feedback particularly the evolutionary scale in the solution space which could be utilized for performance improvement has not been fully utilized to adjust the search strategy. This motivates the proposal of the evolutionary scale adaptation (ESA) in this paper. In ESA, the successful evolutionary scale between the target vectors and trial vectors is first measured, which is normalized as an evolutionary scale indicator. A large and a small indicator value means that a large and a small evolutionary scale tends to be more promising respectively and will then be considered in the offspring generation. Experimental results demonstrate the effectiveness of ESA for strategy adaptation and its superior performance over several classic and state-of-the-art strategy adaptation methods.

The rest of this paper is organized as follows. Section 2 briefly describes the basic procedures of DE, along with a review of multi-strategy adaptation methods. The details of the proposed approach are described in Section 3. In Section 4, we compare and analyze the experimental results on benchmark functions and real-world problems to assess the performance and effectiveness. Finally, Section 5 is devoted to conclusions.

2. Background

2.1 Differential evolution

At the beginning, DE initializes a population according to Eq. (1):

$$x_{i,j,0} = \underline{x}_j + rand_{i,j}(0,1)\cdot\left(\overline{x}_j - \underline{x}_j\right), i = 1,\ 2, \ldots, NP \qquad (1)$$

where $x_{i,j,0}$ represents the j-th dimension of the i-th individual at the initial generation, $\underline{x}_j$ and $\overline{x}_j$ are the lower and upper bounds of the j-th dimension respectively and $rand_{i,j}(0,1)$ is a uniformly distributed random number within [0, 1]. There are NP individuals, and each individual is a D-dimensional vector.

Mutation: Mutation is performed to generate a mutant vector $\mathbf{V}_{i,g}$ by multiplying a scaling factor F with one or more differential vectors, which is then added to a base vector. Several common mutation strategies are as follows:

"DE/rand/1":

$$\mathbf{V}_{i,g} = \mathbf{X}_{r_1,g} + F\cdot\left(\mathbf{X}_{r_2,g} - \mathbf{X}_{r_3,g}\right) \qquad (2)$$

"DE/best/1":

$$\mathbf{V}_{i,g} = \mathbf{X}_{best,g} + F\cdot\left(\mathbf{X}_{r_1,g} - \mathbf{X}_{r_2,g}\right) \qquad (3)$$

"DE/current-to-*p*best/1":

$$\mathbf{V}_{i,g} = \mathbf{X}_{i,g} + F\cdot\left(\mathbf{X}_{pbest,g} - \mathbf{X}_{i,g}\right) + F\cdot\left(\mathbf{X}_{r_1,g} - \mathbf{X}_{r_2,g}\right) \qquad (4)$$

where r_1, r_2, r_3 and i are mutually different random integers from the range of [1, NP]. $\mathbf{X}_{best,g}$ and $\mathbf{X}_{pbest,g}$ are the fittest and one of the top 100p% fittest solutions from the current population. "DE/rand/1" is based on a random search variation, "DE/best/1" focuses on searching around the best solution while "DE/current-to-*p*best/1" guides the target solution towards $\mathbf{X}_{pbest,g}$.

Crossover: Following mutation, a trial vector $\mathbf{U}_{i,\,g}$ is generated by exchanging the dimensions between the mutant vector $\mathbf{V}_{i,\,g}$ and the target (current) vector $\mathbf{X}_{i,\,g}$ with a crossover operation. The classic binomial crossover is as follows:

$$u_{i,j,g} = \begin{cases} v_{i,j,g} & \text{if } rand_{i,j}(0,1) \le CR \text{ or } j = j_{rand} \\ x_{i,j,g} & \text{otherwise} \end{cases} \qquad (5)$$

where the crossover rate CR determines the probability of replacing the dimensions of the target vector with the corresponding dimensions of the mutant vector. The presence of the random index j_{rand} which is a random integer from [1, D] ensures that at least one dimension of $\mathbf{U}_{i,\,g}$ comes from $\mathbf{V}_{i,\,g}$.

Selection: After crossover, the fitness $f(\mathbf{U}_{i,\,g})$ of each trial vector is evaluated. During the selection, $f(\mathbf{U}_{i,\,g})$ is compared with the fitness $f(\mathbf{X}_{i,\,g})$ of the target vector. The better one will be retained as a target vector for the next generation. The selection operation for a minimization problem is as follows:

$$\mathbf{X}_{i,g+1} = \begin{cases} \mathbf{U}_{i,g} & \text{if } f\left(\mathbf{U}_{i,g}\right) \le f\left(\mathbf{X}_{i,g}\right) \\ \mathbf{X}_{i,g} & \text{if } f\left(\mathbf{U}_{i,g}\right) > f\left(\mathbf{X}_{i,g}\right) \end{cases} \qquad (6)$$

2.2 Multi-strategy adaptation

Different types of strategy adaptation methods have been proposed for DE to adjust the search behavior at different evolutionary stages to better fit the optimization, some of which are summarized as follows:

(1) Multi-strategy method based on competition

This type of method adjusts strategies based on the success experience of strategies involved in the competition of generating successful offspring. In strategy adaptive DE (SaDE) [15], four different strategies are selected to form a candidate pool, and a suitable mutation strategy is chosen based on the success rate obtained from a learning period. In ensemble of parameter and strategy DE (EPSDE) [23] , there are parameter and strategy candidate pools respectively, which are randomly configured together, and if successful, they are retained. In multi-population ensemble DE (MPEDE) [16], the population is divided into sub-populations, with each sub-population applying different mutation strategies, and the strategy with the most fitness improvements is assigned to a reward sub-population. In strategy adaptation mechanism (SaM) [17], each strategy is assigned an index, and the strategy adaptation is treated as a parameter adaptation problem. In multiple variants coordination (MVC-DE) [24], different DE variants are adaptively utilized in different evolutionary segments. In chaotic local search-based DE (CLSDE) [25], different kinds of chaotic local search are adaptively employed in the evolutionary process.

(2) Multi-strategy method based on observation

This type of method allocates strategies based on the information derived from the population, such as the fitness and the evolutionary status of the population. Fitness ranking has been successfully applied for developing new mechanisms. In [18], Tang *et al*. proposed an individual-dependent DE (IDE), which includes an individual-dependent parameter mechanism for setting control parameters and an individual-dependent mutation strategy mechanism for adjusting mutation strategies. In [26], Cui *et al*. presented a self-adaptive DE where the population is divided into sub-populations based on fitness, and each sub-population owns a specific mutation strategy. In [27], Mohamed *et al*. introduced two novel mutation strategies, and the selection of the vectors in the mutation strategies is based on fitness ranking. In historical and heuristic-based DE (HHDE) [28], the assignment of mutation strategy for each solution depends on both successful experience and fitness ranking. In multi-layer competitive-cooperative DE (MLCCDE) [29], strategies and computing resources are allocated according to the fitness ranking. In objective and dimension feedback DE (ODFDE) [30], search information collected from the objective and dimension spaces is utilized to assign strategies at the dimensional level. Besides fitness, population information has also been utilized for adjusting search strategies. In neighborhood-based DE (NDE) [31], strategies are employed according to the neighborhood information of each current solution. In explicit adaptive DE (EaDE) [32], the comparison of fitness improvements between superior and inferior individuals serves as an indicator for strategy adaptation.

(3) Multi-strategy method based on pre-selection rule

This type of method generates multiple trial vectors by utilizing multiple strategies, and one of them is selected as the final trial vector using pre-selection rules. In composite DE (CoDE) [20], three trial vectors are randomly generated by combining strategies and parameters from candidate pools, and the best one in terms of fitness is selected and compared with the target vector to update the population. In cheap surrogate model-based DE (CSM-DE) [21], multiple trial vectors are evaluated using a cheap surrogate model, and one of them is selected as the final trial vector. In underestimation-based multi-mutation DE (UM-DE) [33], offspring selection relies on the underestimation model. In selective-candidate framework with a similarity selection rule-based DE (SCSS-DE) [22], the selection of the final trial vectors is based on the fitness ranking and the Euclidean distance between target vectors and trial vectors. In global-local cooperate DE (GLCDE) [34], trial vectors are generated using different types of strategies and evaluated with an improved underestimation model. In ensembling populations-based JADE (EJADE) [35], two sets of mutation and crossover operators are employed to generate offspring, and the fitter one is filtered as the final trial vector.

3 Proposed method

The proposed ESADE consists of two mechanisms: the successful scale estimation (SSE) mechanism and the evolutionary scale adaptation (ESA) mechanism.

3.1 Successful scale estimation

The successful scale estimation (SSE) mechanism measures the successful scale of the evolutionary process by evaluating how far the successful trial vectors deviate from their target vectors. Specifically, it calculates the Euclidean distance between the trial vector and its corresponding target vector at each generation as follows:

$$d_{i,g} = \| \mathbf{U}_{i,g} - \mathbf{X}_{i,g} \|, \; i = 1, 2, ..., NP \tag{7}$$

Then, the *NP* distance values are sorted in ascending order and assigned a ranking index γ_i, where γ_i = 1 and γ_i = *NP* represent the smallest and largest distances respectively. Following, γ_i is normalized to ensure that it falls within a specific range, typically between 0 and 1 by Eq. (8):

$$\gamma_i = \frac{\gamma_i}{NP} \tag{8}$$

Subsequently, if the fitness value of a trial vector is better than that of its corresponding target vector, the normalized rank index γ of the successful trial vector is saved to the set *M*, as follows:

$$\begin{aligned} &\text{if } \; f\left(\mathbf{U}_{i,g}\right) \le f\left(\mathbf{X}_{i,g}\right) \\ &\quad \gamma_i \to M \\ &\text{end} \end{aligned} \tag{9}$$

By saving the γ values of successful trial vectors, SSE can capture the information about the appropriate evolutionary scale for generating improved solutions. Finally, the weighted Lehmer mean ζ of *M* is calculated as follows:

$$\zeta = \frac{\sum_{m=1}^{|M|} w_m \cdot M^2_m}{\sum_{m=1}^{|M|} w_m \cdot M_m} \tag{10}$$

$$w_m = \frac{\Delta f_m}{\sum_{m=1}^{|M|} \Delta f_m} \tag{11}$$

where M_m is the m-th element of M, $\Delta f_m = \left| f(\mathbf{U}_{mk,g}) - f(\mathbf{X}_{mk,g}) \right|$ and mk is the index of the current solution corresponding to m. w_m is used to emphasize the contributions of large fitness improvements. ζ named appropriate evolutionary scale factor, provides a quantitative measure of the suitability and potential of different evolutionary scales for different optimization problems or even different evolutionary stages. In some problems or at certain evolutionary stages, a smaller evolutionary scale tends to be successful. In this case, the successful γ values are relatively small, and thus the ζ is relatively small. In the other case, a larger evolutionary scale is more promising for generating successful solutions and thus the ζ is relatively large. Therefore, ζ measures whether a large or small scale is appropriate. With the above procedures, the pseudo-code of SSE is shown in **Algorithm** 1.

Algorithm 1: **SSE**

```
Input: X_g: current population at generation g;
       U_g : trial population at generation g;
       NP: population size;
       M = ∅:  M is initialized as empty.
Output: ζ : appropriate evolutionary scale factor.
1: For i = 1 : NP
2:     Calculate the distance $d_i$ by Eq. (7);
3: End For
4: Sort d in ascending order and obtain the normalized ranking
       γ by Eq. (8);
5: For i = 1 : NP
6:     If $f(\mathbf{U}_{i,g}) \le f(\mathbf{X}_{i,g})$
7:         $\gamma_i \to M$ ;
8:     End If
9: End For
10: Calculate the appropriate evolutionary scale factor ζ by
       Eq. (10).
```

3. 2 Evolutionary scale adaptation

In DE, a generation strategy refers to how the trial vectors are generated, introducing new genetic materials to explore the searching space. However, if the strategy is too random or too greedy, it would lead to a loss of directionality or suboptimal solutions. To address this issue, the proposed evolutionary scale adaptation (ESA) mechanism dynamically adjusts the greediness of the optimization based on the current search requirement. An evolutionary scale indicator ψ is introduced to measure the degree of successful evolution in the solution space. ψ is initialized at the beginning and updated at each generation by utilizing the appropriate evolutionary scale factor ζ according to Eq. (12), where a = 0.1 is a constant.

$$\psi = (1-a)\cdot\psi + a\cdot\zeta \tag{12}$$

If ψ is smaller than a preset threshold T, it indicates that the recently successful individuals have small γ and a small evolutionary scale is more favorable. In this case, for generating offspring, a closer one from each current solution would be more promising to narrow down the search range. Otherwise, if $\psi \ge T$, a farther one from each current solution is more preferred to expand the search range.

With the above considerations, the pseudo-code of ESA is shown in **Algorithm** 2. Lines 1−3 calculate the distance between each current solution and each of the corresponding generated trial solutions by K candidate strategies. The generation strategy herein is a generalized concept of the procedure of producing trial vectors from target vectors, which could be the low-level ones, such as the mutation and crossover strategy, or the high-level ones, such as different DE variants. Line 4 compares ψ with T to determine the choice of a small or large evolutionary scale. If $\psi < T$, then for each current solution, the closest trial solution with the smallest evolutionary scale is selected as the final offspring (lines 6 and 7). Otherwise, the farthest trial solution with the largest evolutionary scale is selected as the final offspring (lines 11 and 12). The offspring is evaluated in line 15 and then ζ is calculated in line 16. Finally, line 17 updates ψ by using Eq. (12). Note that if the exploitation and exploration features of the generation strategies are known, lines 1−3 could be removed, and lines 5−8 and lines 10−13 could be replaced by the offspring generation of an exploitative and an explorative strategy respectively.

Algorithm 2: **ESA**

```
Input: X_g: current population at generation g;
       $\mathbf{Z}^k_{,g}$ (k = 1, 2, ⋯, K): the K trial population generated by
       K strategies at generation g;
       NP: population size;
       ψ : evolutionary scale indicator;
       T:  threshold for ψ ;
       a:  a constant for updating ψ .
Output: U: final offspring;
        ψ : updated evolutionary scale indicator.
1: For i = 1 : NP
2:     Measure the distance $L_i^k$ between $\mathbf{X}_{i,g}$ and each of the
       corresponding trial solutions $\mathbf{Z}_{i,g}^k$ (k = 1, 2, ⋯, K);
3: End For
4: If ψ < T
5:     For i = 1 : NP
6:         $index = \arg\min_k (L_i^k)$;
7:         $\mathbf{U}_{i,g} = \mathbf{Z}_{i,g}^{index}$;
8:     End For
9: Else
10:    For i = 1 : NP
11:        $index = \arg\max_k (L_i^k)$;
12:        $\mathbf{U}_{i,g} = \mathbf{Z}_{i,g}^{index}$;
```

```
13:   End For
14: End If
15:  Evaluate the fitness of $\mathbf{U}_g$;
16:  Calculate $\zeta$ according to SSE (Algorithm 1);
17: Update $\psi = (1-a)\cdot\psi + a\cdot\zeta$.
```

3.3 Overall framework

Based on ESA, the overall framework of evolutionary scale adaptive DE (ESADE) is shown in **Algorithm** 3. Line 1 initializes the population. Afterwards, at each generation *g*, line 3 first performs the generation procedure given by the *K* strategies to generate *K* trial population. Line 4 then performs the ESA method to obtain the final offspring for each individual and update the evolutionary scale indicator ψ. Finally, DE's selection is performed between $\mathbf{X}_g$ and $\mathbf{U}_g$ (line 5).

Algorithm 3: **ESADE**

```
Input:  K strategies;
        $\psi$ = 0.5: initial value for $\psi$;
        T:  threshold for $\psi$;
        $g_{max}$: maximum number of generations.
Output:  $x_b$: found best solution.
1:  Initialize the population $\mathbf{X}_0$, set the generation count $g = 0$;
2:  While $g \leqslant g_{max}$
3:     Perform DE's mutation and crossover given by the K strategies for $\mathbf{X}_g$ to generate K trial population $\mathbf{Z}^k{}_g$ ($k = 1, 2, \cdots, K$);
4:     Perform ESA (Algorithm 2) to obtain the final offspring $\mathbf{U}_g$ and update the evolutionary scale indicator $\psi$;
5:     Perform DE's selection between $\mathbf{X}_g$ and $\mathbf{U}_g$ to obtain the population $\mathbf{X}_{g+1}$ for the next generation;
6:     $g = g + 1$;
7:  End While
8:  Obtain the best solution $x_b$ from the final population.
```

3.4 Novelty of ESA

With the above descriptions, we would like to highlight the novelty and contribution of the proposed ESA method.

(1) ESA utilizes the successful evolutionary scale in the solution space for strategy adaptation, which is commonly neglected in the existing multi-strategy adaptation methods for DE.

(2) Most multi-strategy adaptation methods are based on the competition of strategies and the derived successful experience of each strategy. While ESA focuses on the evolutionary scale instead of the identity of the strategy.

(3) The final offspring in ESA is adaptively selected from the candidate solutions by the distance measure, which explicitly controls the evolutionary scale of each solution. Although some multi-strategy methods [36][22] also considered the distance between trial vectors and target vectors, ESA owns significant differences. In [36], Bujok proposed to use the farthest trial vectors from target vectors when the ratio *FES*/*maxFES* is relatively small and to use the closest trial vectors from the fittest solution otherwise, where *FES* is the currently consumed function evaluations and *maxFES* is the maximum function evaluations. Therefore, the adjustment of mutation strategy in [36] depends on the evolutionary process and is not in an adaptive manner. In [22], the pre-selection of the closest or farthest trial vectors from target vectors is based on the fitness ranking of target vectors. Thus, it is also not an adaptive method that could dynamically adjust strategy according to different problems.

3.5 Time complexity

The complexity of generating the *K* candidates is $O(K\bullet NP\bullet D)$, while the distance calculation between the candidates and the corresponding current solution is $O(K\bullet NP\bullet D)$. The complexity of the distance ranking is $O(NP\bullet\log_2 NP)$. Thus, the overall time complexity of ESA at one generation is $O(K\bullet NP\bullet D + NP\bullet\log_2 NP)$. And it becomes $O(NP\bullet D + NP\bullet\log_2 NP)$ if the exploitation and exploration features of the generation strategies are known.

4 Simulation

In this section, experiments are performed to verify the effectiveness of the proposed ESA method. The CEC2017 benchmark suite [37], which consists of 29 functions is employed. Functions F1 and F3 are unimodal functions while F4−F10 are simple multimodal functions. F11−F20 are hybrid functions and F21−F30 are composition functions. The performance of an algorithm is evaluated by the solution error value *SE*, which is defined as $f(\boldsymbol{x}) - f(\boldsymbol{x}^*)$, where $\boldsymbol{x}$ represents the best solution found with the maximum number of function evaluations of $10{,}000 \times D$ and $\boldsymbol{x}^*$ is the optimal solution. For each algorithm on each problem, 51 runs are performed, and the mean and standard deviation of *SE* are reported. In addition, the Wilcoxon rank-sum test [38] at a significance level of 0.05 is used to test the statistical significance of the performance between two algorithms. The results denoted as "+/=/–" indicate that our algorithm performs significantly better than (i.e., win), comparable to (i.e., tie), or worse than (i.e., lose) the compared algorithm respectively. The parameter *T* of ESA is set to 0.5.

4.1 Effectiveness of ESA

To validate the effectiveness of ESA, it is first incorporated into two baseline DEs, i.e., the jSO [39] and the L-SHADE_cnEpsin [40] algorithms. The resultant variant ESADE[1] is respectively compared with the baselines. Note that the selection of strategies for ESA is not arbitrary because they should have advantages for different types of problems. Hence, jSO and L-SHADE_cnEpsin are appropriate choices.

[1] The source code of ESADE is publicly available at https://zsxhomepage.github.io

Table 1 Performance comparison of ESADE with the baselines on 30-D, 50-D and 100-D CEC2017 benchmark set over 51 independent runs

	30-D							
	jSO			L-SHADE_ cnEpSin			ESADE	
	mean	std	sig	mean	std	sig	mean	std
F1	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00
F3	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00
F4	5.86E+01	3.11E-14	+	4.12E+01	3.49E+00	-	4.96E+01	1.90E+00
F5	7.55E+00	1.98E+00	=	1.19E+01	1.81E+00	+	7.57E+00	1.45E+00
F6	2.32E-08	6.21E-08	=	8.27E-09	2.81E-08	=	5.40E-08	1.28E-07
F7	3.87E+01	1.88E+00	=	4.33E+01	2.68E+00	+	3.85E+01	1.62E+00
F8	8.09E+00	2.07E+00	=	1.28E+01	2.31E+00	+	7.58E+00	1.46E+00
F9	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00
F10	1.63E+03	2.80E+02	+	1.45E+03	2.44E+02	=	1.54E+03	1.93E+02
F11	4.58E+00	1.17E+01	-	1.41E+01	2.01E+01	+	7.72E+00	1.50E+01
F12	1.47E+02	9.72E+01	+	3.73E+02	2.11E+02	+	1.01E+02	8.54E+01
F13	1.88E+01	4.34E+00	=	1.63E+01	9.82E+00	-	1.74E+01	5.93E+00
F14	2.18E+01	1.11E+00	+	2.15E+01	3.74E+00	+	1.96E+01	4.49E+00
F15	1.32E+00	1.05E+00	-	3.05E+00	1.57E+00	=	2.76E+00	2.06E+00
F16	5.49E+01	7.20E+01	+	1.91E+01	1.84E+01	=	2.13E+01	2.48E+01
F17	3.61E+01	7.83E+00	+	2.79E+01	7.46E+00	-	3.35E+01	5.88E+00
F18	2.09E+01	4.21E-01	+	2.11E+01	7.15E-01	=	2.06E+01	3.40E-01
F19	4.43E+00	1.54E+00	-	5.48E+00	1.60E+00	=	5.59E+00	1.03E+00
F20	3.44E+01	5.19E+00	+	3.04E+01	4.90E+00	=	3.20E+01	5.70E+00
F21	2.08E+02	2.22E+00	=	2.13E+02	2.69E+00	+	2.08E+02	1.60E+00
F22	1.00E+02	1.44E-14	=	1.00E+02	1.44E-14	=	1.00E+02	1.08E-13
F23	3.50E+02	2.78E+00	=	3.54E+02	2.91E+00	+	3.50E+02	2.20E+00
F24	4.26E+02	2.30E+00	=	4.28E+02	2.89E+00	+	4.26E+02	1.65E+00
F25	3.87E+02	6.29E-03	+	3.87E+02	8.01E-03	=	3.87E+02	3.90E-03
F26	9.35E+02	3.71E+01	+	9.43E+02	4.45E+01	+	8.87E+02	2.93E+01
F27	4.97E+02	7.20E+00	=	5.02E+02	5.26E+00	+	4.98E+02	6.43E+00
F28	3.04E+02	2.13E+01	-	3.19E+02	4.26E+01	=	3.13E+02	3.65E+01
F29	4.46E+02	1.41E+01	+	4.35E+02	7.48E+00	=	4.36E+02	1.19E+01
F30	1.97E+03	1.13E+01	=	1.98E+03	4.52E+01	=	1.98E+03	3.25E+01
W	**11**			**11**				
T	**14**			**15**				
L	**4**			**3**				

	50-D							
	jSO			L-SHADE_ cnEpSin			ESADE	
	mean	std	sig	mean	std	sig	mean	std
F1	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00
F3	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00
F4	5.29E+01	4.64E+01	=	4.91E+01	4.28E+01	-	7.56E+01	5.21E+01
F5	1.49E+01	3.10E+00	+	2.57E+01	6.28E+00	+	1.25E+01	2.06E+00
F6	1.71E-07	3.18E-07	-	8.72E-07	7.35E-07	=	1.02E-06	1.24E-06
F7	6.65E+01	2.84E+00	+	7.67E+01	5.87E+00	+	6.42E+01	2.13E+00
F8	1.46E+01	3.69E+00	=	2.78E+01	6.28E+00	+	1.34E+01	2.24E+00
F9	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00
F10	3.57E+03	4.52E+02	+	3.11E+03	2.42E+02	=	3.04E+03	3.47E+02
F11	2.49E+01	3.84E+00	+	2.19E+01	1.75E+00	=	2.16E+01	2.49E+00
F12	1.79E+03	5.25E+02	+	1.37E+03	4.05E+02	+	8.55E+02	3.13E+02
F13	3.56E+01	2.66E+01	=	7.24E+01	3.69E+01	+	4.43E+01	2.93E+01
F14	2.34E+01	1.73E+00	=	2.65E+01	2.04E+00	+	2.35E+01	1.25E+00
F15	2.23E+01	1.72E+00	+	2.65E+01	3.95E+00	+	2.03E+01	1.80E+00
F16	3.74E+02	1.56E+02	+	3.05E+02	1.11E+02	=	3.00E+02	1.25E+02
F17	2.59E+02	9.56E+01	+	2.21E+02	7.46E+01	=	2.09E+02	8.32E+01
F18	2.41E+01	1.93E+00	+	2.46E+01	2.84E+00	+	2.27E+01	1.15E+00
F19	1.26E+01	2.97E+00	=	1.73E+01	3.09E+00	+	1.32E+01	1.87E+00
F20	1.51E+02	7.80E+01	+	1.12E+02	3.13E+01	=	1.21E+02	5.21E+01
F21	2.15E+02	4.06E+00	=	2.28E+02	6.91E+00	+	2.15E+02	2.35E+00
F22	1.55E+03	1.91E+03	=	1.33E+03	1.68E+03	-	2.51E+03	1.67E+03
F23	4.31E+02	5.53E+00	+	4.39E+02	7.30E+00	+	4.27E+02	4.89E+00
F24	5.07E+02	3.48E+00	=	5.12E+02	5.39E+00	+	5.07E+02	3.27E+00
F25	4.80E+02	1.62E+00	+	4.80E+02	1.77E+00	-	4.81E+02	2.81E+00
F26	1.14E+03	5.12E+01	+	1.22E+03	1.04E+02	+	1.09E+03	4.91E+01
F27	5.11E+02	1.07E+01	-	5.31E+02	1.25E+01	+	5.17E+02	1.27E+01
F28	4.59E+02	2.10E-13	+	4.58E+02	6.86E+00	-	4.58E+02	1.73E-01
F29	3.72E+02	1.60E+01	+	3.53E+02	8.58E+00	-	3.59E+02	1.05E+01
F30	6.08E+05	2.95E+04	=	6.47E+05	5.62E+04	=	6.31E+05	5.25E+04
W	**15**			**14**				
T	**12**			**10**				
L	**2**			**5**				

Table 1 (Continued) Performance comparison of ESADE with the baselines on 30-D, 50-D and 100-D CEC2017 benchmark set over 51 independent runs

	100-D							
	jSO			L-SHADE_ cnEpSin			ESADE	
	mean	std	sig	mean	std	sig	mean	std
F1	7.36E-10	3.70E-09	=	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00
F3	3.72E-06	4.19E-06	+	0.00E+00	0.00E+00	-	2.11E-07	2.35E-07
F4	1.98E+02	1.08E+01	=	1.99E+02	7.77E+00	-	2.03E+02	1.14E+01
F5	3.65E+01	8.17E+00	+	5.87E+01	1.34E+01	+	2.57E+01	3.41E+00
F6	1.63E-04	4.77E-04	+	6.11E-05	2.43E-05	+	1.97E-05	1.20E-05
F7	1.43E+02	7.38E+00	+	1.62E+02	6.55E+00	+	1.27E+02	3.32E+00
F8	3.59E+01	8.02E+00	+	5.48E+01	6.72E+00	+	2.64E+01	3.90E+00
F9	7.02E-03	2.43E-02	+	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00
F10	1.07E+04	8.40E+02	+	1.05E+04	4.94E+02	+	8.97E+03	4.66E+02
F11	9.13E+01	2.87E+01	+	5.72E+01	3.97E+01	=	4.85E+01	3.20E+01
F12	1.80E+04	9.69E+03	+	4.53E+03	7.12E+02	-	4.95E+03	8.95E+02
F13	1.56E+02	4.71E+01	+	1.11E+02	3.56E+01	+	7.13E+01	2.76E+01
F14	5.38E+01	7.36E+00	+	4.98E+01	7.19E+00	+	3.91E+01	3.77E+00
F15	1.59E+02	3.98E+01	+	8.89E+01	3.11E+01	=	7.98E+01	3.14E+01
F16	1.75E+03	3.60E+02	+	1.18E+03	2.58E+02	-	1.45E+03	2.13E+02
F17	1.32E+03	2.57E+02	+	9.22E+02	1.68E+02	-	1.03E+03	1.95E+02
F18	1.79E+02	3.51E+01	+	7.28E+01	1.69E+01	+	5.57E+01	1.42E+01
F19	8.90E+01	1.74E+01	+	5.53E+01	7.47E+00	+	4.79E+01	5.57E+00
F20	1.55E+03	2.76E+02	+	1.11E+03	1.74E+02	-	1.19E+03	1.38E+02
F21	2.56E+02	9.67E+00	+	2.77E+02	5.02E+00	+	2.51E+02	4.41E+00
F22	1.16E+04	7.14E+02	+	1.06E+04	7.36E+02	+	9.78E+03	4.93E+02
F23	5.67E+02	7.94E+00	+	5.95E+02	9.19E+00	+	5.52E+02	1.09E+01
F24	8.99E+02	7.05E+00	=	9.17E+02	1.46E+01	+	8.99E+02	5.65E+00
F25	7.35E+02	3.45E+01	+	6.73E+02	4.11E+01	=	6.64E+02	4.34E+01
F26	3.21E+03	7.78E+01	+	3.08E+03	1.39E+02	=	3.10E+03	8.69E+01
F27	5.84E+02	1.83E+01	=	5.87E+02	1.92E+01	+	5.76E+02	2.02E+01
F28	5.23E+02	2.17E+01	+	5.12E+02	1.90E+01	=	5.13E+02	2.55E+01
F29	1.24E+03	1.76E+02	=	1.07E+03	1.37E+02	-	1.22E+03	1.56E+02
F30	2.35E+03	1.39E+02	+	2.39E+03	1.64E+02	+	2.28E+03	1.76E+02
W	**24**			**15**				
T	**5**			**7**				
L	**0**			**7**				

Table 2
Overall performance comparisons of ESADE with jSO and L-SHADE_cnEpSin according to Holm, Hochberg and Hommel procedures

v.s.	unadjusted *p*	p_{Holm}	$p_{Hochberg}$	p_{Hommel}
jSO	**0.000069**	**0.000138**	**0.000138**	**0.000138**
L-SHADE _cnEpSin	**0.000274**	**0.000274**	**0.000274**	**0.000274**

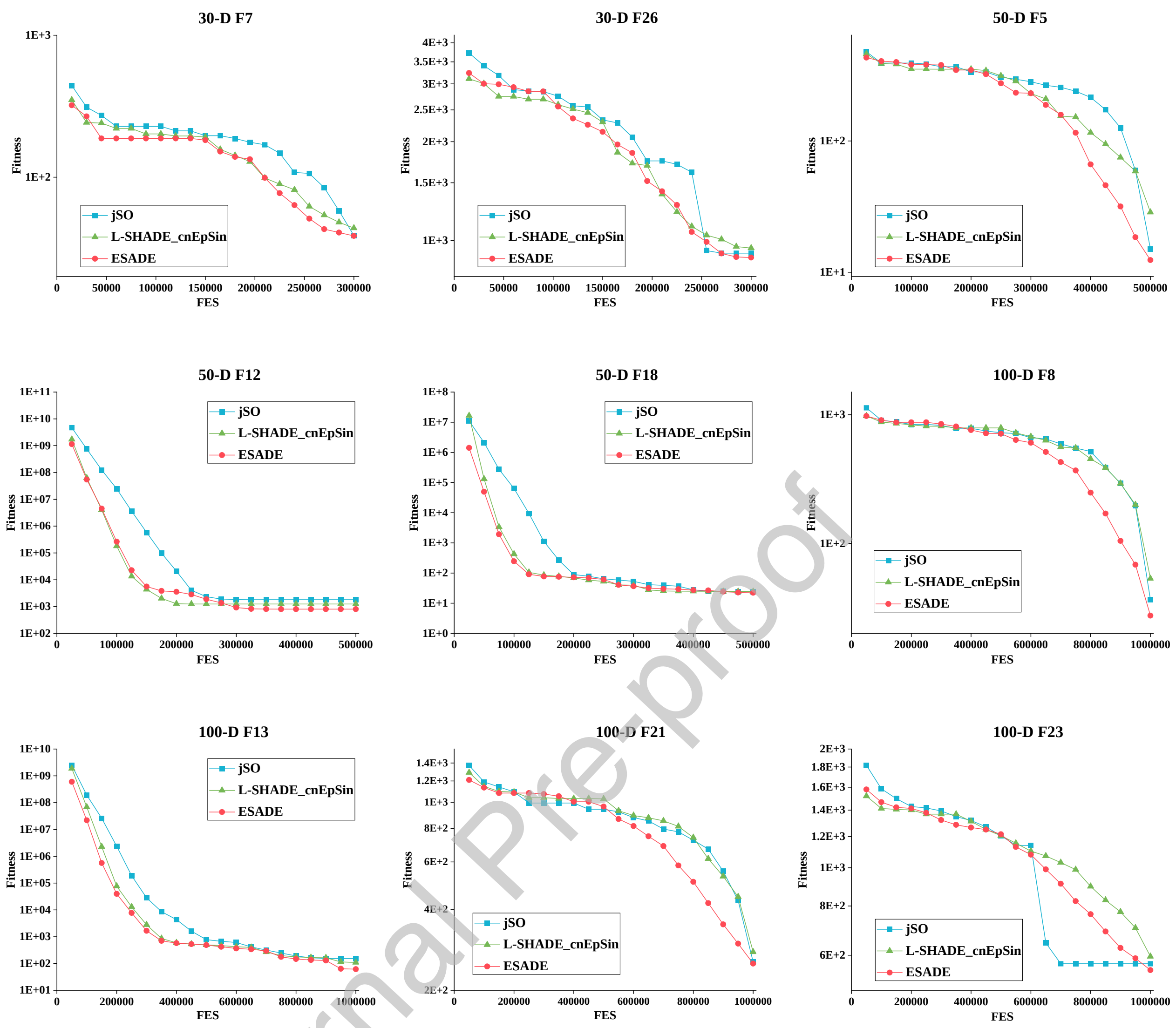

Fig. 1 Convergence curves of ESADE and the baselines on the nine selected functions in the run with the median error value.

Performance comparisons on 30-D, 50-D and 100-D functions are shown in Table 1. From Table 1, it is evident from our evaluations that the overall performance of the proposed ESADE is superior to the other two methods. Specifically, ESADE performs significantly better than jSO on 11, 15 and 24 functions and loses on 4, 2 and 0 functions in the cases of 30-D, 50-D and 100-D respectively. Compared with L-SHADE_cnEpSin, ESADE outperforms on 11, 14, 15 and underperforms on 3, 5 and 7 functions respectively. Notably, the performance superiority of ESADE becomes more prominent as dimensionality *D* increases. For instance, ESADE exhibits remarkable superiority in more than half of the tested functions in the 100-D case. These results demonstrate that ESADE is capable of effectively handling complex optimization problems, thereby showcasing the robustness and suitability for a wide range of applications.

Except the above single problem comparison, multi-problem statistical tests have also been performed, as shown in Table 2. According to the Holm, Hochberg and Hommel procedures, ESADE is significantly better than the other two algorithms with the outcome *p* values smaller than 0.05.

Fig. 1 shows the convergence curves on nine selected functions. As seen, ESADE achieves the best final solutions on eight functions, including 30-D F26, 50-D F5, 50-D F12, 50-D F18, 100-D F8, 100-D F13, 100-D F21, 100-D F23 and competitive performance on one function, i.e., 30-D F7 when compared with jSO.

4.2 Comparison with state-of-the-art DE variants

To demonstrate the performance of ESADE, it is further compared with the following five state-of-the-art DE variants:

PaDE [41]: An improved L-SHADE algorithm with a novel parameter adaptation method.

SCSS-L-SHADE [22]: A similarity selection-based improved L-SHADE.

EaDE [32]: An adaptive DE algorithm based on an explicit adaptation mechanism.

Table 3
Comparison results with state-of-the-art DEs according to single problem Wilcoxon's test

v.s.	30-D			50-D			100-D		
	win	tie	lose	win	tie	lose	win	tie	lose
PaDE	**12**	11	6	**17**	8	4	**24**	2	3
SCSS-L-SHADE	**8**	17	4	**11**	13	5	**23**	5	1
EaDE	**8**	11	10	**11**	13	5	**21**	8	0
L-SHADE-RSP	**9**	14	6	**10**	13	6	**17**	12	0
EJADE	**17**	7	5	**22**	5	2	**27**	1	1
DISH	**12**	14	3	**13**	12	4	**19**	6	4
SCSS-jSO	**6**	16	7	**7**	15	7	**16**	11	2
DTDE	**8**	9	12	**9**	5	15	**15**	3	11

Table 4
Overall performance comparisons of ESADE with state-of-the-art DEs according to Holm, Hochberg and Hommel procedures

v.s.	unadjusted p	p_{Holm}	$p_{Hochberg}$	p_{Hommel}
PaDE	**<1e-8**	**<1e-8**	**<1e-8**	**<1e-8**
SCSS-L-SHADE	**0.000095**	**0.00057**	**0.00057**	**0.00057**
EaDE	**0.004169**	**0.01167**	**0.01167**	**0.01167**
L-SHADE-RSP	**0.044757**	0.134271	0.134271	0.134271
EJADE	**<1e-8**	**<1e-8**	**<1e-8**	**<1e-8**
DISH	**0.002922**	**0.014611**	**0.014611**	**0.011689**
SCSS-jSO	0.579828	1.159655	0.944827	0.944827
DTDE	0.944827	1.159655	0.944827	0.944827

Table 5
Overall performance ranking of the considered DEs

Algorithm	Ranking
PaDE	6.08
SCSS-L-SHADE	5.35
EaDE	4.91
L-SHADE-RSP	4.56
EJADE	7.67
DISH	4.96
SCSS-jSO	3.95
DTDE	3.75
ESADE	**3.72**

L-SHADE-RSP [42]: An improved jSO algorithm with a selective pressure strategy.

EJADE [35]: A state-of-the-art improved JADE algorithm with a dual trial vector mechanism.

DISH [14]: An enhanced jSO algorithm with distance-based parameter adaptation.

SCSS-jSO [22]: An advanced jSO algorithm with a similarity selection mechanism.

Table 6
U-score and U-rank achieved by the DEs

Algorithm	30-D		50-D		100-D	
	U-Score	U-Rank	U-Score	U-Rank	U-Score	U-Rank
jSO	349702	10	384301	9	339521	9
L-SHADE _cnEpSin	407812	6	397470	8	453775	5
PaDE	396981	8	338322	10	260048	10
SCSS-L-SHADE	405813	7	407668	6	345990	8
EaDE	454086	4	424471	5	383472	7
L-SHADE -RSP	408174	5	451270	4	457422	4
EJADE	329764	11	209727	11	206853	11
DISH	351611	9	401501	7	448084	6
SCSS-jSO	460624	3	474042	3	515557	3
DTDE	**526588**	**1**	**556489**	**1**	540905	2
ESADE	465848	2	511742	2	**605376**	**1**

DTDE [43]: A state-of-the-art DE variant based on domain transform.

To ensure a fair comparison, parameters recommended in the corresponding literature are adopted as they have originally been tuned for the same CEC2017 functions. Performance comparisons on 30-D, 50-D and 100-D functions are shown in Table S1 in the supplementary file. Based on the results summarized in Table 3, it can be observed that ESADE demonstrates favorable performance across most scenarios compared to other variants. The only exceptions are in 30-D, where it is outperformed by EaDE, SCSS-jSO and DTDE, and in 50-D, where it is inferior to DTDE. However, in the remaining cases, ESADE consistently showcases superior performance. Notably, when *D* increases to 100, ESADE exhibits significant advancement over PaDE, SCSS-L-SHADE, EaDE, L-SHADE-RSP, EJADE, DISH, SCSS-jSO and DTDE on 24, 23, 21, 17, 27, 19, 16 and 15 functions respectively. This implies that ESADE performs uniquely well when dealing with complex problems in high-dimensional spaces, and it might be explained by the capability of ESA to capture the successful evolutionary scale in high-dimensional problems for better adjusting the exploitation and exploration directions.

According to the *p*-values given by the multi-problem test in Table 4, it is evident that ESADE statistically outperforms PaDE, SCSS-L-SHADE, EaDE, EJADE and DISH as the *p*-values are below 0.05 for all cases. Compared with L-SHADE-RSP, the unadjusted *p* value is below 0.05 but it is not significant according to the Holm, Hochberg and Hommel procedures. In comparison to SCSS-jSO and DTDE, the superiority is not significant. Note that when compared with DTDE, the *p*-values are obtained using DTDE as the control algorithm. Overall, it can be inferred that ESADE performs significantly better than most of the compared DE variants. The performance ranking given by Friedman test is shown in Table 5, from which ESADE achieves the ranking of 3.72, which is competitive to DTDE and much higher than the rest DEs.

Apart from the above statistical analyses, the trial-based dominance approach proposed by Price et al. [44] that could measure both the speed and accuracy of stochastic optimizers is also considered for performance comparison of the DEs. The target error value EVmin for the approach is set to 1E−8 in this experiment. The U-score [44] and its ranking U-rank achieved in the 30-D, 50-D, 100-D are shown in Table 6. From Table 6, ESADE obtains the 1st rank in the 100-D case, the 2nd rank in the 30-D and 50-D cases, while DTDE obtains the 1st rank in the 30-D and 50-D cases.

4.3 Comparison with state-of-the-art multi-strategy methods

To address the inefficiency of solving different problems at different stages with a single fixed strategy, scholars have proposed several multi-strategy methods. The following six methods are considered for comparison with ESA.

Sa [15]: A memory-based strategy adaptation method from the SaDE algorithm.

EPS [23]: A success-based strategy adaptation method from the EPSDE algorithm.

SaM [17]: An index-based strategy adaptation by the parameter adaptation of the JADE algorithm.

CSM [21]: A multi-strategy method that selects offspring based on a cheap surrogate model.

UM [33]: A multi-strategy method that selects offspring based on an underestimation model.

SCSS [22]: A multi-strategy method that selects offspring based on a fitness-based similarity selection rule.

To facilitate direct performance comparisons, all the methods are respectively implemented on the two baselines, i.e., jSO and L-SHADE_cnEpSin. Among all the methods, EPS, SaM, CSM and SCSS are parameter-free and therefore, there are no parameters to tune for the CEC2017 functions. UM has been tuned for the CEC2017 functions in [33]. While for Sa, the learning period *LP* is set as {10, 30, 50, 70, 90} respectively and the one (*LP* = 50) with the best overall performance given by Friedman test is adopted.

From Table S2 in the supplementary file and Table 7, in the 30-D, ESA performs similarly to Sa, EPS and SaM, outperforms CSM and UM but underperforms SCSS. While in the 50-D and 100-D cases, ESA is consistently the best methods. For instance, in the 100-D case, ESA outperforms on 16, 18, 15, 20, 23, 14 functions and loses on 4, 4, 6, 6, 0, 4 functions respectively. These findings indicate that when dealing with large search space, the ESA mechanism allows for appropriate strategy matching to adjust the search range and improve the optimization efficiency.

Furthermore, Table 8 reports the achieved *p*-values according to the Holm, Hochberg and Hommel procedures. It can be concluded that ESA demonstrates its significant superiority over five methods including Sa, EPS, SaM, CSM and UM with *p* value smaller than 0.05 except for SCSS. The overall performance ranking of all methods is shown in Table 9, from which ESA is the first-ranked.

Table 7
Comparison results of ESA with other multi-strategy methods according to single problem Wilcoxon's test

v.s.	30-D			50-D			100-D		
	win	tie	lose	win	tie	lose	win	tie	lose
Sa	**6**	17	6	**12**	11	6	**16**	9	4
EPS	**4**	21	4	**13**	11	5	**18**	7	4
SaM	**7**	14	8	**12**	10	7	**15**	8	6
CSM	**19**	6	4	**17**	7	5	**20**	3	6
UM	**16**	11	2	**18**	9	2	**23**	6	0
SCSS	**4**	17	8	**11**	14	4	**14**	11	4

Table 8
Overall performance comparisons of ESA with multi-strategy methods according to Holm, Hochberg and Hommel procedures

v.s.	unadjusted *p*	p_{Holm}	$p_{Hochberg}$	p_{Hommel}
Sa	**0.003789**	**0.011368**	**0.011368**	**0.011368**
EPS	**0.000584**	**0.002334**	**0.002334**	**0.002334**
SaM	**0.007651**	**0.015303**	**0.015303**	**0.015303**
CSM	**<1e-8**	**<1e-8**	**<1e-8**	**<1e-8**
UM	**<1e-8**	**0.000001**	**0.000001**	**0.000001**
SCSS	0.50492	0.50492	0.50492	0.50492

Table 9
Overall performance ranking of the considered methods

Method	Ranking
Sa	3.94
EPS	4.12
SaM	3.87
CSM	5.14
UM	4.68
SCSS	3.21
ESA	**3**

4.4 Working mechanism of ESA

To investigate the working process of ESA, the preference of a small or large evolutionary scale which is determined by the ψ value, is first analyzed. Following, the standard ESADE is compared with several of its variants to demonstrate the reasonability of the component designs from the perspective of performance.

(1) Preference of small or large evolutionary scale

Fig. 2 plots the selection of a small or large evolutionary scale over the evolution process in ESADE on four 50-D functions F5, F13, F15 and F23, from which it is observed that:

1) For all four functions, a large evolutionary scale is more favored at the early stage.

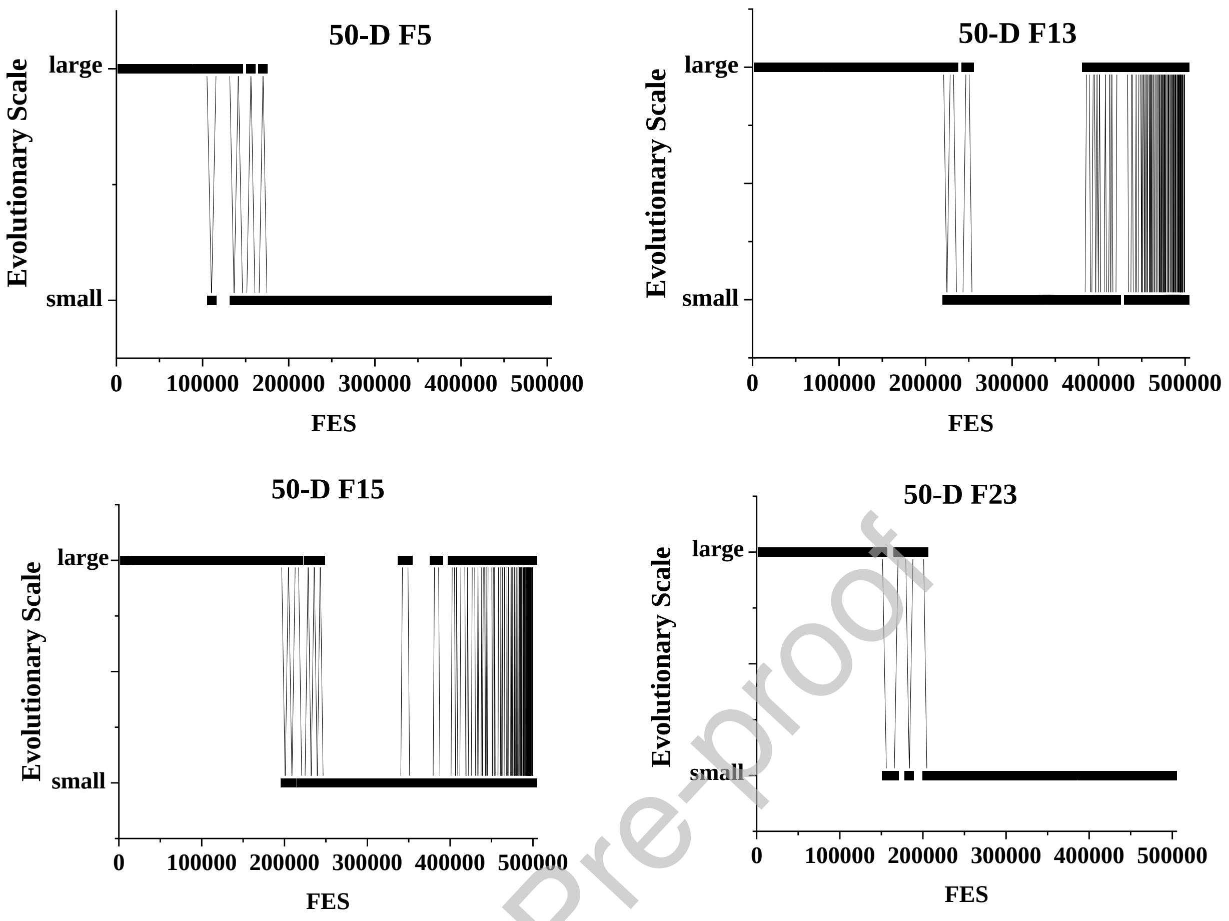

Fig. 2 Plots of the selection of a small or large evolutionary scale over the evolution process in the run with the median error value.

2) On F5 and F23, a large evolutionary scale is adopted at the early stage, while at the latter stage which occupies most of the function evaluations, a small scale is consistently employed.
3) On F13 and F15, in the middle of the evolution process, a small scale is consecutively used. While at the late stage, small and large scales are non-dominated by each other, and are competitive.

In summary, ESA could dynamically adjust the evolutionary scale according to different evolutionary stages and different problems.

(2) Comparison with the variants

To verify the effectiveness of the components of ESA, three variants are constructed as follows:

Variant-reverse: It reverses the decision on a small or large evolutionary scale. Specifically, when $\psi < T$, a large evolutionary scale is adopted. Otherwise, a small evolutionary scale is used.

Variant-random: It randomly decides the employment of a small or large evolutionary scale.

Variant-Amean: Instead of using the Lehmer mean, it uses the arithmetic mean to calculate the ζ of Eq. (10).

Other settings are unaltered for a fair comparison. Comparisons with the standard ESADE are shown in Table S3 and the results are summarized in Table 10. ESADE performs much better than the variants. Specifically, the effectiveness of ESA is confirmed by comparison with Variant-reverse and Variant-random with the "win/lose" metric of "23/2" and "14/4" respectively. The superiority is much more significant when compared with Variant-reverse because the behavior of Variant-reverse is exactly opposite to ESA. Compared with Variant-Amean, the Lehmer mean in ESA could generate a larger ζ, which is beneficial to maintaining population diversity for solving the complicated multi-modal functions.

Table 10
Comparison results of ESADE with the variants according to single problem Wilcoxon's test

	win	tie	lose
Variant- reverse	**23**	4	2
Variant- random	**14**	11	4
Variant- Amean	**19**	9	1

4.5 Application in real-world problems

Table 11
Effectiveness of ESA on the real-world problems

	jSO			L-SHADE_cnEpSin			ESADE	
	mean	std	sig	mean	std	sig	mean	std
Parameter Estimation for Frequency Modulated (FM) Sound Waves	2.32E+00	4.50E+00	=	2.67E+00	4.63E+00	+	**5.98E-01**	**2.42E+00**
Lennard-Jones Potential Problem	-2.75E+01	5.63E-01	+	-2.75E+01	4.90E-01	+	**-2.79E+01**	**4.22E-01**
Tersoff Potential for model Si (B)	**-3.68E+01**	**2.67E-01**	−	-3.68E+01	2.42E-01	+	-3.68E+01	2.57E-01
Tersoff Potential for model Si (C)	-2.92E+01	4.37E-04	+	-2.92E+01	3.01E-04	+	**-2.92E+01**	**6.11E-05**
Circular Antenna Array Design Problem	-2.16E+01	7.34E-02	+	-2.16E+01	8.53E-02	=	**-2.17E+01**	**8.94E-02**
ELD Problems: DED instance 1	4.76E+04	2.82E+02	+	**4.54E+04**	**3.03E+02**	−	4.61E+04	3.75E+02
ELD Instance 4	1.23E+05	3.60E+02	+	1.23E+05	4.36E+02	+	**1.22E+05**	**3.03E+02**
Hydrothermal Scheduling Instance 1	9.26E+05	6.16E+02	+	**9.24E+05**	**4.14E+02**	−	9.25E+05	5.41E+02
+	**6**			**5**				
=	1			1				
−	1			2				

Table 12
Performance comparison of ESA with other multi-strategy methods on the real-word problems

	Sa	EPS	SaM	CSM	UM	SCSS
+	**5**	**5**	**5**	**5**	**5**	**4**
=	3	3	3	2	3	2
−	0	0	0	1	0	2

To assess the performance of ESA on real-world optimization, it is applied to eight representative problems from the CEC2011 test suite [45], namely the Parameter Estimation for Frequency Modulated (FM) Sound Waves (P1), the Lennard-Jones Potential Problem (P2), the Tersoff Potential for model Si (B) Problem (P3), the Tersoff Potential for model Si (C) (P4), the Circular Antenna Array Design Problem (P5), the DED instance 1 (P6), the ELD instance 4 (P7) and the Hydrothermal Scheduling Instance 1 (P8).

The effectiveness of ESA could be clearly observed from Table 11. Compared with jSO, ESADE advances the performance on P2, P4−P8 but loses on P3. Compared with L-SHADE_cnEpSin, ESADE outperforms on five problems, including P1−P4 and P7 but loses on P6 and P8. Overall, ESADE achieves the best performance on five problems while jSO and L-SHADE_cnEpSin show advantages on one (P3) and two (P6, P8) problems respectively.

Further, Tables S4 and 12 show the comparisons with other multi-strategy methods. As seen, ESA exhibits superior performance over Sa, EPS, SaM, CSM, UM and SCSS on 5, 5, 5, 5, 5, 4 problems and inferior performance on 0, 0, 0, 1, 0, 2 problems respectively, which is the best-performing among all the considered methods.

4.6 More investigations on ESA

(1) Parameter sensitivity

To investigate the performance sensitivity to T, another eight settings are examined, which range within [0.1, 0.9] with a step of 0.1. From the comparison results as shown in Table 13, the standard setting of 0.5 performs significantly better than the eight settings. The superiority is much more significant compared with a small or large T value. Since a small and a large T would be a large and small evolutionary scale-favored setting respectively, the results indicate that a median T could better balance the local exploitation and exploration capabilities. Besides T, performance sensitivity to the setting of a in Eq. (12) and the initial ψ value is also studied, as shown in Tables 14 and 15 respectively. From Table 14, the performance degrades as a increases. A large a value will result in a lack of sufficient historical information to update ψ . From Table 15, the initial ψ value does not have a significant impact on the performance as all the settings perform similarly on almost all the functions. The result indicates that the initial ψ has a little influence on the adaptation.

Table 13
Performance comparison of T = 0.5 with other settings on 50-D functions

	0.1	0.2	0.3	0.4	0.6	0.7	0.8	0.9
+	**19**	**20**	**14**	**7**	**8**	**20**	**20**	**20**
=	9	8	13	19	18	7	7	7
−	1	1	2	3	3	2	2	2

Table 14
Performance comparison of a = 0.1 with other settings on 50-D functions

	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
+	**1**	**2**	**3**	**3**	**4**	**6**	**8**	**7**
=	27	25	26	26	**25**	22	**20**	22
−	1	2	0	0	0	1	1	0

Table 15
Performance comparison of initial ψ = 0.5 with other settings on 50-D functions

	0.1	0.2	0.3	0.4	0.6	0.7	0.8	0.9
+	**0**	**0**	**0**	**1**	**1**	**0**	**1**	**0**
=	29	29	29	27	**27**	29	27	29
−	0	0	0	1	1	0	1	0

(2) Performance on low-level adaptation

The previous experiments show the effectiveness of ESA for high-level adaptation, i.e., it is integrated with two DE variants. To study its performance for low-level adaptation of mutation strategies, ESA is implemented with "DE/rand/1" and "DE/current-to-*p*best/1". The parameter adaptation of *F* and *CR* is from SHADE [12] and the population size *NP* is set to 100. Thus, the three algorithms are denoted as "SHADE-rand/1", "SHADE-current-to-*p*best/1" and ESA-SHADE, respectively. As shown in Tables S5 and 16, ESA-SHADE performs significantly better than the other two algorithms with a single mutation strategy. The total "win/lose" results in 30-D, 50-D and 100-D cases are "30/3", "41/2", "42/6" respectively, which confirms the effectiveness of ESA for low-level adaptation.

Table 16
Comparison results of ESA with single mutation strategy on 30-D, 50-D and 100-D CEC2017 benchmark set

v.s.	30-D			50-D			100-D		
	win	tie	lose	win	tie	lose	win	tie	lose
SHADE-rand/1	**20**	9	0	**23**	6	0	**27**	2	0
SHADE-current-to-*p*best/1	**10**	16	3	**18**	9	2	**15**	8	6

5 Conclusion

In this paper, we have proposed an evolutionary scale adaptive DE, named ESADE for global optimization. With the idea that the recently successful evolutionary scale information could be further utilized to guide the evolution, ESA first measures the overall successful evolutionary scale of the population by a normalized ranking mechanism. An appropriate evolutionary scale factor is obtained and further used to update an evolutionary scale indicator. The evolutionary scale indicator is then compared with a threshold to determine the employment of a small or large evolutionary scale, i.e., to select a close or a far trial vector.

The effectiveness of the proposed ESA method has been confirmed by incorporating with state-of-the-art DEs. The resultant ESADE performs statistically better than each of the baseline DEs and several state-of-the-art DEs on the CEC2017 benchmark problems. The contribution of ESA has also been illustrated by comparisons with six other multi-strategy adaptation methods. Further comparison on real-world problems has also demonstrated the advantage of the proposed ESA method.

As for future works, ESA would be extended to other state-of-the-art baselines. Besides, its compatibility with other types of optimizations, such as multimodal [46], multiobjective [47] and multi-task [48] optimizations also needs to be studied.

Acknowledgements

The work described in this paper was supported by the National Natural Science Foundation of China (No. 62201227; No. 62071503) and the Guangzhou Basic and Applied Basic Research (No. SL2022A04J01366).

References

[1] R. Storn, K. Price, Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces, J. of Global Optim. 11 (1997) 341–359.
[2] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.
[3] S. Das, S. S. Mullick, P. N. Suganthan, Recent advances in differential evolution—An updated survey, Swarm Evol. Comput. 27 (2016) 1–30.
[4] K. R. Opara, J. Arabasb, Differential evolution: A survey of theoretical analyses, Swarm Evol. Comput 44 (2019) 546–558.
[5] R. D. Al-Dabbagh, F. Neri, N. Idris, M. S. Baba, Algorithm design issues in adaptive differential evolution: review and taxonomy, Swarm Evol. Comput. 43 (2018) 284–311.
[6] G. Li, Z. Wang, M. Gong, Expensive optimization via surrogate-assisted and model-free evolutionary optimization, IEEE Trans. Syst., Man, Cybern., Syst. DOI: 10.1109/TSMC.2022.3219080, 2022.
[7] G. Wu, R. Mallipeddi, P. N. Suganthan, Ensemble strategies for population-based optimization algorithms – A survey, Swarm Evol. Comput. 44 (2019) 695–711.
[8] L. M. Zheng, S. X. Zhang, K. S. Tang, S. Y. Zheng, Differential evolution powered by collective information, Inf. Sci. 399 (2017) 13–29.
[9] J. Zhang, A. C. Sanderson, JADE: Adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (2009) 945–958.
[10] S.-M. Guo, C.-C. Yang, Enhancing differential evolution utilizing eigenvector-based crossover operator, IEEE Trans. Evol. Comput. 19 (2015) 31–49.
[11] X. Qiu, K. C. Tan, J.-X. Xu, Multiple exponential recombination for differential evolution, IEEE Trans. Cybernet. 47 (2017) 995–1006.
[12] R. Tanabe and A. S. Fukunaga, "Reviewing and benchmarking parameter control methods in differential evolution," IEEE Trans. Cybern., vol. 50, pp. 1170-1184, 2020.
[13] R. Tanabe, A. S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in Proc. IEEE Congr. Evol. Comput., Beijing, China, 2014, pp. 1658–1665.
[14] A. Viktorin, R. Senkerik, M. Pluhacek, T. Kadavy, A. Zamuda, Distance based parameter adaptation for success-history based differential evolution, Swarm Evol. Comput. 50 (2019) 100462.
[15] A. K. Qin, V. L. Huang, P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2009) 398–417.
[16] G. Wu, R, Mallipeddi, P. N. Suganthan et al, Differential evolution with multi-population based ensemble of mutation strategies, Inf. Sci. 329 (2016) 329–345.
[17] W. Gong, Z. Cai, C. X. Ling, H. Li, Enhanced differential evolution with adaptive strategies for numerical optimization, IEEE Trans. Syst., Man, Cybern., Cybern. 41 (2011) 397–413.
[18] L. Tang, Y. Dong, J. Liu, Differential evolution with an individual-dependent mechanism, IEEE Trans. Evol. Comput. 19 (2015) 560–574.
[19] Z. H. Zhan, Z. J. Wang, H. Jin et al, Adaptive distributed differential evolution, IEEE Trans. Cybernet. 11 (2020) 4633–4647.
[20] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (2011) 55–66.
[21] W. Gong, A. Zhou, Z. Cai, A multi-operator search strategy based on cheap surrogate models for evolutionary optimization, IEEE Trans. Evol. Comput. 19 (2015) 746–758.
[22] S. X. Zhang, W. S. Chan, Z. K. Peng, S. Y. Zheng, K. S. Tang, Selective-candidate framework with similarity selection rule for evolutionary optimization, Swarm Evol. Comput. 56 (2020) 100696.
[23] R. Mallipeddi, P. Suganthan, Q. Pan, M. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2011) 1679–1696.

[24] S. X. Zhang, S. Y. Zheng, L. M. Zheng, An efficient multiple variants coordination framework for differential evolution, IEEE Trans. Cybernet. 47 (2017) 2780–2793.
[25] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng and M. Zhou, Chaotic local search-based differential evolution algorithms for optimization, IEEE Trans. Syst., Man, Cybern., Syst. 51 (2021), 3954-3967.
[26] L. Cui, G. Li, Q. Lin, J. Chen, and N. Lu, Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations, Comput. Oper. Res. 67 (2016), 155−173.
[27] A. W. Mohamed, A. A. Hadi, and K. M. Jambi, Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization, Swarm Evol. Comput. 50 (2019), 100455.
[28] X-F. Liu, et al, Historical and heuristic-based adaptive differential evolution, IEEE Trans. Syst., Man, Cybern., Syst. 49 (2018) 2623–2635.
[29] S. X. Zhang, L. M. Zheng, K. S. Tang, S. Y. Zheng, W. S. Chan, Multi-layer competitive-cooperative framework for performance enhancement of differential evolution, Inf. Sci. 482 (2019) 86–104.
[30] S. X. Zhang, S. Y. Zheng, L. M. Zheng, Differential evolution with objective and dimension knowledge utilization, Swarm Evol. Comput., 80 (2023) 101322.
[31] M. Tian, and X. Gao. Differential evolution with neighborhood-based adaptive evolution mechanism for numerical optimization, Inf. Sci. 478 (2019), 422−448
[32] S. X. Zhang, W. S. Chan, K. S. Tang and S. Y. Zheng, Adaptive strategy in differential evolution via explicit exploitation and exploration controls, Appl. Soft Comput. 107 (2021), 107494.
[33] X. Zhou, G. Zhang. Differential evolution with underestimation-based multimutation strategy, IEEE Trans. Cybernet. 49 (2018) 1353–1364.
[34] X. -G. Zhou, C. -X. Peng, J. Liu, Y. Zhang and G. -J. Zhang, Underestimation-assisted global-local cooperative differential evolution and the application to protein structure prediction, IEEE Trans. Evol. Comput. 24 (2020), 536-550.
[35] W. Yi, Y. Chen, Z. Pei, J. Lu, Adaptive differential evolution with ensembling operators for continuous optimization problems, Swarm Evol. Comput. 69 (2022) 100994.
[36] P. Bujok, Improving the convergence of differential evolution, In: Numerical analysis and its applications. NAA 2016. Lecture Notes in Computer Science, 10187. Springer, Cham. https://doi.org/10.1007/978-3-319-57099-0_26.
[37] N.H. Awad, M.Z. Ali, J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Nanyang Technol. Univ., Singapore, Nov, 2016.
[38] D. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures. London, U.K./Boca Raton, FL: Chapman & Hall/CRC, 2003.
[39] J. Brest, M. S Maučec, B. Bošković, Single objective real-parameter optimization: algorithm jSO, in Proc. IEEE Congr. Evol. Comput., San Sebastian, 2017, pp. 1311–1318.
[40] N. H. Awad, M. Z. Ali, and P. N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems, in Proc. IEEE Congr. Evol. Comput. Jun, pp. 372–379, 2017.
[41] Z. Meng, J.-S. Pan, K.-K. Tseng, PaDE: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization, Knowl. Based Syst. 168 (2019) 80–99.
[42] V. Stanovov, S. Akhmedova, E. Semenkin, LSHADE algorithm with rank-based selective pressure strategy for solving CEC 2017 benchmark problems, in Proc. IEEE Congr. Evol. Comput., Rio de Janeiro, 2018, pp. 1–8.
[43] S. X. Zhang, Y. N. Wen, Y. H. Liu, L. M. Zheng and S. Y. Zheng, Differential evolution with domain transform, IEEE Trans. Evol. Comput. 27 (2023), 1440−1455.
[44] K. V. Price, A. Kumar, P. N. Suganthan, Trial-based dominance for comparing both the speed and accuracy of stochastic optimizers with standard non-parametric tests, Swarm Evol. Comput. 78 (2023) 101287.
[45] S. Das, P.N. Suganthan, Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems, Jadavpur University, Nanyang Technological University, Technical Report, 2010.
[46] C. Yue, B. Qu, J. Liang, A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems, IEEE Trans. Evol. Comput., 22 (2017) 805–817.
[47] K. Li, Á. Fialho, S. Kwong, Q. Zhang, Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 18 (2013) 114–130.
[48] G. Li, Q. Zhang, Z. Wang, Evolutionary competitive multitasking optimization, IEEE Trans. Evol. Comput., 26 (2022) 278–289.

CRediT author statement

Sheng Xin Zhang: Conceptualization, Methodology, Software, Writing - Original Draft, Writing - Review & Editing, Funding acquisition.
Xin Rou Hu：Software, Formal Analysis, Writing - Original Draft.
Shao Yong Zheng: Writing - Review & Editing, Funding acquisition.

Declaration of interests

☒ 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.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: