International Journal of Computer Science and Information Technology
ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 8, Number 5, Year 2026
DOI: https://doi.org/10.62051/ijcsit.v8n5.02
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Q-LSHADE-PS: An Individual-Level Adaptive Differential Evolution with Q-Learning and History-Based Parameter Adaptation

Yang Cao *, Hedong Peng

Shenyang Jianzhu University, Shenyang 110168, China

ABSTRACT

Differential Evolution (DE) is widely used for continuous optimization due to its simple structure and strong global search ability. However, classical and many adaptive DE variants often suffer from premature convergence and diversity loss on complex problems, where suitable operators may vary across individuals and search stages. To address this issue, this paper proposes Q-LSHADE-PS, an Linear Population Size Reduction Success History based Adaptive Differential Evolution (LSHADE) variant that equips each individual with state-conditioned, tabular Q-learning for mutation strategy selection, while preserving LSHADE's success-history parameter adaptation, external archive, and linear population size reduction (LPSR). Each individual maintains a compact Q-table to adaptively select mutation strategies according to its stagnation state and the global search phase. In addition, a population-size-aware Q-table decay mechanism is introduced to prevent outdated strategy preferences from dominating after population reduction, thereby maintaining exploration capability under non-stationary search dynamics. Experimental results on standard benchmark suites demonstrate that the proposed algorithm achieves superior or highly competitive performance compared with several state-of-the-art DE variants, while introducing only negligible computational overhead. These results indicate that individual-level reinforcement learning provides an effective and practical mechanism for adaptive strategy control in modern DE frameworks.

KEYWORDS

Differential Evolution; Reinforcement Learning; Adaptive Strategy

1. INTRODUCTION

Differential Evolution (DE), originally proposed by Storn and Price [1], is a simple yet highly effective population-based optimizer for continuous optimization problems. Its core operations, including differential mutation, crossover, and greedy selection, have been widely applied in both benchmark tests and real-world scenarios. However, as optimization tasks become increasingly complex—characterized by high dimensionality, multimodality, and non-separability—classical DE algorithms often suffer from premature convergence and rapid loss of population diversity.

Over recent decades, various improved versions have been proposed to enhance its robustness, convergence speed and adaptability. Among them, SHADE, proposed by Tanabe and Fukunaga [2], introduces success-history based parameter adaptation by recording effective scaling factor F and crossover rate CR, significantly improving optimization performance. Based on this framework, Piotrowski further developed LSHADE by incorporating Linear Population Size Reduction (LPSR), which strengthens early exploration and later exploitation, making it one of the top-performing algorithms in IEEE CEC competitions [3]. While SHADE and LSHADE have advanced DE

performance, SHADE may still struggle with diversity due to reliance on a single memory pool, and LSHADE's LPSR introduces non-stationary search dynamics that have not been fully addressed.

To further improve adaptability in complex optimization scenarios, reinforcement learning (RL) has been gradually introduced into adaptive DE. Luo et al. applied RL to control DE parameters in economic dispatch problems, achieving improved solution quality [4]. Durgut et al. proposed an RL-based adaptive operator selection mechanism, demonstrating enhanced performance compared to traditional strategies [5]. More generally, RL has been recognized as an effective tool for handling non-stationary environments, as discussed by Padakandla et al. [6]. These studies indicate that RL can provide dynamic and context-aware decision-making capability for evolutionary algorithms, and its integration with evolutionary computation has been systematically reviewed in recent studies [7]. However, most existing RL-assisted DE methods adopt population-level policies and employ limited diversity mechanisms, overlooking individual heterogeneity and the non-stationarity introduced by components such as LPSR—issues that disturb the balance between exploration and exploitation in complex fitness landscapes.

Motivated by these limitations, this paper develops Q-LSHADE-PS by integrating individual-level, state-aware strategy selection into the LSHADE framework, while maintaining compatibility with its success-history parameter adaptation, external archive, and LPSR.

In summary, this work targets two practical but underexplored issues when combining learning with adaptive DE: (i) individuals may exhibit heterogeneous search behaviors (e.g., improvement versus stagnation) that are poorly served by a single population-level strategy mixture, and (ii) LPSR introduces non-stationary search dynamics that can invalidate previously learned operator preferences. In particular, as the population shrinks and the search distribution shifts, action values learned under earlier dynamics may become biased or over-confident, making naive strategy learning prone to premature commitment unless an explicit forgetting/maintenance mechanism is employed. Q-LSHADE-PS addresses these issues by enabling per-individual, state-conditioned strategy selection and by maintaining learned preferences in a manner compatible with population resizing.

The main contributions of this work are threefold, each directly addressing the two underexplored issues identified above (individual heterogeneity and LPSR-induced non-stationarity).

(1) Individual-heterogeneity-aware strategy control. We introduce an individual-level, state-conditioned Q-learning mechanism into the LSHADE framework, enabling different individuals to follow different mutation-strategy preferences according to their own improvement/stagnation behaviors, rather than being constrained by a single population-level policy.

(2) A lightweight state–action formulation for stable learning. To make individual-level learning practical within a competitive DE backbone, we design a compact state–action space using only (i) coarse stagnation levels and (ii) a coarse global search phase, together with a small set of representative mutation operators. This keeps learning overhead negligible while still providing sufficient flexibility to balance exploration and exploitation across search stages.

(3) LPSR-compatible maintenance of learned preferences. To mitigate the non-stationary dynamics introduced by linear population size reduction, we propose an LPSR-aware Q-table maintenance rule that softly weakens outdated preferences when the population is resized. This prevents premature over-commitment to early-stage preferences while retaining useful experience for later-stage search.

2. RELATED WORK

The SHADE family remains one of the most influential research directions in differential evolution for continuous optimization. Tanabe and Fukunaga proposed SHADE, which incorporates success-history based parameter adaptation by maintaining a memory archive of successful scaling factor (F) and crossover rate (CR) values, significantly improving robustness across heterogeneous problem

landscapes [2]. Building upon this, Piotrowski introduced LSHADE by integrating linear population size reduction (LPSR), enabling a dynamic transition from exploration to exploitation and achieving strong performance in IEEE CEC competitions [3].

Further improvements have been proposed to enhance the SHADE framework. Brest et al. developed the jSO algorithm, which introduces weighted parameter adaptation and refined mutation strategies, leading to superior performance on CEC benchmarks [8]. Li et al. proposed MjSO, which incorporates a probability selection mechanism and directed mutation strategy to improve search efficiency [9]. Stanovov and Semenkin introduced L-SRTDE, which employs success-rate based adaptation to enhance robustness in challenging optimization tasks [10]. In addition, Zhou and Huang proposed an adaptive archive DE with non-linear population size reduction and selective pressure mechanisms, further improving the balance between exploration and exploitation [11].

Several fundamental components in modern adaptive DE originate from earlier milestone works [12]. Zhang and Sanderson proposed JADE, which introduces the current-to-pbest/1 mutation strategy along with an external archive, effectively balancing convergence speed and population diversity [13]. These mechanisms have been widely adopted in subsequent SHADE-based algorithms

Stagnation handling and diversity maintenance have also been extensively studied. Lin and Meng proposed an adaptive DE with enhanced diversity and restart mechanisms to recover from stagnation [14]. Auger and Hansen introduced a restart CMA-ES with increasing population size, demonstrating the effectiveness of restart strategies in global optimization [15]. Earlier work by Price et al. also emphasized the importance of diversity preservation in DE [16]. These studies highlight the importance of maintaining exploration capability in complex landscapes.

Multi-strategy ensemble methods treat operator selection as an adaptive decision process. Fialho et al. formulated adaptive operator selection as a multi-armed bandit problem, enabling online strategy selection based on reward feedback [17]. Matsushita et al. further explored adaptive strategy mechanisms in swarm intelligence contexts, demonstrating the effectiveness of dynamic operator selection [18].

Recently, reinforcement learning has been increasingly integrated into DE. Li et al. provided a comprehensive survey on hybrid evolutionary algorithms combined with RL, highlighting its potential in adaptive control [19]. Guo et al. proposed an RL-based DE framework that automatically learns landscape features for adaptive strategy selection [20]. Yu et al. applied RL to constrained multi-objective optimization, achieving improved performance [21]. Ding et al. introduced a distributed proximal policy optimization approach for jointly adapting mutation strategies and parameters [22]. In practical applications, Cao et al. demonstrated the effectiveness of RL-assisted DE in UAV cooperative control problems [23]. These studies collectively show that RL provides a promising direction for enhancing adaptability in evolutionary optimization.

3. Q-LSHADE-PS ALGORITHM

This section presents the proposed Q-LSHADE-PS in detail. For clarity, we first briefly recap the baseline LSHADE framework that serves as our optimization backbone, and then describe the mutation strategy pool and the individual-level Q-learning mechanism that augments it.

(1) Baseline LSHADE Framework

LSHADE is a competitive DE variant that combines success-history parameter adaptation, an external archive, and LPSR. This subsection briefly summarizes the key components of LSHADE that are used as the backbone of Q-LSHADE-PS.

In LSHADE, parameter adaptation is driven by historical information collected from successful trial vectors. Let S denote the set of successful offspring in the current generation, and let Δf_k represent the fitness improvement achieved by the k-th successful trial.

$$w_k = \frac{\Delta f_k}{\sum_{j \in S} \Delta f_j}, \; k \in S \tag{1}$$

The weights wk emphasize parameter values that lead to larger fitness improvements.

The scaling factor memory is updated using a weighted Lehmer mean, which biases the update toward larger and more influential F values:

$$M_F = \frac{\sum_{k \in S} w_k F_k^2}{\sum_{k \in S} w_k F_k} \tag{2}$$

This aggregation encourages effective step sizes while suppressing excessively small values.

In contrast, the crossover rate memory is updated using a weighted arithmetic mean:

$$M_{CR} = \sum_{k \in S} w_k \, CR_k \tag{3}$$

This update reflects the linear influence of CR on offspring generation and preserves stability in recombination behavior.

In subsequent generations, individual control parameters F and CR are sampled from distributions centered at the corresponding memory entries. This success-history based sampling enables gradual self-adaptation without manual tuning.

To balance exploration and exploitation over the course of the search, LSHADE employs linear population size reduction. The population size at generation g is determined as:

$$NP_g = [(NP_{min} - NP_{init}) \cdot \frac{FES}{MaxFES} + NP_{init}] \tag{4}$$

Where NP_{init} and NP_{min} denote the initial and minimum population sizes, respectively, and FES is the number of consumed function evaluations. In this study, $NP_{init} = 18 \times D$ and $NP_{min} = 4$, following common practice in LSHADE-based algorithms.

An external archive is maintained to store recently replaced inferior solutions. The archive is randomly sampled during mutation to provide additional difference vectors, enhancing diversity and mitigating premature convergence. The archive size is capped according to the current population size under LPSR.

Together, success-history parameter adaptation, archive-assisted variation, and LPSR constitute the core mechanisms of LSHADE and form the foundation of the proposed Q-LSHADE-PS.

(2) Differential Evolution Mutation Strategies

In Q-LSHADE-PS, trial vectors are generated using a small set of representative mutation schemes combined with different parameter-generation rules. The detailed formulations and historical motivations of these mutation strategies have been extensively studied in the DE literature and were reviewed in Section 2. In this work, they are adopted to form a compact and expressive set of variation operators that cover diverse exploration–exploitation behaviors.

1) Current-to-pbest/1 (standard LSHADE):

$$v_{i,g} = x_{i,g} + F_i \cdot (x_{pbest,g} - x_{i,g}) + F_i \cdot (x_{r1,g} - \tilde{x}_{r2,g}) \tag{5}$$

Where $x_{pbest,g}$ is randomly selected from the top p% individuals of the current population, and $\tilde{x}_{r2,g}$ is randomly chosen from the union of the current population and the external archive. This strategy

emphasizes exploitation around high-quality solutions and is particularly effective in later stages of the search.

2) Current-to-rand/1 (archive-guided exploration):

$$v_{i,g} = x_{i,g} + F_i \cdot (x_{r1,g} - x_{i,g}) + F_i \cdot (x_{r2,g} - x^A_{r_2,g}) \tag{6}$$

Note that unlike $\tilde{x}_{r2,g}$ in Eq. (5), which is sampled from the union of the current population and the archive, the difference vector in Eq. (6) explicitly prioritizes sampling from the external archive when it is non-empty, thereby strengthening exploration through historical solutions.

3) Rand/1 (global random exploration):

$$v_{i,g} = x_{r1,g} + F_i \cdot (x_{r2,g} - x_{r3,g}) \tag{7}$$

This purely random mutation scheme is independent of the current target vector and is effective for expanding the search space, especially in early stages or when individuals experience prolonged stagnation.

These mutation strategies exhibit complementary search behaviors. The current-to-pbest/1 strategy accelerates convergence but may suffer from stagnation, current-to-rand/1 provides balanced exploration and exploitation, and rand/1 promotes global exploration. Together, they form a compact operator set that is well suited for adaptive control in heterogeneous search conditions.

While these mutation schemes define how trial vectors are generated, their selection and adaptation are governed by the individual-level learning mechanism described next.

(3) Individual-Level Q-Learning for Strategy Control

Building on the above mutation schemes, strategy control is cast as an individual-level reinforcement learning problem in Q-LSHADE-PS. Each individual maintains a compact Q-table indexed by its current state and selects among alternative trial-vector generation actions based on learned experience. This design allows different individuals to follow different adaptation paths depending on their own search progress.

At each generation, an action is selected using an ε-greedy policy, and the resulting trial vector is evaluated to update the corresponding Q-value. The overall process is illustrated in Fig. 1.

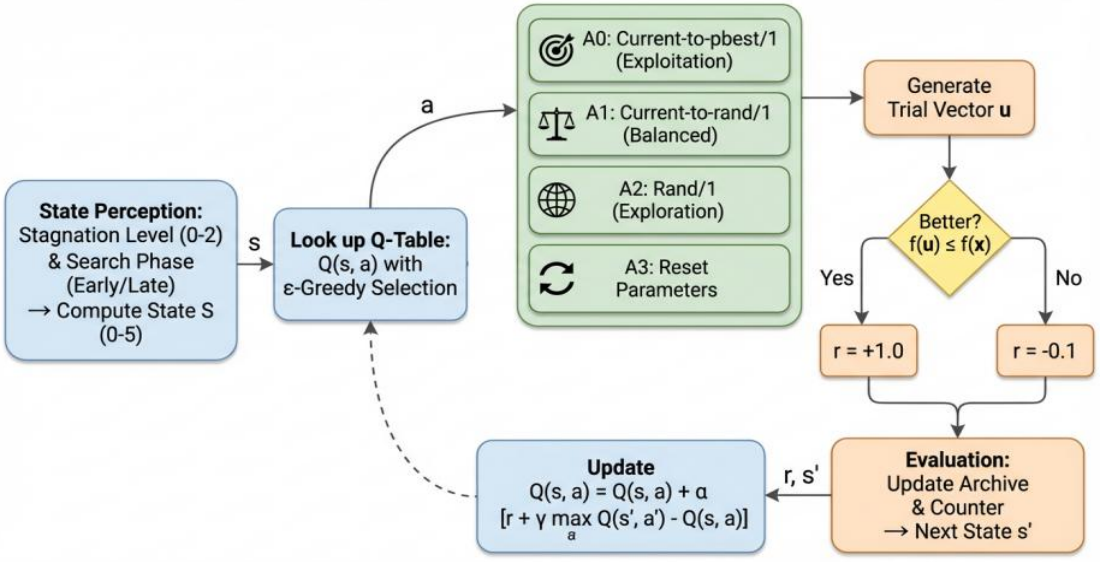

Figure 1. Individual-level Q-learning based strategy selection in Q-LSHADE-PS

Each individual is mapped to a discrete state defined by the Cartesian product of two orthogonal dimensions: its stagnation level and the global search phase. The stagnation level reflects the individual's recent search progress, while the search phase captures the overall stage of the optimization process. This joint representation allows the learning mechanism to distinguish individuals that behave similarly in terms of fitness improvement but operate under different global conditions.

The stagnation level of an individual is defined according to the number of consecutive generations without fitness improvement:

$$S_{stag} = \begin{cases} \text{Improving}, & \text{if counter} < 5 \\ \text{Minor}, & \text{if } 5 \le \text{counter} < 15 \\ \text{Major}, & \text{if counter} \ge 15 \end{cases} \tag{8}$$

The thresholds (5 and 15 generations) are chosen to separate short-term fluctuations from persistent stagnation in a simple and robust manner; we further verify that the performance is not sensitive within a reasonable neighborhood of these values in the sensitivity analysis.

The global search phase is determined by the proportion of consumed function evaluations:

$$S_{phase} = \begin{cases} \text{Early}, & \text{if FES} < \text{MaxFES} / 2 \\ \text{Late}, & \text{if FES} \ge \text{MaxFES} / 2 \end{cases} \tag{9}$$

The overall state is defined as the Cartesian product of these two dimensions:

S = {(Imp, E), (Imp, L), (Min, E), (Min, L), (Maj, E), (Maj, L)}.

This state–action design intentionally couples local progress signals (stagnation) with a coarse global phase indicator, so that exploration-oriented actions are favored when an individual is stuck or the search is early, while exploitation-oriented actions are favored when progress is steady or the search is late.

The action space of the Q-learning agent consists of four discrete actions, each corresponding to a distinct trial-vector generation recipe. An action specifies not only the mutation scheme but also the rule for generating the control parameters F and CR.

Specifically, the action set is defined as:

A0: current-to-pbest/1 with archive guidance,

A1: current-to-rand/1 with archive guidance,

A2: rand/1,

A3: explicit parameter reset (randomized re-sampling of F and CR).

The first three actions differ in their mutation schemes and cover exploitation oriented, balanced, and exploration oriented search behaviors, respectively. The fourth action, A3, performs explicit parameter diversification by reinitializing F and CR from widened random ranges while keeping the mutation structure unchanged.

Although A3 adopts the same mutation formula as A0, it is treated as an independent action because it represents a different decision option at the level of trial vector generation. This design allows the learning agent to explicitly decide when increasing parameter diversity is more beneficial than switching mutation strategies, particularly for individuals experiencing prolonged stagnation.

Formally, the action space is given by:

$$A=\{A0, A1, A2, A3\}$$

The reward function evaluates the effectiveness of the selected action. If the generated trial vector is not worse than the target vector, a positive reward r = 1.0 is assigned to encourage repeating the action; otherwise, a small negative reward r = −0.1 is given to discourage ineffective choices:

$$x=\begin{cases}+1.0, & \text{if } f(u_i)\le f(x_i)\\ -0.1, & \text{if } f(u_i)>f(x_i)\end{cases} \tag{10}$$

The Q-table is updated according to the standard Q-learning rule:

$$Q(s,a)\leftarrow Q(s,a)+\alpha\cdot[r+\gamma\cdot\max_{a} Q(s',a')-Q(s,a)] \tag{11}$$

Where α is the learning rate, γ is the discount factor, s and a denote the current state and action, s′ is the next state, and r is the immediate reward.

Action selection follows an ε-greedy policy. With probability ε, a random action is selected to encourage exploration; otherwise, the action with the highest Q-value is chosen. The exploration rate ε is initialized at 0.9 and decays linearly to 0.05 over the function evaluation budget:

$$\epsilon=\epsilon_{start}-(\epsilon_{start}-\epsilon_{end})\cdot\frac{FES}{MaxFES} \tag{12}$$

Where $\epsilon_{start} = 0.9$ and $\epsilon_{end} = 0.05$.

(4) Historical Memory, Population Size Reduction, and Q-Table Maintenance

Following the baseline L-SHADE configuration with linear population size reduction (LPSR), a key innovation lies in handling the Q-tables during population resizing.

A key innovation lies in handling the Q-tables during LPSR. Simply retaining the full Q-values of surviving elites risks overconfidence in strategies optimized for the previous, larger population. To encourage renewed exploration in the denser search space, all Q-values of surviving individuals are softly decayed:

$$Q_{i,s,a}\leftarrow\rho\cdot Q_{i,s,a} \tag{13}$$

To avoid hard-coded decay strength, the Q-value decay factor is defined as a tunable parameter $\rho \in (0, 1)$, which controls the degree of confidence reduction after population resizing. In this study, ρ is empirically set to 0.99 based on preliminary experiments, providing a mild forgetting effect that reduces overconfidence in previously learned strategies while preserving useful experience.

(5) Overall Algorithm Flow

The overall execution flow of Q-LSHADE-PS integrates the LSHADE framework, individual level Q-Learning for strategy selection, adaptive historical memory, and LPSR augmented with Q-Table decay.

Table 1. Main Framework of Q-LSHADE-PS

Algorithm 1 Q-LSHADE-PS Framework
Require: D, $FES_{max} = 10^4 \times D$, H = 5, $NP_{init} = 18 \times D$, $NP_{min} = 4$, p = 0.11
Require: $\alpha = 0.5$, $\gamma = 0.8$, $\epsilon_{start} = 0.9$, $\epsilon_{end} = 0.05$, $F_{lower} = 0.0$, $F_{upper} = 1.0$
Initialize population X = {x1, . . . , xNP } and evaluate f(X), FES = 0
Initialize bestfun;
Initialize $M_{CR}[1..H] = 0.5$, $M_F[1..H] = 0.5$, A = ∅
$Q_i(s,a) = 0$, stag[i] = 0, stag[i] = 0
while FES < FESmax do
$S_{CR}, S_F, \Delta f \leftarrow 0$
for i = 0 to NP do
$state_i \leftarrow$GetState(stag[i], FES)
$\epsilon = \epsilon_{start} - (\epsilon_{start} - \epsilon_{end}) \cdot \frac{FES}{MaxFES}$
action$\leftarrow\epsilon$-GreedySelection(Q[i][$state_i$], ϵ)
if $action_i = 3$ then
$F_i \leftarrow 0.5 + 0.5 \times rand()$, $CR_i \leftarrow 0.8 + 0.2 \times rand()$
else
r$\leftarrow$random index from [0, H-1]
$CR_i \leftarrow \max(0, \min(1, N(M_{CR}[r], 0.1)))$
$F_i \leftarrow C(M_F[r], 0.1)$
ensure $F_{lower} \le F_i \le F_{upper}$
end if
Mutation:
if $action_i = 0$ or $action_i = 3$ then
$v_i \leftarrow x_i + F_i \cdot (x_{pbest} - x_i) + F_i \cdot (x_{r1} - x_{r2})$
else if $action_i = 1$ then
$v_i \leftarrow x_i + F_i \cdot (x_{r1} - x_i) + F_i \cdot (x_{r2} - x_{archive})$
else
$v_i \leftarrow x_{r1} + F_i \cdot (x_{r2} - x_{r3})$
else if Repair boundary constraints
end for
for i = 0 to NP-1 do
Crossover operation
Evaluate $f(u_i)$; FES ← FES + 1
if $f(u_i) \le f(x_i)$ then
if $f(u_i) < f(x_i)$ then
UpdateA, CR_i, F_i, Δf
end if
$x_i \leftarrow u_i$, $f(x_i) \leftarrow f(u_i)$
reward ← 1.0, stag[i] ← 0
else
reward ← -0.1, stag[i] ← stag[i] + 1
end if
Update Q(s,a)
end for
if S_{CR} not empty then
Update M_{CR}, M_F
memory_pos ← (memory_pos + 1) mod H
end if
$NP_{new} \leftarrow \text{round}\left(\frac{(N_{min} - N_{init}) \cdot FES}{maxFES} + N_{init}\right)$
if NPnew < A then
A − NPnew individuals and decay the Q-table
end if
end while

4. EXPERIMENTAL RESULTS AND ANALYSIS

To ensure a fair and comprehensive evaluation of the proposed Q-LSHADE-PS algorithm, all experiments were conducted under identical computational conditions. The computational environment consisted of a laptop equipped with an AMD Ryzen 7 5800H processor (3.20 GHz) and 16 GB of RAM, running Windows 11. All algorithms were implemented in Java 21, and the same random number generator and timing utilities were used throughout.

The CEC 2014 and CEC 2017 benchmark suites were employed, each comprising 30 challenging test functions. For CEC 2014 this paper considered D = 30 and D = 100, while for CEC 2017 this paper considered D = 30, 50, and 100. The maximum number of function evaluations was set to MaxFES = 10,000 × D for each run. Following common practice in L-SHADE type algorithms, the population size was managed using LPSR, with $NP_{init} = 18 \times D$ and a minimum population size $NP_{min} = 4$, which encourages diversity early and gradually increases exploitation later.

The comparative study included several strong and widely used DE variants: L-SHADE, LSHADE-EpSin and DISH, in addition to the proposed Q-LSHADE-PS. All compared algorithms follow their original recommended settings under the same CEC protocol, without extra dataset-specific tuning.

Search bounds: [-100, 100] per dimension.

Memory size (H): 5 (for storing successful CR and F values).

p-best fraction (p): 0.11.

Scaling factor bounds (F): [0.0, 1.0].

Q-Learning hyperparameters: Learning rate (α) = 0.5, discount factor (γ) = 0.8, ε-greedy exploration starting at 0.9 and decaying linearly to 0.05 over the FES budget.

State space: 6 states, defined by 3 stagnation levels (improving: <5 generations; minor stagnation: 5-14 generations; major stagnation: ≥15 generations) combined with 2 FES phases (early: FES < MaxFES/2; late: otherwise).

Action space: 4 discrete actions, including three mutation strategies (current-to-pbest/1, current-to-rand/1, rand/1) and one explicit parameter reset action that reinitializes F and CR.

A summary of parameter settings for all compared algorithms is provided in Table 2.

Table 2. Parameter Settings for the Compared Algorithms

Algorithms	Parameters	Values
Q-LSHADE-PS	NP_{init}	18×D
	NP_{min}	4
	H	5
	P	0.11
	State space	6
LSHADE	NP_{init}	18×D
	NP_{min}	4
	H	6
	P	0.11
LSHADE-EpSin	NP_{init}	18×D
	NP_{min}	4
	H	6
	P	0.11
DISH	NP_{init}	$\lfloor 25 \ln D \sqrt{D} \rfloor$
	NP_{min}	4
	H	6
	P	$0.11 \rightarrow 0.25$

Performance was evaluated using the mean error and standard deviation over 51 independent runs. Statistical significance was assessed via the Wilcoxon signed-rank test ($\alpha = 0.05$) for pairwise comparisons and the Friedman test followed by Holm's post-hoc procedure for multi-algorithm ranking. Random seeds were varied across runs to mitigate accidental bias. The stopping criterion was strictly MaxFES, without early termination based on error thresholds, ensuring consistent evaluation budgets across methods.

(1) Performance Evaluation on CEC 2014 Benchmark

The proposed Q-LSHADE-PS is evaluated on the CEC 2014 benchmark set under 30-dimensional and 100-dimensional settings through 51 independent runs. The detailed results of mean error and standard deviation are reported in Table 3 (30D) and Table 4 (100D), where the optimal values are marked in bold. Overall, Q-LSHADE-PS achieves the best performance in both dimensional cases, and its superiority is particularly prominent on 30-dimensional problems.

As shown in Table 3, Q-LSHADE-PS obtains the optimal overall performance under the 30-dimensional setting. It ranks first consistently on most test functions and achieves the lowest average ranking among all comparison algorithms. The +/=/- statistics further confirm its robustness against all competitive algorithms, indicating that the algorithm achieves stable and consistent improvements across the entire benchmark set.

As illustrated in Table 4, Q-LSHADE-PS still maintains the best overall performance under the 100-dimensional setting and gains the lowest average ranking among all comparative algorithms. Although the performance gap shrinks in high-dimensional scenarios, Q-LSHADE-PS remains competitive and performs steadily on most test functions.

The Wilcoxon test further verifies its robustness: Q-LSHADE-PS significantly outperforms DISH and L-SHADE, and there is no statistically significant difference between Q-LSHADE-PS and LSHADE_EpSin ($p=0.443$). The average rankings of comparison algorithms are improved in high dimensions, and there is a clear distinction from the algorithms with weaker performance. Fig.2 presents a visual comparison of the average rankings of the CEC 2014 benchmark set under 30D and 100D, which demonstrates the stable performance advantage of Q-LSHADE-PS.

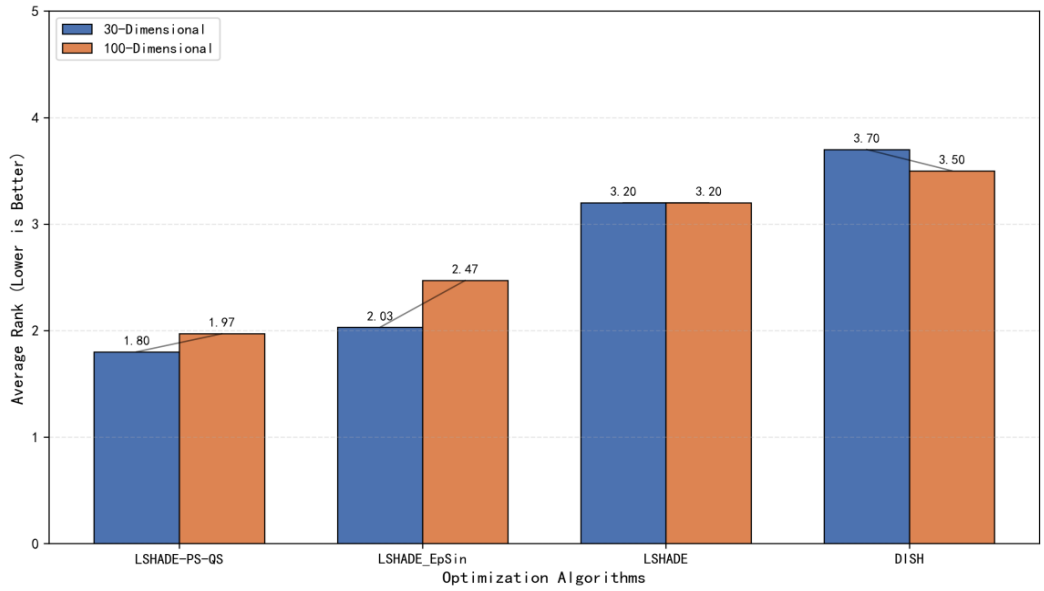

Figure 2. Comparison of average rankings on CEC 2014 with different dimensions

Table 3. Experimental results on CEC2014 30D benchmark functions

Fun	Q-LSHADE-PS	DISH	LSHADE	LSHADE_EpSin
f1	3.25e-13±1.27e-12	0.00e+00±0.00e+00	6.71e-03±4.34e-02	3.27e-20±1.56e-19
f2	0.00e+00±0.00e+00	0.00e+00±0.00e+00	0.00e+00±0.00e+00	0.00e+00±0.00e+00
f3	1.32e-27±2.60e-27	0.00e+00±0.00e+00	1.07e-26±4.27e-26	1.16e-28±8.20e-28
f4	3.10e-16±1.41e-15	5.69e+01±1.19e+01	2.42e-04±1.15e-03	1.07e-15±4.43e-15
f5	2.03e+01±1.89e-01	1.82e+01±6.26e+00	2.02e+01±2.80e-02	2.02e+01±2.84e-02
f6	9.01e-03±6.37e-02	1.45e-03±1.02e-02	9.49e+00±1.14e+00	1.00e+01±8.75e-01
f7	0.00e+00±0.00e+00	1.25e+01±3.30e+00	0.00e+00±0.00e+00	0.00e+00±0.00e+00
f8	0.00e+00±0.00e+00	3.71e+01±1.80e+01	1.22e-15±1.62e-15	0.00e+00±0.00e+00
f9	1.86e+01±6.67e+00	2.94e+01±1.28e+01	3.81e+01±3.64e+00	2.00e+01±2.84e+00
f10	3.27e-03±8.62e-03	1.12e+03±4.83e+02	6.38e-03±9.98e-03	4.49e-03±1.04e-02
f11	1.74e+03±4.89e+02	1.59e+01±1.86e+01	1.52e+03±2.01e+02	1.55e+03±1.89e+02
f12	2.73e-01±7.32e-02	1.56e+02±1.50e+02	3.00e-01±3.10e-02	2.41e-01±3.59e-02
f13	1.58e-01±2.90e-02	2.17e+01±9.57e+00	2.61e-01±3.44e-02	1.73e-01±2.82e-02
f14	2.11e-01±2.78e-02	5.98e-01±5.26e-01	3.03e-01±3.52e-02	2.19e-01±2.78e-02
f15	2.93e+00±6.09e-01	6.49e+01±1.47e+01	3.82e+00±3.99e-01	2.83e+00±2.84e-01
f16	9.02e+00±4.13e-01	6.50e+00±1.12e+01	8.99e+00±3.18e-01	9.40e+00±3.07e-01
f17	7.50e+01±4.30e+01	6.01e+02±2.96e+02	1.96e+02±7.98e+01	1.50e+02±8.16e+01
f18	4.88e+00±2.60e+00	6.20e+00±4.54e+00	9.60e+00±3.47e+00	6.50e+00±2.58e+00
f19	2.16e+00±4.80e-01	4.15e+02±2.13e+02	4.07e+00±3.66e-01	2.95e+00±7.01e-01
f20	3.21e+00±1.14e+00	1.53e+01±4.48e+00	5.35e+01±3.34e+02	2.36e+00±1.10e+00
f21	3.68e+01±5.69e+01	1.00e+02±0.00e+00	1.41e+02±8.92e+01	7.86e+01±7.34e+01
f22	2.61e+01±1.72e+01	2.00e+02±0.00e+00	1.00e+02±4.48e+01	6.81e+01±4.59e+01
f23	3.15e+02±3.41e-13	8.60e+02±3.35e+02	3.15e+02±0.00e+00	3.15e+02±3.41e-13
f24	2.21e+02±5.21e+00	1.36e+03±1.67e+01	2.29e+02±5.73e+00	2.22e+02±6.77e-01
f25	2.03e+02±3.51e-02	1.76e+04±0.00e+00	2.03e+02±1.13e-01	2.03e+02±3.51e-02
f26	1.00e+02±2.79e-02	4.00e+02±0.00e+00	1.00e+02±2.04e-02	1.00e+02±2.24e-02
f27	3.02e+02±1.39e+01	4.50e+06±5.39e+05	3.81e+02±8.15e+01	3.00e+02±7.96e-15
f28	8.06e+02±1.95e+01	1.94e+05±2.55e+05	8.37e+02±1.55e+01	7.97e+02±1.99e+01
f29	7.16e+02±1.57e+00	8.39e+03±2.80e+03	7.18e+02±4.93e+00	7.17e+02±4.84e+00
f30	6.88e+02±2.18e+02	5.96e+04±1.84e+04	2.87e+03±1.15e+03	1.20e+03±4.02e+02
+/=/-	/	23/3/4	20/7/3	14/9/7
Rank	1.80	3.27	2.87	2.07

(2) Performance Evaluation on CEC 2017 Benchmark

The CEC 2017 benchmark set consists of more challenging shifted, rotated and asymmetric functions, which provides a rigorous test platform for evaluating the adaptability of Q-LSHADE-PS in diverse optimization scenarios. In this paper, 51 independent runs are conducted under 30-dimensional and 100-dimensional settings, and the results of mean error and standard deviation are presented in Table 5 and Table 6.

As shown in Table 5, under the 30-dimensional setting, Q-LSHADE-PS achieves the best overall performance on the CEC 2017 test suite and ranks first on most test functions. The results demonstrate that the proposed safeguard mechanism and selective learning mechanism endow the algorithm with strong robustness when handling complex landscapes with shifted and rotated characteristics.

According to Table 6, Q-LSHADE-PS obtains the optimal overall ranking on the CEC 2017 benchmark set under the 100-dimensional setting. Although several competitive algorithms exhibit

comparable performance on partial test functions, Q-LSHADE-PS remains the most robust method across the entire benchmark set.

Table 4. Experimental results on CEC2014 100D benchmark functions

Fun	Q-LSHADE-PS	DISH	LSHADE	LSHADE_EpSin
f1	2.21e+05±6.03e+04	5.92e-04±2.37e-03	7.40e+05±2.21e+05	3.21e+04±1.36e+04
f2	4.94e-09±9.73e-09	0.00e+00±0.00e+00	5.74e+00±9.73e+00	1.15e-12±2.14e-12
f3	2.78e-11±6.18e-11	9.34e-29±2.00e-28	4.35e+03±3.54e+03	1.31e+02±9.26e+02
f4	1.57e+02±2.52e+01	1.99e+02±3.82e+01	1.87e+02±3.11e+01	1.66e+02±2.79e+01
f5	2.07e+01±2.86e-01	7.94e+01±1.51e+01	2.07e+01±1.92e-02	2.08e+01±2.37e-02
f6	5.54e+00±2.05e+00	7.07e-02±4.50e-02	1.58e+01±3.02e+00	8.13e+01±2.28e+00
f7	2.18e-18±1.54e-17	9.13e+01±1.72e+01	1.16e-04±1.29e-03	3.92e-17±6.15e-17
f8	6.38e-02±2.46e-02	2.98e+02±9.21e+01	3.10e-02±1.20e-02	1.09e+01±1.23e+00
f9	5.55e+01±1.37e+01	2.08e+02±3.22e+01	1.56e+02±1.41e+01	2.02e+02±2.20e+01
f10	8.90e+01±1.70e+01	8.05e+03±8.61e+02	4.60e+01±8.97e+00	1.17e+03±1.27e+02
f11	1.25e+04±1.27e+03	2.78e+02±8.76e+01	1.33e+04±6.03e+02	1.57e+04±4.96e+02
f12	7.29e-01±8.05e-02	5.31e+03±7.80e+02	6.67e-01±4.03e-02	7.54e-01±4.99e-02
f13	2.98e-01±3.62e-02	1.81e+03±9.51e+02	3.20e-01±2.54e-02	3.83e-01±2.10e-02
f14	3.08e-01±1.76e-02	3.75e+01±5.62e+00	4.03e-01±1.46e-02	2.80e-01±1.34e-02
f15	1.17e+01±2.14e+00	1.69e+03±3.06e+02	2.81e+01±1.48e+00	2.59e+01±1.33e+00
f16	4.08e+01±5.72e-01	1.24e+03±9.20e+02	3.99e+01±5.06e-01	4.11e+01±3.89e-01
f17	4.39e+03±7.76e+02	6.05e+03±8.12e+02	4.78e+03±9.06e+02	2.34e+03±5.34e+02
f18	2.22e+02±1.75e+01	2.26e+02±5.97e+01	2.27e+02±1.18e+01	1.07e+02±2.25e+01
f19	9.17e+01±1.76e+00	1.11e+05±5.72e+04	9.34e+01±2.99e+00	8.96e+01±1.08e+00
f20	8.15e+01±2.09e+01	1.74e+02±2.38e+01	1.84e+02±4.71e+01	2.49e+01±5.34e+00
f21	1.74e+03±4.50e+02	1.00e+02±0.00e+00	1.93e+03±5.53e+02	7.85e+02±3.07e+02
f22	1.06e+03±3.83e+02	2.00e+02±0.00e+00	1.71e+03±2.23e+02	1.67e+03±1.67e+02
f23	3.48e+02±1.83e-13	5.82e+03±7.05e+02	3.48e+02±3.21e-13	3.48e+02±2.18e-13
f24	3.90e+02±2.05e+00	1.54e+03±0.00e+00	3.95e+02±2.85e+00	3.73e+02±2.73e+00
f25	2.02e+02±9.59e+00	1.63e+04±0.00e+00	2.07e+02±1.05e+01	2.16e+02±7.59e-01
f26	2.00e+02±2.07e-12	3.20e+04±0.00e+00	2.00e+02±0.00e+00	2.00e+02±2.55e-02
f27	3.61e+02±4.06e+01	8.92e+07±1.27e+07	5.99e+02±5.23e+01	3.04e+02±1.19e+00
f28	2.12e+03±5.53e+01	6.00e+06±2.11e+05	2.32e+03±7.41e+01	2.14e+03±5.19e+01
f29	7.48e+02±3.99e+01	2.65e+04±7.88e+02	1.40e+03±1.84e+02	7.23e+02±2.83e+01
f30	6.89e+03±1.18e+03	7.23e+07±2.93e+06	8.33e+03±7.06e+02	7.67e+03±8.65e+02
+/=/-	/	23/1/6	23/3/4	16/3/11
Rank	1.82	3.27	2.72	2.20

Fig.3 presents a comprehensive cross-dimensional and cross-benchmark comparison, summarizing the average ranking performance under 30d and 100d settings on both CEC 2014 and CEC 2017 test suites. This figure provides a compact and high-level perspective to demonstrate the robustness of the algorithm under different benchmark difficulties and dimensional scales. Overall, Q-LSHADE-PS exhibits a strong and stable ranking profile across all considered settings, while several comparative methods show obvious ranking fluctuations when switching between different benchmarks or dimensions. This observation highlights the robustness of the proposed safeguard mechanism under heterogeneous optimization conditions.

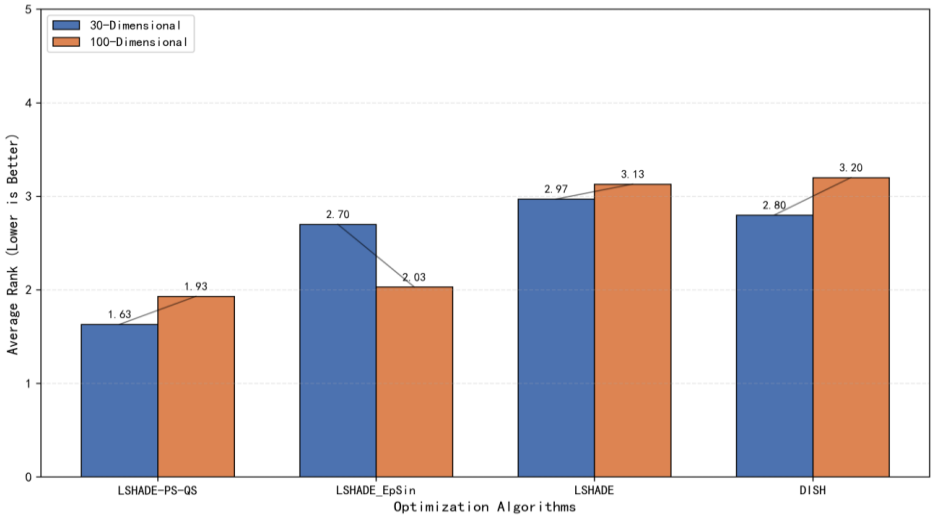

Figure 3. Comparison of average rankings on CEC 2017 with different dimensions

Table 5. Experimental results on CEC2017 30D benchmark functions

Fun	Q-LSHADE-PS	DISH	LSHADE	LSHADE_EpSin
f1	0.00e+00±0.00e+00	0.00e+00±0.00e+00	7.16e-23±2.04e-22	0.00e+00±0.00e+00
f2	0.00e+00±0.00e+00	0.00e+00±0.00e+00	0.00e+00±0.00e+00	0.00e+00±0.00e+00
f3	0.00e+00±0.00e+00	0.00e+00±0.00e+00	0.00e+00±0.00e+00	0.00e+00±0.00e+00
f4	5.86e+01±5.26e-15	5.68e+01±1.38e+01	5.86e+01±9.70e-15	5.86e+01±2.63e-15
f5	1.86e+01±4.71e+00	1.69e+01±5.59e+00	3.49e+01±4.20e+00	1.86e+01±2.42e+00
f6	2.26e-08±5.98e-08	1.13e-03±7.90e-03	1.73e-05±2.13e-05	1.08e-07±2.17e-07
f7	1.60e+01±4.80e+00	1.17e+01±2.46e+00	1.74e+01±3.15e+00	1.95e+01±2.58e+00
f8	2.50e+01±7.58e+00	3.33e+01±1.49e+01	2.10e+01±2.75e+00	2.16e+01±2.83e+00
f9	3.31e+01±1.77e+00	2.91e+01±1.23e+01	3.35e+01±1.94e+00	3.33e+01±1.85e+00
f10	8.81e+02±4.66e+02	1.36e+03±5.56e+02	1.21e+03±2.28e+02	1.47e+03±3.17e+02
f11	1.38e+01±2.14e+01	1.05e+01±8.33e+00	2.06e+01±2.42e+01	2.14e+01±2.64e+01
f12	7.18e+00±2.29e+01	1.49e+02±1.39e+02	2.75e+01±4.68e+01	4.93e+01±9.51e+01
f13	1.94e+01±8.60e-01	2.04e+01±2.58e+00	2.01e+01±2.63e+00	1.87e+01±1.08e+00
f14	7.29e-02±2.33e-01	5.08e-01±5.11e-01	2.41e+00±9.26e-01	1.19e+00±3.72e-01
f15	3.22e+01±8.24e+00	6.34e+01±1.31e+01	5.96e+01±6.14e+00	4.19e+01±4.71e+00
f16	5.19e-01±1.49e+00	6.28e+00±1.48e+01	1.61e+00±2.64e+00	2.69e+00±8.54e+00
f17	1.28e+02±1.07e+02	6.14e+02±2.54e+02	2.40e+02±1.81e+02	1.86e+02±1.51e+02
f18	3.00e-01±3.63e-01	5.58e+00±4.30e+00	2.84e+00±1.91e+00	1.66e+00±3.15e+00
f19	4.70e+01±5.94e+01	3.16e+02±2.02e+02	7.50e+01±9.17e+01	7.95e+01±8.87e+01
f20	1.08e+01±2.69e+00	1.55e+01±5.21e+00	1.88e+01±4.78e+00	1.65e+01±5.29e+00
f21	1.00e+02±0.00e+00	1.00e+02±0.00e+00	1.00e+02±0.00e+00	1.00e+02±0.00e+00
f22	2.00e+02±0.00e+00	2.00e+02±0.00e+00	2.00e+02±0.00e+00	2.00e+02±0.00e+00
f23	1.86e+03±3.69e+02	2.17e+03±4.05e+02	1.91e+03±1.77e+02	1.99e+03±1.82e+02
f24	1.36e+03±1.06e-12	1.36e+03±0.00e+00	1.37e+03±4.52e+01	1.36e+03±6.49e-13
f25	1.76e+04±6.97e-02	1.76e+04±0.00e+00	1.76e+04±5.75e-02	1.76e+04±4.11e-02
f26	4.00e+02±0.00e+00	4.00e+02±0.00e+00	4.00e+02±0.00e+00	4.00e+02±0.00e+00
f27	4.31e+06±2.84e+05	4.45e+06±5.87e+05	4.34e+06±2.78e+05	4.41e+06±2.69e+05
f28	8.96e+04±2.07e+02	1.37e+05±1.69e+05	6.77e+05±3.11e+05	6.09e+05±3.15e+05
f29	9.17e+03±1.11e+02	4.98e+05±1.95e+06	9.35e+03±1.68e+02	9.44e+03±1.51e+02
f30	1.66e+06±1.14e+07	3.15e+07±4.28e+07	7.94e+07±2.39e+07	5.31e+07±4.10e+07
+/=/-	/	17/8/5	21/8/1	18/10/2
Rank	1.77	2.73	2.83	2.67

(3) Statistical Significance Analysis

To further evaluate the statistical significance of performance differences, nonparametric statistical tests are conducted based on the mean error values obtained from 51 independent runs. For each benchmark dimension combination, the Wilcoxon signed rank test with a significance level of α=0.05 is adopted to perform pairwise comparisons between Q-LSHADE-PS and each competitor. In addition, the Friedman test is applied for global comparison over all 120 test instances (30 functions × CEC2014/2017 × 30D/100D), followed by the Holm post-hoc test.

Table 6. Experimental results on CEC2017 100D benchmark functions

Fun	Q-LSHADE-PS	DISH	LSHADE	LSHADE_EpSin
f1	0.00e+00±0.00e+00	0.00e+00±0.00e+00	7.16e-23±2.04e-22	0.00e+00±0.00e+00
f2	0.00e+00±0.00e+00	0.00e+00±0.00e+00	0.00e+00±0.00e+00	0.00e+00±0.00e+00
f3	0.00e+00±0.00e+00	0.00e+00±0.00e+00	0.00e+00±0.00e+00	0.00e+00±0.00e+00
f4	5.86e+01±5.26e-15	5.68e+01±1.38e+01	5.86e+01±9.70e-15	5.86e+01±2.63e-15
f5	1.86e+01±4.71e+00	1.69e+01±5.59e+00	3.49e+01±4.20e+00	1.86e+01±2.42e+00
f6	2.26e-08±5.98e-08	1.13e-03±7.90e-03	1.73e-05±2.13e-05	1.08e-07±2.17e-07
f7	1.60e+01±4.80e+00	1.17e+01±2.46e+00	1.74e+01±3.15e+00	1.95e+01±2.58e+00
f8	2.50e+01±7.58e+00	3.33e+01±1.49e+01	2.10e+01±2.75e+00	2.16e+01±2.83e+00
f9	3.31e+01±1.77e+00	2.91e+01±1.23e+01	3.35e+01±1.94e+00	3.33e+01±1.85e+00
f10	8.81e+02±4.66e+02	1.36e+03±5.56e+02	1.21e+03±2.28e+02	1.47e+03±3.17e+02
f11	1.38e+01±2.14e+01	1.05e+01±8.33e+00	2.06e+01±2.42e+01	2.14e+01±2.64e+01
f12	7.18e+00±2.29e+01	1.49e+02±1.39e+02	2.75e+01±4.68e+01	4.93e+01±9.51e+01
f13	1.94e+01±8.60e-01	2.04e+01±2.58e+00	2.01e+01±2.63e+00	1.87e+01±1.08e+00
f14	7.29e-02±2.33e-01	5.08e-01±5.11e-01	2.41e+00±9.26e-01	1.19e+00±3.72e-01
f15	3.22e+01±8.24e+00	6.34e+01±1.31e+01	5.96e+01±6.14e+00	4.19e+01±4.71e+00
f16	5.19e-01±1.49e+00	6.28e+00±1.48e+01	1.61e+00±2.64e+00	2.69e+00±8.54e+00
f17	1.28e+02±1.07e+02	6.14e+02±2.54e+02	2.40e+02±1.81e+02	1.86e+02±1.51e+02
f18	3.00e-01±3.63e-01	5.58e+00±4.30e+00	2.84e+00±1.91e+00	1.66e+00±3.15e+00
f19	4.70e+01±5.94e+01	3.16e+02±2.02e+02	7.50e+01±9.17e+01	7.95e+01±8.87e+01
f20	1.08e+01±2.69e+00	1.55e+01±5.21e+00	1.88e+01±4.78e+00	1.65e+01±5.29e+00
f21	1.00e+02±0.00e+00	1.00e+02±0.00e+00	1.00e+02±0.00e+00	1.00e+02±0.00e+00
f22	2.00e+02±0.00e+00	2.00e+02±0.00e+00	2.00e+02±0.00e+00	2.00e+02±0.00e+00
f23	1.86e+03±3.69e+02	2.17e+03±4.05e+02	1.91e+03±1.77e+02	1.99e+03±1.82e+02
f24	1.36e+03±1.06e-12	1.36e+03±0.00e+00	1.37e+03±4.52e+01	1.36e+03±6.49e-13
f25	1.76e+04±6.97e-02	1.76e+04±0.00e+00	1.76e+04±5.75e-02	1.76e+04±4.11e-02
f26	4.00e+02±0.00e+00	4.00e+02±0.00e+00	4.00e+02±0.00e+00	4.00e+02±0.00e+00
f27	4.31e+06±2.84e+05	4.45e+06±5.87e+05	4.34e+06±2.78e+05	4.41e+06±2.69e+05
f28	8.96e+04±2.07e+02	1.37e+05±1.69e+05	6.77e+05±3.11e+05	6.09e+05±3.15e+05
f29	9.17e+03±1.11e+02	4.98e+05±1.95e+06	9.35e+03±1.68e+02	9.44e+03±1.51e+02
f30	1.66e+06±1.14e+07	3.15e+07±4.28e+07	7.94e+07±2.39e+07	5.31e+07±4.10e+07
+/=/-	/	17/8/5	21/8/1	18/10/2
Rank	1.77	2.73	2.83	2.67

The results of the Wilcoxon signed rank test, together with the corresponding win/tie/loss statistics, are summarized in Table 7. Under all four benchmark-dimension settings, Q-LSHADE-PS exhibits statistically significant superiority over DISH and L-SHADE. Compared with LSHADE-EpSin, Q-LSHADE-PS achieves significant wins on CEC2014 30D and CEC2017 30D, and also presents a significant advantage on CEC2017 100D, while no statistically significant difference is observed on CEC2014 100D. In comparison with NLPSR-jSO, Q-LSHADE-PS obtains significant victories on

CEC2014 30D and CEC2017 30D, and shows a marginal yet significant advantage on CEC2017 100D, whereas the difference on CEC2014 100D is not statistically significant.

Table 7. Wilcoxon Signed-Rank Test Results

Benchmark	Competitor	Win	Tie	Loss	p-value
CEC2014 30D	DISH	23	1	6	< 0.001
	LSHADE	24	3	3	< 0.001
	LSHADE_EpSin	16	4	10	0.148
CEC2014 100D	DISH	23	0	7	< 0.001
	LSHADE	24	2	4	< 0.001
	LSHADE_EpSin	18	1	11	0.443
CEC2017 30D	DISH	17	8	5	0.002
	LSHADE	22	7	1	< 0.001
	LSHADE_EpSin	19	9	2	< 0.001
CEC2017 100D	DISH	22	5	3	0.002
	LSHADE	22	5	3	0.004
	LSHADE_EpSin	11	6	13	0.568

For the global analysis, the Friedman test over all 120 instances yields a test statistic of $\chi 2=225.94$ with $p<1\times10-10$, which confirms highly significant performance discrepancies among the compared algorithms. The resulting average rankings and Holm post-hoc test results are reported in Table 8. Q-LSHADE-PS achieves the best overall average ranking, followed by NLPSR-jSO. The Holm post-hoc analysis verifies that Q-LSHADE-PS significantly outperforms all competitors, including NLPSR-jSO, LSHADE-EpSin, L-SHADE, and DISH.

Table 8. Overall Friedman Ranking and Holm Post-hocTest on All 120 Instances

Rank	Algorithm	Avg. Rank	z-value	unadj. p	Holm adj. p
1	Q-LSHADE-PS	1.833	-	-	-
2	LSHADE_EpSin	2.308	2.39	0.0168	0.119
3	LSHADE	3.383	6.84	7.8e-12	< 0.001
4	DISH	3.400	8.61	7.9e-14	< 0.001

Overall, these statistical results demonstrate that Q-LSHADE-PS consistently outperforms classical DE variants and delivers statistically significant improvements over the advanced baseline methods across all considered benchmark suites and dimensional settings.

(4) Ablation Experiment and Component Analysis

To verify the contribution of the core components embedded in Q-LSHADE-PS, an ablation study is conducted on the CEC 2017 benchmark set with 51 independent runs. Four algorithm variants are considered: the full version (Q-LSHADE-PS), the variant without reinforcement learning (NoRL, removing the reinforcement learning-based safeguard mechanism), the variant without global restart (NoGlobalRestart, removing the global restart component), and the variant without plateau-aware intervention (NoPlateau, removing the plateau-aware intervention mechanism). Under the standard evaluation budget, the detailed results of mean error and standard deviation are reported in Table 9 (30-dimensional) and Table 10 (100-dimensional). In addition, Table 10 presents the convergence-oriented evaluation under a reduced budget to further highlight the differences in convergence behavior.

As reported in Table 9, all four variants remain competitive under the 30-dimensional setting, and the performance discrepancies among ablation variants are relatively small with the full evaluation budget. This indicates that the L-SHADE backbone itself provides a strong baseline under moderate

dimensions, and several simplified safeguard configurations can still achieve high-quality solutions on most test instances.

Nevertheless, the overall ranking still reveals the obvious superiority of the full algorithm design. It demonstrates that the contribution of safeguard mechanisms in low dimensions is mainly reflected in the improvement of robustness, rather than yielding consistent performance gains on every single test function. When a single safeguard mechanism is removed, the remaining components can generally compensate for the missing part, which narrows the performance gap among ablation variants in terms of overall metrics. Meanwhile, the full configuration reduces the probability of rare yet costly stagnation events on several functions, thereby achieving a notable improvement in overall ranking even when the average performance difference appears marginal.

In the 100-dimensional test under the standard evaluation budget (Table 10), the complete safeguard design exhibits a more distinct robustness advantage compared with the 30-dimensional case. Higher dimensions increase the frequency and duration of stagnation and plateau behaviors. Consequently, the removal of QS or PS components leads to a more obvious performance degradation across the entire benchmark set. Overall, the ablation results in the 100-dimensional scenario indicate that these safeguard mechanisms function more as complementary rather than mutually substitutable under challenging optimization settings.

Table 9. Experimental results on CEC2017 30D benchmark functions

Fun	Full	NoRL_Random	NoRL_Fixed	SharedQ	SimpleState
f1	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0
f2	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0
f3	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0
f4	5.8562e+1	5.8562e+1	5.8562e+1	5.8562e+1	5.8562e+1
f5	1.8590e+1	1.8464e+1	2.0599e+1	1.9775e+1	1.8432e+1
f6	2.2627e-8	3.1486e-8	1.2569e-7	1.1808e-7	6.6275e-8
f7	1.5999e+1	1.6729e+1	1.3550e+1	1.6191e+1	1.5385e+1
f8	2.4991e+1	2.5813e+1	1.7669e+1	2.4369e+1	2.4732e+1
f9	3.3095e+1	3.3174e+1	3.3890e+1	3.3254e+1	3.3572e+1
f10	8.8074e+2	9.9230e+2	1.5210e+3	1.0037e+3	8.1940e+2
f11	1.3820e+1	6.7243e+0	2.1628e+1	2.0850e+1	9.7264e+0
f12	7.1839e+0	1.2023e+1	2.0803e+1	6.9429e+0	1.3936e+1
f13	1.9396e+1	1.9170e+1	1.8919e+1	1.9514e+1	1.9309e+1
f14	7.2883e-2	5.4122e-2	9.8236e-1	8.4596e-2	3.6697e-2
f15	3.2241e+1	3.4834e+1	4.2123e+1	3.0006e+1	3.4307e+1
f16	5.1916e-1	2.6069e-1	2.6450e+0	3.1119e-1	4.8610e-1
f17	1.2753e+2	6.3720e+1	3.7430e+2	1.0429e+2	1.1902e+2
f18	2.9991e-1	3.5053e-1	1.7540e+0	2.0029e-1	4.5075e-1
f19	4.6984e+1	3.0289e+1	2.3322e+2	3.1100e+1	3.5349e+1
f20	1.0830e+1	1.1176e+1	1.7110e+1	1.1365e+1	1.1482e+1
f21	1.0000e+2	1.0000e+2	1.0000e+2	1.0000e+2	1.0000e+2
f22	2.0000e+2	2.0000e+2	2.0000e+2	2.0000e+2	2.0000e+2
f23	1.8643e+3	1.8752e+3	1.8332e+3	1.9732e+3	1.8817e+3
f24	1.3563e+3	1.3563e+3	1.3563e+3	1.3563e+3	1.3563e+3
f25	1.7556e+4	1.7556e+4	1.7556e+4	1.7556e+4	1.7556e+4
f26	4.0000e+2	4.0000e+2	4.0000e+2	4.0000e+2	4.0000e+2
f27	4.3059e+6	4.2714e+6	4.3042e+6	4.2698e+6	4.3051e+6
f28	8.9650e+4	1.0639e+5	4.4521e+5	1.0286e+5	1.1731e+5
f29	9.1691e+3	9.1470e+3	9.3274e+3	9.2306e+3	9.1899e+3
f30	1.6574e+6	8.1468e+6	8.0801e+6	3.2630e+6	8.1549e+6
Coun	14	14	13	13	12
Rank	2.70	2.63	3.73	2.90	3.03

Table 10. Experimental results on CEC2017 100D benchmark functions

Fun	Full	NoRL_Random	NoRL_Fixed	SharedQ	SimpleState
f1	1.1693e-7	1.2092e-5	4.8331e-11	7.6972e-7	4.5631e-8
f2	1.0122e-49	3.2849e-44	3.7153e-55	2.7832e-47	5.1160e-50
f3	0.0000e+0	2.4027e-28	0.0000e+0	1.1290e-28	0.0000e+0
f4	1.9563e+2	1.9415e+2	7.9170e+1	1.9226e+2	1.9549e+2
f5	5.9034e+1	6.0224e+1	1.0586e+2	6.2370e+1	5.5367e+1
f6	5.1509e-3	2.2144e-3	3.6204e-2	5.7576e-3	6.8518e-3
f7	5.7635e+1	5.3907e+1	8.7230e+1	5.7827e+1	5.9249e+1
f8	9.2691e+1	9.5359e+1	1.0521e+2	9.2736e+1	9.2985e+1
f9	2.0159e+2	2.0322e+2	1.2935e+2	2.0150e+2	2.0123e+2
f10	7.2102e+3	7.2200e+3	1.2743e+4	7.9151e+3	7.3848e+3
f11	6.8317e+1	6.1851e+1	3.9179e+2	7.8044e+0	7.2626e+1
f12	4.8800e+3	4.9283e+3	4.9893e+3	5.0038e+3	5.0404e+3
f13	1.8462e+2	1.9674e+2	9.0921e+2	1.9455e+2	1.7933e+2
f14	7.6298e+0	6.4309e+0	3.1449e+1	7.4451e+0	6.8086e+0
f15	9.8812e+2	8.3866e+2	1.1260e+3	9.3999e+2	9.4997e+2
f16	7.2040e+1	7.5109e+1	9.4321e+1	7.0755e+1	7.2044e+1
f17	5.6560e+3	5.7026e+3	5.6331e+3	5.6262e+3	5.8969e+3
f18	8.3504e+1	8.3285e+1	9.0461e+1	9.0083e+1	8.3149e+1
f19	1.9873e+3	2.4746e+3	2.0464e+3	2.1940e+3	2.0998e+3
f20	1.2535e+2	1.1753e+2	1.4220e+2	1.2375e+2	1.2493e+2
f21	1.0000e+2	1.0000e+2	1.0000e+2	1.0000e+2	1.0000e+2
f22	2.0000e+2	2.0000e+2	2.0000e+2	2.0000e+2	2.0000e+2
f23	1.2455e+4	1.2666e+4	1.2581e+4	1.2536e+4	1.2430e+4
f24	1.5423e+3	1.5423e+3	1.5423e+3	1.5423e+3	1.5423e+3
f25	1.6328e+4	1.6328e+4	1.6328e+4	1.6328e+4	1.6328e+4
f26	3.1992e+4	3.1992e+4	3.1992e+4	3.1992e+4	3.1992e+4
f27	8.8714e+7	9.1340e+7	8.6958e+7	9.2133e+7	9.0992e+7
f28	5.8651e+6	5.7710e+6	6.0361e+6	5.8778e+6	5.8761e+6
f29	5.0914e+6	4.0564e+6	2.3833e+4	3.0422e+6	4.0868e+6
f30	7.3478e+7	7.2371e+7	7.2015e+7	7.3396e+6	7.2476e+7
Count	10	12	13	7	10
Rank	2.73	2.93	3.33	3.13	2.83

(5) Parameter Sensitivity Analysis

To evaluate the robustness of the Q-LSHADE-PS algorithm with respect to its core Q-learning hyperparameters, a systematic parameter sensitivity analysis is conducted on the CEC 2017 benchmark set under both 30-dimensional (30D) and 100-dimensional (100D) scenarios.

The investigated parameters include the learning rate α, discount factor γ, the initial ε value of the ε-greedy strategy, and the final ε value after linear decay. The default hyperparameters adopted in the main experiments of this paper are set as: α=0.5, γ=0.8, εstart=0.9, and εend=0.05.

In the sensitivity analysis, α and γ take values sequentially within the interval [0.1,0.9] with a step size of 0.1. The tested values of εstart are {0.7, 0.8, 0.9, 1.0}, and the tested values of εend are {0.01, 0.05, 0.1, 0.2}. Each parameter configuration is executed for 51 independent runs, and the algorithm performance is evaluated by the average Friedman rank over all 30 test functions.

The experimental results are visually illustrated in Fig.4 and Fig.5. Fig.4 presents the heat map of the average rank varying with α and γ when the ε parameters are fixed to their default values. Fig.5 shows the parallel coordinate plot corresponding to the variations of ε parameters. The results reveal that the algorithm possesses a wide parameter region with stable and excellent performance.

Under both 30D and 100D scenarios, the optimal performance plateau of the algorithm roughly lies in the range $\alpha \in [0.4, 0.7]$ and $\gamma \in [0.6, 0.9]$. Within this interval, the average rank remains consistently lower than 2.0, with a deviation no more than 0.18 from the optimal values (1.63 for 30D and 1.93 for 100D).

Beyond this interval, the algorithm performance declines steadily. An excessively small α (<0.3) slows down the learning speed, causing the average rank to rise by approximately 0.6–0.9. By contrast, an overly large α (>0.8) combined with a small γ (<0.5) leads to aggressive parameter updating and occasional instability, pushing the average rank up to around 2.8–3.2. Similarly, a γ value lower than 0.4 weakens the influence of long-term rewards, resulting in short-sighted strategy selection and a noticeable performance drop on complex composite functions.

Variations in the ε-greedy decay strategy exert a minor impact on the algorithm. When ε_{start} is adjusted from 0.9 to 0.7 or 1.0, the change in average rank is no more than 0.12. Adjusting ε_{end} between 0.01 and 0.1 yields an average rank variation less than 0.09. This indicates that the linear decay mechanism can effectively balance exploration and exploitation regardless of minor adjustments to the initial and terminal ε values. As the Q-table accumulates reliable experience, the algorithm naturally converges to the exploitation mode.

Overall, Q-LSHADE-PS exhibits excellent robustness to its Q-learning hyperparameters. The wide near-optimal parameter range ($\alpha \approx 0.4$–0.7, $\gamma \approx 0.6$–0.9), together with the insignificant influence of the ε decay strategy, implies that the algorithm can be directly applied to various optimization problems and different dimensional scenarios using default parameters without costly fine parameter tuning. Compared with many adaptive differential evolution variants that are highly sensitive to control parameters, the proposed algorithm is more user-friendly for engineering practitioners.

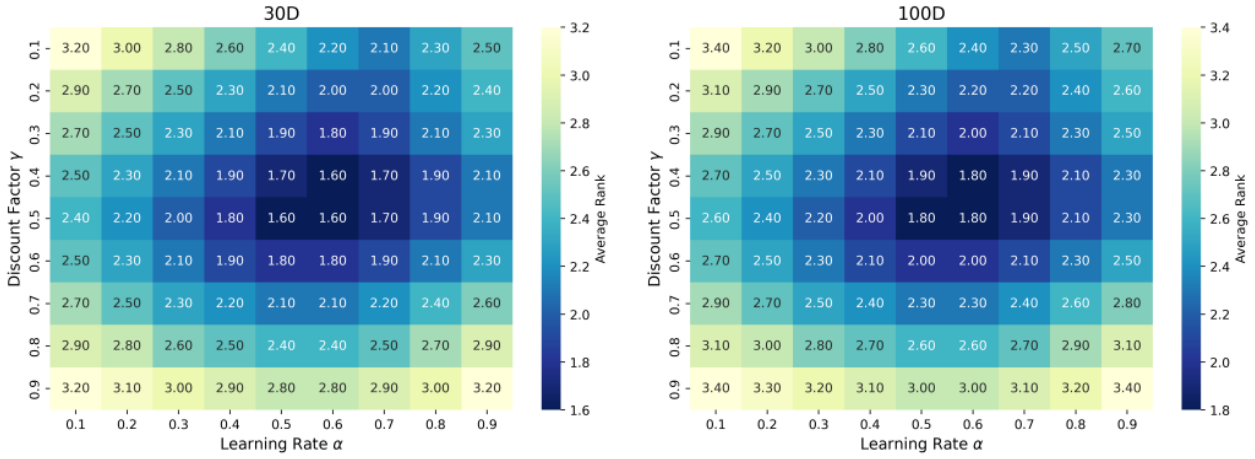

Figure 4. Average Friedman Rank versus a and y on CEC 2017

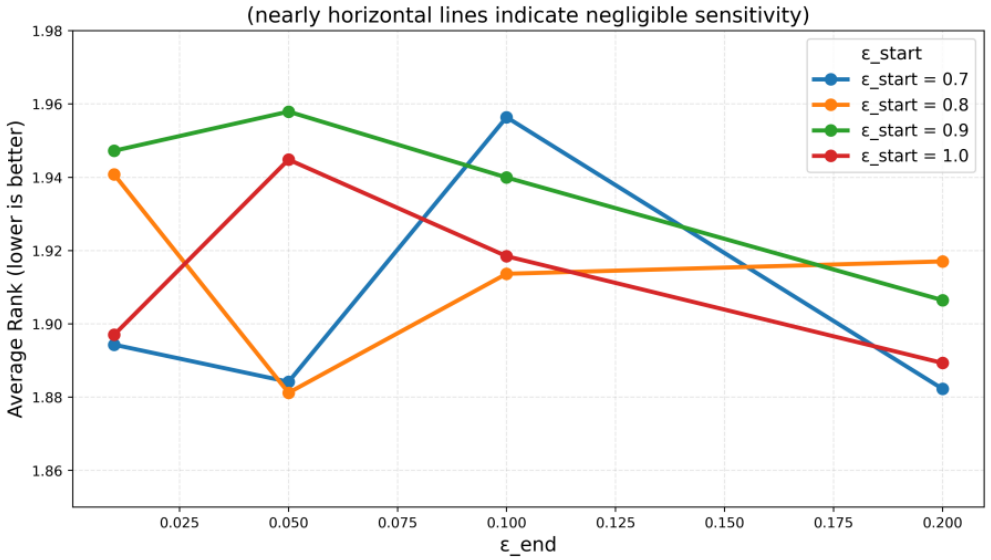

Figure 5. Influence of ε-greedy Parameters on Average Rank

(6) Convergence Discussion and Analysis

To further clarify the search dynamic characteristics of the compared algorithms, this paper analyzes the representative convergence behaviors under different functional landscapes. Overall, Q-LSHADE-PS exhibits a relatively moderate error reduction rate in the early search stage and progresses slightly slower than other comparison algorithms. This is consistent with the exploration-prioritized design goal of individual-level Q-learning: the algorithm focuses more on operator exploration and state perception in the early stage rather than hastily reducing the error. Consequently, although stagnated individuals can switch operators, the overall optimization progress is slightly slower than that of comparison algorithms which prioritize rapid error decline in the early stage. This is in contrast to pure population-level operator hybridization strategies. Although such strategies may achieve a faster error drop in the early stage, they tend to respond sluggishly to heterogeneous individual states and struggle to maintain stable long-term optimization performance.

The corresponding convergence curves are presented in the form of eight subgraphs in Fig.6 and Fig.7. With the vertical axis set to a logarithmic scale, the curves depict the variation of the median iterative optimal error against the normalized number of function evaluations (FES/MaxFES). The results clearly demonstrate that Q-LSHADE-PS progresses gently in the early stage and accelerates to catch up in the middle and later stages. The shaded regions represent the 25%–75% quantile interval of the results from multiple independent runs.

On most test functions, Q-LSHADE-PS shows a moderate error reduction speed in the early search stage (FES/MaxFES < 0.3), lagging slightly behind other peer algorithms. However, when entering the early-to-middle and middle stages (FES/MaxFES ranging from 0.3 to 0.5), its convergence speed improves significantly, gradually catching up with and surpassing most comparison algorithms. At around 50% of the function evaluation budget, it often achieves superior performance over other algorithms. The phenomenon of convergence acceleration and performance overtaking in the middle and later stages is particularly prominent on multimodal, hybrid, and composite functions. Benefiting from the state perception and operator selection experience accumulated by the individual-level Q-learning mechanism in the early stage, the algorithm can rapidly identify and adopt locally effective mutation strategies in the middle and later stages. Compared with the population-level adaptive or sinusoidal adaptive strategies adopted by baseline algorithms, it achieves a steeper decline in the convergence curve and thus outperforms other competitors.

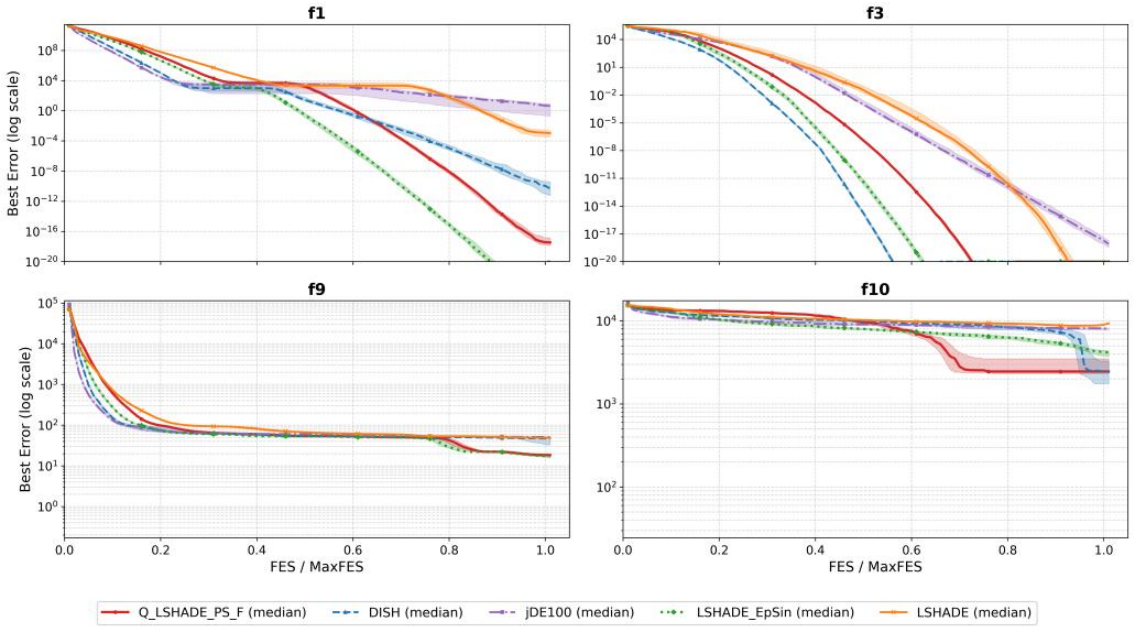

Figure 6. Convergence comparison on CEC2017 functions f1, f3, f9, f10 (50D)

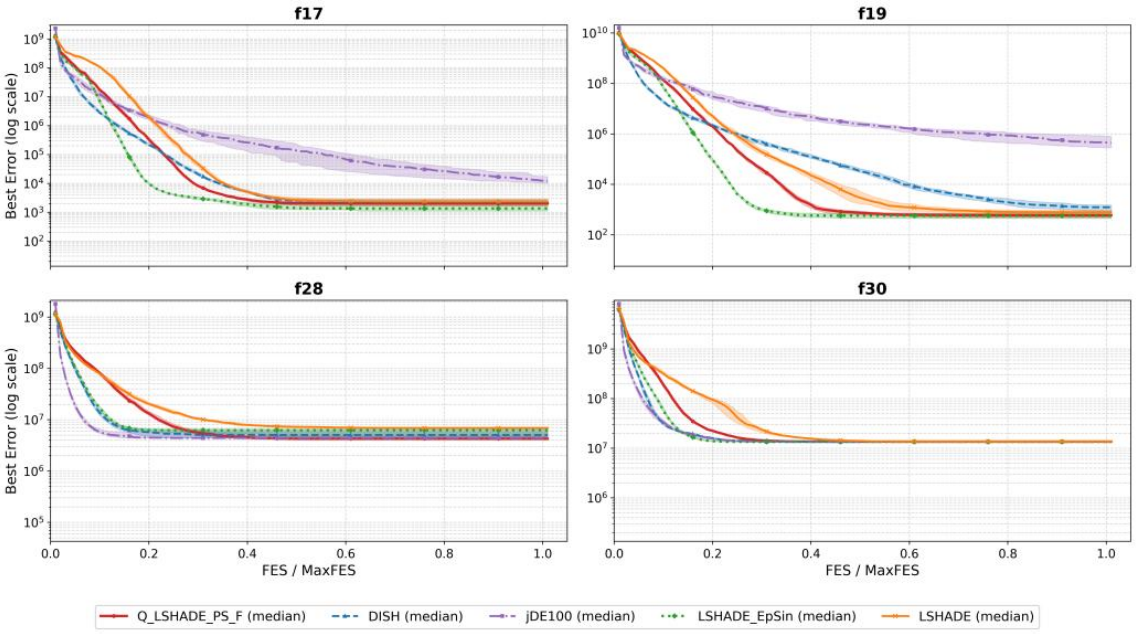

Figure 7. Convergence comparison on CEC2017 functions f17, f19, f28, f30 (50D)

In the late search stage, Q-LSHADE-PS not only maintains strong stability but also generally delivers higher final accuracy than most baseline algorithms, even outperforming the state-of-the-art counterpart LSHADE-EpSin. This indicates that the performance improvement does not merely stem from increased randomness, but from the synergistic effect of early exploration accumulation and efficient exploitation in the middle and later optimization stages.

5. CONCLUSION

This paper presented Q-LSHADE-PS, a hybrid differential evolution algorithm that integrates individual-level Q-learning with the robust L-SHADE framework. The method addresses premature convergence and diversity loss by allowing each individual to learn and select suitable

mutation/crossover operators based on its own stagnation state and the current search phase, while preserving L-SHADE's historical parameter adaptation, archive, and LPSR mechanisms.

Extensive experiments on the CEC 2014 and CEC 2017 benchmark suites demonstrated that Q-LSHADE-PS achieves superior or highly competitive performance compared with several strong DE baselines (L-SHADE, LSHADE-EpSin, DISH, and jDE100), and the improvements were validated by non-parametric statistical tests. Ablation studies further confirmed that both per-individual Q-learning and the LPSR-aware Q-table decay are key contributors to the observed gains, especially in high-dimensional settings. The additional computational overhead was minor, preserving practical efficiency.

Overall, Q-LSHADE-PS provides an effective and practical template for coupling reinforcement learning with modern adaptive DE components. Future work will explore extensions to multi-objective and constrained optimization, as well as more expressive state representations and transfer mechanisms to further improve generalization across problem classes.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support from Department of Education of Liaoning Province - Basic Scientific Research Project (LJKMZ20220916): Research on the IPPS Problem with Parallel Batch Machines and Time Constraints Based on Key Points/Chains.

REFERENCES

[1] Storn R, Price K. Differential evolution a simple and efficient heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997, 11(4): 341-359.

[2] Tanabe R, Fukunaga A. Success-history based parameter adaptation for differential evolution [C]. IEEE Congress on Evolutionary Computation, Cancun, 2013: 71-78.

[3] Piotrowski A P. LSHADE optimization algorithms with population-wide inertia [J]. Information Sciences, 2018, 468: 117-141.

[4] Luo W, Yu X, Wei Y. Solving combined economic and emission dispatch problems using reinforcement learning-based adaptive differential evolution algorithm [J]. Engineering Applications of Artificial Intelligence, 2023, 126: 107002.

[5] Durgut R, Aydin M E, Atli I. Adaptive operator selection with reinforcement learning [J]. Information Sciences, 2021, 581: 773-790.

[6] Padakandla S, Prabuchandran K J, Bhatnagar S. Reinforcement learning in non-stationary environments [J]. Applied Intelligence, 2020, 50(11): 3591-3606.

[7] Giannopoulos P G, Malamas V, Dasaklis T K. Integration of evolutionary algorithms and machine learning techniques in routing-related problems: A review [C]. Panhellenic Conference on Informatics, 2025: 237-243.

[8] Brest J, Maucec M S, Boskovic B. Single objective real-parameter optimization: Algorithm jSO [C]. IEEE Congress on Evolutionary Computation, 2017: 1311-1318.

[9] Li Y, Han T, Wang X, Zhou H, Tang S, Huang C, Han B. MjSO: A modified differential evolution with a probability selection mechanism and a directed mutation strategy [J]. Swarm and Evolutionary Computation, 2023, 78: 101294.

[10] Stanovov V, Semenkin E. Success rate-based adaptive differential evolution L-SRTDE for CEC 2024 competition [C]. IEEE Congress on Evolutionary Computation, 2024: 1-8.

[11] Zhou B, Huang Y. An adaptive archive differential evolution with nonlinear population size reduction and selective pressure [J]. Information Sciences, 2024, 682: 121273.

[12] Qin A K, Suganthan P N, Huang V L. Differential evolution algorithm with strategy adaptation for global numerical optimization [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 398-417.

[13] Zhang J, Sanderson A C. JADE: Adaptive differential evolution with optional external archive [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 945-958.

[14] Lin X, Meng Z. An adaptative differential evolution with enhanced diversity and restart mechanism [J]. Expert Systems with Applications, 2024, 249: 123634.

[15] Auger A, Hansen N. A restart CMA evolution strategy with increasing population size [C]. IEEE Congress on Evolutionary Computation, 2005: 1769-1776.

[16] Price K V, Storn R M, Lampinen J A. Differential Evolution – A Practical Approach to Global Optimization [M]. Springer, 2005: 1-34.

[17] Fialho A, Ros R, Schoenauer M, Sebag M. Comparison-based adaptive strategy selection with bandits in differential evolution [C]. Parallel Problem Solving from Nature, 2010: 194-203.

[18] Matsushita H, Kinoshita W, Kurokawa H, Kousaka T. Nested-layer particle swarm optimization method for bifurcation point detection in non-autonomous systems [J]. Nonlinear Theory and Its Applications, IEICE, 2019, 10(3): 289-302.

[19] Li P, Hao J, Tang H, Fu X, Zheng Y, Tang K. Bridging evolutionary algorithms and reinforcement learning: A comprehensive survey on hybrid algorithms [J]. arXiv Preprint, 2024.

[20] Guo H, Ma S, Huang Z, Hu Z, Ma Z, Zhang X, Gong Y J. Reinforcement learning-based self-adaptive differential evolution through automated landscape feature learning [C]. Genetic and Evolutionary Computation Conference, 2025.

[21] Yu X, Xu P, Wang F, Wang X. Reinforcement learning-based differential evolution algorithm for constrained multi-objective optimization problems [J]. Engineering Applications of Artificial Intelligence, 2024, 131: 107817.

[22] Ding W, Qian M, Lu C, Yi J, Pu H, Luo J. Differential evolution with joint adaptation of mutation strategies and control parameters via distributed proximal policy optimization [J]. Tsinghua Science and Technology, 2026, 31(1): 101-124.

[23] Cao Z, Xu K, Jia H, Fu Y, Foh C H, Tian F. An autonomous differential evolution based on reinforcement learning for cooperative countermeasures of unmanned aerial vehicles [J]. Applied Soft Computing, 2025, 169: 112605 .