Swarm and Evolutionary Computation 98 (2025) 102125

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Differential evolution algorithm with local and global parameter adaptation

Xiao Lin Jin [a], Sheng Xin Zhang [a,*], Li Ming Zheng [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

ARTICLE INFO

Keywords:
Differential evolution
Parameter adaptation
Local and global search
Exploitation and exploration
Global optimization

ABSTRACT

Differential Evolution (DE) is an effective meta-heuristic algorithm for numerical optimization. However, it suffers from persistent limitations such as sensitivity to parameter settings and premature convergence tendencies. This paper presents a novel Local and Global Parameter Adaptation (LGP) mechanism to mitigate these deficiencies through two key innovations. First, we develop a dual historical memory strategy that dynamically classifies successful control parameters into local or global historical record based on the Euclidean distance between parent-offspring vector pairs, the local and global historical memory are updated accordingly at each generation. Second, we introduce a parameter adaptation strategy that adaptively selects elements from appropriate historical memory for the generation of new control parameters to maintain exploitation-exploration balance. Extensive experimental validation demonstrates LGP's effectiveness. When integrated with four DE variants, LGP consistently improves their performance, and the LGP-enhanced algorithm demonstrates remarkable performance compared with seven State-of-the-Art DE algorithms. Results confirm that LGP improves solution accuracy and prevents premature convergence simultaneously.

1. Introduction

Evolutionary algorithms (EAs) have emerged as powerful optimization tools for complex global optimization problems [1], particularly for problems with large-scale dimensionality, multi-modal solution landscapes and highly nonlinear objective functions [2]. Among various EAs, differential evolution (DE) [3], introduced by Storn and Price, has gained prominence due to its conceptual simplicity, robust performance and remarkable convergence characteristics [4–6], and has been widely applied in various domains, including chemical engineering, artificial intelligence, neural networks and image processing [7–10].

The canonical DE operates through three fundamental operations: mutation, crossover and selection. Among these components, the mutation strategy and control parameters (particularly the scaling factor F and crossover rate CR) predominantly determine the algorithm's search behavior. Although DE has demonstrated remarkable success across various applications, its conventional implementation suffers from a well-documented limitation: the algorithm's performance exhibits strong dependence on parameter configuration. These critical parameters demonstrate problem-dependent variability and evolutionary stage-specific dynamics, thus necessitating effective adaptation schemes. This stands in contrast to traditional DE where parameters remain fixed throughout evolution or adjusted through heuristic rules [4].

In response to this challenge, researchers have increasingly focused on developing advanced parameter control methods [11], leading to substantial improvements in DE's optimization performance. This research thrust has yielded numerous advanced DE variants. One approach involves the random combination of multiple parameter settings to improve the algorithm's robustness and global search capability, as exemplified by CoDE (Composite DE) [12] and EPSDE (Ensemble of Mutation Strategies and Parameters in DE) [13]. Another strategy employs multi-modal distribution or multi-stage parameter tuning to dynamically regulate parameters within an appropriate range, such as CoBiDE (DE based on Covariance Matrix Learning and Bimodal Distribution Parameter Setting) [14] and DE-NPS (DE with a Novel Perturbation Strategy) [15]. Adaptive parameter adjustment mechanisms, which typically leverage historical evolutionary information to enhance adaptability, have also gained prominence. Notable examples include SaDE (Self-adaptive DE) [16], JADE (Adaptive DE) [17], SHADE (Success-history based Adaptive DE) [18], and DISH (Distance based Parameter Adaptation for SHADE) [19], along with other advanced techniques [20–22]. Several State-of-the-Art DE variants have achieved outstanding performance in the IEEE Congress on Evolutionary Computation (CEC) competitions, including jSO (Single-objective Optimization Algorithm) [23], LSHADE-cnEpSin (Ensemble Sinusoidal Differential Covariance Matrix Adaptation with Euclidean

* Corresponding authors.
E-mail addresses: zhangsx@jnu.edu.cn (S.X. Zhang), lmzheng@jnu.edu.cn (L.M. Zheng).

https://doi.org/10.1016/j.swevo.2025.102125
Received 19 May 2025; Received in revised form 3 August 2025; Accepted 8 August 2025
Available online 28 August 2025

Neighborhood for LSHADE (SHADE using Linear Population Size Reduction)) [24], and L-SRTDE (Success Rate-based Adaptive DE) [25]. In addition, the rapid advancement of machine learning has opened new avenues for the design and application of meta-heuristic algorithms. Driven by this trend, many research has adopted reinforcement learning techniques to achieve parameter adaptation, providing innovative solutions for DE [26–28].

While the above mentioned DE variants have made significant strides in enhancing optimization performance, critical limitations still persist. First, suboptimal parameter configuration often disrupts the crucial balance between exploitation and exploration [29], particularly in complex optimization problems. Second, as problem dimensionality increases and solution landscapes become more complex, some DE variants frequently exhibit insufficient capability to escape local optima and premature convergence tendencies [6]. These observed limitations motivate our development of a novel parameter adaptation mechanism specifically designed to maintain exploitation-exploration equilibrium throughout the evolutionary process. In response, this paper proposes a Local and Global Parameter Adaptation (LGP) mechanism with three key contributions:

(a) A dual historical memory strategy is proposed. Successful parameters F and CR are stored in local or global historical record based on the Euclidean distance between parent vector and offspring vector. In each generation, the weighted Lehmer mean of local and global historical record are stored in local and global historical memory respectively, which stores successful exploitation and exploration historical information.
(b) A parameter adaptation strategy is proposed to select elements from local or global historical memory adaptively for the generation of new F and CR, further improving the balance between local exploitation and global exploration capabilities.
(c) The efficacy of the LGP mechanism is evaluated through comprehensive experimental studies. First, we assess the generalizability of LGP by integrating it with four DE variants and conducting comparative studies against their baseline counterparts. Second, we develop an enhanced algorithm that incorporates LGP, and performing extensive comparisons with seven State-of-the-Art DE algorithms. The experimental results consistently demonstrate that our LGP mechanism exhibits remarkable optimization capability.

The remainder of this paper is organized as follows: Section 2 reviews the basic process of DE and some related works. Section 3 introduces the details of the proposed LGP mechanism. In Section 4, the experimental results and analysis of the LGP mechanism are presented. Finally, Section 5 concludes this paper.

2. Background

This section is divided into two main parts: First, we present a step-by-step explanation of the classical DE algorithm's operational procedures, including initialization, mutation, crossover and selection; Second, a review of adaptive parameter control method for DE is performed.

2.1. Differential evolution

Differential evolution (DE) iteratively searches for the optimal solution through initialization and repeated application of three genetic operations: mutation, crossover and selection, until the termination criterion is met.

2.1.1. Initialization

The initial population $\mathbf{P}_G$ at generation $G = 0$ consists of NP D-dimensional individuals randomly sampled in the searching space. The lower and upper bounds of each dimension are defined as:

$$\mathbf{x}_{min} = \left\{x_{1,min}, x_{2,min}, \dots, x_{D,min}\right\} \tag{1}$$

$$\mathbf{x}_{max} = \left\{x_{1,max}, x_{2,max}, \dots, x_{D,max}\right\} \tag{2}$$

The j-th ($j \in 1, 2, \dots, D$) dimension of the i-th ($i \in 1, 2, \dots, NP$) individual, also called target vector, is generated as follows:

$$x_{j,i,0} = x_{j,\min} + rand_{i,j}[0,1] \cdot (x_{j,\max} - x_{j,\min}) \tag{3}$$

where $rand_{i,j}[0,1]$ is a uniformly distributed random number within [0, 1].

2.1.2. Mutation

Mutation is performed by perturbing a base vector through the addition of one or more differential vectors, thereby generating a mutant vector. Some widely used mutation strategies are as follows:

DE/rand/1:

$$\mathbf{v}_{i,G} = \mathbf{x}_{r_1,G} + F \cdot (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}) \tag{4}$$

DE/rand/2:

$$\mathbf{v}_{i,G} = \mathbf{x}_{r_1,G} + F \cdot (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}) + F \cdot (\mathbf{x}_{r_4,G} - \mathbf{x}_{r_5,G}) \tag{5}$$

DE/best/1:

$$\mathbf{v}_{i,G} = \mathbf{x}_{best,G} + F \cdot (\mathbf{x}_{r_1,G} - \mathbf{x}_{r_2,G}) \tag{6}$$

DE/current-to-best/1:

$$\mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F \cdot (\mathbf{x}_{best,G} - \mathbf{x}_{i,G}) + F \cdot (\mathbf{x}_{r_1,G} - \mathbf{x}_{r_2,G}) \tag{7}$$

DE/current-to-$p$best/1 (with archive):

$$\mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F \cdot (\mathbf{x}_{pbest,G} - \mathbf{x}_{i,G}) + F \cdot (\mathbf{x}_{r_1,G} - \tilde{\mathbf{x}}_{r_2,G}) \tag{8}$$

Here, the indices r_1, r_2, r_3, r_4 and r_5 are mutually different integers that randomly chosen from $\{1, 2, \dots, NP\}$ and distinct from the index i, $\mathbf{x}_{best,G}$ and $\mathbf{x}_{pbest,G}$ represent the best individual and one of the top $100p\%$ ($p \in (0,1]$) fittest individuals in the G-th generation respectively, and $\tilde{\mathbf{x}}_{r_2,G}$ is randomly chosen from the union of the current population and the archive. F is a scaling factor constrained to the interval (0, 1].

2.1.3. Crossover

The target vector $\mathbf{x}_{i,G}$ (parent) and the mutant vector $\mathbf{v}_{i,G}$ undergo recombination of dimensions to produce the trial vector $\mathbf{u}_{i,G}$ (offspring). The classical DE algorithm provides two crossover strategies: exponential crossover and binomial crossover, with the latter being more prevalent in practice. The binomial crossover operation is defined as follows:

$$\mathbf{u}_{j,i,G} = \begin{cases} \mathbf{v}_{j,i,G}, & \text{if} \quad rand_{i,j}\ (0,1) \le CR \quad \text{or } j = j_{\text{rand}} \\ \mathbf{x}_{j,i,G}, & \text{otherwise.} \end{cases} \tag{9}$$

where $rand_{i,j}(0,1)$ is a uniformly distributed random number within (0, 1), j_{rand} is an integer randomly selected from $\{1, 2, \dots, D\}$ to ensure that at least one variable inherits from the mutant vector. CR is a crossover rate within [0, 1].

2.1.4. Selection

The selection operation determines the next generation by retaining the fitter one between the trial vector $\mathbf{u}_{i,G}$ and the target vector $\mathbf{x}_{i,G}$, as expressed in the following equation:

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G}, & \text{if } f(\mathbf{u}_{i,G}) \le f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G}, & \text{otherwise.} \end{cases} \tag{10}$$

where $f(\cdot)$ denotes the fitness function that is to be minimized.

2.2. Parameter adaptation for DE

It is well-established that the effectiveness of DE is critically dependent on its control parameters (scaling factor F and crossover rate CR). F controls the magnitude of differential vector's perturbation to the base vector during mutation. When F assumes smaller values, it promotes localized refinement of solutions, though this may result in premature convergence to suboptimal solutions. Conversely, larger F values enhance the algorithm's ability to escape local optima and traverse the search space, albeit at the potential cost of search stability and convergence precision. Regarding CR, this parameter determines the expected proportion of components inherited from the mutant vector. Lower CR values preserve more parental characteristics, favoring intensive local search (exploitation), whereas higher CR values increase genetic diversity in offspring, thereby expanding the exploration of novel regions in the search space.

The conventional parameter control method in DE relies on fixed values determined by trial-and-error experimentation, maintaining unchanged throughout the optimization process [3]. While this method offers implementation simplicity, its effectiveness is heavily dependent on the chosen parameter values, lacking adaptability across different problem domains [30], and the requirement for extensive preliminary parameter tuning imposes significant computational overhead.

In contrast to the fixed parameter approach, deterministic parameter control method enables parameters change dynamically governed by predefined rules during algorithm execution, effectively mitigating premature convergence issues associated with fixed parameter configurations. Draa et al. [31] developed an enhanced DE variant that employs novel sinusoidal formulas to dynamically adjust the values of F and CR. This approach effectively maintains a good balance between exploitation and exploration.

However, the deterministic parameter approach exhibits fundamental limitations due to its undirected nature, where arbitrarily generated parameters often prove inadequate for addressing distinct search requirements across different optimization problems. These inherent limitations have led to the development of more sophisticated adaptive parameter control method, enabling parameters to be adjusted dynamically based on the search feedback during the evolutionary process. Over the past twenty years, substantial advancements have been made in enhancing DE's performance through the development of adaptive parameter control method. These approaches can be classified into the following categories.

2.2.1. Fitness-based adaptation

This category encompasses parameter adaptation methods guided by fitness-based evaluations at either individual or population level. Yu et al. [32] proposed a two-level adaptive parameter control scheme that simultaneously optimizes parameters at both population and individual levels, enabling dynamic adjustment according to the overall state of the population and the characteristics of individuals. Sarker et al. [33] introduced a novel parameter selection mechanism that dynamically identifies optimal combination of parameters (F, CR and NP) by recording the success rate of each random parameter combination throughout different evolutionary stages for diverse optimization problems. Tang et al. [34] proposed an individual-dependent parameter setting method where the parameters are dynamically assigned to each individual based on the differences in their fitness values. Gupta and Su [35] proposed a fitness-based adaptive parameter control method by analyzing the evolutionary states of individuals involved during the generation of trial vectors, effectively maintaining an appropriate balance between population diversity and convergence speed. Yu et al. [36] incorporated a fitness-independent parameter adaptation mechanism into the framework of surrogate-assisted DE as an efficient parameter adaptation method to search for optimum within local regions. Ren et al. [37] proposed a fitness-based weighting strategy by employing a novel method which utilizes fitness information to avoid premature convergence and stagnation. Adalja et al. [38] developed a weighted average algorithm that computes weighted average based on top-performing candidates (defined by fitness values) to guide population movement, aiming to improve solution quality. Aljaidi et al. [39] proposed a multi-objective RIME algorithm that incorporates a Hard-Rime puncture mechanism for exploitation, with solution updates guided by normalized fitness-based selection probabilities, which effectively guides the agents toward high-quality solutions. Zhang et al. [40] designed a fully informed fuzzy logic system (FLS) to improve DE's performance in noisy environment, which dynamically adjusts DE's control parameters using normalized noisy fitness ranks and diversity ranks as inputs, generating a superiority indicator to guide adaptation.

2.2.2. Success-history-based adaptation

This category of adaptive methods maintain a historical record of parameters that generate successful offspring vectors, subsequently leveraging historical information to guide the generation of future parameters. A typical example is the adaptive method proposed by Zhang and Sanderson [17], where F_i and CR_i for each individual $\mathbf{x}_i$ is independently generated via the following equations:

$$F_i = \mathrm{randc}_i(\mu_F, 0.1) \tag{11}$$

$$CR_i = \mathrm{randn}_i(\mu_{CR}, 0.1) \tag{12}$$

where $\mathrm{randc}_i(\mu_F, 0.1)$ and $\mathrm{randn}_i(\mu_{CR}, 0.1)$ are the Cauchy and normal distribution with mean values of μ_F and μ_{CR} and variances of 0.1, respectively. μ_F and μ_{CR} are updated at the end of each generation as:

$$\mu_F = (1-c)\cdot \mu_F + c\cdot mean_L(S_F) \tag{13}$$

$$\mu_{CR} = (1-c)\cdot \mu_{CR} + c\cdot mean_A(S_{CR}) \tag{14}$$

where S_F and S_{CR} are the set of all successful F_i and CR_i at the current generation, respectively. c is a constant belongs to $[0,1]$, $mean_A(\cdot)$ is the arithmetic mean and $mean_L(\cdot)$ is the Lehmer mean defined as:

$$\mathrm{mean}_L(S_F) = \frac{\sum_{F\in S_F} F^2}{\sum_{F\in S_F} F} \tag{15}$$

Tanabe and Fukunaga [18] proposed an improved version of [17] named success-history-based parameter adaptation, the mean values used for the generation of new F and CR are randomly selected from historical memory M_F and M_{CR} (each with LP entries), respectively. M_F and M_{CR} are generated by the weighted Lehmer mean of S_F ($\mathrm{mean}_{WL}(S_F)$) and weighted mean of S_{CR} ($\mathrm{mean}_{WA}(S_{CR})$) respectively. Success-History based Adaptive DE (SHADE) [18,41,42] uses different parameter adaptation mechanisms based on this success-history-based parameter adaptation. Tanabe and Fukunaga [43] proposed L-SHADE by integrating the Linear Population Size Reduction (LPSR) rule into SHADE 1.1 [42], the latest version of SHADE which uses weighted Lehmer mean instead of weighted mean for the generation of M_{CR}, won the competition in CEC2014. Subsequently, researchers developed enhanced parameter adaptive mechanisms derived from SHADE 1.1. Stanovov et al. [44] introduced a linear bias adjustment for F and CR, enabling diverse methods for mean calculation. Chen et al. [45] introduced a data fusion-based parameter adaptation technique for CR to enhance the reliability of adaptation. In addition, Song et al. [15] developed a bi-stage parameter adaptation for F which employs a tripuls function to dynamically regulate F within an appropriate range during the initial evolutionary stage. Xu and Meng [22] proposed a multi-stage parameter adaptation scheme incorporating a progressive weighting strategy for historical memory updates. Specifically, F follows an improved Mexican hat wavelet basis function, Laplace distribution and Cauchy distribution in the first, second and third stage, respectively.

Concurrently, CR follows a Gaussian distribution that adjusted according to different stages of evolution. Zhong et al. [46] introduced an independent success history adaptation scheme based on the skeleton of efficient optimizer Competitive DE (CDE) for F, aiming to accelerate optimization. Khishe et al. [47] employed the hierarchical population-based DE for parameter generation. Specifically, the ordinary layer utilizes semi-fixed Cauchy and Gaussian distribution for F and CR respectively, whereas the elite layer employs wavelet-based function and Gaussian-based method for F and CR to enhance the algorithm's local exploitation capability, respectively. Aljaidi et al. [48] introduced a fitness-independent parameter control approach to improve effectiveness, which divides the population into K groups with F following a semi-fixed Cauchy distribution, and dynamically updates the group's probabilities through a success-failure counting strategy.

2.2.3. Reinforcement learning-based adaptation

Beyond the aforementioned approaches, recent studies have explored reinforcement learning-based methods for parameter adaptation. Huynh et al. [49] integrated the reinforcement learning Q-learning model into DE as an adaptive parameter controller, which dynamically adjusts the control parameters during execution by leveraging historical optimization information from preceding iterations. Sun et al. [27] modeled the parameter control problem as a finite-horizon Markov decision process, and applied a reinforcement learning algorithm named policy gradient to provide control parameters adaptively during the search procedure. Tan et al. [50] implemented a reinforcement learning-based algorithm to select parameters adaptively during evolution, effectively balancing population convergence with diversity maintenance. Gradient-based parameter adaptation with line search (GPALS) is a method for online parameter adaptation based on dynamic performance estimation of the population [51], Tatsis and Parsopoulos [28] proposed an enhanced version of [51], which dynamically tunes the tuner's parameters by using the reinforcement learning paradigm. Alkhalidy et al. [52] proposed a hybrid DE algorithm that dynamically adjusts parameters via reinforcement learning agent (combining Q-learning and policy gradient methods) using real-time vehicular data, improving task distribution and decryption efficiency. Wang et al. [53] proposed a DE framework that employs Q-learning from reinforcement learning technique to dynamically adjust control parameters and select mutation strategies, thereby enhancing algorithmic adaptability and overall performance.

3. Proposed method

3.1. Motivation

As described above, the representative success-history-based parameter adaptation (SHA) uses the parameter adaptation strategy based on the historical record of successful parameters, S_F and S_{CR}, the mean values of S_F and S_{CR} at each generation are stored in historical memory M_F and M_{CR}, respectively. However, all successful parameters F and CR are stored in historical record without distinguishing between exploitation and exploration capabilities, leading to underutilization of local exploitation and global exploration information. Moreover, the elements in historical memory are randomly selected for the generation of new F and CR, without the capability to adaptively choose appropriate elements according to search requirements.

In order to further improve the SHA's optimization searching ability, allowing the algorithm preserves effective exploitation and exploration information during the evolutionary process, we propose Local and Global Parameter Adaptation (LGP) mechanism in this paper, an improved version of SHA which incorporates a dual historical memory strategy and a parameter adaptation strategy. In the LGP mechanism, successful parameters F and CR are stored either in local historical record S_F^L, S_{CR}^L or global historical record S_F^G, S_{CR}^G based on the Euclidean distance between parent vector and offspring vector. The mean values of S_F^L and S_{CR}^L at each generation are stored in local historical memory M_F^L and M_{CR}^L, respectively; Similarly, the mean values of S_F^G and S_{CR}^G are stored in global historical memory M_F^G and M_{CR}^G, respectively. The generation of new F and CR utilizes this information adaptively. This will further improve the balance between local exploitation and global exploration capabilities.

In this section, we first elaborate on the generation of local and global historical record and memory, which store successful exploitation and exploration historical information in the dual historical memory strategy (Section 3.2). Subsequently, we detail the parameter adaptation strategy employed in the LGP mechanism in Section 3.3. The complete LGP mechanism is then described in Section 3.4, followed by an analysis of its novelty in Section 3.5.

3.2. Dual historical memory strategy

In each generation, the scaling factor $F_{i,G}$ and the crossover rate $CR_{i,G}$ for each individual $\mathbf{x}_i$ are generated via the following equations:

$$F_{i,G} = \text{randc}_i(M_{F,r_i}, 0.1) \tag{16}$$

$$CR_{i,G} = \text{randn}_i(M_{CR,r_i}, 0.1) \tag{17}$$

where r_i is an index randomly selected from 1 to the history length H; $\text{randc}_i(M_{F,r_i}, 0.1)$ and $\text{randn}_i(M_{CR,r_i}, 0.1)$ are the Cauchy and normal distribution with mean values of M_{F,r_i} and M_{CR,r_i} and variances of 0.1, respectively. M_{F,r_i} and M_{CR,r_i} are selected from local historical memory M_F^L, M_{CR}^L or global historical memory M_F^G, M_{CR}^G adaptively. In generation G, the k^L-th element in local historical memory and the k^G-th element in global historical memory are updated as follows:

$$M^L_{F,k^L,G+1} = \begin{cases} \text{mean}_{WL}(S_F^L) & \text{if } S_F^L \neq \emptyset \\ M^L_{F,k^L,G} & \text{otherwise.} \end{cases} \tag{18}$$

$$M^L_{CR,k^L,G+1} = \begin{cases} \text{mean}_{WL}(S_{CR}^L) & \text{if } S_{CR}^L \neq \emptyset \\ M^L_{CR,k^L,G} & \text{otherwise.} \end{cases} \tag{19}$$

$$M^G_{F,k^G,G+1} = \begin{cases} \text{mean}_{WL}(S_F^G) & \text{if } S_F^G \neq \emptyset \\ M^G_{F,k^G,G} & \text{otherwise.} \end{cases} \tag{20}$$

$$M^G_{CR,k^G,G+1} = \begin{cases} \text{mean}_{WL}(S_{CR}^G) & \text{if } S_{CR}^G \neq \emptyset \\ M^G_{CR,k^G,G} & \text{otherwise.} \end{cases} \tag{21}$$

where S_F^L and S_{CR}^L are local historical record which stores successful local control parameters $F_{i,G}$ and $CR_{i,G}$, respectively; Similarly, the successful global control parameters are stored in global historical record S_F^G and S_{CR}^G, with which the offspring $\mathbf{u}_{i,G}$ is fitter than the parent $\mathbf{x}_{i,G}$. k^L and k^G ($k^L, k^G \in [1, H]$) are the indices for updating the local and global historical memory, respectively. The weighted Lehmer mean $\text{mean}_{WL}(S^L)$ and $\text{mean}_{WL}(S^G)$ are calculated as follows (S^L refers to either S_F^L or S_{CR}^L; S^G refers to either S_F^G or S_{CR}^G):

$$\text{mean}_{WL}(S^L) = \frac{\sum_{k=1}^{|S^L|} w_k^L \cdot (S_k^L)^2}{\sum_{k=1}^{|S^L|} w_k^L \cdot S_k^L} \tag{22}$$

$$\text{mean}_{WL}(S^G) = \frac{\sum_{k=1}^{|S^G|} w_k^G \cdot (S_k^G)^2}{\sum_{k=1}^{|S^G|} w_k^G \cdot S_k^G} \tag{23}$$

$$w_k^L = \frac{\Delta f_k}{\sum_{l=1}^{|S_{CR}^L|} \Delta f_l} \tag{24}$$

$$w_k^G = \frac{\Delta f_k}{\sum_{l=1}^{|S_{CR}^G|} \Delta f_l} \tag{25}$$

where the fitness improvement $\Delta f_k = |f(\mathbf{u}_{k,G}) - f(\mathbf{x}_{k,G})|$.

We utilize the Euclidean distance between parent vector $\mathbf{x}_{i,G}$ and offspring vector $\mathbf{u}_{i,G}$ to determine whether to store successful parameters $F_{i,G}$ and $CR_{i,G}$ into local or global historical record. First, at each generation G, calculate the Euclidean distance between $\mathbf{x}_{i,G}$ and $\mathbf{u}_{i,G}$ for $i = 1, 2, \dots, NP$ by the following equation:

$$d(\mathbf{u}_{i,G}, \mathbf{x}_{i,G}) = \sqrt{\sum_{j=1}^{D} (\mathbf{u}_{i,j,G} - \mathbf{x}_{i,j,G})^2} \tag{26}$$

where D is the dimensionality of the target problem. Subsequently, sort $d(\mathbf{u}_{i,G}, \mathbf{x}_{i,G})$ for $i = 1, 2, \dots, NP$ in ascending order and assign the Euclidean distance ranking $Rank_{i,G}$ from 1 to NP. In order to eliminate the effect of magnitude and make the features comparable, we finally normalize the Euclidean distance ranking by dividing by the population size NP:

$$dist_{i,G} = Rank_{i,G}/NP \tag{27}$$

Under the circumstances of $\mathbf{u}_{i,G}$ fitter than $\mathbf{x}_{i,G}$, we introduce a distance threshold value $t_2 \in [0, 1]$ for proximity determination. If $dist_{i,G}$ ($dist_{i,G} \in [0, 1]$) less than or equal to t_2, means there are potentially superior solutions for regions closer to $\mathbf{x}_{i,G}$ in the context of a continuous landscape, then we should strengthen the local exploitation capability to achieve a steady improvements for promising areas, $F_{i,G}$ and $CR_{i,G}$ with the ability to produce close $\mathbf{u}_{i,G}$ will be stored in local historical record S_F^L and S_{CR}^L, respectively. Otherwise, the regions away from $\mathbf{x}_{i,G}$ have potentially superior solutions, the global exploration capability should be strengthened to encourage exploration to new searching areas, $F_{i,G}$ and $CR_{i,G}$ with the ability to generate distant $\mathbf{u}_{i,G}$ will be stored in global historical record S_F^G and S_{CR}^G, respectively.

3.3. Parameter adaptation strategy

In SHA, the mean values M_{F,r_i} and M_{CR,r_i} in Eqs. (16) and (17) are randomly selected from M_F and M_{CR}, respectively. Apparently, the major challenge in LGP is how to determine M_{F,r_i} and M_{CR,r_i} from local historical memory M_F^L, M_{CR}^L or global historical memory M_F^G, M_{CR}^G. Generally, if the CR values in generation G are universally smaller, means fewer exchange of dimensions happened between the parent and mutant vector, the offspring vector retains more variables from the parent vector, suggesting that there is a possibility of searching for a superior solution in the localized region of the parent vector, therefore we should select M_{F,r_i} and M_{CR,r_i} from local historical memory to encourage local exploitation capability for the generation of $F_{i,G+1}$ and $CR_{i,G+1}$. In contrast, there is a greater likelihood of searching for a superior solution in the region far from the parent vector, the selection of M_{F,r_i} and M_{CR,r_i} from global historical memory can help strengthen the global exploration capability, avoiding algorithm falling into local optimal. With this principle, we propose a parameter adaptation strategy to select M_{F,r_i} and M_{CR,r_i} automatically, as follows:

If $rand(0,1) < t_1 - mean(CR)$
  $M_{F,r_i} = M_F^L(r_i)$, $M_{CR,r_i} = M_{CR}^L(r_i)$;
Else
  $M_{F,r_i} = M_F^G(r_i)$, $M_{CR,r_i} = M_{CR}^G(r_i)$;
End If

where $rand(0,1)$ is a uniformly distributed random number within (0, 1) and t_1 is a threshold value belongs to [1, 2]. According to this parameter adaptation strategy, the smaller the mean value of CR in generation G, the higher the probability of using the local historical memory, while in the situation of larger mean value of CR, the choice of M_{F,r_i} and M_{CR,r_i} are likely to utilize the global historical memory.

3.4. The complete LGP mechanism

As shown in Fig. 1, the LGP mechanism maintains a local historical memory (M_F^L and M_{CR}^L) and a global historical memory (M_F^G and M_{CR}^G), each with H entries in the dual historical memory strategy, which updated based on local historical record (S_F^L and S_{CR}^L) and global historical record (S_F^G, S_{CR}^G), respectively. The parameter adaptation strategy is applied to select M_{F,r_i} and M_{CR,r_i} adaptively for the generation of new F and CR.

Algorithm 1 shows the overall pseudo-code of benchmark DE which uses the current-to-$p$best/1/bin strategy incorporating the LGP mechanism (LGP-based), which consists of two components, i.e. the dual historical memory strategy and the parameter adaptation strategy as displayed in Fig. 1. In each generation, the DE control parameters $F_{i,G}$ and $CR_{i,G}$ for each individual $\mathbf{x}_i$ are generated based on the parameter adaptation strategy (lines 12–20). After mutation and crossover, $dist_{i,G}$ is calculated according to the Euclidean distance between $\mathbf{x}_{i,G}$ and $\mathbf{u}_{i,G}$ for $i = 1, 2, \dots, NP$ (line 23), then for $F_{i,G}$ and $CR_{i,G}$ which make $\mathbf{u}_{i,G}$ fitter than $\mathbf{x}_{i,G}$, store them in local or global historical record based on $dist_{i,G}$ (lines 30–37). Finally, line 40 updates the local and global historical memory (**Algorithm** 2). This process is repeated until the termination criterion is achieved.

Algorithm 1 LGP-based DE algorithm

———————— Initialization ————————
1: $G = 0$;
2: Initialize population $\mathbf{P}_0 = (\mathbf{x}_{1,0}, \dots, \mathbf{x}_{NP,0})$ randomly;
3: Initialize CR value for each individual to 0.5;
4: Set all values in local memory M_F^L, M_{CR}^L to 0.5; Set all values in global memory M_F^G, M_{CR}^G to 0.5;
5: Archive $\mathbf{A} = \emptyset$;
6: Index counter of local and global memory $k^L = 1$ and $k^G = 1$; History length $H = 5$;
7: Threshold values $t_1 = 1.5$, $t_2 = 0.8$;
———————— Main loop ————————
8: **while** The termination criteria are not met **do**
9:   $S_F^L = \emptyset$, $S_{CR}^L = \emptyset$, $S_F^G = \emptyset$ and $S_{CR}^G = \emptyset$;
10:   **for** $i = 1 : NP$ **do**
11:     r_i = Select from $[1, H]$ randomly;
12:     **if** $rand(0,1) < t_1 - mean(CR)$ **then**
13:       $M_{F,r_i} = M_F^L(r_i)$;
14:       $M_{CR,r_i} = M_{CR}^L(r_i)$;
15:     **else**
16:       $M_{F,r_i} = M_F^G(r_i)$;
17:       $M_{CR,r_i} = M_{CR}^G(r_i)$;
18:     **end if**
19:     $F_{i,G} = randc_i(M_{F,r_i}, 0.1)$;
20:     $CR_{i,G} = randn_i(M_{CR,r_i}, 0.1)$;
21:     Generate trial vector $\mathbf{u}_{i,G}$ by current-to-$p$best/1/bin;
22:   **end for**
23:   Calculate $dist_{i,G}$ according to Eq. (27);
24:   **for** $i = 1 : NP$ **do**
25:     **if** $f(\mathbf{u}_{i,G}) \le f(\mathbf{x}_{i,G})$ **then**
26:       $\mathbf{x}_{i,G+1} = \mathbf{u}_{i,G}$;
27:     **else**
28:       $\mathbf{x}_{i,G+1} = \mathbf{x}_{i,G}$;
29:     **end if**
30:     **if** $f(\mathbf{u}_{i,G}) < f(\mathbf{x}_{i,G})$ **then**
31:       $\mathbf{x}_{i,G} \to \mathbf{A}$;
32:       **if** $dist_{i,G} \le t_2$ **then**
33:         $F_{i,G} \to S_F^L$, $CR_{i,G} \to S_{CR}^L$;
34:       **else**
35:         $F_{i,G} \to S_F^G$, $CR_{i,G} \to S_{CR}^G$;
36:       **end if**
37:     **end if**
38:   **end for**
39:   If necessary, delete randomly selected individuals from the archive such that the archive size is $|\mathbf{A}|$;
40:   Update memories M_F^L, M_{CR}^L, M_F^G and M_{CR}^G (**Algorithm** 2);
41:   $G = G + 1$;
42: **end while**

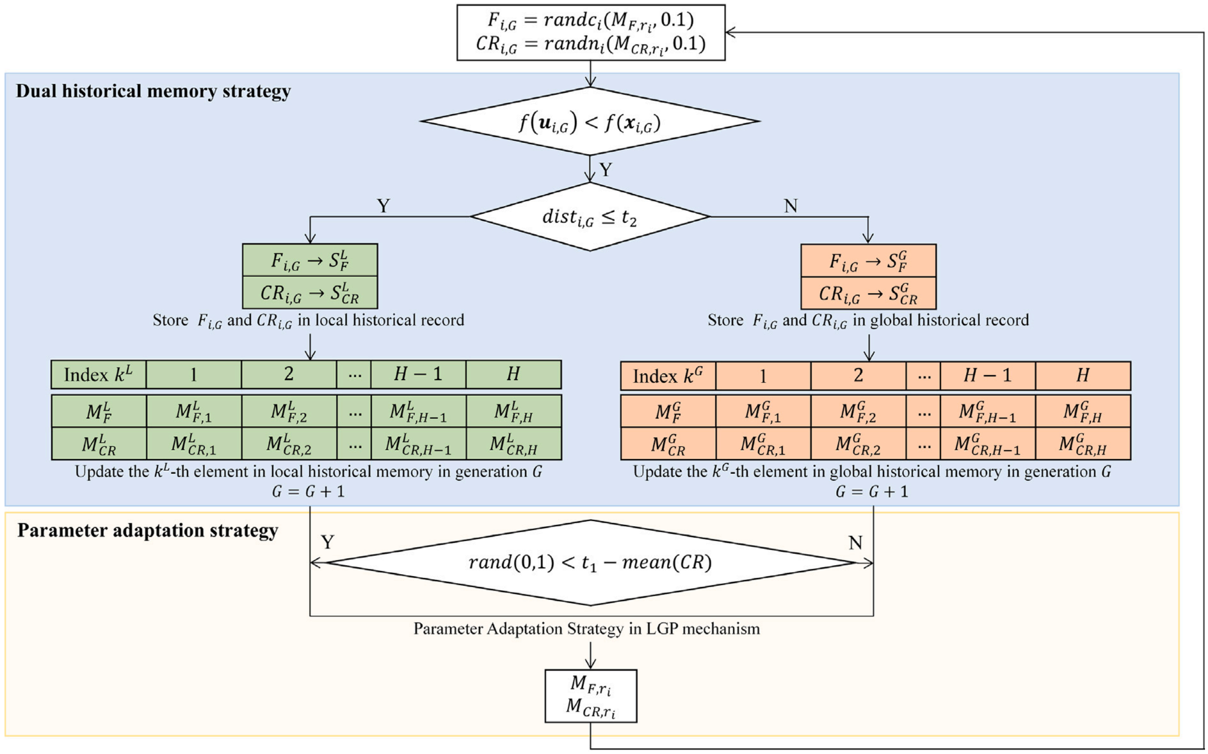

Fig. 1. The flow diagram of LGP mechanism.

Algorithm 2 Memory update in LGP-based DE

if $S_F^L \neq \emptyset$ **and** $S_{CR}^L \neq \emptyset$ **then**
  Update M^L_{F,k^L} according to Eq. (18);
  Update M^L_{CR,k^L} according to Eq. (19);
  $k^L = k^L + 1$;
  if $k^L > H$ **then**
    $k^L \leftarrow 1$;
  end if
end if
if $S_F^G \neq \emptyset$ **and** $S_{CR}^G \neq \emptyset$ **then**
  Update M^G_{F,k^G} according to Eq. (20);
  Update M^G_{CR,k^G} according to Eq. (21);
  $k^G = k^G + 1$;
  if $k^G > H$ **then**
    $k^G \leftarrow 1$;
  end if
end if

3.5. Novelty of LGP

Based on the aforementioned descriptions, we highlight the novelty of the proposed LGP mechanism.

(a) The dual historical memory strategy in LGP categorizes successful parameters F and CR into either local or global historical record by evaluating the Euclidean distance between parent-offspring vector pairs. This approach effectively leverages both exploitation and exploration information embedded in the parameters, mitigating a critical limitation prevalent in current parameter adaptation methodologies.

(b) The SHADE-based algorithm randomly selects elements from historical memory for generating new F and CR. In contrast, LGP employs an adaptive parameter adaptation strategy that intelligently chooses appropriate elements from either local or global historical memory based on current search requirements, further enhancing the exploitation-exploration equilibrium.

(c) The selection of successful parameters stored in either local or global historical record is governed by distance measures within the LGP mechanism, effectively balancing exploitation and exploration. While previous parameter adaptation methods [19,54,55] have also incorporated the Euclidean distance between parent and offspring vectors, LGP owns significant differences. In [54], M candidates are first generated for each current solution independently. Subsequently, the final candidate is determined by a Euclidean distance-based similarity selection rule. The approach in [19] employs a distance-based parameter adaptation method, which utilizes weighted means for F and CR adaptation, with weights derived from the Euclidean distance between parent-offspring vector pairs instead of fitness improvements, thereby promoting more intensive search space exploration. In [55], this concept is further advanced by introducing squared Euclidean distance in the update formulae for F and CR, amplifying the distance's influence on parameter adaptation. However, these existing methods treat all parameters uniformly, lacking LGP's critical differentiation between local and global parameter storage.

4. Simulation

To validate the effectiveness of the proposed LGP mechanism, comprehensive experiments are conducted using the CEC2017 benchmark suite [56]. The 29 functions can be classified into four types based on their mathematical characteristics, where $F1$ and $F3$ are unimodal functions, $F4 - F10$ are simple multimodal functions, $F11 - F20$ are hybrid functions and $F21 - F30$ are composition functions. It is important to notice that $F2$ has been removed from CEC2017 due to its unstable behavior in high-dimensional cases. The performance of the compared algorithms is evaluated by the solution error value defined as $F(\mathbf{x}) - F(\mathbf{x}^*)$, where $F(\mathbf{x})$ is the smallest fitness for minimization problem

Table 1
Performance comparison with L-SHADE on 30-D, 50-D and 100-D CEC2017 functions.

D Problem	30-D L-SHADE	30-D L-SHADE-LGP	50-D L-SHADE	50-D L-SHADE-LGP	100-D L-SHADE	100-D L-SHADE-LGP
F1	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)
F3	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	**9.34e−7 (1.19e−6)** −	1.28e−3 (1.24e−3)
F4	5.86e+1 (2.54e−14) ≈	5.86e+1 (2.69e−14)	7.36e+1 (4.72e+1) ≈	6.33e+1 (5.05e+1)	1.97e+2 (1.60e+1) ≈	2.00e+2 (8.80e+0)
F5	**6.73e+0 (1.30e+0)** −	7.59e+0 (2.44e+0)	**1.21e+1 (2.01e+0)** −	1.31e+1 (2.38e+0)	3.75e+1 (5.02e+0) ≈	3.78e+1 (5.40e+0)
F6	1.07e−8 (3.72e−8) ≈	3.35e−9 (1.97e−8)	2.06e−5 (1.46e−4) +	**4.23e−8 (1.26e−7)**	6.91e−3 (5.02e−3) +	**7.11e−4 (8.76e−4)**
F7	**3.72e+1 (1.34e+0)** −	3.87e+1 (1.79e+0)	**6.34e+1 (1.95e+0)** −	6.63e+1 (2.69e+0)	**1.42e+2 (4.76e+0)** −	1.43e+2 (5.88e+0)
F8	**7.18e+0 (1.57e+0)** −	8.81e+0 (2.73e+0)	**1.21e+1 (2.31e+0)** −	1.43e+1 (2.34e+0)	3.79e+1 (5.29e+0) ≈	3.61e+1 (6.28e+0)
F9	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	5.41e−1 (4.99e−1) +	**7.02e−3 (2.43e−2)**
F10	1.47e+3 (1.56e+2) ≈	1.42e+3 (2.44e+2)	3.15e+3 (3.05e+2) ≈	3.26e+3 (2.85e+2)	1.04e+4 (5.17e+2) ≈	1.05e+4 (4.65e+2)
F11	2.43e+1 (2.72e+1) +	**1.55e+1 (2.37e+1)**	4.83e+1 (7.94e+0) +	**2.90e+1 (4.33e+0)**	4.36e+2 (1.05e+2) +	**1.10e+2 (4.12e+1)**
F12	9.32e+2 (3.41e+2) +	**3.08e+2 (1.78e+2)**	2.12e+3 (4.39e+2) +	**1.92e+3 (4.81e+2)**	2.27e+4 (8.30e+3) +	**1.38e+4 (5.08e+3)**
F13	1.78e+1 (1.13e+1) ≈	1.73e+1 (5.34e+0)	6.27e+1 (3.09e+1) +	**4.12e+1 (2.40e+1)**	4.03e+2 (1.55e+2) +	**1.45e+2 (4.23e+1)**
F14	**2.18e+1 (1.35e+0)** −	2.22e+1 (6.02e+0)	2.97e+1 (3.34e+0) +	**2.72e+1 (2.56e+0)**	2.53e+2 (2.99e+1) +	**6.11e+1 (1.03e+1)**
F15	3.00e+0 (2.06e+0) +	**2.55e+0 (1.25e+0)**	3.99e+1 (8.31e+0) +	**2.44e+1 (3.96e+0)**	2.47e+2 (5.34e+1) +	**1.98e+2 (5.13e+1)**
F16	4.71e+1 (4.49e+1) ≈	6.55e+1 (6.92e+1)	4.09e+2 (1.32e+2) ≈	4.12e+2 (1.56e+2)	**1.69e+3 (2.58e+2)** −	1.80e+3 (2.90e+2)
F17	3.22e+1 (4.89e+0) ≈	3.16e+1 (7.23e+0)	2.43e+2 (7.33e+1) ≈	2.52e+2 (7.98e+1)	**1.11e+3 (1.97e+2)** −	1.19e+3 (2.05e+2)
F18	2.19e+1 (1.26e+0) +	**2.03e+1 (3.98e+0)**	3.99e+1 (1.37e+1) +	**2.53e+1 (2.25e+0)**	2.16e+2 (5.63e+1) ≈	2.09e+2 (3.68e+1)
F19	4.81e+0 (1.40e+0) ≈	5.10e+0 (1.49e+0)	2.42e+1 (7.22e+0) +	**1.40e+1 (2.92e+0)**	1.78e+2 (2.63e+1) +	**1.26e+2 (2.48e+1)**
F20	3.10e+1 (6.24e+0) +	**2.89e+1 (6.14e+0)**	**1.36e+2 (4.55e+1)** −	1.96e+2 (8.30e+1)	1.56e+3 (1.92e+2) ≈	1.61e+3 (2.57e+2)
F21	**2.07e+2 (1.48e+0)** −	2.09e+2 (2.47e+0)	2.13e+2 (2.14e+0) ≈	2.14e+2 (3.49e+0)	2.60e+2 (5.46e+0) ≈	2.60e+2 (5.86e+0)
F22	1.00e+2 (1.44e−14) ≈	1.00e+2 (1.44e−14)	2.31e+3 (1.76e+3) ≈	2.60e+3 (1.72e+3)	**1.13e+4 (4.75e+2)** −	1.16e+4 (5.60e+2)
F23	**3.50e+2 (2.64e+0)** −	3.52e+2 (3.70e+0)	4.30e+2 (3.60e+0) ≈	4.32e+2 (5.99e+0)	5.69e+2 (9.35e+0) +	**5.58e+2 (9.31e+0)**
F24	**4.26e+2 (1.77e+0)** −	4.28e+2 (3.13e+0)	5.07e+2 (2.79e+0) ≈	5.08e+2 (5.30e+0)	9.10e+2 (7.51e+0) +	**9.00e+2 (7.60e+0)**
F25	3.87e+2 (2.23e−2) +	**3.87e+2 (9.29e−3)**	4.84e+2 (1.21e+1) +	**4.81e+2 (3.57e+0)**	7.48e+2 (3.18e+1) +	**7.31e+2 (4.19e+1)**
F26	9.25e+2 (3.89e+1) ≈	9.41e+2 (5.39e+1)	1.13e+3 (4.61e+1) ≈	1.13e+3 (4.94e+1)	3.30e+3 (7.60e+1) +	**3.17e+3 (6.81e+1)**
F27	5.04e+2 (5.47e+0) +	**4.98e+2 (8.24e+0)**	5.33e+2 (1.38e+1) +	**5.20e+2 (1.30e+1)**	6.27e+2 (1.96e+1) +	**6.07e+2 (2.26e+1)**
F28	3.35e+2 (5.36e+1) ≈	3.26e+2 (4.70e+1)	4.68e+2 (1.93e+1) +	**4.59e+2 (1.60e−13)**	5.24e+2 (2.30e+1) ≈	5.31e+2 (2.52e+1)
F29	4.33e+2 (7.01e+0) ≈	4.35e+2 (1.52e+1)	**3.51e+2 (1.27e+1)** −	3.57e+2 (1.16e+1)	1.29e+3 (1.69e+2) ≈	1.30e+3 (1.92e+2)
F30	1.99e+3 (6.70e+1) ≈	1.97e+3 (2.77e+1)	6.86e+5 (1.15e+5) +	**6.25e+5 (5.64e+4)**	2.39e+3 (1.29e+2) +	**2.31e+3 (1.13e+2)**
+/−/≈	7/7/15		12/5/12		14/5/10	

obtained after $10000 \cdot D$ function evaluations and $F(\mathbf{x}^*)$ is the fitness of the global optimal $\mathbf{x}^*$, the solution error value is considered as zero if it is smaller than 10^{-8}. For each test function, the number of dimensions $D = 30, 50, 100$ and the search range is $[-100, 100]^D$, 51 independent runs are performed, while the mean and standard deviation of the solution error values are reported. In order to draw statistical conclusions, Wilcoxon rank-sum test [57] with 5% significance level is applied to compare the performance. The symbols "+", "≈" and "−" represent that the algorithm incorporating LGP performs significantly better than, similar to or worse than the compared algorithm, respectively.

4.1. Effectiveness of the LGP mechanism

The proposed LGP mechanism is first integrated with the following four DE algorithms to verify its effectiveness, performance of the resulting variants, L-SHADE-LGP, jSO-LGP, DTDE-LGP and ODF-SCSS-jSO-LGP are compared with the baseline algorithms, respectively.

L-SHADE [43]: An improved version of SHADE algorithm with linear population size reduction.

jSO [23]: An improved version of iL-SHADE [58] algorithm that introduces a new weighted mutation strategy.

DTDE [59]: An improved SCSS-L-SHADE algorithm with domain transform methodology.

ODF-SCSS-jSO [60]: An improved jSO algorithm based on the feedback of objective and dimension.

All parameter configurations in the aforementioned algorithms strictly follow the specifications provided in their original publications, guaranteeing a fair comparison. In addition, the threshold values $t_1 = 1.5$ and $t_2 = 0.8$ are adopted in the LGP mechanism, which are based on the experimental findings given later in Section 4.4. Tables 1, 2, 3 and 4 present the comparisons of L-SHADE, jSO, DTDE and ODF-SCSS-jSO with their corresponding variants on the 30-D, 50-D and 100-D CEC2017 benchmark suite, respectively. The entries which are significantly better are marked in **bold-face**. Table 5 shows the summary comparison results of them.

As observed from Table 5, in higher dimensions (50-D and 100-D), the LGP mechanism achieves notable gains over the competing algorithms, but its performance remains average in 30-D. Out of the 232 cases in 50-D and 100-D, LGP variant wins in 101 (=12+12+11+11+14+14+16+11) and loses in 22 (=5+1+3+1+5+1+3+3), while for the 116 cases in 30-D, LGP variant wins in 22 (=7+6+5+4) and loses in 16 (=7+1+4+4). In detail, from Table 1, the LGP mechanism improves the performance of L-SHADE on 7, 12 and 14 functions and is inferior on 7, 5 and 5 functions in the cases of 30-D, 50-D and 100-D, respectively. According to Table 2, jSO-LGP outperforms jSO on 6, 12 and 14 functions in 30-D, 50-D and 100-D cases, respectively, while being outperformed on just 1 function in each dimensional case. Compared with DTDE, DTDE-LGP wins 5, 11 and 16 functions but loses 4, 3 and 3 functions in the cases of 30-D, 50-D and 100-D from Table 3, respectively. As shown in Table 4, our ODF-SCSS-jSO-LGP exhibits superior performance compared with the original ODF-SCSS-jSO, achieving better results on 4 (30-D), 11 (50-D) and 11 (100-D) functions, whereas the baseline algorithm succeeds in only 4, 1 and 3 functions for the respective dimensions.

Furthermore, Fig. 2 presents the overall performance ranking of the four baseline algorithms and their corresponding LGP-integrated variants based on Friedman's test [61]. The results demonstrate that all LGP-integrated versions outperform their corresponding original counterparts, achieving substantially higher rankings. Notably, ODF-SCSS-jSO-LGP attains the highest ranking (3.45) among all evaluated methods.

Apart from the single-problem comparison, Table 6 reports the comparison results according to the multi-problem Wilcoxon signed-rank test [61]. From this table, L-SHADE-LGP, jSO-LGP, DTDE-LGP and ODF-SCSS-jSO-LGP achieve a much larger $R+$ than $R-$ when compared with their baseline algorithms, and the significant difference between LGP-integrated variants and their original counterparts can be observed at the significant level of 5% except for L-SHADE-LGP, indicating that the superiority of LGP is essentially significant.

The experimental results demonstrate that the LGP mechanism generates meaningful performance enhancements across different algorithm architectures, with especially significant improvements observed

Table 2
Performance comparison with jSO on 30-D, 50-D and 100-D CEC2017 functions.

D	30-D	30-D	50-D	50-D	100-D	100-D
Problem	jSO	jSO-LGP	jSO	jSO-LGP	jSO	jSO-LGP
F1	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)
F3	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	**6.28e−7 (6.55e−7)** −	3.34e−6 (5.57e−6)
F4	5.86e+1 (2.60e−14) ≈	5.86e+1 (2.13e−14)	6.03e+1 (4.96e+1) ≈	6.37e+1 (4.68e+1)	2.00e+2 (9.24e+0) ≈	1.99e+2 (9.86e+0)
F5	8.91e+0 (1.61e+0) ≈	8.05e+0 (1.85e+0)	1.58e+1 (2.74e+0) +	**1.37e+1 (2.60e+0)**	3.85e+1 (6.50e+0) +	**3.14e+1 (5.02e+0)**
F6	1.54e−8 (5.31e−8) ≈	1.62e−8 (5.52e−8)	6.15e−7 (1.38e−6) ≈	4.73e−7 (1.72e−6)	9.41e−5 (3.15e−4) ≈	2.92e−5 (2.33e−5)
F7	3.87e+1 (1.53e+0) +	**3.79e+1 (1.62e+0)**	6.67e+1 (3.39e+0) +	**6.38e+1 (1.98e+0)**	1.42e+2 (7.54e+0) +	**1.32e+2 (3.32e+0)**
F8	8.69e+0 (1.68e+0) ≈	8.02e+0 (1.63e+0)	1.66e+1 (2.85e+0) +	**1.45e+1 (2.29e+0)**	3.64e+1 (6.00e+0) +	**2.96e+1 (4.98e+0)**
F9	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	1.94e−2 (6.86e−2) +	**1.76e−3 (1.25e−2)**
F10	1.46e+3 (2.68e+2) ≈	1.54e+3 (2.05e+2)	3.11e+3 (4.24e+2) ≈	3.08e+3 (2.93e+2)	9.72e+3 (6.93e+2) +	**9.44e+3 (5.64e+2)**
F11	1.00e+1 (1.90e+1) ≈	4.51e+0 (8.30e+0)	2.71e+1 (3.04e+0) +	**2.58e+1 (3.14e+0)**	1.08e+2 (2.53e+1) +	**6.81e+1 (2.38e+1)**
F12	2.26e+2 (1.35e+2) +	**1.15e+2 (8.77e+1)**	1.77e+3 (4.67e+2) +	**1.48e+3 (4.02e+2)**	1.76e+4 (7.73e+3) +	**1.16e+4 (5.49e+3)**
F13	1.65e+1 (5.11e+0) ≈	1.54e+1 (4.92e+0)	3.69e+1 (2.08e+1) +	**2.83e+1 (1.84e+1)**	1.69e+2 (5.08e+1) +	**1.18e+2 (3.49e+1)**
F14	2.17e+1 (1.51e+0) ≈	2.17e+1 (1.22e+0)	2.40e+1 (2.08e+0) ≈	2.43e+1 (2.32e+0)	6.37e+1 (9.71e+0) +	**4.23e+1 (5.87e+0)**
F15	1.16e+0 (7.43e−1) +	**9.21e−1 (7.08e−1)**	2.35e+1 (2.46e+0) +	**2.11e+1 (2.21e+0)**	1.69e+2 (3.54e+1) +	**1.07e+2 (3.78e+1)**
F16	**6.03e+1 (7.27e+1)** −	7.27e+1 (6.87e+1)	4.39e+2 (1.21e+2) ≈	4.29e+2 (1.19e+2)	1.75e+3 (3.01e+2) ≈	1.72e+3 (3.52e+2)
F17	3.25e+1 (7.72e+0) ≈	3.45e+1 (7.07e+0)	2.70e+2 (1.08e+2) ≈	2.96e+2 (8.39e+1)	1.27e+3 (2.22e+2) ≈	1.23e+3 (1.80e+2)
F18	2.08e+1 (4.43e−1) ≈	2.08e+1 (2.82e−1)	2.47e+1 (1.99e+0) +	**2.26e+1 (1.30e+0)**	1.93e+2 (3.95e+1) +	**1.13e+2 (3.06e+1)**
F19	4.64e+0 (1.82e+0) ≈	3.92e+0 (1.21e+0)	1.38e+1 (2.39e+0) +	**1.26e+1 (2.98e+0)**	1.17e+2 (2.25e+1) +	**5.76e+1 (7.62e+0)**
F20	3.00e+1 (4.94e+0) ≈	3.19e+1 (4.68e+0)	**1.19e+2 (6.25e+1)** −	1.39e+2 (5.90e+1)	1.31e+3 (2.88e+2) ≈	1.40e+3 (1.70e+2)
F21	2.09e+2 (1.76e+0) +	**2.08e+2 (1.65e+0)**	2.18e+2 (2.75e+0) +	**2.15e+2 (3.18e+0)**	2.61e+2 (8.07e+0) +	**2.58e+2 (5.12e+0)**
F22	1.00e+2 (1.44e−14) ≈	1.00e+2 (6.39e−14)	1.38e+3 (1.76e+3) +	**8.72e+2 (1.47e+3)**	1.03e+4 (6.17e+2) ≈	1.01e+4 (5.60e+2)
F23	3.53e+2 (4.15e+0) ≈	3.53e+2 (4.25e+0)	4.35e+2 (4.30e+0) ≈	4.34e+2 (4.96e+0)	5.60e+2 (1.09e+1) ≈	5.60e+2 (9.61e+0)
F24	4.29e+2 (2.78e+0) +	**4.28e+2 (2.05e+0)**	5.13e+2 (4.04e+0) +	**5.11e+2 (3.10e+0)**	9.19e+2 (9.66e+0) ≈	9.19e+2 (8.21e+0)
F25	3.87e+2 (7.26e−3) ≈	3.87e+2 (6.99e−3)	4.82e+2 (4.03e+0) ≈	4.81e+2 (2.27e+0)	7.37e+2 (3.19e+1) ≈	7.31e+2 (3.74e+1)
F26	9.58e+2 (4.77e+1) +	**9.42e+2 (3.45e+1)**	1.18e+3 (4.77e+1) ≈	1.18e+3 (5.06e+1)	3.36e+3 (9.88e+1) ≈	3.36e+3 (1.23e+2)
F27	5.00e+2 (5.69e+0) ≈	5.01e+2 (6.00e+0)	5.13e+2 (7.99e+0) ≈	5.19e+2 (1.56e+1)	5.96e+2 (1.67e+1) ≈	5.91e+2 (1.88e+1)
F28	3.13e+2 (3.54e+1) ≈	3.13e+2 (3.60e+1)	4.59e+2 (2.05e−13) ≈	4.64e+2 (1.47e+1)	5.31e+2 (2.40e+1) +	**5.16e+2 (1.86e+1)**
F29	4.39e+2 (1.03e+1) ≈	4.41e+2 (1.32e+1)	3.54e+2 (1.53e+1) ≈	3.62e+2 (1.60e+1)	1.35e+3 (1.93e+2) ≈	1.36e+3 (1.70e+2)
F30	1.97e+3 (2.69e+1) ≈	1.97e+3 (2.62e+1)	6.16e+5 (3.22e+4) ≈	6.13e+5 (4.57e+4)	2.33e+3 (1.43e+2) ≈	2.33e+3 (1.32e+2)
+/ − / ≈	6/1/22		12/1/16		14/1/14	

Table 3
Performance comparison with DTDE on 30-D, 50-D and 100-D CEC2017 functions.

D	30-D	30-D	50-D	50-D	100-D	100-D
Problem	DTDE	DTDE-LGP	DTDE	DTDE-LGP	DTDE	DTDE-LGP
F1	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	**0.00e+0 (0.00e+0)** −	4.56e−9 (1.55e−8)
F3	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	5.53e−10 (2.82e−9)	**1.17e−4 (1.35e−4)** −	1.90e−1 (1.80e−1)
F4	**5.86e+1 (2.41e−14)** −	5.86e+1 (2.87e−14)	8.17e+1 (5.14e+1) ≈	6.93e+1 (5.21e+1)	1.97e+2 (1.16e+1) ≈	1.99e+2 (1.26e+1)
F5	1.44e+0 (1.20e+0) ≈	1.64e+0 (1.22e+0)	**1.72e+0 (1.44e+0)** −	2.36e+0 (1.38e+0)	5.33e+0 (2.26e+0) +	**3.73e+0 (1.66e+0)**
F6	**2.68e−9 (1.92e−8)** −	4.71e−8 (2.14e−7)	**1.14e−8 (2.31e−8)** −	1.45e−7 (1.99e−7)	2.02e−3 (1.58e−3) +	**2.88e−4 (5.70e−4)**
F7	3.36e+1 (8.79e−1) ≈	3.41e+1 (8.21e−1)	5.67e+1 (1.00e+0) ≈	5.68e+1 (1.13e+0)	**1.11e+2 (1.33e+0)** −	1.13e+2 (1.60e+0)
F8	1.72e+0 (1.20e+0) ≈	1.62e+0 (1.26e+0)	1.68e+0 (1.25e+0) ≈	2.24e+0 (1.56e+0)	4.57e+0 (2.26e+0) +	**3.32e+0 (2.01e+0)**
F9	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	9.38e−2 (1.62e−1) +	**7.02e−3 (2.43e−2)**
F10	3.38e+1 (6.83e+1) ≈	6.43e+1 (1.07e+2)	1.95e+2 (1.31e+2) +	**1.60e+2 (8.15e+1)**	3.92e+2 (2.41e+2) ≈	3.61e+2 (2.68e+2)
F11	2.21e+1 (2.66e+1) ≈	2.03e+1 (2.62e+1)	3.18e+1 (4.91e+0) +	**2.48e+1 (2.95e+0)**	1.67e+2 (4.28e+1) +	**1.29e+2 (5.74e+1)**
F12	5.91e+2 (2.28e+2) +	**2.58e+2 (1.35e+2)**	1.99e+3 (4.35e+2) +	**1.64e+3 (4.77e+2)**	1.50e+4 (5.62e+3) ≈	1.56e+4 (7.19e+3)
F13	1.87e+1 (5.24e+0) ≈	1.91e+1 (4.85e+0)	5.03e+1 (3.19e+1) +	**4.05e+1 (2.15e+1)**	1.66e+2 (3.85e+1) +	**1.11e+2 (3.45e+1)**
F14	2.23e+1 (5.65e+0) ≈	2.27e+1 (6.52e+0)	2.57e+1 (1.93e+0) ≈	2.46e+1 (5.31e+0)	7.69e+1 (1.62e+1) +	**3.84e+1 (4.90e+0)**
F15	5.53e+0 (2.12e+0) ≈	5.38e+0 (2.04e+0)	2.63e+1 (4.11e+0) +	**2.27e+1 (2.32e+0)**	2.57e+2 (4.86e+1) +	**1.37e+2 (3.71e+1)**
F16	1.41e+1 (1.45e+0) ≈	1.40e+1 (1.91e+0)	1.40e+2 (4.25e+1) ≈	1.40e+2 (4.15e+1)	2.29e+2 (1.18e+2) ≈	2.29e+2 (1.42e+2)
F17	2.12e+1 (4.90e+0) ≈	1.99e+1 (6.54e+0)	5.03e+1 (6.33e+1) ≈	6.05e+1 (7.15e+1)	9.40e+1 (9.94e+1) ≈	6.12e+1 (5.70e+1)
F18	2.63e+1 (3.25e+0) ≈	2.73e+1 (6.12e+0)	**3.42e+1 (6.81e+0)** −	3.81e+1 (7.34e+0)	2.11e+2 (4.49e+1) +	**1.79e+2 (4.16e+1)**
F19	5.23e+0 (1.65e+0) ≈	5.15e+0 (1.88e+0)	1.04e+1 (2.70e+0) +	**8.02e+0 (2.30e+0)**	1.66e+2 (2.69e+1) +	**5.61e+1 (1.38e+1)**
F20	1.74e+1 (8.40e+0) ≈	1.47e+1 (9.73e+0)	2.30e+1 (3.38e+0) ≈	2.42e+1 (3.45e+0)	1.83e+2 (6.42e+1) ≈	1.75e+2 (5.60e+1)
F21	2.02e+2 (1.74e+0) ≈	2.03e+2 (1.55e+0)	2.03e+2 (2.42e+0) ≈	2.03e+2 (2.12e+0)	2.27e+2 (3.95e+0) +	**2.25e+2 (2.69e+0)**
F22	**1.00e+2 (1.44e−14)** −	1.00e+2 (2.22e−13)	3.45e+2 (1.38e+2) ≈	3.48e+2 (1.80e+2)	1.05e+3 (3.40e+2) ≈	1.02e+3 (3.49e+2)
F23	3.47e+2 (3.29e+0) ≈	3.47e+2 (3.28e+0)	4.21e+2 (7.96e+0) +	**4.18e+2 (6.02e+0)**	5.49e+2 (8.82e+0) +	**5.38e+2 (7.37e+0)**
F24	4.24e+2 (1.14e+0) ≈	4.23e+2 (1.43e+0)	5.01e+2 (2.13e+0) +	**4.98e+2 (2.49e+0)**	8.96e+2 (7.35e+0) +	**8.83e+2 (5.11e+0)**
F25	3.87e+2 (1.74e−2) +	**3.87e+2 (7.91e−3)**	4.81e+2 (3.48e+0) +	**4.81e+2 (3.19e+0)**	7.37e+2 (3.88e+1) ≈	7.38e+2 (3.92e+1)
F26	8.90e+2 (4.29e+1) +	**8.40e+2 (6.51e+1)**	1.03e+3 (5.08e+1) +	**9.53e+2 (7.33e+1)**	3.14e+3 (8.66e+1) +	**2.93e+3 (9.36e+1)**
F27	5.02e+2 (6.71e+0) +	**4.98e+2 (6.64e+0)**	5.27e+2 (1.47e+1) +	**5.19e+2 (1.19e+1)**	6.16e+2 (1.56e+1) +	**5.99e+2 (1.75e+1)**
F28	**3.17e+2 (4.04e+1)** −	3.21e+2 (4.36e+1)	4.63e+2 (1.33e+1) ≈	4.61e+2 (9.58e+0)	5.33e+2 (2.79e+1) ≈	5.23e+2 (2.17e+1)
F29	4.12e+2 (6.79e+0) +	**4.10e+2 (6.86e+0)**	3.03e+2 (8.09e+0) ≈	3.02e+2 (8.74e+0)	8.26e+2 (1.48e+2) +	**7.75e+2 (1.10e+2)**
F30	1.97e+3 (3.23e+1) ≈	1.97e+3 (2.53e+1)	6.55e+5 (6.56e+4) ≈	6.40e+5 (6.36e+4)	2.37e+3 (1.21e+2) ≈	2.44e+3 (2.39e+2)
+/ − / ≈	5/4/20		11/3/15		16/3/10	

Table 4
Performance comparison with ODF-SCSS-jSO on 30-D, 50-D and 100-D CEC2017 functions.

D Problem	30-D ODF-SCSS-jSO	30-D ODF-SCSS-jSO-LGP	50-D ODF-SCSS-jSO	50-D ODF-SCSS-jSO-LGP	100-D ODF-SCSS-jSO	100-D ODF-SCSS-jSO-LGP
F1	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)
F3	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	**2.76e−8 (3.29e−8)** −	1.49e−6 (2.86e−6)
F4	5.86e+1 (1.80e−14) ≈	5.86e+1 (1.14e−14)	4.37e+1 (3.66e+1) +	**3.54e+1 (3.39e+1)**	2.04e+2 (9.73e+0) ≈	2.04e+2 (9.85e+0)
F5	4.79e+0 (1.44e+0) ≈	4.61e+0 (2.03e+0)	8.02e+0 (1.98e+0) +	**6.09e+0 (2.35e+0)**	**1.31e+1 (3.72e+0)** −	1.52e+1 (4.11e+0)
F6	1.25e−7 (6.26e−7) ≈	7.55e−8 (1.95e−7)	1.73e−6 (2.47e−6) ≈	2.71e−6 (3.93e−6)	**7.46e−6 (2.62e−6)** −	6.66e−5 (3.11e−5)
F7	3.55e+1 (1.15e+0) ≈	3.56e+1 (1.19e+0)	5.93e+1 (1.54e+0) ≈	5.93e+1 (1.23e+0)	1.17e+2 (2.30e+0) ≈	1.18e+2 (2.28e+0)
F8	5.50e+0 (1.70e+0) +	**4.19e+0 (1.69e+0)**	8.06e+0 (2.46e+0) +	**6.07e+0 (2.45e+0)**	1.38e+1 (2.96e+0) ≈	1.51e+1 (4.04e+0)
F9	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)
F10	1.47e+3 (2.03e+2) ≈	1.48e+3 (1.93e+2)	2.91e+3 (3.29e+2) ≈	3.06e+3 (3.13e+2)	8.66e+3 (5.51e+2) ≈	8.78e+3 (5.86e+2)
F11	2.27e+1 (2.83e+1) +	**1.09e+1 (2.16e+1)**	2.46e+1 (3.73e+0) ≈	2.38e+1 (3.68e+0)	3.59e+1 (2.71e+1) +	**2.93e+1 (2.42e+1)**
F12	1.78e+2 (1.31e+2) +	**1.37e+2 (1.09e+2)**	1.25e+3 (3.88e+2) +	**9.60e+2 (3.68e+2)**	1.22e+4 (3.66e+3) +	**1.10e+4 (5.43e+3)**
F13	1.80e+1 (3.67e+0) ≈	1.71e+1 (3.13e+0)	2.55e+1 (1.53e+1) ≈	2.25e+1 (1.61e+1)	1.10e+2 (4.72e+1) +	**5.74e+1 (2.40e+1)**
F14	**2.06e+1 (7.36e−1)** −	2.05e+1 (2.92e+0)	2.28e+1 (1.47e+0) ≈	2.28e+1 (1.45e+0)	3.63e+1 (4.07e+0) +	**3.26e+1 (2.83e+0)**
F15	5.58e−1 (3.91e−1) ≈	6.13e−1 (4.40e−1)	1.96e+1 (1.56e+0) ≈	1.91e+1 (1.02e+0)	6.77e+1 (3.47e+1) +	**4.55e+1 (3.07e+1)**
F16	3.99e+1 (5.24e+1) ≈	2.90e+1 (2.26e+1)	3.75e+2 (1.20e+2) +	**3.05e+2 (1.31e+2)**	1.39e+3 (2.90e+2) +	**1.21e+3 (3.84e+2)**
F17	3.23e+1 (5.50e+0) ≈	3.43e+1 (5.37e+0)	2.24e+2 (6.28e+1) ≈	2.18e+2 (8.31e+1)	9.25e+2 (2.28e+2) ≈	9.20e+2 (2.42e+2)
F18	**2.03e+1 (2.84e+0)**	2.07e+1 (2.89e−1)	2.26e+1 (1.50e+0) +	**2.19e+1 (1.10e+0)**	7.87e+1 (2.66e+1) +	**5.88e+1 (1.65e+1)**
F19	4.08e+0 (1.40e+0) ≈	3.70e+0 (1.35e+0)	1.08e+1 (2.51e+0) +	**9.25e+0 (2.01e+0)**	4.77e+1 (6.53e+0) +	**3.76e+1 (4.00e+0)**
F20	**3.18e+1 (4.55e+0)** −	3.50e+1 (7.14e+0)	1.22e+2 (5.51e+1) ≈	1.22e+2 (5.34e+1)	1.15e+3 (1.66e+2) ≈	1.13e+3 (1.78e+2)
F21	2.06e+2 (1.61e+0) ≈	2.06e+2 (1.41e+0)	2.11e+2 (2.31e+0) +	**2.09e+2 (2.48e+0)**	2.43e+2 (3.26e+0) ≈	2.44e+2 (4.97e+0)
F22	1.00e+2 (1.44e−14) ≈	1.00e+2 (1.44e−14)	1.42e+3 (1.65e+3) ≈	1.51e+3 (1.63e+3)	8.62e+3 (1.20e+3) ≈	8.75e+3 (6.58e+2)
F23	3.52e+2 (3.07e+0) +	**3.50e+2 (3.09e+0)**	**4.32e+2 (5.19e+0)** −	4.35e+2 (7.06e+0)	5.68e+2 (8.50e+0) ≈	5.68e+2 (8.89e+0)
F24	4.26e+2 (1.72e+0) ≈	4.26e+2 (1.66e+0)	5.10e+2 (2.68e+0) ≈	5.10e+2 (2.66e+0)	9.08e+2 (6.80e+0) ≈	9.09e+2 (6.09e+0)
F25	3.87e+2 (9.12e−3) ≈	3.87e+2 (7.12e−3)	4.80e+2 (1.62e+0) +	**4.80e+2 (1.62e+0)**	7.47e+2 (3.46e+1) +	**7.24e+2 (4.67e+1)**
F26	**9.36e+2 (3.75e+1)** −	9.58e+2 (4.07e+1)	1.15e+3 (4.62e+1) ≈	1.17e+3 (5.84e+1)	3.25e+3 (8.14e+1) ≈	3.22e+3 (8.79e+1)
F27	5.02e+2 (4.89e+0) ≈	5.04e+2 (4.74e+0)	5.15e+2 (7.17e+0) +	**5.11e+2 (7.38e+0)**	5.86e+2 (1.46e+1) +	**5.77e+2 (1.24e+1)**
F28	3.13e+2 (3.54e+1) ≈	3.04e+2 (2.23e+1)	4.59e+2 (2.59e−13) +	**4.59e+2 (2.00e−13)**	5.20e+2 (1.08e+1) ≈	5.16e+2 (1.02e+1)
F29	4.35e+2 (6.80e+0) ≈	4.39e+2 (7.68e+0)	3.56e+2 (1.26e+1) ≈	3.56e+2 (1.11e+1)	1.16e+3 (1.71e+2) ≈	1.16e+3 (1.33e+2)
F30	1.98e+3 (3.25e+1) ≈	1.98e+3 (3.81e+1)	5.92e+5 (1.74e+4) ≈	5.98e+5 (2.43e+4)	2.42e+3 (1.36e+2) +	**2.35e+3 (1.03e+2)**
+/ − / ≈	4/4/21		11/1/17		11/3/15	

Table 5
Summary performance comparisons of LGP-integrated variants with L-SHADE, jSO, DTDE and ODF-SCSS-jSO on 30-D, 50-D and 100-D CEC2017 functions according to single-problem Wilcoxon rank-sum test.

+/ − / ≈	30-D	50-D	100-D
L-SHADE	**7**/7/15	**12**/5/12	**14**/5/10
jSO	**6**/1/22	**12**/1/16	**14**/1/14
DTDE	**5**/4/20	**11**/3/15	**16**/3/10
ODF-SCSS-jSO	**4**/4/21	**11**/1/17	**11**/3/15

Table 6
Comparison results with the corresponding baseline algorithms on CEC2017 functions according to multi-problem Wilcoxon signed-rank test.

Algorithm	v.s.	$R+$	$R-$	p-value
L-SHADE-LGP	L-SHADE	2224.0	1517.0	0.1274
jSO-LGP	jSO	2379.0	1362.0	**0.0284**
DTDE-LGP	DTDE	2763.5	977.5	**0.0001**
ODF-SCSS-jSO-LGP	ODF-SCSS-jSO	2376.5	1364.5	**0.0292**

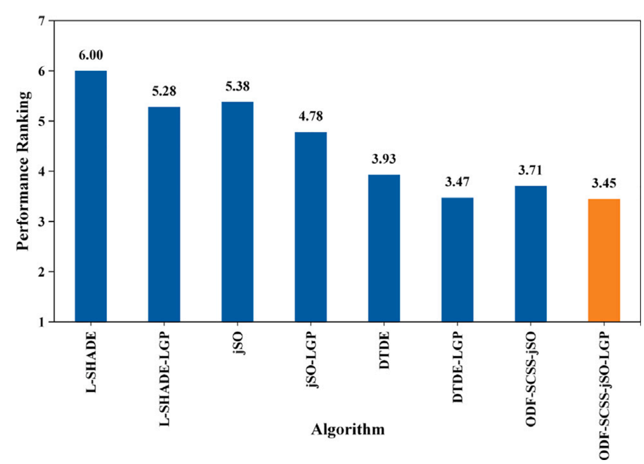

Fig. 2. Overall performance ranking of the four algorithms and their variants.

in high-dimensional scenarios. The consistent performance improvement across diverse testing conditions demonstrates the robustness and versatility of the LGP mechanism.

4.2. Comparisons with state-of-the-art DEs

To further assess the performance of ODF-SCSS-jSO-LGP, we use the following seven State-of-the-Art DE algorithms for comparison.

EJADE [62]: An adaptive DE algorithm with ensembling operators.

SCSS-L-SHADE [54]: A variant of L-SHADE algorithm based on a similarity selection rule.

DISH [19]: An improved jSO algorithm utilizing a novel distance based parameter adaptation mechanism.

L-SHADE-cnEpSin [24]: Shorten as "CnEpSin". An enhanced L-SHADE algorithm with an ensemble of sinusoidal approaches based on performance adaptation and covariance matrix learning for the crossover operator.

L-SHADE-RSP [63]: Shorten as "RSP". An improved jSO algorithm with a rank-based selective pressure strategy for mutation.

ESADE [64]: An adaptive DE algorithm based on evolutionary scale adaptation.

CELDE [65]: An adaptive DE algorithm based on collective ensemble learning paradigm.

For convenience, ODF-SCSS-jSO-LGP is written as LGPDE in the following text. The parameter settings for the aforementioned seven algorithms are meticulously maintained according to their source references to ensure equitable evaluation. Tables 7–9 present the comparisons of LGPDE with other DE algorithms on the 30-D, 50-D and 100-D CEC2017 benchmark suite respectively, where the best performance (smallest mean error value) for each function is highlighted with a gray background. Furthermore, Table 10 shows the summary comparison results of them.

Table 7
Performance comparison of LGPDE with State-of-the-Art DEs on 30-D CEC2017 functions.

Problem	EJADE	SCSS-L-SHADE	DISH	CnEpSin	RSP	ESADE	CELDE	LGPDE
F1	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)
F3	7.35e−5 (4.26e−4) +	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)
F4	5.07e+1 (2.13e+1) ≈	5.86e+1 (2.90e−14) +	5.86e+1 (2.54e−14) +	4.08e+1 (3.42e+0) −	5.86e+1 (1.97e−14) ≈	5.02e+1 (1.93e+0) −	4.41e+1 (8.80e+0) −	5.86e+1 (1.14e−14)
F5	1.92e+1 (5.19e+0) +	6.52e+0 (1.52e+0) +	7.65e+0 (2.11e+0) +	1.17e+1 (2.37e+0) +	6.96e+0 (2.04e+0) +	7.32e+0 (1.23e+0) +	7.89e+0 (2.16e+0) +	4.61e+0 (2.03e+0)
F6	0.00e+0 (0.00e+0) −	0.00e+0 (0.00e+0) −	1.41e−8 (3.76e−8) ≈	1.75e−8 (4.48e−8) ≈	5.37e−9 (2.68e−8) −	1.22e−7 (2.35e−7) ≈	0.00e+0 (0.00e+0) −	7.55e−8 (1.95e−7)
F7	4.91e+1 (4.91e+0) +	3.82e+1 (1.74e+0) +	3.93e+1 (1.90e+0) +	4.31e+1 (2.11e+0) +	3.85e+1 (1.72e+0) +	3.82e+1 (1.63e+0) +	3.94e+1 (1.88e+0) +	3.56e+1 (1.19e+0)
F8	2.05e+1 (4.73e+0) +	7.57e+0 (1.78e+0) +	7.54e+0 (2.11e+0) +	1.26e+1 (2.28e+0) +	7.49e+0 (2.13e+0) +	7.59e+0 (1.70e+0) +	8.68e+0 (1.99e+0) +	4.19e+0 (1.69e+0)
F9	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)
F10	1.59e+3 (4.27e+2) ≈	1.44e+3 (2.22e+2) ≈	1.65e+3 (2.80e+2) +	1.41e+3 (2.16e+2) ≈	1.58e+3 (2.46e+2) +	1.52e+3 (2.41e+2) ≈	1.49e+3 (2.61e+2) ≈	1.48e+3 (1.93e+2)
F11	1.34e+1 (2.13e+1) +	2.30e+1 (2.66e+1) +	4.03e+0 (8.40e+0) +	1.74e+1 (2.22e+1) +	9.56e+0 (1.90e+1) +	1.16e+1 (2.00e+1) +	4.73e+0 (1.04e+1) ≈	1.09e+1 (2.16e+1)
F12	1.31e+3 (1.84e+3) +	5.40e+2 (2.72e+2) +	1.00e+2 (7.51e+1) −	3.90e+2 (2.29e+2) +	1.34e+2 (9.26e+1) ≈	1.08e+2 (8.48e+1) ≈	1.51e+2 (1.20e+2) ≈	1.37e+2 (1.09e+2)
F13	2.21e+1 (1.02e+1) +	1.47e+1 (5.98e+0) ≈	2.06e+1 (3.86e+0) +	1.61e+1 (9.64e+0) ≈	1.86e+1 (3.92e+0) +	1.71e+1 (5.65e+0) +	1.78e+1 (4.06e+0) +	1.71e+1 (3.13e+0)
F14	1.13e+1 (8.54e+0) −	2.19e+1 (2.77e+0) +	2.18e+1 (1.24e+0) +	2.06e+1 (4.62e+0) +	2.14e+1 (1.17e+0) +	1.91e+1 (5.47e+0) ≈	1.90e+1 (6.44e+0) ≈	2.05e+1 (2.92e+0)
F15	5.47e+0 (2.90e+0) +	1.86e+0 (1.07e+0) +	2.14e+0 (1.25e+0) +	2.82e+0 (1.64e+0) +	1.03e+0 (6.69e−1) +	2.47e+0 (2.06e+0) +	1.19e+0 (1.14e+0) +	6.13e−1 (4.40e−1)
F16	2.60e+2 (1.63e+2) +	5.10e+1 (5.12e+1) +	2.88e+1 (4.05e+1) −	2.29e+1 (3.13e+1) −	2.03e+1 (1.93e+1) −	1.96e+1 (1.75e+1) −	2.18e+1 (3.51e+1) −	2.90e+1 (2.26e+1)
F17	1.53e+1 (9.27e+0) −	3.46e+1 (5.36e+0) ≈	3.75e+1 (6.74e+0) +	2.82e+1 (6.59e+0) −	3.32e+1 (7.08e+0) ≈	3.51e+1 (5.15e+0) ≈	3.24e+1 (5.69e+0) ≈	3.43e+1 (5.37e+0)
F18	2.49e+1 (1.51e+1) +	2.09e+1 (5.36e−1) −	2.08e+1 (2.04e−1) +	2.13e+1 (9.68e−1) ≈	2.08e+1 (2.61e−1) ≈	1.98e+1 (3.94e+0) −	1.99e+1 (3.98e+0) −	2.07e+1 (2.89e−1)
F19	5.45e+0 (2.20e+0) +	5.00e+0 (1.76e+0) +	4.54e+0 (1.31e+0) +	5.65e+0 (1.50e+0) +	3.41e+0 (7.34e−1) ≈	5.72e+0 (1.00e+0) +	4.88e+0 (1.72e+0) +	3.70e+0 (1.35e+0)
F20	2.44e+1 (4.36e+1) −	2.97e+1 (5.03e+0) −	3.33e+1 (8.40e+0) ≈	3.07e+1 (5.36e+0) −	3.23e+1 (5.21e+0) ≈	3.29e+1 (4.55e+0) ≈	3.01e+1 (4.75e+0) −	3.50e+1 (7.14e+0)
F21	2.22e+2 (4.86e+0) +	2.08e+2 (1.50e+0) +	2.08e+2 (1.98e+0) +	2.13e+2 (2.37e+0) +	2.07e+2 (2.49e+0) +	2.08e+2 (1.33e+0) +	2.09e+2 (1.84e+0) +	2.06e+2 (1.41e+0)
F22	1.00e+2 (1.44e−14) ≈	1.00e+2 (1.44e−14) ≈	1.00e+2 (1.24e−13) +	1.00e+2 (1.44e−14) ≈	1.00e+2 (1.44e−14) ≈	1.00e+2 (1.08e−13) ≈	1.00e+2 (1.44e−14) ≈	1.00e+2 (1.44e−14)
F23	3.69e+2 (6.48e+0) +	3.50e+2 (2.54e+0) ≈	3.51e+2 (3.73e+0) ≈	3.55e+2 (3.26e+0) +	3.51e+2 (3.47e+0) +	3.50e+2 (3.28e+0) ≈	3.50e+2 (3.01e+0) ≈	3.50e+2 (3.09e+0)
F24	4.37e+2 (5.88e+0) +	4.26e+2 (1.68e+0) ≈	4.27e+2 (3.71e+0) ≈	4.28e+2 (3.07e+0) +	4.27e+2 (1.94e+0) ≈	4.26e+2 (1.72e+0) ≈	4.25e+2 (1.84e+0) −	4.26e+2 (1.66e+0)
F25	3.87e+2 (8.64e−2) +	3.87e+2 (1.48e−2) +	3.87e+2 (6.59e−3) ≈	3.87e+2 (7.73e−3) −	3.87e+2 (5.70e−3) ≈	3.87e+2 (6.66e−3) −	3.87e+2 (9.34e−3) −	3.87e+2 (7.12e−3)
F26	1.10e+3 (6.50e+1) +	9.23e+2 (3.38e+1) −	9.29e+2 (4.34e+1) −	9.43e+2 (5.81e+1) ≈	9.34e+2 (3.89e+1) −	8.88e+2 (4.48e+1) −	8.94e+2 (3.95e+1) −	9.58e+2 (4.07e+1)
F27	5.01e+2 (7.44e+0) ≈	5.01e+2 (5.32e+0) ≈	4.96e+2 (7.43e+0) −	5.05e+2 (5.35e+0) ≈	5.00e+2 (6.55e+0) −	5.00e+2 (6.58e+0) −	5.00e+2 (6.66e+0) −	5.04e+2 (4.74e+0)
F28	3.34e+2 (5.16e+1) ≈	3.10e+2 (3.56e+1) ≈	3.00e+2 (2.36e−13) +	3.13e+2 (3.54e+1) ≈	3.02e+2 (1.45e+1) ≈	3.17e+2 (4.04e+1) ≈	3.15e+2 (3.86e+1) ≈	3.04e+2 (2.23e+1)
F29	4.14e+2 (1.29e+1) −	4.38e+2 (1.33e+1) ≈	4.46e+2 (1.61e+1) +	4.34e+2 (1.18e+1) −	4.46e+2 (1.19e+1) +	4.37e+2 (9.77e+0) ≈	4.36e+2 (8.97e+0) ≈	4.39e+2 (7.68e+0)
F30	2.01e+3 (1.04e+2) +	1.98e+3 (4.10e+1) ≈	1.97e+3 (9.76e+0) ≈	1.98e+3 (4.65e+1) ≈	1.97e+3 (1.76e+1) ≈	1.97e+3 (1.13e+1) ≈	1.97e+3 (3.46e+1) ≈	1.98e+3 (3.81e+1)
+/ − / ≈	17/5/7	12/4/13	16/4/9	11/6/12	11/4/14	8/6/15	7/9/13	

Table 8
Performance comparison of LGPDE with State-of-the-Art DEs on 50-D CEC2017 functions.

Problem	EJADE	SCSS-L-SHADE	DISH	CnEpSin	RSP	ESADE	CELDE	LGPDE
F1	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)
F3	1.57e+2 (4.66e+2) +	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)
F4	4.78e+1 (4.63e+1) ≈	7.83e+1 (5.21e+1) +	5.03e+1 (4.48e+1) +	5.31e+1 (4.71e+1) ≈	4.19e+1 (4.16e+1) +	5.91e+1 (5.01e+1) ≈	4.48e+1 (4.05e+1) ≈	3.54e+1 (3.39e+1)
F5	4.18e+1 (7.68e+0) +	1.17e+1 (2.21e+0) +	1.31e+1 (3.38e+0) +	2.46e+1 (6.06e+0) +	1.28e+1 (4.23e+0) +	1.25e+1 (2.29e+0) +	1.54e+1 (3.26e+0) +	6.09e+0 (2.35e+0)
F6	6.07e−7 (2.13e−6) −	6.86e−8 (2.35e−7) −	1.73e−7 (3.01e−7) −	7.54e−7 (7.63e−7) −	8.63e−8 (1.62e−7) −	1.27e−6 (1.47e−6) ≈	5.88e−9 (1.56e−8) −	2.71e−6 (3.93e−6)
F7	8.80e+1 (7.83e+0) +	6.37e+1 (2.20e+0) +	6.84e+1 (3.62e+0) +	7.81e+1 (7.15e+0) +	6.61e+1 (3.65e+0) +	6.38e+1 (2.15e+0) +	6.67e+1 (3.28e+0) +	5.93e+1 (1.23e+0)
F8	4.31e+1 (8.85e+0) +	1.12e+1 (2.36e+0) +	1.29e+1 (3.67e+0) +	2.72e+1 (6.95e+0) +	1.34e+1 (3.08e+0) +	1.32e+1 (2.21e+0) +	1.52e+1 (3.62e+0) +	6.07e+0 (2.45e+0)
F9	5.03e−1 (6.34e−1) +	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)
F10	3.22e+3 (6.19e+2) ≈	3.19e+3 (2.79e+2) +	3.61e+3 (4.07e+2) +	3.17e+3 (3.01e+2) +	3.40e+3 (3.99e+2) +	3.09e+3 (3.77e+2) ≈	3.25e+3 (3.99e+2) +	3.06e+3 (3.13e+2)
F11	4.42e+1 (1.01e+1) +	3.25e+1 (4.27e+0) +	2.32e+1 (3.54e+0) ≈	2.17e+1 (2.14e+0) −	2.40e+1 (3.66e+0) ≈	2.16e+1 (2.25e+0) −	2.09e+1 (3.12e+0) −	2.38e+1 (3.68e+0)
F12	8.52e+3 (8.71e+3) +	1.97e+3 (5.52e+2) +	1.41e+3 (4.16e+2) +	1.31e+3 (3.57e+2) +	1.63e+3 (3.87e+2) +	9.38e+2 (3.04e+2) ≈	9.31e+2 (3.39e+2) ≈	9.60e+2 (3.68e+2)
F13	7.70e+1 (5.93e+1) +	4.24e+1 (2.89e+1) +	3.15e+1 (2.55e+1) +	6.79e+1 (4.20e+1) +	2.47e+1 (1.67e+1) +	5.34e+1 (3.42e+1) +	3.77e+1 (2.53e+1) +	2.25e+1 (1.61e+1)
F14	5.26e+1 (2.46e+1) +	2.57e+1 (1.91e+0) +	2.42e+1 (2.41e+0) +	2.65e+1 (2.18e+0) +	2.29e+1 (2.04e+0) ≈	2.36e+1 (1.25e+0) +	2.41e+1 (1.81e+0) +	2.28e+1 (1.45e+0)
F15	6.60e+1 (6.54e+1) +	2.79e+1 (4.73e+0) +	2.11e+1 (2.06e+0) +	2.55e+1 (3.28e+0) +	2.05e+1 (1.68e+0) +	1.97e+1 (1.25e+0) +	1.90e+1 (1.28e+0) ≈	1.91e+1 (1.02e+0)
F16	7.03e+2 (2.83e+2) +	3.50e+2 (1.29e+2) ≈	3.93e+2 (1.55e+2) +	2.62e+2 (1.05e+2) ≈	3.51e+2 (1.66e+2) ≈	3.19e+2 (1.02e+2) ≈	2.74e+2 (1.08e+2) ≈	3.05e+2 (1.31e+2)
F17	4.61e+2 (1.89e+2) +	2.36e+2 (8.54e+1) ≈	2.84e+2 (1.18e+2) +	2.31e+2 (6.36e+1) ≈	2.26e+2 (9.47e+1) ≈	2.08e+2 (8.57e+1) ≈	2.12e+2 (6.95e+1) ≈	2.18e+2 (8.31e+1)
F18	1.18e+2 (1.06e+2) +	2.72e+1 (3.32e+0) +	2.27e+1 (1.28e+0) +	2.46e+1 (2.07e+0) +	2.27e+1 (1.14e+0) +	2.24e+1 (1.17e+0) +	2.22e+1 (1.29e+0) ≈	2.19e+1 (1.10e+0)
F19	3.58e+1 (3.66e+1) +	1.51e+1 (2.96e+0) +	1.16e+1 (2.51e+0) +	1.69e+1 (2.40e+0) +	1.06e+1 (2.18e+0) +	1.27e+1 (2.03e+0) +	1.14e+1 (2.67e+0) +	9.25e+0 (2.01e+0)
F20	3.60e+2 (1.83e+2) +	1.69e+2 (7.55e+1) +	1.63e+2 (9.65e+1) ≈	1.10e+2 (2.12e+1) ≈	1.32e+2 (5.55e+1) ≈	1.05e+2 (3.19e+1) −	1.03e+2 (3.80e+1) −	1.22e+2 (5.34e+1)
F21	2.43e+2 (9.03e+0) +	2.13e+2 (2.54e+0) +	2.15e+2 (3.82e+0) +	2.26e+2 (6.98e+0) +	2.13e+2 (3.98e+0) +	2.15e+2 (2.58e+0) +	2.17e+2 (3.41e+0) +	2.09e+2 (2.48e+0)
F22	3.19e+3 (1.64e+3) +	2.64e+3 (1.62e+3) +	1.40e+3 (1.88e+3) +	1.28e+3 (1.61e+3) ≈	1.82e+3 (1.86e+3) +	2.19e+3 (1.77e+3) +	1.95e+3 (1.79e+3) +	1.51e+3 (1.63e+3)
F23	4.65e+2 (1.21e+1) +	4.27e+2 (4.21e+0) −	4.31e+2 (6.64e+0) −	4.39e+2 (7.43e+0) +	4.33e+2 (5.70e+0) ≈	4.27e+2 (4.26e+0) −	4.30e+2 (5.06e+0) −	4.35e+2 (7.06e+0)
F24	5.30e+2 (1.07e+1) +	5.06e+2 (2.96e+0) −	5.08e+2 (3.66e+0) −	5.13e+2 (5.91e+0) ≈	5.08e+2 (3.74e+0) −	5.07e+2 (3.06e+0) −	5.06e+2 (3.35e+0) −	5.10e+2 (2.66e+0)
F25	5.18e+2 (3.31e+1) +	4.82e+2 (4.56e+0) +	4.80e+2 (1.24e−2) −	4.81e+2 (2.68e+0) −	4.81e+2 (2.36e+0) −	4.80e+2 (8.71e−1) −	4.81e+2 (2.37e+0) −	4.80e+2 (1.62e+0)
F26	1.45e+3 (1.15e+2) +	1.12e+3 (5.49e+1) −	1.12e+3 (5.18e+1) −	1.21e+3 (9.61e+1) +	1.13e+3 (4.37e+1) −	1.08e+3 (5.20e+1) −	1.08e+3 (4.65e+1) −	1.17e+3 (5.84e+1)
F27	5.26e+2 (1.83e+1) +	5.29e+2 (1.55e+1) +	5.07e+2 (9.43e+0) −	5.30e+2 (1.52e+1) +	5.14e+2 (9.03e+0) ≈	5.19e+2 (1.37e+1) +	5.18e+2 (7.21e+0) +	5.11e+2 (7.38e+0)
F28	4.86e+2 (2.26e+1) +	4.64e+2 (1.51e+1) ≈	4.59e+2 (1.84e−13) −	4.58e+2 (9.55e+0) −	4.59e+2 (1.92e−13) −	4.58e+2 (2.12e−1) −	4.58e+2 (9.61e+0) −	4.59e+2 (2.00e−13)
F29	3.41e+2 (5.74e+1) −	3.59e+2 (1.33e+1) ≈	3.74e+2 (1.62e+1) +	3.54e+2 (9.81e+0) ≈	3.66e+2 (1.59e+1) +	3.58e+2 (1.34e+1) ≈	3.61e+2 (1.35e+1) +	3.56e+2 (1.11e+1)
F30	6.01e+5 (2.54e+4) ≈	6.41e+5 (5.64e+4) +	6.01e+5 (2.66e+4) ≈	6.53e+5 (7.55e+4) +	6.08e+5 (3.62e+4) ≈	6.46e+5 (7.34e+4) +	6.07e+5 (4.02e+4) ≈	5.98e+5 (2.43e+4)
+/ − / ≈	23/2/4	18/4/7	16/7/6	15/4/10	13/5/11	12/7/10	11/8/10	

Table 10 demonstrates that LGPDE achieves competitive or superior performance relative to other algorithms across most test functions. The 30-D evaluation reveals favorable performance, with "+/–" metrics of 17/5 (v.s. EJADE), 12/4 (v.s. SCSS-L-SHADE), 16/4 (v.s. DISH), 11/6 (v.s. CnEpSin), 11/4 (v.s. RSP), 8/6 (v.s. ESADE) and 7/9 (v.s. CELDE). The advantage of LGPDE becomes particularly pronounced in higher-dimensional scenarios. For 50-D cases, it achieves "+/–" metric of 23/2 against EJADE, 18/4 against SCSS-L-SHADE, and 13/5 against RSP. This superiority further escalates in 100-D tests, with metrics reaching 26/1 (v.s. EJADE), 22/2 (v.s. SCSS-L-SHADE) and 16/5 (v.s. RSP). These results collectively indicate that LGPDE maintains robust performance across dimensional scales, with particularly outstanding results in high-dimensional optimization. Furthermore, the type-specific functional analysis yields additional insights:

(a) Unimodal Functions: LGPDE achieves competitive performance with other algorithms on $F1$ across all dimensions and $F3$ in 30-D and 50-D cases. However, its performance on $F3$ in 100-D is suboptimal, where CnEpSin emerges as the top performer.

(b) Simple Multimodal Functions: LGPDE demonstrates consistently strong results, matching or exceeding the performance of competitors. Specifically, LGPDE achieves the best performance compared with other algorithms on 4 (30-D), 6 (50-D) and 5 (100-D) functions. Particularly noteworthy is its sustained excellent performance on $F5$ and $F7$–$F9$ across all dimensional scenarios.

(c) Hybrid Functions: LGPDE shows superior performance on 1 ($F15$, 30-D), 4 ($F13$, $F14$, $F18$ and $F19$, 50-D), and 2 ($F11$ and $F17$, 100-D) functions, with its most dominant performance observed in the 50-D cases.

(d) Composition Functions: For functions $F21$–$F30$, LGPDE achieves better results than competitors on 3 (30-D), 2 (50-D) and 2 (100-D) functions. Notably, LGPDE maintains optimal performance on $F21$ consistently across all dimensions.

Fig. 3 provides a detailed description of the convergence curves of LGPDE and the compared DE algorithms on eight selected representative functions in the run with the median error value. On 100-D $F6$, CELDE achieves the best result, for the remaining seven functions, LGPDE achieves either optimal or competitive performance, including

Table 9
Performance comparison of LGPDE with State-of-the-Art DEs on 100-D CEC2017 functions.

Problem	EJADE	SCSS-L-SHADE	DISH	CnEpSin	RSP	ESADE	CELDE	LGPDE
F1	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	3.60e−8 (6.01e−8) +	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)
F3	3.48e+3 (4.24e+3) +	1.04e−4 (1.32e−4) +	4.51e−5 (6.07e−5) +	0.00e+0 (0.00e+0) −	2.76e−7 (5.66e−7) −	1.22e−7 (1.27e−7) −	1.06e−7 (1.32e−7) −	1.49e−6 (2.86e−6)
F4	3.93e+1 (5.94e+1) −	1.95e+2 (1.56e+1) ≈	1.99e+2 (1.01e+1) ≈	1.99e+2 (8.35e+0) −	2.00e+2 (9.72e+0) −	1.98e+2 (1.44e+1) −	1.96e+2 (9.95e+0) −	2.04e+2 (9.85e+0)
F5	1.22e+2 (1.63e+1) +	2.84e+1 (4.73e+0) +	2.54e+1 (9.93e+0) +	5.49e+1 (7.51e+0) +	3.10e+1 (7.75e+0) +	2.67e+1 (4.39e+0) +	3.14e+1 (6.48e+0) +	1.52e+1 (4.11e+0)
F6	6.41e−2 (4.42e−2) +	1.56e−3 (1.56e−3) +	6.30e−6 (5.56e−6) −	6.08e−5 (1.90e−5) ≈	1.64e−5 (1.16e−5) −	1.90e−5 (9.74e−6) −	5.92e−7 (5.88e−7) −	6.66e−5 (3.11e−5)
F7	2.17e+2 (1.34e+1) +	1.31e+2 (3.13e+0) +	1.38e+2 (9.41e+0) +	1.62e+2 (5.47e+0) +	1.38e+2 (8.21e+0) +	1.28e+2 (2.94e+0) +	1.30e+2 (6.23e+0) +	1.18e+2 (2.28e+0)
F8	1.14e+2 (2.00e+1) +	2.74e+1 (3.73e+0) +	2.45e+1 (8.76e+0) +	5.36e+1 (5.92e+0) +	2.79e+1 (9.17e+0) +	2.69e+1 (3.46e+0) +	3.13e+1 (5.65e+0) +	1.51e+1 (4.04e+0)
F9	2.42e+1 (1.62e+1) +	1.38e−1 (2.28e−1) +	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	3.51e−3 (1.76e−2) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0) ≈	0.00e+0 (0.00e+0)
F10	1.01e+4 (1.34e+3) +	9.82e+3 (4.94e+2) +	1.11e+4 (6.25e+2) +	1.03e+4 (4.62e+2) +	1.08e+4 (7.56e+2) +	9.03e+3 (4.83e+2) +	9.42e+3 (6.34e+2) +	8.78e+3 (5.86e+2)
F11	4.82e+2 (3.59e+2) +	1.74e+2 (4.23e+1) +	6.09e+1 (3.01e+1) +	6.11e+1 (3.95e+1) +	8.17e+1 (3.40e+1) +	6.14e+1 (4.39e+1) +	3.37e+1 (3.42e+1) +	2.93e+1 (2.42e+1)
F12	3.70e+4 (2.50e+4) +	1.68e+4 (6.78e+3) +	1.19e+4 (4.62e+3) ≈	4.60e+3 (6.06e+2) −	1.12e+4 (4.28e+3) ≈	5.33e+3 (1.09e+3) −	6.45e+3 (2.55e+3) −	1.10e+4 (5.43e+3)
F13	4.98e+2 (6.91e+2) +	1.80e+2 (6.82e+1) +	1.16e+2 (3.48e+1) +	1.07e+2 (3.42e+1) +	1.35e+2 (4.13e+1) +	7.54e+1 (2.52e+1) +	3.71e+1 (1.35e+1) −	5.74e+1 (2.40e+1)
F14	3.35e+2 (2.16e+2) +	8.59e+1 (1.83e+1) +	3.81e+1 (3.88e+0) +	5.21e+1 (7.51e+0) +	4.42e+1 (5.30e+0) +	3.67e+1 (3.64e+0) +	3.19e+1 (3.83e+0) ≈	3.26e+1 (2.83e+0)
F15	4.26e+2 (4.02e+2) +	2.50e+2 (5.57e+1) +	1.15e+2 (3.63e+1) +	9.49e+1 (3.19e+1) +	1.27e+2 (3.69e+1) +	7.42e+1 (3.37e+1) +	4.34e+1 (3.02e+1) ≈	4.55e+1 (3.07e+1)
F16	1.96e+3 (4.91e+2) +	1.47e+3 (2.81e+2) +	1.71e+3 (3.36e+2) +	1.21e+3 (1.91e+2) ≈	1.46e+3 (3.37e+2) +	1.37e+3 (2.67e+2) ≈	1.41e+3 (2.79e+2) +	1.21e+3 (3.84e+2)
F17	1.62e+3 (4.57e+2) +	1.03e+3 (1.98e+2) +	1.31e+3 (2.88e+2) +	9.52e+2 (1.59e+2) ≈	1.15e+3 (2.90e+2) +	1.02e+3 (1.86e+2) +	1.00e+3 (2.11e+2) +	9.20e+2 (2.42e+2)
F18	4.60e+3 (4.48e+3) +	2.11e+2 (4.91e+1) +	1.07e+2 (2.54e+1) +	7.01e+1 (1.68e+1) +	1.65e+2 (3.55e+1) +	5.63e+1 (1.54e+1) ≈	4.69e+1 (1.78e+1) −	5.88e+1 (1.65e+1)
F19	1.74e+2 (6.40e+1) +	1.63e+2 (1.92e+1) +	5.86e+1 (8.04e+0) +	5.55e+1 (5.67e+0) +	6.49e+1 (1.07e+1) +	4.90e+1 (4.50e+0) +	3.63e+1 (4.24e+0) ≈	3.76e+1 (4.00e+0)
F20	1.65e+3 (4.50e+2) +	1.47e+3 (2.02e+2) +	1.55e+3 (3.25e+2) +	1.17e+3 (1.58e+2) ≈	1.37e+3 (2.51e+2) +	1.13e+3 (1.65e+2) ≈	1.15e+3 (1.55e+2) ≈	1.13e+3 (1.78e+2)
F21	3.35e+2 (1.64e+1) +	2.52e+2 (5.08e+0) +	2.46e+2 (7.38e+0) ≈	2.77e+2 (1.12e+1) +	2.51e+2 (7.90e+0) +	2.50e+2 (3.97e+0) +	2.55e+2 (5.92e+0) +	2.44e+2 (4.97e+0)
F22	1.12e+4 (1.35e+3) +	1.09e+4 (6.14e+2) +	1.17e+4 (7.40e+2) +	1.07e+4 (5.60e+2) +	1.12e+4 (7.07e+2) +	9.74e+3 (4.97e+2) +	1.00e+4 (6.48e+2) +	8.75e+3 (6.58e+2)
F23	6.26e+2 (1.90e+1) +	5.60e+2 (9.11e+0) −	5.64e+2 (1.06e+1) −	5.98e+2 (1.01e+1) +	5.67e+2 (1.01e+1) ≈	5.52e+2 (8.48e+0) −	5.54e+2 (7.92e+0) −	5.68e+2 (8.89e+0)
F24	9.89e+2 (2.48e+1) +	9.03e+2 (7.32e+0) −	8.94e+2 (6.78e+0) −	9.19e+2 (1.50e+1) +	9.00e+2 (7.67e+0) −	8.98e+2 (6.21e+0) −	8.97e+2 (6.41e+0) −	9.09e+2 (6.09e+0)
F25	7.54e+2 (4.77e+1) +	7.31e+2 (3.86e+1) ≈	7.07e+2 (4.28e+1) −	6.90e+2 (4.20e+1) −	7.20e+2 (4.97e+1) ≈	6.71e+2 (4.07e+1) −	7.00e+2 (4.69e+1) −	7.24e+2 (4.67e+1)
F26	4.21e+3 (2.59e+2) +	3.22e+3 (8.78e+1) ≈	3.11e+3 (8.13e+1) −	3.07e+3 (1.15e+2) −	3.15e+3 (9.45e+1) −	3.09e+3 (7.97e+1) −	3.08e+3 (7.40e+1) −	3.22e+3 (8.79e+1)
F27	6.52e+2 (2.48e+1) +	6.24e+2 (1.64e+1) +	5.76e+2 (1.43e+1) ≈	5.91e+2 (1.84e+1) +	5.81e+2 (1.94e+1) ≈	5.73e+2 (1.88e+1) −	5.80e+2 (1.65e+1) ≈	5.77e+2 (1.24e+1)
F28	5.20e+2 (5.26e+1) ≈	5.36e+2 (2.40e+1) +	5.26e+2 (2.24e+1) +	5.12e+2 (1.91e+1) −	5.24e+2 (2.70e+1) +	5.16e+2 (2.79e+1) ≈	5.24e+2 (2.32e+1) ≈	5.16e+2 (1.02e+1)
F29	1.66e+3 (3.75e+2) +	1.27e+3 (1.57e+2) +	1.25e+3 (1.68e+2) +	1.14e+3 (1.09e+2) ≈	1.22e+3 (1.83e+2) ≈	1.16e+3 (1.74e+2) ≈	1.16e+3 (1.42e+2) ≈	1.16e+3 (1.33e+2)
F30	2.50e+3 (2.14e+2) +	2.33e+3 (1.24e+2) ≈	2.33e+3 (1.30e+2) ≈	2.38e+3 (1.81e+2) ≈	2.34e+3 (1.61e+2) ≈	2.30e+3 (1.38e+2) −	2.29e+3 (1.73e+2) −	2.35e+3 (1.03e+2)
+/−/≈	26/1/2	22/2/5	18/5/6	15/6/8	16/5/8	12/10/7	9/11/9	

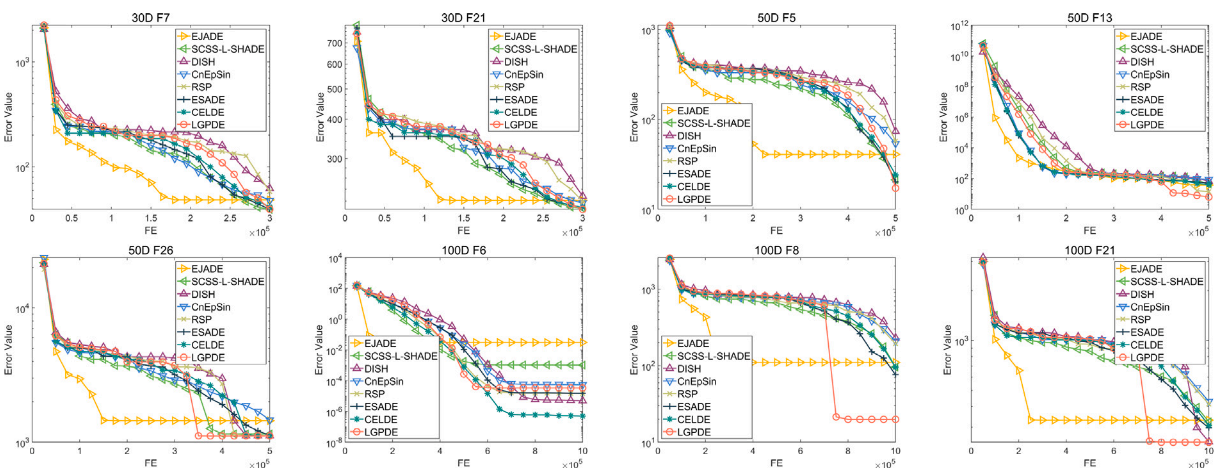

Fig. 3. Convergence curves of the compared DE algorithms on eight selected functions.

Table 10
Summary performance comparisons of LGPDE with State-of-the-Art DEs on 30-D, 50-D and 100-D CEC2017 functions according to single-problem Wilcoxon rank-sum test.

+/−/≈	30-D	50-D	100-D
EJADE	**17**/5/7	**23**/2/4	**26**/1/2
SCSS-L-SHADE	**12**/4/13	**18**/4/7	**22**/2/5
DISH	**16**/4/9	**16**/7/6	**18**/5/6
CnEpSin	**11**/6/12	**15**/4/10	**15**/6/8
RSP	**11**/4/14	**13**/5/11	**16**/5/8
ESADE	**8**/6/15	**12**/7/10	**12**/10/7
CELDE	**7**/9/13	**11**/8/10	**9**/11/9

30-D $F7$, 30-D $F21$, 50-D $F5$, 50-D $F13$, 50-D $F26$, 100-D $F8$ and 100-D $F21$. Notably, while LGPDE exhibits relatively slower initial convergence compared with some competitors, it demonstrates superior global search capability by effectively avoiding premature convergence to local optima, this trend is especially pronounced in 50-D $F13$ and 100-D $F8$ cases.

Furthermore, the overall performance ranking of the compared DE algorithms is depicted in Fig. 4. Following comprehensive examination, it is conspicuous that LGPDE achieves an impressive ranking of 3.37, demonstrating near-equivalent performance to the top-ranked CELDE (3.31), with only a marginal 0.06 difference separating the two leading algorithms. The subsequent rankings show ESADE following closely behind, with RSP, DISH, CnEpSin, SCSS-L-SHADE and EJADE completing the competitive field. In addition, to validate the statistical significance of performance differences, we perform multi-problem statistical analysis employing multiple testing procedures (Bonferroni, Holm, Hochberg and Hommel) through the KEEL software. As presented in Table 11, LGPDE performs comparably to CELDE with the adjusted p-values (p_{Bonf} is derived from Friedman's test with Bonferroni correction for family-wise error rate control) larger than 0.05, while CELDE performs significantly better than EJADE, SCSS-L-SHADE, DISH, CnEpSin and RSP.

Moreover, the statistical comparisons between LGPDE and the compared DE algorithms are further validated through the multi-problem Wilcoxon signed-rank test. As presented in Table 12, where $R+$ and $R-$ denote the rank sums for cases where LGPDE outperforms and underperforms its competitors respectively, LGPDE obtains higher $R+$ than $R-$ across all test cases, and performs comparably to both ESADE and CELDE ($p > 0.05$), while showing statistically superior results relative to all other algorithms in the comparison ($p < 0.05$). Both ESADE and CELDE demonstrate promising performance due to their strategies for combining the strengths of two competing algorithms, jSO and CnEpSin. ESADE employs an effective evolutionary scale adaptation mechanism that combines the strengths of jSO and CnEpSin, thereby

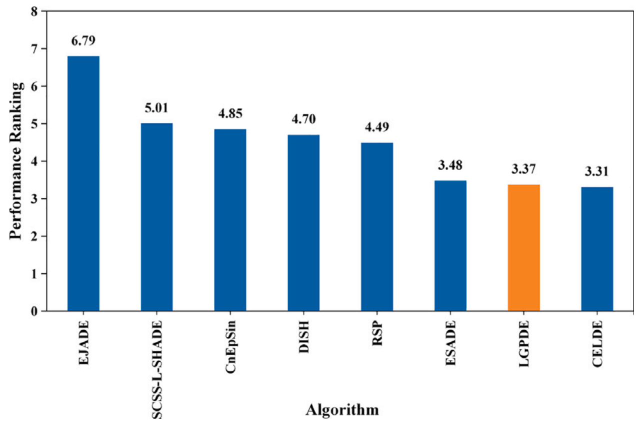

Fig. 4. Overall performance ranking of the compared DE algorithms.

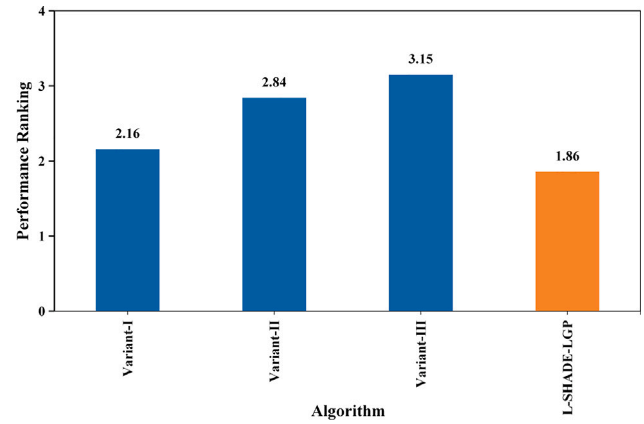

Fig. 5. Overall performance ranking of L-SHADE-LGP and its variant algorithms.

Table 11
Overall performance comparisons of the DE algorithms on CEC2017 functions according to Bonferroni, Holm, Hochberg and Hommel procedures with CELDE as the control method.

v.s.	Unadjusted p	p_{Bonf}	p_{Holm}	$p_{Hochberg}$	p_{Hommel}
EJADE	**<1e−6**	**<1e−6**	**<1e−6**	**<1e−6**	**<1e−6**
SCSS-L-SHADE	**0.000005**	**0.000035**	**0.000030**	**0.000030**	**0.000030**
DISH	**0.000311**	**0.002180**	**0.001246**	**0.001246**	**0.001246**
CnEpSin	**0.000034**	**0.000236**	**0.000168**	**0.000168**	**0.000168**
RSP	**0.001359**	**0.009512**	**0.004077**	**0.004077**	**0.004077**
ESADE	0.733524	5.134669	1.467048	0.889236	0.889236
LGPDE	0.889236	6.224649	1.467048	0.889236	0.889236

Table 12
Comparison results of LGPDE with the compared DE algorithms on CEC2017 functions according to multi-problem Wilcoxon signed-rank test.

v.s.	$R+$	$R-$	p-value
EJADE	3503.0	238.0	**<1e−6**
SCSS-L-SHADE	3080.0	661.0	**<1e−6**
DISH	2696.5	1131.5	**0.0009**
CnEpSin	2690.0	1051.0	**0.0004**
RSP	2874.5	953.5	**0.0000**
ESADE	2170.0	1571.0	0.1964
CELDE	1985.0	1756.0	0.6205

meeting the search requirements at different evolutionary stages. Meanwhile, CELDE utilizes a collective ensemble learning paradigm to integrate the advantages of jSO and CnEpSin, within a single individual for offspring generation.

4.3. Effectiveness of the components in LGP mechanism

To explicitly assess the contribution of each component in LGP mechanism, we develop the following three distinct L-SHADE-LGP variants:

Variant-I: This variant is developed to evaluate the efficacy of the dual historical memory strategy (The upper part in Fig. 1). The generation of local and global historical record follows an opposing approach.

Variant-II: This variant is designed to assess the effectiveness of the parameter adaptation strategy (The lower part in Fig. 1). M_{F,r_i} and M_{CR,r_i} are exclusively selected from local historical memory M_F^L and M_{CR}^L, respectively.

Variant-III: This variant is also designed to assess the effectiveness of the parameter adaptation strategy. Both M_{F,r_i} and M_{CR,r_i} are drawn solely from global historical memory M_F^G and M_{CR}^G, respectively.

Aside from the aforementioned differences, all other experimental settings remain identical to those in L-SHADE-LGP. Fig. 5 presents the overall performance ranking of L-SHADE-LGP and its three variant algorithms. As evident from the results, L-SHADE-LGP achieves the highest performance ranking with an outstanding score of 1.86, followed by Variant-I (2.16), Variant-II (2.84) and Variant-III (3.15). The performance advantage of L-SHADE-LGP over Variant-I validates the efficacy of the dual historical memory strategy. Furthermore, when comparing L-SHADE-LGP with Variant-II and III separately, the effectiveness of the novel parameter adaptation strategy becomes clearly apparent.

Additionally, Fig. 6 illustrates the convergence curves of L-SHADE-LGP and its variant algorithms on four selected functions, 30-D $F15$, 50-D $F4$, 50-D $F26$ and 100-D $F13$. The results demonstrate that L-SHADE-LGP achieves significantly faster convergence and higher solution accuracy compared with the three variants.

Table 13
Comparison results of L-SHADE-LGP with different threshold value settings for t_1 on 50-D CEC2017 functions according to single-problem Wilcoxon rank-sum test.

t_1	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
+	**21**	**18**	**14**	**12**	**4**	/	**8**	**12**	**17**	**21**	**19**
−	2	2	4	4	2	/	0	1	1	1	2
≈	6	9	11	13	23	/	21	16	11	7	8

Table 14
Comparison results of L-SHADE-LGP with different threshold value settings for t_2 on 50-D CEC2017 functions according to single-problem Wilcoxon rank-sum test.

t_2	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
+	**25**	**24**	**23**	**23**	**22**	**19**	**8**	/	**1**	**18**
−	1	2	2	1	2	2	2	/	1	1
≈	3	3	4	5	5	8	19	/	27	10

4.4. Parameter sensitivity analysis

The threshold values t_1 and t_2 are critical to the parameter adaptation strategy and the dual historical memory strategy, respectively. To analyze the performance sensitivity to t_1 and t_2, this subsection examines ten discrete values of t_1 (ranging from 1.0 to 2.0 in 0.1 increments) and nine discrete values of t_2 (ranging from 0.1 to 1.0 in 0.1 increments) based on algorithm L-SHADE-LGP. For the 50-D CEC2017 benchmark suite, we evaluate L-SHADE-LGP's performance by comparing the results under each threshold setting with the standard parameter configuration where $t_1 = 1.5$ and $t_2 = 0.8$ using the Wilcoxon rank-sum test, the experimental results are illustrated in Tables 13 and 14, respectively.

The analysis reveals that the algorithm's performance significantly degrades when t_1 deviates substantially from its standard value (1.5), particularly at extreme settings (1.0, 1.1, 1.9 and 2.0). Similarly, lower

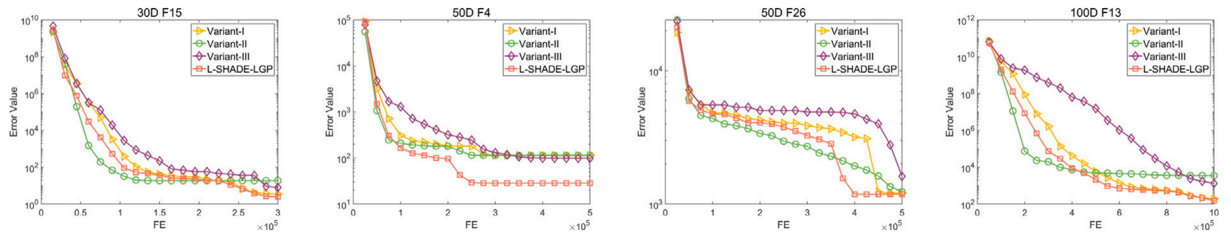

Fig. 6. Convergence curves of L-SHADE-LGP and its variant algorithms on four selected functions.

Table 15
Performance ranking and p_{Bonf} of different settings for t_1 according to Friedman's test with Bonferroni correction.

t_1	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
Ranking	8.02	7.09	5.81	5.19	4.17	3.71*	4.84	5.64	6.98	7.64	6.91
p_{Bonf}	**0.0000**	**0.0010**	0.1573	0.8868	5.9302	/	1.9139	0.2662	**0.0017**	**0.0001**	**0.0023**

Table 16
Performance ranking and p_{Bonf} of different settings for t_2 according to Friedman's test with Bonferroni correction.

t_2	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Ranking	8.71	8.02	7.57	6.53	5.74	4.50	3.09	3.02*	3.05	4.78
p_{Bonf}	**0.0000**	**0.0000**	**0.0000**	**0.0001**	**0.0055**	0.5598	8.3779	/	8.6887	0.2428

values of t_2 (0.1–0.5) demonstrate notably worse performance compared with the standard setting (0.8). This occurs because improper settings of t_1 and t_2 disrupt the balance between local exploitation and global exploration, causing either premature convergence to local optima or diminished exploitation capability in promising regions.

To further evaluate the performance of different threshold value configurations, we conduct Friedman's test with Bonferroni correction, analyzing both performance ranking of each setting and statistical significance relative to the top-ranked setting. The comprehensive results of this analysis for t_1 and t_2 are presented in Tables 15 and 16 respectively, where the top-ranked setting is marked with $*$, bold adjusted p_{Bonf} indicates that the null hypothesis is rejected with 5% significance level. The analysis demonstrates that the standard configuration (1.5 for t_1 and 0.8 for t_2) yields optimal performance, attaining the highest ranking in their respective categories.

4.5. Time complexity analysis

To assess whether the LGP mechanism increases the algorithm's computational complexity, we evaluate the complexity of L-SHADE and L-SHADE-LGP using the index $(\hat{T}2-T1)/T0$ following the methodology established in [56]. Where $T0$ denotes the execution time of a reference test program described below:

$$
\begin{aligned}
&\textbf{for } i = 1 : 1000000\\
&\quad x = 0.55 + (double)i; x = x + x; x = x/2; x = x * x;\\
&\quad x = \sqrt{x}; x = log(x); x = exp(x); x = x/(x+2);\\
&\textbf{end}
\end{aligned}
$$

$T1$ represents the time required for 200,000 evaluations of 30-D $F18$ optimization problem; and $\hat{T}2$ is the average runtime of the algorithm over five independent optimizations under the same condition as $T1$. $(\hat{T}2 - T1)/T0$ serves as a complexity indicator: smaller values correspond to lower algorithmic complexity. The experiments are conducted using MATLAB R2024a on a PC equipped with an Intel(R) Core(TM) i7-14700 @2.10 GHz CPU and Windows 11 24H2 OS.

As illustrated in Table 17, L-SHADE-LGP maintains comparable computational complexity to L-SHADE, though with a marginally higher time complexity, demonstrating that the incorporation of the LGP mechanism enhances the performance of the original algorithm without significantly increasing its computational overhead.

When dealing with different optimization problems, assuming the evaluation time is m times that of 30-D $F18$ (i.e., $m \cdot T1$), the runtime of the algorithm under the same condition would be $\hat{T}2 - T1 + m \cdot T1$. We define the runtime ratio between L-SHADE-LGP and L-SHADE as n (i.e., $n = T_{L-SHADE-LGP}/T_{L-SHADE}$). When $m = 10$, n equals 1.0128; when $m = 100$, n becomes 1.0014. The closer n is to 1, the lower the additional computational overhead introduced by LGP. Therefore, when optimizing problems with longer evaluation times, LGP does not incur significant extra computational costs.

Table 17
Comparison of time complexity between L-SHADE and L-SHADE-LGP ($T0$, $T1$ and $\hat{T}2$ in seconds).

Algorithm	$T0$	$T1$	$\hat{T}2$	$(\hat{T}2 - T1)/T0$
L-SHADE	0.0482	0.3117	0.6324	6.6535
L-SHADE-LGP			0.6763	7.5643

Table 18
Summary of LGPDE's performance compared with State-of-the-Art DEs on CEC2011 problems.

	EJADE	SCSS-L-SHADE	DISH	CnEpSin	RSP	ESADE	CELDE
+	**11**	**8**	**9**	**11**	**9**	**6**	**11**
−	2	4	2	2	2	4	2
≈	9	10	11	9	11	12	9

4.6. Real-world applications

To evaluate the performance of LGPDE on real-world optimization problems, we conduct experiments using the 22 problems ($P1$–$P22$) from the CEC2011 test suite [66]. Each problem is evaluated with a maximum of 150,000 function evaluations, and the statistical results (mean, standard deviation, best and worst objective function values) are reported over 25 independent runs. The comparative analysis between LGPDE and the seven State-of-the-Art competitors is presented in Table 18, with detailed experimental results for $P1$–$P13$ provided in Table 19 and $P14$–$P22$ provided in Table 20.

The experimental results demonstrate LGPDE's remarkable competitiveness. As shown in Table 18, LGPDE outperforms its competitors on the majority of test problems, achieving superior results on 11, 8, 9, 11, 9, 6 and 11 problems against EJADE, SCSS-L-SHADE, DISH, CnEpSin, RSP, ESADE and CELDE respectively, while exhibiting inferior performance on only 2, 4, 2, 2, 2, 4 and 2 problems, highlighting its stability and generalization capability.

Table 19
Performance comparison of LGPDE with State-of-the-Art DEs on CEC2011 problems ($P1$–$P13$).

Problem		Algorithm							
		EJADE	SCSS-L-SHADE	DISH	CnEpSin	RSP	ESADE	CELDE	LGPDE
$P1$	Mean	**0.0000e+00**	1.9191e−10	**0.0000e+00**	4.4829e−01	1.2201e+00	4.0670e−01	**0.0000e+00**	1.8475e−06
	STD	**0.0000e+00**	9.5931e−10	**0.0000e+00**	2.2415e+00	3.3722e+00	2.0335e+00	**0.0000e+00**	8.9690e−06
	Best	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	Worst	0.0000e+00	4.7966e−09	0.0000e+00	1.1207e+01	1.0167e+01	1.0167e+01	0.0000e+00	4.4880e−05
	+/−/≈	≈	≈	≈	≈	≈	≈	≈	/
	p-value	1.6151e−01	1.6151e−01	1.6151e−01	6.0413e−01	5.7616e−01	6.0413e−01	1.6151e−01	/
$P2$	Mean	−2.3067e+01	**−2.6391e+01**	−2.3484e+01	−2.6008e+01	−2.4459e+01	−2.6308e+01	−2.6089e+01	−2.6262e+01
	STD	1.7174e+00	**7.1654e−01**	1.6179e+00	5.5630e−01	9.2648e−01	4.3412e−01	4.3512e−01	4.7971e−01
	Best	−2.7426e+01	−2.8121e+01	−2.6342e+01	−2.7321e+01	−2.6747e+01	−2.7308e+01	−2.7004e+01	−2.7265e+01
	Worst	−2.0738e+01	−2.5118e+01	−2.0295e+01	−2.5046e+01	−2.2270e+01	−2.5330e+01	−2.5230e+01	−2.5161e+01
	+/−/≈	+	≈	+	≈	+	≈	≈	/
	p-value	2.1801e−04	2.7646e−01	3.6663e−04	6.8977e−01	2.7468e−03	8.6526e−01	8.2737e−02	/
$P3$	Mean	**1.1515e−05**	1.1515e−05	1.1515e−05	1.1515e−05	**1.1515e−05**	**1.1515e−05**	**1.1515e−05**	**1.1515e−05**
	STD	**1.2633e−19**	3.1438e−19	1.0393e−19	2.8267e−19	**1.5170e−19**	**1.5170e−19**	**1.5170e−19**	**1.5170e−19**
	Best	1.1515e−05	1.1515e−05	1.1515e−05	1.1515e−05	1.1515e−05	1.1515e−05	1.1515e−05	1.1515e−05
	Worst	1.1515e−05	1.1515e−05	1.1515e−05	1.1515e−05	1.1515e−05	1.1515e−05	1.1515e−05	1.1515e−05
	+/−/≈	≈	+	+	+	≈	≈	≈	/
	p-value	8.0715e−02	4.8013e−05	3.1849e−03	2.5657e−08	1.0000e+00	1.0000e+00	1.0000e+00	/
$P4$	Mean	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**
	STD	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**
	Best	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	Worst	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	+/−/≈	≈	≈	≈	≈	≈	≈	≈	/
	p-value	1.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00	/
$P5$	Mean	−3.4713e+01	−3.6070e+01	−3.4407e+01	**−3.6295e+01**	−3.5061e+01	−3.5837e+01	−3.5918e+01	−3.5921e+01
	STD	8.6444e−01	7.5113e−01	1.1724e+00	**6.2036e−01**	8.8250e−01	8.7805e−01	8.3867e−01	9.9846e−01
	Best	−3.6788e+01	−3.6838e+01	−3.5872e+01	−3.6801e+01	−3.6688e+01	−3.6869e+01	−3.6845e+01	−3.6831e+01
	Worst	−3.3769e+01	−3.4554e+01	−3.1366e+01	−3.4363e+01	−3.3965e+01	−3.4434e+01	−3.4342e+01	−3.4295e+01
	+/−/≈	+	≈	+	≈	+	≈	≈	/
	p-value	8.7263e−04	9.3666e−02	8.6433e−04	4.1379e−01	2.4146e−04	9.7365e−02	9.7635e−01	/
$P6$	Mean	−2.8346e+01	**−2.9163e+01**	−2.7971e+01	−2.9079e+01	−2.8458e+01	−2.9091e+01	−2.9159e+01	−2.9091e+01
	STD	7.8149e−01	**1.9071e−03**	1.6883e+00	3.4520e−01	1.2979e+00	3.4655e−01	6.2730e−03	3.4686e−01
	Best	−2.9166e+01	−2.9166e+01	−2.9157e+01	−2.9163e+01	−2.9152e+01	−2.9165e+01	−2.9166e+01	−2.9165e+01
	Worst	−2.6270e+01	−2.9158e+01	−2.2923e+01	−2.7424e+01	−2.3001e+01	−2.7429e+01	−2.9137e+01	−2.7426e+01
	+/−/≈	+	≈	+	≈	+	≈	≈	/
	p-value	6.2782e−03	3.2783e−01	7.3266e−05	1.5368e−01	2.3652e−04	6.5275e−01	1.6353e−01	/
$P7$	Mean	7.4849e−01	1.1490e+00	1.1525e+00	9.1391e−01	1.1263e+00	1.0434e+00	**5.0000e−01**	1.1609e+00
	STD	2.2133e−01	8.1883e−02	1.0853e−01	2.4239e−01	1.2727e−01	8.5495e−02	**0.0000e+00**	8.8838e−02
	Best	5.0000e−01	9.6419e−01	9.0392e−01	5.0000e−01	8.7848e−01	8.8524e−01	5.0000e−01	9.8972e−01
	Worst	1.3661e+00	1.3069e+00	1.3546e+00	1.1398e+00	1.3800e+00	1.2013e+00	5.0000e−01	1.2652e+00
	+/−/≈	−	≈	≈	−	≈	−	−	/
	p-value	4.1024e−07	3.3198e−01	4.6093e−01	3.1529e−06	4.3768e−01	7.5523e−05	9.7285e−11	/
$P8$	Mean	**2.2000e+02**	**2.2000e+02**	**2.2000e+02**	**2.2000e+02**	**2.2000e+02**	**2.2000e+02**	**2.2000e+02**	**2.2000e+02**
	STD	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**	**0.0000e+00**
	Best	2.2000e+02	2.2000e+02	2.2000e+02	2.2000e+02	2.2000e+02	2.2000e+02	2.2000e+02	2.2000e+02
	Worst	2.2000e+02	2.2000e+02	2.2000e+02	2.2000e+02	2.2000e+02	2.2000e+02	2.2000e+02	2.2000e+02
	+/−/≈	≈	≈	≈	≈	≈	≈	≈	/
	p-value	1.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00	/
$P9$	Mean	2.0761e+03	2.5143e+03	2.6033e+03	3.3184e+03	2.1349e+03	2.4395e+03	6.7843e+03	**1.4629e+03**
	STD	5.4504e+02	5.4113e+02	6.7649e+02	2.2965e+03	4.9077e+02	7.9101e+02	2.7959e+03	**3.3466e+02**
	Best	1.1300e+03	1.5474e+03	1.3289e+03	1.4108e+03	1.3043e+03	1.1596e+03	3.2115e+03	8.8439e+02
	Worst	3.7651e+03	3.5716e+03	4.7637e+03	1.3630e+04	3.1132e+03	4.1249e+03	1.5775e+04	1.9932e+03
	+/−/≈	+	+	+	+	+	+	+	/
	p-value	1.8026e−05	5.8547e−09	3.2057e−08	1.4648e−08	2.9240e−06	5.6183e−06	1.4157e−09	/
$P10$	Mean	−2.1652e+01	−2.1660e+01	**−2.1668e+01**	−2.1638e+01	−2.1662e+01	−2.1644e+01	−2.1638e+01	−2.1638e+01
	STD	3.8961e−02	5.4956e−02	**6.5735e−02**	8.6212e−02	7.5668e−02	3.7442e−04	8.6222e−02	8.6212e−02
	Best	−2.1839e+01	−2.1843e+01	−2.1843e+01	−2.1843e+01	−2.1843e+01	−2.1645e+01	−2.1843e+01	−2.1843e+01
	Worst	−2.1644e+01	−2.1644e+01	−2.1644e+01	−2.1439e+01	−2.1480e+01	−2.1644e+01	−2.1439e+01	−2.1439e+01
	+/−/≈	≈	≈	≈	≈	≈	≈	≈	/
	p-value	5.8767e−01	2.1106e−01	3.4212e−01	5.3879e−01	6.6122e−01	8.5112e−01	9.3761e−01	/
$P11$	Mean	5.2000e+04	**5.1866e+04**	5.2196e+04	5.2301e+04	5.2027e+04	5.1978e+04	5.2910e+04	5.2052e+04
	STD	4.6235e+02	**4.6244e+02**	5.2645e+02	5.1957e+02	5.1640e+02	7.1066e+02	5.6516e+02	4.2872e+02
	Best	5.0811e+04	5.0918e+04	5.1327e+04	5.1106e+04	5.0826e+04	5.0663e+04	5.1878e+04	5.1257e+04
	Worst	5.2765e+04	5.2941e+04	5.3486e+04	5.3005e+04	5.3215e+04	5.3409e+04	5.4198e+04	5.3027e+04
	+/−/≈	≈	≈	≈	≈	≈	≈	+	/
	p-value	6.8367e−01	1.5106e−01	4.4922e−01	5.7239e−02	7.7102e−01	7.5622e−01	1.9973e−06	/
$P12$	Mean	1.0723e+06	**1.0708e+06**	1.0722e+06	1.0751e+06	1.0719e+06	1.0726e+06	1.0764e+06	1.0719e+06
	STD	1.6765e+03	**1.1965e+03**	1.7122e+03	4.2330e+03	8.7064e+02	1.3628e+03	1.8905e+03	8.0738e+02
	Best	1.0679e+06	1.0692e+06	1.0689e+06	1.0712e+06	1.0701e+06	1.0699e+06	1.0730e+06	1.0705e+06
	Worst	1.0750e+06	1.0734e+06	1.0757e+06	1.0895e+06	1.0740e+06	1.0751e+06	1.0798e+06	1.0733e+06
	+/−/≈	≈	−	≈	+	≈	+	+	/
	p-value	9.1402e−02	8.4600e−04	3.2240e−01	1.3564e−06	9.5358e−01	2.5659e−02	1.8002e−09	/
$P13$	Mean	1.5445e+04	1.5444e+04	1.5444e+04	1.5445e+04	1.5444e+04	1.5445e+04	1.5445e+04	**1.5444e+04**
	STD	6.2513e+00	8.7932e−01	1.1119e−06	5.8989e−01	9.9854e−07	6.4682e+00	6.4682e+00	**8.6342e+00**
	Best	1.5444e+04	1.5444e+04	1.5444e+04	1.5444e+04	1.5444e+04	1.5444e+04	1.5444e+04	1.5444e+04
	Worst	1.5475e+04	1.5448e+04	1.5444e+04	1.5446e+04	1.5444e+04	1.5476e+04	1.5476e+04	1.5475e+04
	+/−/≈	+	+	≈	+	≈	≈	+	/
	p-value	3.4438e−02	7.8561e−03	9.8452e−01	3.8987e−05	9.1402e−02	1.1160e−01	9.0701e−04	/

5. Conclusion

This paper proposes a novel Local and Global Parameter Adaptation (LGP) mechanism to mitigate the persistent deficiencies of maintaining exploitation-exploration balance in existing parameter adaptation methods. The LGP mechanism introduces a dual historical memory strategy that classifies successful F and CR into either local or global historical record based on the Euclidean distance between parent and offspring vectors, the corresponding local and global historical memory are updated at the end of each generation. Meanwhile, adaptive selection of M_{F,r_i} and M_{CR,r_i} from appropriate historical memory is achieved by the proposed parameter adaptation strategy. To validate the generalizability and effectiveness of LGP, we first integrate it with four representative DE variants for performance evaluation, then develop LGPDE by incorporating LGP into the ODF-SCSS-jSO algorithm and conduct comprehensive comparisons with seven State-of-the-Art DE algorithms on the CEC2011 and CEC2017 benchmark suites. Our experimental results reveal that (1) LGP consistently enhances algorithmic performance across different architectures, confirming its robustness and versatility; (2) LGPDE exhibits remarkable performance, particularly excelling in high-dimensional problems; and (3) LGP achieves significantly improved convergence rate and solution precision without substantial computational overhead.

Table 20
Performance comparison of LGPDE with State-of-the-Art DEs on CEC2011 problems ($P14$–$P22$).

Problem		Algorithm							
		EJADE	SCSS-L-SHADE	DISH	CnEpSin	RSP	ESADE	CELDE	LGPDE
$P14$	Mean	1.8236e+04	1.8092e+04	1.8096e+04	1.8139e+04	1.8097e+04	**1.8086e+04**	1.8105e+04	1.8095e+04
	STD	2.5876e+02	3.1125e+01	2.3310e+01	3.2980e+01	2.5875e+01	**1.5787e+01**	4.0293e+01	3.6322e+01
	Best	1.8019e+04	1.8075e+04	1.8076e+04	1.8075e+04	1.8075e+04	1.8075e+04	1.8035e+04	1.8019e+04
	Worst	1.8882e+04	1.8184e+04	1.8166e+04	1.8201e+04	1.8186e+04	1.8128e+04	1.8194e+04	1.8187e+04
	+/ − / ≈	≈	≈	≈	+	≈	≈	≈	/
	p-value	8.2341e−01	6.2502e−02	2.6043e−01	2.4528e−04	6.6947e−01	6.5283e−02	7.4244e−02	/
$P15$	Mean	3.2827e+04	**3.2740e+04**	3.2741e+04	3.2883e+04	3.2741e+04	3.2741e+04	3.2744e+04	3.2742e+04
	STD	5.1354e+01	**2.9701e−01**	4.5431e−01	1.6046e+01	3.8658e−01	2.6027e−01	4.0816e+00	1.4983e+00
	Best	3.2743e+04	3.2740e+04	3.2740e+04	3.2850e+04	3.2740e+04	3.2740e+04	3.2741e+04	3.2740e+04
	Worst	3.2928e+04	3.2742e+04	3.2742e+04	3.2910e+04	3.2742e+04	3.2741e+04	3.2761e+04	3.2746e+04
	+/ − / ≈	+	−	−	+	−	−	+	/
	p-value	6.5743e−09	6.1473e−07	9.7208e−04	1.4157e−09	4.6094e−05	2.5489e−05	4.0837e−03	/
$P16$	Mean	1.3378e+05	1.2355e+05	1.2370e+05	1.2650e+05	1.2396e+05	**1.2351e+05**	1.2443e+05	1.2399e+05
	STD	5.3323e+03	4.6926e+02	4.2244e+02	2.5437e+03	6.1044e+02	**4.1520e+02**	6.7958e+02	5.2170e+02
	Best	1.2466e+05	1.2268e+05	1.2303e+05	1.2309e+05	1.2267e+05	1.2280e+05	1.2273e+05	1.2308e+05
	Worst	1.4282e+05	1.2445e+05	1.2459e+05	1.3105e+05	1.2504e+05	1.2431e+05	1.2526e+05	1.2504e+05
	+/ − / ≈	+	−	≈	+	≈	−	+	/
	p-value	2.2857e−09	3.3915e−03	5.9826e−02	3.8768e−06	9.0732e−01	2.0352e−03	5.2059e−03	/
$P17$	Mean	2.0042e+06	1.8629e+06	1.8614e+06	1.8691e+06	1.8587e+06	1.8468e+06	1.8858e+06	**1.8318e+06**
	STD	1.0677e+05	1.3357e+04	1.4285e+04	1.1707e+04	1.0569e+04	9.8207e+03	1.1235e+04	**1.0629e+04**
	Best	1.8596e+06	1.8263e+06	1.8399e+06	1.8467e+06	1.8366e+06	1.8238e+06	1.8691e+06	1.8150e+06
	Worst	2.2622e+06	1.8934e+06	1.8951e+06	1.8952e+06	1.8745e+06	1.8744e+06	1.9060e+06	1.8608e+06
	+/ − / ≈	+	+	+	+	+	+	+	/
	p-value	1.5967e−09	3.5800e−08	2.5677e−08	3.6690e−09	2.5677e−08	1.0605e−05	1.4157e−09	/
$P18$	Mean	9.6263e+05	9.3270e+05	9.3375e+05	9.3683e+05	9.3238e+05	9.3390e+05	9.3745e+05	**9.3101e+05**
	STD	6.6808e+04	2.0414e+03	1.6935e+03	1.8063e+03	1.7508e+03	1.5476e+03	2.2533e+03	**1.2422e+03**
	Best	9.3298e+05	9.2942e+05	9.3101e+05	9.3327e+05	9.2884e+05	9.3077e+05	9.3342e+05	9.2882e+05
	Worst	1.2442e+06	9.3730e+05	9.3760e+05	9.4051e+05	9.3582e+05	9.3795e+05	9.4207e+05	9.3308e+05
	+/ − / ≈	+	+	+	+	+	+	+	/
	p-value	1.8002e−09	1.9063e−03	4.5414e−07	1.4157e−09	4.6140e−03	8.5468e−08	1.4157e−09	/
$P19$	Mean	1.0751e+06	9.4067e+05	9.4114e+05	9.4217e+05	9.4002e+05	9.4080e+05	9.4297e+05	**9.3881e+05**
	STD	2.9837e+05	1.3397e+03	9.3777e+02	1.4296e+03	1.1190e+03	1.4904e+03	1.8703e+03	**1.0350e+03**
	Best	9.4009e+05	9.3861e+05	9.3964e+05	9.4018e+05	9.3776e+05	9.3820e+05	9.3801e+05	9.3645e+05
	Worst	2.0320e+06	9.4361e+05	9.4327e+05	9.4717e+05	9.4211e+05	9.4390e+05	9.4630e+05	9.4070e+05
	+/ − / ≈	+	+	+	+	+	+	+	/
	p-value	2.0288e−09	6.7488e−06	2.5677e−08	5.2120e−09	3.8418e−04	6.7488e−06	2.5677e−08	/
$P20$	Mean	9.9324e+05	9.3285e+05	9.3332e+05	9.3692e+05	9.3246e+05	9.3426e+05	9.3858e+05	**9.3078e+05**
	STD	1.2536e+05	1.1780e+03	1.4014e+03	1.4231e+03	1.3586e+03	1.3760e+03	2.0409e+03	**1.2189e+03**
	Best	9.3421e+05	9.3078e+05	9.3140e+05	9.3368e+05	9.3034e+05	9.3152e+05	9.3477e+05	9.2911e+05
	Worst	1.3728e+06	9.3478e+05	9.3577e+05	9.3954e+05	9.3547e+05	9.3652e+05	9.4243e+05	9.3351e+05
	+/ − / ≈	+	+	+	+	+	+	+	/
	p-value	1.4157e−09	2.6597e−06	4.1024e−07	1.4157e−09	7.5523e−05	1.4648e−08	1.4157e−09	/
$P21$	Mean	1.5001e+01	1.6456e+01	1.5649e+01	**1.4379e+01**	1.5974e+01	1.5755e+01	1.6003e+01	1.5283e+01
	STD	9.3963e−01	5.8186e−01	7.5733e−01	**1.7699e+00**	6.2875e−01	7.1195e−01	9.4226e−01	8.3350e−01
	Best	1.2954e+01	1.5380e+01	1.3853e+01	1.0395e+01	1.3629e+01	1.4292e+01	1.3415e+01	1.3277e+01
	Worst	1.6333e+01	1.8030e+01	1.6643e+01	1.6842e+01	1.6994e+01	1.7109e+01	1.8016e+01	1.6317e+01
	+/ − / ≈	≈	+	≈	≈	+	≈	+	/
	p-value	3.7211e−01	6.7951e−07	6.8172e−02	8.7738e−02	3.0746e−04	6.5289e−02	6.2228e−03	/
$P22$	Mean	1.5365e+01	**1.3546e+01**	1.5415e+01	1.3721e+01	1.5330e+01	1.3689e+01	1.3631e+01	1.7350e+01
	STD	2.3023e+00	**2.3505e+00**	1.5858e+00	1.3510e+00	1.6531e+00	1.4183e+00	1.4015e+00	2.0904e+00
	Best	9.7890e+00	8.6167e+00	1.2755e+01	1.1832e+01	1.2755e+01	1.1832e+01	1.1832e+01	1.2755e+01
	Worst	1.8940e+01	1.7009e+01	1.8146e+01	1.7486e+01	1.8603e+01	1.7486e+01	1.7486e+01	2.1243e+01
	+/ − / ≈	−	−	−	−	−	−	−	/
	p-value	1.9063e−03	5.6183e−06	7.3522e−04	6.1473e−07	2.2732e−04	9.1194e−07	5.0097e−07	/

This research establishes a novel paradigm for historical parameter utilization through exploitation-exploration balance, significantly advancing the performance of DE. Future investigations should explore more effective parameter classification schemes through multidimensional characteristic analysis, and systematically integrate complementary strategies to enhance solution accuracy and stability.

CRediT authorship contribution statement

Xiao Lin Jin: Writing – original draft, Validation, Software, Methodology. **Sheng Xin Zhang:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Li Ming Zheng:** Supervision, Funding acquisition. **Shao Yong Zheng:** Supervision, Funding acquisition.

Declaration of competing 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.

Acknowledgments

The work described in this paper was supported by the National Natural Science Foundation of China under Grant 62201227 and Grant 62071503, the National Special Project Number for International Cooperation under Grant 2015DFR11050, the Applied Science and Technology Research and Development Special Fund Project of Guangdong Province under Grant 2016B010126004 and the Guangzhou Basic and Applied Basic Research under Grant SL2022A04J01366.

Data availability

Data will be made available on request.

References

[1] A.E. Eiben, J. Smith, From evolutionary computation to the evolution of things, Nature 521 (7553) (2015) 476–482.
[2] S. Das, S.S. Mullick, P.N. Suganthan, Recent advances in differential evolution–an updated survey, Swarm Evol. Comput. 27 (2016) 1–30.
[3] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997) 341–359.
[4] S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2010) 4–31.
[5] R.D. Al-Dabbagh, F. Neri, N. Idris, M.S. Baba, Algorithmic design issues in adaptive differential evolution schemes: Review and taxonomy, Swarm Evol. Comput. 43 (2018) 284–311.
[6] K.R. Opara, J. Arabas, Differential evolution: A survey of theoretical analyses, Swarm Evol. Comput. 44 (2019) 546–558.
[7] Q. Yang, P. Liu, J. Zhang, N. Dong, Combined heat and power economic dispatch using an adaptive cuckoo search with differential evolution mutation, Appl. Energy 307 (2022) 118057.
[8] Z. Liao, S. Li, Solving nonlinear equations systems with an enhanced reinforcement learning based differential evolution, Complex Syst. Model. Simul. 2 (1) (2022) 78–95.
[9] Y. Xue, Y. Tong, F. Neri, An ensemble of differential evolution and Adam for training feed-forward neural networks, Inf. Sci. 608 (2022) 453–471.
[10] C. Rajesh, S. Kumar, An evolutionary block based network for medical image denoising using differential evolution, Appl. Soft Comput. 121 (2022) 108776.
[11] R. Tanabe, A.S. Fukunaga, Reviewing and benchmarking parameter control methods in differential evolution, IEEE Trans. Cybern. 50 (2020) 1170–1184.
[12] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (1) (2011) 55–66.
[13] R. Mallipeddi, P.N. Suganthan, Q.-K. Pan, M.F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2) (2011) 1679–1696.
[14] Y. Wang, H.-X. Li, T. Huang, L. Li, Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, Appl. Soft Comput. 18 (2014) 232–247.
[15] Z. Song, L. Sun, N. Matsveichuk, Y. Sotskov, Diversity enhancement-based differential evolution with a novel perturbation strategy, Swarm Evol. Comput. 92 (2025) 101822.
[16] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2008) 398–417.
[17] J. Zhang, A.C. Sanderson, JADE: Adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.
[18] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, in: Proc. IEEE Congr. Evol. Comput., IEEE, 2013, pp. 71–78.
[19] 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.
[20] Z. Meng, C. Yang, Hip-DE: Historical population based mutation strategy in differential evolution with parameter adaptive mechanism, Inf. Sci. 562 (2021) 44–77.
[21] V. Stanovov, S. Akhmedova, E. Semenkin, Biased parameter adaptation in differential evolution, Inf. Sci. 566 (2021) 215–238.
[22] Q. Xu, Z. Meng, Differential evolution with multi-stage parameter adaptation and diversity enhancement mechanism for numerical optimization, Swarm Evol. Comput. 92 (2025) 101829.
[23] J. Brest, M.S. Maučec, B. Bošković, Single objective real-parameter optimization: Algorithm jSO, in: Proc. IEEE Congr. Evol. Comput., IEEE, 2017, pp. 1311–1318.
[24] N.H. Awad, M.Z. Ali, P.N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving CEC2017 benchmark problems, in: Proc. IEEE Congr. Evol. Comput., 2017, pp. 372–379.
[25] V. Stanovov, E. Semenkin, Success rate-based adaptive differential evolution L-SRTDE for CEC 2024 competition, in: Proc. IEEE Congr. Evol. Comput., IEEE, 2024, pp. 1–8.
[26] Z. Xu, G. Han, L. Liu, M. Martinez-Garcia, Z. Wang, Multi-energy scheduling of an industrial integrated energy system by reinforcement learning-based differential evolution, IEEE Trans. Green Commun. Netw. 5 (3) (2021) 1077–1090.
[27] J. Sun, X. Liu, T. Bäck, Z. Xu, Learning adaptive differential evolution algorithm from optimization experiences by policy gradient, IEEE Trans. Evol. Comput. 25 (4) (2021) 666–680.
[28] V.A. Tatsis, K.E. Parsopoulos, Reinforcement learning for enhanced online gradient-based parameter adaptation in metaheuristics, Swarm Evol. Comput. 83 (2023) 101371.
[29] Y. Zhang, G. Chen, L. Cheng, Q. Wang, Q. Li, Methods to balance the exploration and exploitation in differential evolution from different scales: A survey, Neurocomputing 561 (2023) 126899.
[30] Z. Song, C. Ren, Z. Meng, An adaptive differential evolution with opposition-learning based diversity enhancement, Expert Syst. Appl. 243 (2024) 122942.
[31] A. Draa, S. Bouzoubia, I. Boukhalfa, A sinusoidal differential evolution algorithm for numerical optimisation, Appl. Soft Comput. 27 (2015) 99–126.
[32] W.-J. Yu, M. Shen, W.-N. Chen, Z.-H. Zhan, Y.-J. Gong, Y. Lin, O. Liu, J. Zhang, Differential evolution with two-level parameter adaptation, IEEE Trans. Cybern. 44 (2014) 1080–1099.
[33] R.A. Sarker, S.M. Elsayed, T. Ray, Differential evolution with dynamic parameters selection for optimization problems, IEEE Trans. Evol. Comput. 18 (2014) 689–707.
[34] L. Tang, Y. Dong, J. Liu, Differential evolution with an individual-dependent mechanism, IEEE Trans. Evol. Comput. 19 (4) (2014) 560–574.
[35] S. Gupta, R. Su, An efficient differential evolution with fitness-based dynamic mutation strategy and control parameters, Knowl.-Based Syst. 251 (2022) 109280.
[36] L. Yu, C. Ren, Z. Meng, A surrogate-assisted differential evolution with fitness-independent parameter adaptation for high-dimensional expensive optimization, Inf. Sci. 662 (2024) 120246.
[37] C. Ren, Z. Song, Z. Meng, Differential evolution with fitness-difference based parameter control and hypervolume diversity indicator for numerical optimization, Eng. Appl. Artif. Intell. 133 (2024) 108081.
[38] D. Adalja, K. Kalita, L. Čepová, P. Patel, N. Mashru, P. Jangir, Arpita, Advancing truss structure optimization—A multi-objective weighted average algorithm with enhanced convergence and diversity, Results Eng. (2025) 104241.
[39] M. Aljaidi, N. Mashru, P. Patel, D. Adalja, P. Jangir, Arpita, S.B. Pandya, M. Khishe, MORIME: A multi-objective RIME optimization framework for efficient truss design, Results Eng. 25 (2025) 103933.
[40] S.X. Zhang, Y.H. Liu, X.R. Hu, L.M. Zheng, S.Y. Zheng, Fully informed fuzzy logic system assisted adaptive differential evolution algorithm for noisy optimization, IEEE Trans. Fuzzy Syst. 33 (6) (2025) 1876–1888.
[41] R. Tanabe, A. Fukunaga, Evaluating the performance of SHADE on CEC 2013 benchmark problems, in: Proc. IEEE Congr. Evol. Comput., 2013, pp. 1952–1959.
[42] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, 2014, submitted for publication.
[43] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: Proc. IEEE Congr. Evol. Comput., 2014, pp. 1658–1665.
[44] V. Stanovov, S. Akhmedova, E. Semenkin, NL-SHADE-LBC algorithm with linear parameter adaptation bias change for CEC 2022 numerical optimization, in: Proc. IEEE Congr. Evol. Comput., IEEE, 2022, pp. 01–08.
[45] B. Chen, H. Ouyang, S. Li, D. Zou, Differential evolution algorithm with a complementary mutation strategy and data fusion-based parameter adaptation, Inf. Sci. 668 (2024) 120522.
[46] R. Zhong, A.G. Hussien, S. Zhang, Y. Xu, J. Yu, Space mission trajectory optimization via competitive differential evolution with independent success history adaptation, Appl. Soft Comput. (2025) 112777.
[47] M. Khishe, P. Jangir, Arpita, S.P. Agrawal, S.B. Pandya, A. Parmar, L. Abualigah, Innovative diversity metrics in hierarchical population-based differential evolution for PEM fuel cell parameter optimization, Eng. Rep. 7 (1) (2025) e13065.
[48] M. Aljaidi, S.P. Agrawal, P. Jangir, S.B. Pandya, A. Parmar, Arpita, A.F. Alkoradees, B.I. Trivedi, M. Khishe, A two stage differential evolution algorithm for parameter estimation of proton exchange membrane fuel cell, Sci. Rep. 15 (1) (2025) 5354.
[49] T.N. Huynh, D.T. Do, J. Lee, Q-learning-based parameter control in differential evolution for structural optimization, Appl. Soft Comput. 107 (2021) 107464.

[50] Z. Tan, Y. Tang, K. Li, H. Huang, S. Luo, Differential evolution with hybrid parameters and mutation strategies based on reinforcement learning, Swarm Evol. Comput. 75 (2022) 101194.
[51] V.A. Tatsis, K.E. Parsopoulos, Dynamic parameter adaptation in metaheuristics using gradient approximation and line search, Appl. Soft Comput. 74 (2019) 368–384.
[52] M. Alkhalidy, M.B. Taha, R. Chowdhury, C. Talhi, H. Ould-Slimane, A. Mourad, Optimizing CP-ABE decryption in urban VANETs: A hybrid reinforcement learning and differential evolution approach, IEEE Open J. Commun. Soc. (2024).
[53] Y. Wang, X. Yu, W. Zhang, An improved reinforcement learning-based differential evolution algorithm for combined economic and emission dispatch problems, Eng. Appl. Artif. Intell. 140 (2025) 109709.
[54] 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.
[55] Z. Zeng, M. Zhang, H. Zhang, Z. Hong, Improved differential evolution algorithm based on the sawtooth-linear population size adaptive method, Inf. Sci. 608 (2022) 1045–1071.
[56] N. Awad, M. Ali, J. Liang, B. Qu, P. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization, Tech. Rep., Nanyang Technological University Singapore Singapore, 2016, pp. 1–34.
[57] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. 1 (6) (1945) 80–83.
[58] J. Brest, M.S. Maučec, B. Bošković, iL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization, in: Proc. IEEE Congr. Evol. Comput., IEEE, 2016, pp. 1188–1195.
[59] S.X. Zhang, Y.N. Wen, Y.H. Liu, L.M. Zheng, S.Y. Zheng, Differential evolution with domain transform, IEEE Trans. Evol. Comput. 27 (2023) 1440–1455.
[60] S.X. Zhang, S.Y. Zheng, L.M. Zheng, Differential evolution with objective and dimension knowledge utilization, Swarm Evol. Comput. 80 (2023) 101322.
[61] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18.
[62] W. Yi, Y. Chen, Z. Pei, J. Lu, Adaptive differential evolution with ensembling operators for continuous optimization problems, Swarm Evol. Comput. 69 (2022) 100994.
[63] 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., 2018, pp. 1–8.
[64] S.X. Zhang, X.R. Hu, S.Y. Zheng, Differential evolution with evolutionary scale adaptation, Swarm Evol. Comput. 85 (2024) 101481.
[65] S.X. Zhang, Y.H. Liu, L.M. Zheng, S.Y. Zheng, Differential evolution with collective ensemble learning, Swarm Evol. Comput. 87 (2024) 101521.
[66] 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, Kolkata, 2010, pp. 341–359.