Electronic Research Archive

ERA, 33(9): 5536–5590.
DOI: 10.3934/era.2025248
Received: 17 July 2025
Revised: 30 August 2025
Accepted: 04 September 2025
Published: 16 September 2025

https://www.aimspress.com/journal/ERA

Research article

Reinforcement learning-guided Animated Oat Optimization Algorithm with dynamic niching for high-dimensional optimization problems

Jia-Lin Yang[1], Hao-Ran Sun[1], Chai-Rui Chen[1], Ruo-Bin Wang[1,*], Lin Xu[2], Jeng-Shyang Pan[3] and Shu-Chuan Chu[3]

[1] School of Artificial Intelligence and Computer Science, North China University of Technology, Beijing 100144, China
[2] School of Computer & Mathematical Sciences, University of Adelaide, Adelaide 5005, Australia
[3] School of Artificial Intelligence/School of Future Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China

* **Correspondence:** Email: wrb@ncut.edu.cn; Tel: +86-10-88803051.

Abstract: High-dimensional optimization problems face challenges from exponentially expanding search spaces and deceptive local optima resisting metaheuristics. In order to address these issues, a reinforcement learning-guided Animated Oat Optimization Algorithm with a dynamic niching strategy, called RLDN-AOO, is proposed in this research. RLDN-AOO offers the following major novelties: i) a mathematically formulated three-state dynamic niching mechanism that adaptively partitions the population, preserves diversity, and enhances the algorithm's ability to escape local optima, and ii) a reinforcement learning strategy selection mechanism is proposed to address the issue of the algorithm's inadequate dynamic adaptability. We compared it with state-of-the-art algorithms (CEC2017, Dim = 50, 100, 200, 500), including LSHADE-SPACMA, CMA-ES variants, and RL-based optimizers. In addition, we applied it in the optimization of BP neural networks. Experimental results showed that RLDN-AOO achieves competitive performance across most benchmarks and, in some cases, performs comparably to LSHADE-SPACMA variants. The source code of RLDN-AOO is openly accessible via https://github.com/robingit77/RLDN-AOO.

Keywords: Animated Oat Optimization Algorithm; reinforcement learning; dynamic niching; multi-armed bandit; high-dimensional optimization

1. Introduction

High-dimensional optimization (HDO) problems are very common in practical production and applications, such as robot path planning [1], reinforcement learning training acceleration [2], feature selection [3], and discontinuous maneuver trajectory prediction [4]. Beyond these applications, HDO also arises in the numerical simulation of complex physical phenomena, such as sub-diffusion equations on distorted meshes [5], where high-order compact or structure-preserving schemes have been developed to handle computational challenges. These examples further demonstrate the fundamental importance of efficient and scalable methods for high-dimensional problems.

Compared with low-dimensional problems, the intrinsic characteristics of HDO change greatly [6], which causes two core challenges: first, the search space increases exponentially with the number of decision variables [7], which makes optimization difficult. Second, the objective functions of high-dimensional problems are often deceptive; they contain multiple local optimal regions [8], and the objective values at these local optima may be close to, or even better than, the projection of the global optimum in certain subspaces.

Metaheuristic algorithms are a class of commonly used algorithms for solving complex optimization problems [9,10]. For example, the Secretary Bird Optimization Algorithm (SBOA) [11] can be applied to global optimization problems and other real-world engineering design problems. However, they face the curse of dimensionality and premature convergence problems in high-dimensional scenarios [7,12]. The currently leading Animated Oat Optimization algorithm (AOO) [13] outperforms the Particle Swarm Optimization (PSO) [14] by 20%–30% in the 30-dimensional CEC2022 benchmark test, but its accuracy is still significantly reduced in the space of more than 100 dimensions due to the loss of population diversity in the later stage. This is a common challenge faced by most metaheuristics [12].

In response to the above challenges, researchers have proposed a variety of improvement strategies, such as precise elimination and generation mechanisms (mLSHADE-SPACMA) [15] and a mechanism for determining the effectiveness of element-wise SNR estimation based on covariance matrix eigenvector direction (LRA-CMA-ES) [16], among others. Although these improvements have enhanced performance to a certain extent, they lack the ability to automatically adjust and adapt. Reinforcement learning (RL), with its agent–environment interaction framework that triggers strong reasoning ability [17], offers dynamic strategy adaptation. Y. Zhang et al. [18] incorporated RL modification factors and historical population transfer/feedback operators to develop RLNNA, an RL-enhanced neural network algorithm (NNA) for photovoltaic model parameter extraction. Wu et al. [19] proposed the RLTLBO algorithm, which integrates Q-learning into the teaching–learning-based optimization (TLBO) framework.

This study bridges these gaps by proposing a reinforcement learning-guided Animated Oat Optimization Algorithm with dynamic niching (RLDN-AOO):

• Reinforcement learning–driven dynamic strategy: As a reinforcement learning–based model, the multi-armed bandit is introduced to optimize the HDO problems. It explores and adjusts weights in real time, helping to overcome the high computational complexity found in traditional Q-learning and similar methods.

• Adaptive basic algorithm selection: AOO is used as the optimization framework, and its unique adaptive balance mechanism is used to alleviate the high-dimensional degradation problem.

• Dynamic niching enhancement: A three-state niche strategy is designed to maintain population

diversity while avoiding the parameter ossification problem of the traditional niching method.

The rest of this article is arranged as follows. Section 2 reviews reinforcement learning and the Animated Oat Optimization Algorithm, and identifies the corresponding research gaps. In Section 3, the RLDN-AOO algorithm is described in detail. In Section 4, the performance of the algorithm in CEC2017 high-dimensional experiments is shown and discussed. Section 5 provides a conclusion and future direction.

2. Related work

2.1. Reinforcement learning

Reinforcement learning, a key approach in machine learning, allows agents to discover how to map states to actions through continuous interaction with the environment, relying on trial and error to optimize long-term cumulative rewards [20]. Typical RL methods (such as Q-learning) have been successfully applied to improve metaheuristic algorithms. In their work, Ding et al. [21] presented a novel asynchronous reinforcement learning approach, termed APSO-BQSA, which combines the Sarsa algorithm with backward Q-learning and asynchronous particle swarm optimization. Similarly, Jiao et al. [22] proposed OVEA, an evolutionary framework for multi-objective optimization that incorporates adaptive operator selection and reference vectors, where Q-learning is employed to dynamically determine the optimal crossover operator based on offspring performance. However, these methods rely on complex state-action space modeling, which has defects such as large computational overhead and parameter tuning relying on experience.

The multi-armed bandit (MAB) problem [23] is a basic framework in the field of RL and online decision-making. It simplifies the trade-off between exploration and development into a serialized "rocker" selection problem, which is lighter than Q-learning. Classical MAB algorithms such as Upper Confidence Bound (UCB) [23] and Thompson Sampling [24] can achieve regret close to the theoretical lower limit through lightweight calculation.

2.2. Algorithm selection and enhancement

Choosing a suitable algorithm framework is a key point to solving HDO. Zhang et al. [25] proposed the merged biogeography-based optimization (EMBBO) algorithm and demonstrated its superior optimization efficiency against numerous state-of-the-art algorithms. Lyu et al. [26] improved the Marine Predators Algorithm (MPA) [27] with adaptive parameters and proposed the MMPA. Based on the Whale Optimization Algorithm (WOA) [28], Zhang et al. [29] proposed a hybrid WOA with gathering strategies (HWOAG), which performs well on high-dimensional optimized clustering datasets. Building upon the Differential Evolution (DE) [30] algorithm, Li et al. [31] presented a niching-based adaptive differential evolution (DADE) algorithm centered around diversity and utilized it for multimodal optimization problems.

AOO, a new plant-inspired optimization algorithm proposed by Wang et al. [13] in 2025, has unique biological inspiration and efficient optimization performance. Its inspiration comes from the adaptive propagation mechanism of the animated oat. An optimization framework with both exploration and exploitation capabilities is constructed by simulating the behavior of seeds in the natural environment.

Each seed has its own characteristics during dispersal, which are calculated as Eq (1):

$$\begin{cases} m = 0.5 \times \dfrac{r}{\dim} \\ L = N \times \dfrac{r}{\dim} \\ e = 0.5 \times \dfrac{r}{\dim} \\ c = 1 - (\dfrac{t}{T})^3 \end{cases} \tag{1}$$

In the equation, m, L, and e are static attributes of the seed, representing the seed's quality, the length of the main awn, and the eccentric rotation coefficient, respectively. These parameters abstract the seed's initial characteristics and consistently influence the particle's movement throughout the optimization process. The random number $\mathrm{r} \in [0,1]$ is cleverly introduced to introduce dispersion, while the dimension dim and the population size N assign distinct initial weights to particles across different dimensions and populations. The parameter c is a dynamic factor that adaptively adjusts the seed's step size based on the ratio of the current iteration t to the maximum number of iterations T.

The exploration stage simulates the process of random seed dispersal through the medium of wind, water, and animals, and is explored using random perturbations. The specific formulas are as follows:

$$W = \frac{c}{\pi} \times (2 \times r_{\dim} - 1) \otimes UB \tag{2}$$

$$\begin{cases} X_{t+1}(i) = \dfrac{1}{N} \times \sum_{i=1}^{N} X_t(i) + W, & if \ \ \mathrm{mod}(i, N/10) = 0, \\ X_{t+1}(i) = X_{best} + W, & if \ \ \mathrm{mod}(i, N/10) = 1, \\ X_{t+1}(i) = X_t(i) + W, & else. \end{cases} \tag{3}$$

The most critical component in the above formulation is the step size W, which is adjusted using the dynamic factor c, the dim-dimensional random number $\mathrm{r}_{\dim}$, and the upper bound UB. This mechanism allows the step size to adapt to different dimensions and boundary ranges. As shown in Eq (3), the particle's position is updated with offsets based on its mean, optimal, and original positions.

The exploitation phase is divided into two cases, corresponding to movement strategies with and without obstacles, as follows:

Moisture-absorbing rolling (without obstacles): Based on the humidity-responsive deformation of the moisture-absorbing awns, local fine search is achieved by an eccentric rotation model. The specific formulas are defined as follows:

$$A = UB - \left| \frac{UB \times t \times \sin(2 \times \pi \times r)}{T} \right| \tag{4}$$

$$R = (m \times e + L^2) \times \frac{r_{\dim}(-A, A)}{\dim} \tag{5}$$

$$\text{Levy}(\dim) = 0.01 \times \frac{\mu \times \sigma}{|v|^{1/\beta}} \tag{6}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi \times \beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \tag{7}$$

$$X_t(i) = X_{best} + R + c \times Levy(\dim) \otimes X_{best} \tag{8}$$

Energy ejection (encountering obstacles): Simulate the energy storage and release of seeds when encountering obstacles and jump out of the local extremes through the projectile motion model. The formulas are defined as follows:

$$B = UB - \left| \frac{UB \times t \times \cos(2 \times \pi \times r)}{T} \right| \tag{9}$$

$$\begin{cases} k = 0.5 + 0.5 \times r \\ x = 3 \times \dfrac{r}{\dim} \\ \theta = \pi \times r \\ \alpha = \dfrac{1}{\pi} \times e^{\frac{r'}{T}} \end{cases} \tag{10}$$

$$J = \frac{2 \times k \times x^2 \times \sin(2\theta)}{mg} \times \frac{r_{\dim}(-B, B)}{\dim} \times (1 - \alpha) \tag{11}$$

$$X_t(i) = X_{best} + J + c \times Levy(\dim) \otimes X_{best} \tag{12}$$

Both movement strategies are based on the optimal position X_{best} in Eqs (8) and (12), with each being selected with a probability of $1/4$. Both use Levy flights as a strategy to escape local optima, as shown in Eqs (6) and (7). When employing Levy flights, typically only the parameter β needs to be adjusted (in the original AOO, β is fixed at 1.5). The difference between the two methods lies in the offset vectors R and J in Eqs (5) and (11). Here, $r_{\dim}(-A, A)$ denotes a random matrix of dimension dim, with each element sampled uniformly from the interval $[-A, A]$. In Eqs (5) and (9), A, B are key parameters used for building models, participating in subsequent rolling mechanisms, and other calculations. R models the offset generated when the seed moves by "rolling" without encountering obstacles, representing an internal drive. J, on the other hand, models the offset when the seed encounters obstacles and moves by a stored-energy catapult mechanism, representing an external drive. Both R and J are related to the seed's initial attributes. Equation (10) defines the correlation coefficients for seed ejection, where k, x, θ, α represent the elasticity coefficient, the change in awn length during energy storage, the ejection angle, and the air resistance coefficient, respectively.

The pseudocode for the step-by-step execution logic of animated oat optimization is as follows:

Algorithm 1 Step-by-step execution logic of Animated Oat Optimization

Input: *pop_size*: Number of oat seed agents; *dim*: Dimensionality of optimization problem
max_epoch: Total iteration limit; *curr_epoch*: Current iteration count
Output: *global_best_pos*: Optimal position discovered by AOO
global_best_fit: Corresponding fitness value

```
Initialize populations and generate fitness
while curr_epoch ≤ max_epoch do
    for agent_idx = 1 to pop_size do
        Calculate parameters according to Eq (1)
        if rand(0,1) > 0.5 then
            Execute Long-Range Dispersal (Eqs (2) and (3))
        else
            if rand(0,1) > 0.5 then
                Execute Moisture-Absorbing Rolling (Eqs (4) to (8))
            else
                Execute Obstacle-Evading Ejection (Eqs (9) to (12))
            end if
        end if
        Boundary Compliance & Fitness Update
        Local & Global Optima Tracking
    end for
    Epoch Termination & Progression
end while
```

It is found that AOO is not only competitive with other algorithms in low dimensions but also has lower degradation than many similar algorithms in high dimensions [13]. At the same time, to make up for the serious lack of diversity of AOO in the later stage, this paper introduces the niche strategy.

The niche strategy refers to the method of dividing the population into multiple sub-populations (niches) to enable different sub-populations to evolve in their respective "niches" to maintain population diversity. NCOA [32], DADE [31], and other algorithms use the niche strategy. Their approaches demonstrate superior performance in avoiding premature convergence compared to non-niching algorithms [31–33].

In addition to metaheuristic-based enhancements, recent research has increasingly applied machine learning–driven methods to optimization and engineering tasks. For instance, a decoupled physics-informed neural network (dPINN) is proposed to enforce initial and boundary conditions in dynamic PDEs with higher accuracy and efficiency [34]. Deep neural network (DNN) models have also been used to predict material behavior in deep drawing processes [35], quantify uncertainty in bio-inspired porous structures via Bayesian neural networks [36], and enhance dimensional accuracy prediction in flow forming through integration with finite element analysis [37]. Furthermore, hybrid approaches that couple DNNs with evolved metaheuristics, such as the EVARO algorithm, have shown promising results in structural damage detection for bridges [38]. These studies highlight the potential of ML-driven methods in engineering optimization, but they mainly focus on low-to-medium-dimensional problems and domain-specific applications. In contrast, our work targets reinforcement learning–driven niching strategies for high-dimensional optimization, which has not been sufficiently

explored in prior literature.

2.3. Research gaps

2.3.1. Gaps in existing approaches for HDO

Intelligent optimization algorithms: Most metaheuristics often suffer from the "curse of dimensionality", where population diversity decreases rapidly and search efficiency drops sharply. As the dimension increases, they face premature convergence, degraded solution quality, and prohibitive computational costs, limiting their scalability to large-scale problems.

ML-driven optimization: Although PINN, DNN, and other methods have achieved results in applications such as manufacturing, materials, and health monitoring, they are mostly focused on low/medium-dimensional problems, and the problem is specific and does not have universal HDO processing capabilities.

RL in metaheuristics: Existing methods (such as Q-learning and SARSA) are applied to operator selection, but there are problems such as complex modeling, large computational overhead, and being unfriendly to high-dimensional problems.

Niching strategies: Most of the existing niche methods are fixed or two-state mechanisms, which can easily lead to misjudgment due to the rough judgment and may fall into the local optimal region repeatedly, as they do not have a memory mechanism.

2.3.2. Gaps in AOO itself

While the AOO algorithm provides a promising bio-inspired framework, its performance in high-dimensional optimization scenarios can be further enhanced. Specifically:

It is necessary to further strategically maintain population diversity in the development stage: The particles of the AOO algorithm tend to converge quickly to the current optimal solution in the later iteration [e.g., Eqs (3), (8) and (12)]. In the later stage, particles will gather in large numbers, the diversity will decrease, and the tendency to converge to a local optimum will increase significantly in high-dimensional scenarios.

It is desirable to enhance adaptability in strategy switching: At present, the switching between exploration and development stages is usually based on a fixed threshold (such as $mod\left(t,\ N/10\right)$ in Eq (3)). This static switching method is difficult to effectively cope with the changing search environment in HDO problems. There is an urgent need for a more adaptable and agile adjustment mechanism to strengthen the algorithm's responsiveness to intricate optimization scenarios.

Further improvements can be made by adopting adaptive parameter settings: For example, the static setting of key parameters in Levy's flight step generation formula [such as β in Eq (6)] limits the ability of particles to break free from local optima or to identify and exploit potential regions, as a result of inadequate step size adaptability.

2.3.3. Summary of related work

In summary, while machine learning–driven optimization, RL-enhanced metaheuristics, and niching-based algorithms have each demonstrated valuable contributions, they remain limited in handling high-dimensional optimization problems due to either domain-specific applicability,

excessive computational overhead, or lack of adaptive population diversity maintenance. Meanwhile, the baseline AOO algorithm shows promise but suffers from diversity loss and rigid parameter settings when scaled to high dimensions. Therefore, to fill these gaps, we propose RLDN-AOO, which integrates a lightweight multi-armed bandit reinforcement learning framework with a three-state dynamic niching strategy, providing a novel and scalable solution for high-dimensional optimization.

3. General description of RLDN-AOO algorithm

3.1. Overall framework of the algorithm

An overview of the proposed RLDN-AOO algorithm is provided in this section. The algorithm incorporates a dynamic niching (DN) strategy, which divides all populations into multiple niches that are iteratively updated independently. Each niche has its own rise and fall cycle, and different niches will have different focus behaviors in different periods. In addition, the algorithm also introduces the multi-armed bandit in reinforcement learning as the exclusive location update decision center for each niche. This is able to choose the most appropriate location update method for the current situation based on experience, which changes the traditional strategy of artificially setting the probability and improves the adaptive nature of the algorithm.

The structure of each niche is shown in Figure 1, where each dotted circle represents a niche, and each niche possesses its own MAB decision hub. The red star represents the subpopulation optimum of the current niche, the nearby small circles are ordinary particles, and the red circles with forks indicate the taboo regions, i.e., the labeled local optimum regions.

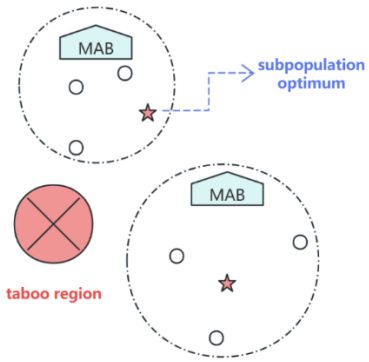


Figure 1. The structure of a single niche.

The flowchart of the proposed algorithm is shown in Figure 2. After entering the main loop, it traverses all the niches and first determines whether the niche is trapped in a local optimum. If it is, it reinitializes the niche as a way of forcing it to escape from the local optimum position. If it is not, then the position update under the multi-armed bandit machine strategy will be performed. Finally, it will judge whether the niche should be reallocated based on the stagnation of the global optimum, to redistribute the resources of each niche and increase the likelihood of jumping out.

In addition to these two main mechanisms, adaptive parameter γ, multi-stage Levy flight, multi-strategy boundary handling, and other mechanisms are also proposed, which will be introduced below.

The introduction to the niching and multi-armed bandit machine strategies will be given in subsections 3.2 and 3.3, respectively.

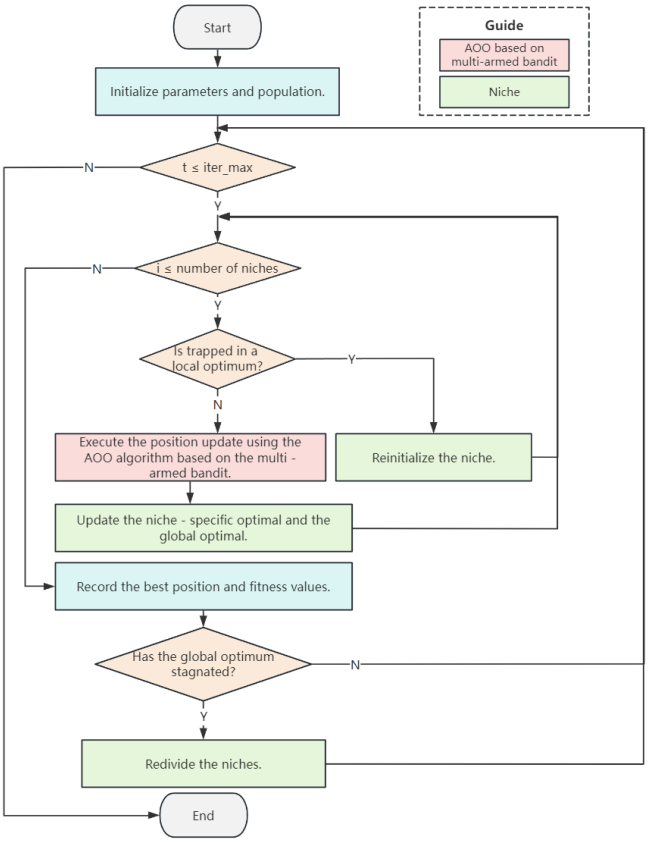


Figure 2. The flowchart of RLDN-AOO.

3.2. Dynamic niching strategy

In the proposed algorithm, a dynamic niching strategy inspired by Li et al. [31] is proposed. Each miniature habitat has three states—healthy, debilitated, and extinct—corresponding to not falling into, possibly falling into, and definitely falling into a local optimum. There are different strategies for each stage, which will be explained in the following subsections.

Based on this mechanism, our niches would have acquired dynamic adaptability. Here, *dynamic adaptability* refers to the algorithm's ability to adjust the survival state of each niche in response to fitness stagnation and diversity collapse.

3.2.1. Healthy state

This paper argues that a healthy niche should have at least one of the following characteristics:

1) $\boldsymbol{D}(\mathcal{N}) \geq \rho$: The diversity of this niche $\boldsymbol{D}(\mathcal{N})$ is not less than the threshold value ρ (in this work, $\rho = 0.05$).

2) $\left| \boldsymbol{f}_{\mathcal{N}}^{\star} - \boldsymbol{f}^{\star} \right| > \boldsymbol{A}_{\boldsymbol{acc}}$: The difference between the subgroup optimum $\boldsymbol{f}_{\mathcal{N}}^{\star}$ and the global optimum $\boldsymbol{f}^{\star}$ for this niche does not exceed the accuracy thresholds A_{acc} (in this work, $A_{acc} = 10$).

The calculation formula for $\boldsymbol{D}(\mathcal{N})$ is as follows:

$$\begin{cases} \boldsymbol{D}(\mathcal{N}) = \dfrac{1}{\boldsymbol{A}(|\mathcal{N}|-1)} \sum\limits_{\boldsymbol{i}\in\mathcal{N}} \left\| \boldsymbol{x}_{\boldsymbol{i}} - \boldsymbol{x}_{\mathcal{N}}^{\star} \right\| \\ \boldsymbol{A} = \|\boldsymbol{u} - \boldsymbol{l}\| \end{cases} \tag{13}$$

where $\mathcal{N} \subseteq \{1,\ldots,\boldsymbol{N}\}$ means the index of the current niche (in this work, $|\mathcal{N}| \geq 3$), $\boldsymbol{l}, \boldsymbol{u} \in \mathbb{R}^{\boldsymbol{d}}$ are lower/upper bounds, and $\left\| \boldsymbol{x}_{\boldsymbol{i}} - \boldsymbol{x}_{\mathcal{N}}^{\star} \right\|$ represents the Euclidean distance between the optimal niche $\boldsymbol{x}_{\mathcal{N}}^{\star}$ and the $\boldsymbol{i}_{\boldsymbol{th}}$ particle $\boldsymbol{x}_{\boldsymbol{i}}$. The Euclidean distance is formulated as Eq (14):

$$d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\| = \sqrt{\sum_{k=1}^{d} (x_k - y_k)^2} \tag{14}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_d)$ are different position vectors of particles, and x_k indicates the value of the k dimension of $\boldsymbol{x}$. $\boldsymbol{y}$ is the same.

The best individual and score are calculated as:

$$\begin{cases} \mathbf{x}_{\mathcal{N}}^{\star} = \arg\min_{\mathbf{x}_{\boldsymbol{i}}, \boldsymbol{i}\in\mathcal{N}} \boldsymbol{f}(\mathbf{x}_{\boldsymbol{i}}) \\ \boldsymbol{f}_{\mathcal{N}}^{\star} = \boldsymbol{f}(\mathbf{x}_{\mathcal{N}}^{\star}) \end{cases} \tag{15}$$

where $\mathbf{x}_{\mathcal{N}}^{\star}$ and $\boldsymbol{f}_{\mathcal{N}}^{\star}$ are the best individual and best score in a certain niche, and $\boldsymbol{f}(\mathbf{x}_{\boldsymbol{i}})$ indicates the fitness of $\mathbf{x}_{\boldsymbol{i}}$.

When feature 1) is satisfied, it indicates that the current niche particles are not concentrated near the subpopulation optimum, have high diversity, and are not trapped in a local optimum.

When feature 2) is satisfied, it means that the current niche is the niche where the global optimum is located, or it is very close to the global optimum, which means that there is a lot of potential, and it does not get stuck in a local optimum.

When the niche is in a healthy state, the $\beta_{\mathcal{N}}$ that controls Levy's flight is kept at 1.6, and the boundary is handled using the strategy of attached boundaries, reflecting a more conservative approach.

3.2.2. Debilitated state

In this paper, when not all the above conditions are satisfied at the same time, it is said to enter a debilitated state. That means:

$$\begin{cases} \boldsymbol{D}(\mathcal{N}) < \rho \\ \left| \boldsymbol{f}_{\mathcal{N}}^{\star} - \boldsymbol{f}^{\star} \right| \leq \boldsymbol{A}_{\boldsymbol{acc}} \end{cases} \tag{16}$$

At this case, it means that the niche is already highly concentrated and has insufficient potential to be separated from the global optimum; it can enter the local optimum, but that is not determined. Thus, we record the number of sustained debilitated times $\boldsymbol{g}_{\mathcal{N}}$. If the above two equations hold continuously, then $\boldsymbol{g}_{\mathcal{N}} \leftarrow \boldsymbol{g}_{\mathcal{N}} + 1$.

When the niche is in a debilitated state, it will affect the value of $\beta_{\mathcal{N}}$ in Levy's flight through $\boldsymbol{g}_{\mathcal{N}}$, increasing the possibility of the niche to escape from the local optimum. See Subsection 3.3.1 for details of the influence. See Eq (17).

$$\beta_{\mathcal{N}} = \beta_{\min} + (\beta_{\max} - \beta_{\min}) \cos\left(\frac{\pi}{2} \cdot \frac{\boldsymbol{g}_{\mathcal{N}}}{\boldsymbol{G}} \right) \tag{17}$$

where $\beta_{\min}=1.2$, $\beta_{\max}=1.6$, and threshold $G=5$ in this work.

Meanwhile, if the particle whose position is updated in this state crosses the boundary, the following boundary processing strategy proposed in [39] is used to get rid of the adhesion of the crossing particle to the boundary and instead look for opportunities in the high potential region. This strategy will be described in detail in Subsection 3.3.6.

3.2.3. Extinct state

When the number of continuous weakening exceeds the debilitated threshold G (in this work, $\boldsymbol{G}=5$), that means $\boldsymbol{g}_{\mathcal{N}}>\boldsymbol{G}$.

In this case, the current niche is considered to have fallen into a local optimum and enters a state of extinction; then, it is necessary to record the current region as a taboo region, i.e., a locally optimal region, and subsequently reinitialize the niche.

The flowchart for marking taboo areas is shown in Figure 3.

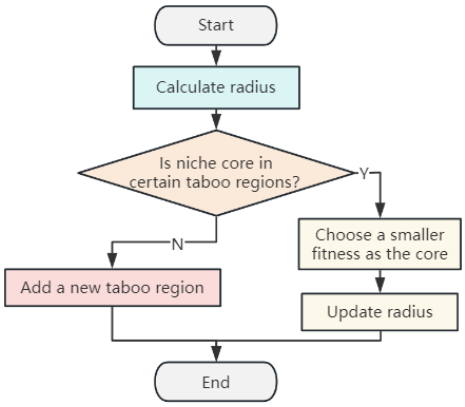


Figure 3. Flowchart of marking the taboo area.

Each taboo region consists of a core point c and a radius r, where the core point is the subpopulation optimum of the niche that has fallen into a locally optimal position, and the radius is the Euclidean distance between the subpopulation optimum of the current niche and its nearest particle. That is:

$$\boldsymbol{r}=\min_{\boldsymbol{i}\in\mathcal{N},\boldsymbol{x_i}\neq\boldsymbol{x}^{\star}_{\mathcal{N}}}\left\|\ \boldsymbol{x_i}-\boldsymbol{x}^{\star}_{\mathcal{N}}\right\| \tag{18}$$

where the $\left\|\ \mathbf{x}_{\boldsymbol{i}}-\mathbf{x}^{\star}_{\mathcal{N}}\right\|$ represents the Euclidean distance between the optimal niche and the $\boldsymbol{i}_{\boldsymbol{th}}$ particle.

When the location of the proposed added taboo region c is found to be within the range of a taboo region, the taboo region is updated. The fitness of the two cores is first compared, and the smaller one is chosen as the new core. The radius is then updated to $d+r$, where d is the Euclidean distance between the two cores.

For reinitialization, this paper uses a reverse learning (OBL) strategy with the following pseudo-code:

Algorithm 2 Pseudo-code for re-initializing niches

Input: *NP*: niche population size; *lb*: lower bound; *ub*: upper bound; R_{taboo}: taboo regions.

Output: the positions and fitness values of individuals in the current niche after reinitializing.

```
Calculate the dimension and population size according to the input population information.
for 1: NP
    Set in_taboo = True
    while in_taboo == True:
        Generate two individuals according to Eq (19);
        Calculate the fitness of both;
        Choose a smaller one as the X_target;
        if ~is_in_taboo_region(X_target, R_taboo)
            Set in_taboo = False;
            Record target position and fitness;
        end if
    end while
end for
```

First, a random position is generated, and then the opposing position of that position in the whole space is generated. It is calculated as Eq (19):

$$\begin{cases} X = lb + (ub - lb) \cdot r \\ X' = (ub + lb) - X \end{cases} \tag{19}$$

where X denotes a random location of $1 \times dim$, r denotes a uniformly distributed random vector on the interval $[0,1]$, and X' denotes the reverse individual.

The fitness of the two is then calculated, and the location with the smaller fitness is chosen as the new location. It is then judged whether the position is in the recorded taboo region; if not, it is recorded, and vice versa, the particle is regenerated, and so on. In this way, a new niche is obtained, and finally, the mechanism of forcing the locally optimal niche to jump out is realized by choosing the particle with the smallest fitness of this niche as the subpopulation optimum.

The diagram in Figure 4 shows the progression of a niche from a healthy state into a debilitated and finally an extinct state. In the debilitated stage, the niche particles are very close to the local optimal position, and the diversity is maintained at an extremely low level. In the extinct stage, the local optimal position of the recorded niche is near the taboo area, and the niche particles are reinitialized and distributed to the position of the non-taboo area.

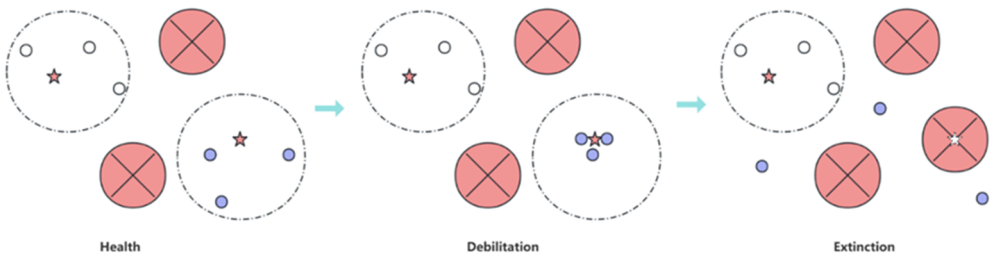


Figure 4. Niche evolution diagram.

3.2.4. Summary of the three-state dynamic niching

a) State transition table

The transition conditions and behaviors of each state in the niche are shown in Table 1.

Table 1. Three-state dynamic niche transition.

State	Condition	Threshold	Actions
Healthy	$\begin{cases} \boldsymbol{D}(\mathcal{N}) \geq \rho \\ \left\| \boldsymbol{f}_{\mathcal{N}}^{\star} - \boldsymbol{f}^{\star} \right\| > \boldsymbol{A}_{acc} \end{cases}$	$\rho = 0.05$ $\boldsymbol{A}_{acc} = 10$	$\begin{cases} \boldsymbol{g}_{\mathcal{N}} = 0 \\ \beta_{\mathcal{N}} = 1.6 \end{cases}$
Debilitated	$\begin{cases} \boldsymbol{D}(\mathcal{N}) < \rho \\ \left\| \boldsymbol{f}_{\mathcal{N}}^{\star} - \boldsymbol{f}^{\star} \right\| \leq \boldsymbol{A}_{acc} \\ 0 < \boldsymbol{g}_{\mathcal{N}} \leq \boldsymbol{G} \end{cases}$	$\rho = 0.05$ $\boldsymbol{A}_{acc} = 10$ $\boldsymbol{G} = 5$	$\begin{cases} \boldsymbol{g}_{\mathcal{N}} \leftarrow \boldsymbol{g}_{\mathcal{N}} + 1 \\ \beta_{\mathcal{N}} = \beta_{\min} + (\beta_{\max} - \beta_{\min}) \cos\left(\frac{\pi}{2} \cdot \frac{\boldsymbol{g}_{\mathcal{N}}}{\boldsymbol{G}} \right) \\ \textbf{\textit{Boundary handling variants}} \end{cases}$
Extinct	$\boldsymbol{g}_{\mathcal{N}} > \boldsymbol{G}$	$\boldsymbol{G} = 5$	$\boldsymbol{Re-initializing\ niche}$

b) Pseudocode of the three-state dynamic niching

Below, we present the proposed niche framework in pseudocode; modifying the framework is also the general process of the entire project.

Algorithm 3 Pseudocode of the three-state dynamic niching

Input: $P = \{x_1, \ldots, x_N\}$: population; *lb,ub*: lower /upper bound; ρ: diversity threshold; G: extinction patience; A_{acc}: near-optimal threshold; β_{min}, β_{max}: β bounds; $S_{threshold}$: global optimal stagnation threshold

Output: global best position x^* and its fitness f^*.

Initialize: partition P into niches $\{N_1, \ldots, N_k\}$ (via adaptive niche division in section 3.2.5);
    for each N_k: set gap counter $g_k \leftarrow 0$.
for t = 1 … T **do**
    for each niche N_k **do**
        Calculate niche parameters like x_k^*, D_k, f_k^* according to Eqs (13), (15) and (17).
        if ($D_k < \rho$) and ($|f_k^* - f^*| > A_{acc}$) then
            $g_k \leftarrow g_k + 1$
        else
            $g_k \leftarrow g_k + 1$
        end if

        # --- State decision ---
        if $g_k = 0$ **then**

Continued on next page

state←Healthy
else if $g_k \leq G$ **then**
state←Debilitated
Else
state←Extinct
end if

# --- state-specific actions ---
if state $\in$ {Healthy, Debilitated} **then**
Dynamically adjust Levy parameter β_k according to Eq (17).

# produce new candidates for members of N_k
for each $x_i \in N_k$ **do**
Calculate X_{target} according to Eq (28). # from Section 3.3.1
Plevy←Levy(β_k) # heavy-tailed step
W ← bounded stochastic step (problem-dependent)

strategy chosen by MAB (UCB) from Section 3.3
$x_{new} \leftarrow GenerateCandidate(x_i, x_{target}, W, Plevy, UCB(N_k))$

# boundary handling (detail in Section 3.3.6)
if state = Healthy **then**
$x_{new} \leftarrow BoundaryCheck(x_{new}, l, u)$
else
$x_{new} \leftarrow BoundaryCheckFollower(x_{new}, x_i, l, u, x_k^*)$
end if

if $f(x_{new}) < f(x_i)$ **then**
replace $x_i \leftarrow x_{new}$ and update x_k^*, f_k^*.
end if
end for

else # state = Extinct
Calculate r according to Eq (18).
$AddOrMergeTabooBall(center = x_k^*, radius = r, score = f_k^*)$

# Scatter niche with OBL while avoiding taboo regions
for each position in N_k **do**
Repeat:
$x_0 \leftarrow Uniform(l, u)$
$x_{opp} \leftarrow (l + u) - x_0$ # opposition-based candidate

Continued on next page

$x_{new} \leftarrow argmin\{ f(x_0), f(x_{opp}) \}$
until $x_{new} \notin any\ TabooBall$
replace $x_i \leftarrow x_{new}$
end for

$g_k \leftarrow 0$ # reset after reinitialization
end if # different states
end for # for each niche N_k

# optional: if no improvement for S iterations, re-partition niches
if $NoImprovementFor(S_{threshold})$ **then**
Update niches $\{N_1, \ldots, N_k\}$ via adaptive niche division (in Section 3.2.5);
end if
end for # for t=1:T

3.2.5. Adaptive niche division

In addition to the division of niches at initialization, all niches are also redivided when the global search stalls. Each niche division will redefine the members and particle states of each niche, facilitating interactions between particles and facilitating the breaking of stalemates.

First, we consider the global search to enter a stalled state when $S_{cnt} > S_{threshold}$, where S_{cnt} denotes the global optimal number of stalls, and $S_{threshold}$ denotes the stalling threshold. When the above conditions are satisfied, we reclassify all niches. The flowchart of reclassification is shown in Figure 5.

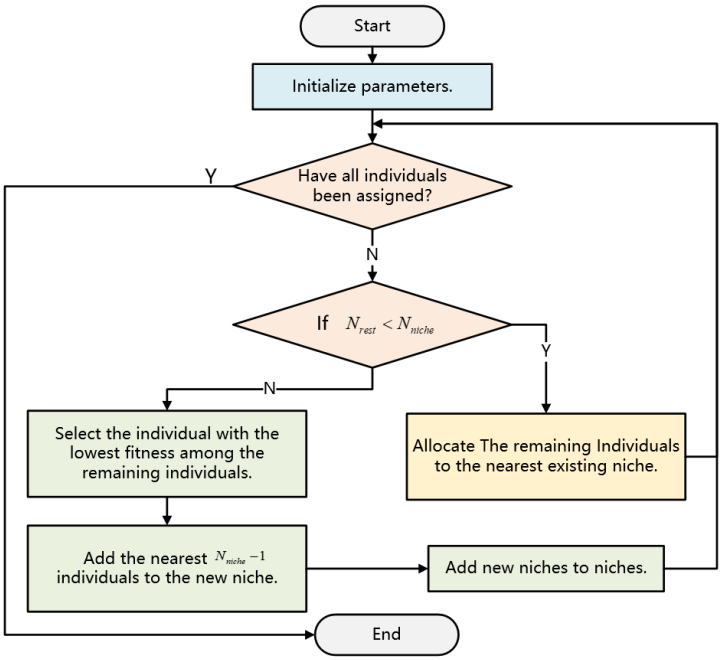


Figure 5. Flowchart of niche rezoning.

During the initialization of the parameters, we compute the dynamic parameter N_{niche}, indicating the current minimum number of particles per niche, which is computed as in Eq (20):

$$N_{niche} = S_{min} + \left(S_{max} - S_{min}\right) \times \frac{t}{T} \tag{20}$$

S_{min} and S_{max} indicate the minimum and maximum number of particles in a single niche, respectively. N_{niche} denotes the number of particles in the niche after rounding. Through the above formula, we can control the increase of N_{niche} with the increase of iteration number, and at the same time, make the number of niches decrease gradually to simulate the survival of the niche. In this paper, the settings of $S_{min} = 3$, $S_{max} = N/2$, and N denote the total population number. According to the above settings, it can achieve multi-niche exploration in the early stage of the population and stable dual niche development in the later stage.

In the course of division, the remaining unassigned individuals will first be ranked in terms of fitness, and the best-fitted among them will be selected as the core of the new niche. The distance of the remaining individuals from this core will be calculated using the Euclidean distance [Eq (14)], and the nearest $N_{niche} - 1$ individuals will be selected as members. x denotes unassigned individuals, and y denotes the core of the new niche.

If the number of remaining particles is found to be less than N_{niche}, then the remaining particles are assigned to the niche. The same Euclidean distance formula is used, except that y represents the center of the proposed niche, which is the average of all the individuals in that niche in each dimension.

An example plot of the population distribution when $N = 30, N_{niche} = 10, \dim = 2$ is shown in Figure 6.

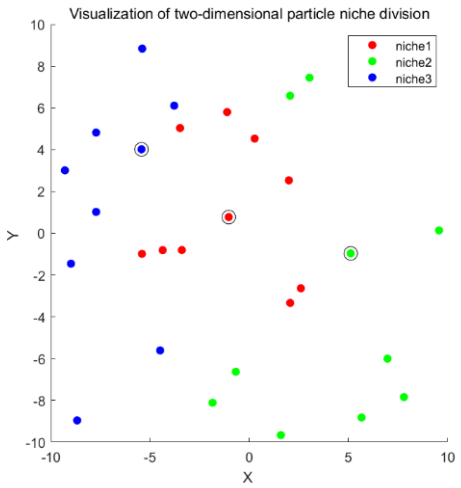


Figure 6. Niche division demonstration diagram.

The distribution of particles for each niche added is shown, where the particles with circles on the outside are the core of the current niche.

3.3. Reinforcement learning framework

In the classic reinforcement learning framework, an agent needs to have state/action/reward. We

did not use RL such as Q-learning/SARSA/DQN, which requires explicit modeling of states, but instead chose the MAB framework, which is essentially a stateless special case of RL, retaining only action space and reward updates. Therefore, it can significantly reduce computational costs and is more suitable for high-dimensional optimization scenarios. This section will introduce the framework of this research design from three aspects: action space, reward function, and learning algorithm.

3.3.1. State representation

In classical multi-armed bandit frameworks, no explicit state is required, which limits their adaptability in dynamic optimization. To address this, we adopt a lightweight contextual bandit view, where each niche $\mathcal{N}$ at iteration t is described by a compact state vector:

$$\boldsymbol{s}_t = (\boldsymbol{D}_{\boldsymbol{norm}}(\mathcal{N}), \boldsymbol{g}_{\mathcal{N}}) \tag{21}$$

where $\boldsymbol{D}_{\boldsymbol{norm}}(\mathcal{N})$ denotes the normalized diversity of the niche, and $\boldsymbol{g}_{\mathcal{N}}$ is a stagnation counter tracking whether the niche is trapped in a local optimum. This state representation introduces minimal overhead while allowing the bandit model to adaptively allocate strategies according to the exploration–exploitation dynamics of each niche.

3.3.2. Action space

In our framework, the reinforcement learning agent does not introduce new operators, but instead integrates directly with the improved AOO mechanism by treating its four update operators as the action space of the multi-armed bandit. In this way, the RL component adaptively selects which improved AOO operator to apply to each individual niche, based on the reward feedback.

The action space consists of four candidate search strategies: $\boldsymbol{a} \in \{1, 2, 3, 4\}$.

For the m_{th} niche, if its $state \in \{Healthy, \cdot\ Debilitated\}$, we will update the position of its particles in one of the following four ways. The action space is described in Table 2.

Table 2. Action space and its characteristics, used by the proposed algorithm.

Action	AOO operator		Features
$a = 1$	$x_{new} = \frac{1}{N} \times \sum_{i=1}^{N} x_i + W$	(22)	Mean-based update
$a = 2$	$x_{new} = x_{target} + W$	(23)	Leader-based update
$a = 3$	$x_{new} = x_i + W$	(24)	Neighborhood perturbation
$a = 4$	$\begin{cases} x_{new} = x_{target} + R + c \times Levy(\dim, \beta_m) \otimes x_{target}, & \text{r<0.5,} \\ x_{new} = x_{target} + J + c \times Levy(\dim, \beta_m) \otimes x_{target}, & else. \end{cases}$	(25)	Levy-flight–enhanced update

Where N represents the current number of particles in the niche, and the calculation formulas for W, R, c, J, β_m are shown in Eqs (2), (4)–(11) and (17), where x_{target} is a mixed guiding particle

used to macroscopically coordinate the direction of particle aggregation. The specific calculation method is as follows.

First, a dynamic parameter γ is proposed in this work to regulate the transition of niche core particles, aiming to effectively balance exploration and exploitation across different phases. The formula is shown in Eq (26).

$$\gamma = \frac{1}{1+\exp\left(k \cdot w \cdot \left(\frac{t}{T} - \frac{1}{2}\right)\right)} \tag{26}$$

where k and w are used to control the steepness and center offset of the curve at γ. The curves of this formula for different k and w are shown in Figure 7.

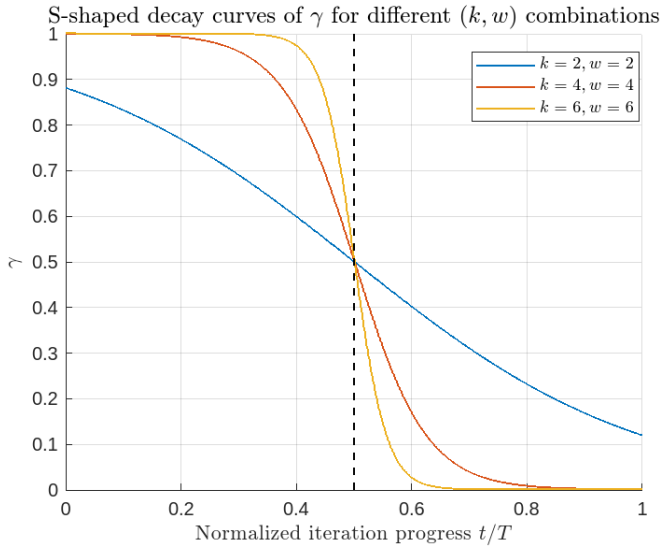


Figure 7. The γ curves under different k and w values.

The change curve of γ is an S curve. When $k = 4$, $w = 4$, γ approaches 1 in the early stage, transitions rapidly in the middle stage, and slowly approaches 0 in the late stage. At the same time, we design the formula for calculating the core position of the niche as Eqs (27) and (28):

$$\gamma' = w + (1-w) \times \gamma \tag{27}$$

$$x_{\text{target}} = \gamma' \times x_k^* + (1-\gamma') \times x^* \tag{28}$$

where $w = 0.1$ (default) is used to regulate the minimum value approaching γ. Through the adjustment of γ, x_{target} can be adjusted to be similar to the current niche optimum x_k^* in the earlier phase and rapidly approach the global optimum x^* in the middle and late stages, but always differentiate from the x^*, which is conducive to maintaining the diversity and balancing the pre-exploration and post-exploitation at the macro level.

3.3.3. Reward function

After applying action a at iteration t, a reward R_t is assigned by combining convergence

improvement and population diversity:

$$\boldsymbol{R}_t = \alpha_t \cdot \boldsymbol{f}_{norm} + (1-\alpha_t) \cdot \boldsymbol{D}_{norm} \tag{29}$$

where $\boldsymbol{f}_{norm} = \frac{\boldsymbol{f}_{\max} - \boldsymbol{f}_{\text{new}}}{\boldsymbol{f}_{\max} - \boldsymbol{f}_{\text{best}}}$ denotes normalized fitness improvement, $\boldsymbol{D}_{norm}$ is normalized niche diversity, and $\alpha_t = \boldsymbol{w}_1 + \boldsymbol{w}_2 \cdot \frac{t}{T_{\max}}$ is a time-dependent weight shifting emphasis from exploration to exploitation, set $w_1 = 0.6, w_2 = 0.3$ (default).

In this paper, the mean of the Euclidean distance between individuals of the niche as a percentage of the diagonal distance in the solution space is used as the normalization method. The formulas for calculating the niche-normalized diversity $\boldsymbol{D}_{norm}$ are Eqs (30)–(32).

$$D_o = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \left\| x_i - x_j \right\| \tag{30}$$

$$D_{\max} = \| \mathbf{ub} - \mathbf{lb} \| = \sqrt{\sum_{d=1}^{\dim} (ub_d - lb_d)^2} \tag{31}$$

$$D_{norm} = \frac{D_o}{D_{\max}} \cdots\cdots\cdots \tag{32}$$

where the calculation of $\left\| x_i - x_j \right\|$, as shown in Eq (14), represents the Euclidean distance between two different particles. D_o denotes the mean Euclidean distance between individuals, n is the size of the niche, and D_{max} represents the diagonal distance in solution space.

3.3.4. Learning algorithm

Incremental update rule: The i_{th} niche maintains action-value estimates $Q(a)$ and selection counts $N(a)$. After each reward, the update rule is:

$$\begin{cases} \boldsymbol{Q}_{t+1}(\boldsymbol{a}) = \boldsymbol{Q}_t(\boldsymbol{a}) + \dfrac{1}{\boldsymbol{N}_i(\boldsymbol{a})} \left(\boldsymbol{R}_t - \boldsymbol{Q}_t(\boldsymbol{a}) \right) \\ \boldsymbol{N}_i(\boldsymbol{a}) \leftarrow \boldsymbol{N}_i(\boldsymbol{a}) + 1 \end{cases} \tag{33}$$

Action selection: The action selection is guided by the UCB criterion:

$$\boldsymbol{UCB}_t(\boldsymbol{a}) = \boldsymbol{Q}_t(\boldsymbol{a}) + \boldsymbol{c} \cdot \sqrt{\frac{\ln \boldsymbol{t}}{\boldsymbol{N}_i(\boldsymbol{a})}} \tag{34}$$

where c is a tunable constant controlling the degree of exploration. The action with the maximum $UCB_t(a)$ is selected at each iteration. That is, $\boldsymbol{a}^* = \arg\max_{\boldsymbol{a}} \left(\boldsymbol{UCB}_t(\boldsymbol{a}) \right)$.

3.3.5. Pseudocode of RL strategy selection

For each execution location update niche ($state \in \{Healthy, Debilitated\}$), its pseudocode is as follows:

Algorithm 4 Pseudo-code of the AOO based on the multi-armed bandit

Input: $P = \{x_1, \ldots, x_N\}$: population; *Dim*: problem dimension;
T: maximum number of iterations; *c*: exploration coefficient in UCB algorithm;
lb,ub: lower /upper bound;

Output: the positions and fitness values of individuals in the current niche after update.

```
Initialize: niches and related parameters (described in Section 3.2);
        Q(a) = 0 , N(a) = 0  for all actions  a
for  t = 1:T  do
    for each niche  i  do
        Evaluate niche status according to Section 3.2
        if  state ∈ {Healthy,· Debilitated}  then
            Compute  UCB_t(a)  for all actions according to Eq (34)
            Select action  a* = argmax UCB_t(a)
            switch  a*
                case 1: Mean-based update as defined by Eq (22);
                case 2: Leader-based update as defined by Eq (23);
                case 3: Neighborhood perturbation as defined by Eq (24);
                case 4: Levy-flight–enhanced update as defined by Eq (25);
            end switch
            if  state = Healthy
                Use Method for boundary handling;
            else
                Use Method for boundary handling;
            end if
            Calculate the normalized reward according to Eq (29)
            Update MAB statistics according to Eq (33)
            Retain the better solution and update the sub-group optimal, global optimal
        Else
            # Processing  state = Extinct  (see section 3.2 for details)
        end if
    end for      # each niche i
end for      # each iteration t
```

3.3.6. Multi-strategy boundary processing

After the particle position is updated, the boundary processing is followed. However, the same boundary treatment method is adopted for different niche states. This lack of a targeted treatment strategy leads to a waste of computing resources and loss of optimization efficiency to a certain extent. This paper holds that when the niche enters a debilitated state, it may enter the local optimal region. Compared with

the ordinary adsorption boundary treatment method, it will be more conducive to the evolution of the niche by allowing the particles to explore the adjacent range of high-quality points.

After the particle position is updated, the state of its niche is first judged.

Method 1: If $g = 0$, it indicates that it is in a healthy state; using the general boundary processing method, the value of the out-of-bound dimension is assigned to ub or lb.

Method 2: If $g \neq 0$, it indicates that it is in a debilitated state and calls the following processing strategy. The specific formula is Eq (35):

$$\begin{cases} x_i^{''} = x_{best} + \left|ub - x_{best}\right| \cdot \dfrac{\left|x_i - ub\right|}{\left|x_i^{'} - x_i\right|}, & x_i^{'} > ub, \\ x_i^{''} = x_{best} - \left|lb - x_{best}\right| \cdot \dfrac{\left|x_i - lb\right|}{\left|x_i^{'} - x_i\right|}, & x_i^{'} < lb. \end{cases} \tag{35}$$

where x_i and $x_i^{'}$ represent the particle's value in the i_{th} dimension before and after the position update, respectively, while x_{best} corresponds to the niche's best value in that dimension. $\left|ub - x_{best}\right|$ and $\left|lb - x_{best}\right|$ control the unit distance of the step, while the subsequent fractions control the number of units, thus enabling the transgressing particle to explore near the niche optimal position and break through the current dilemma.

Its boundary treatment is shown schematically in Figure 8.

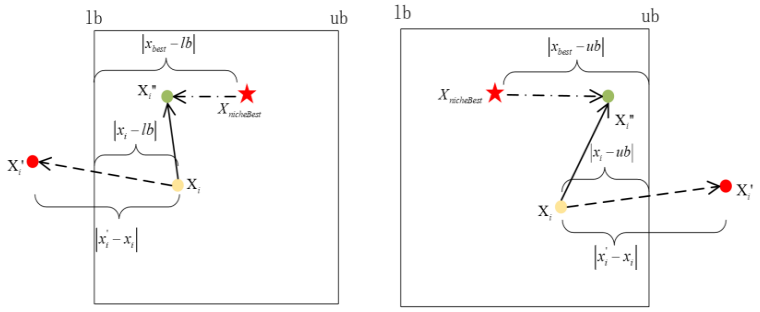


Figure 8. Cross-border processing schematic diagram.

3.4. Time complexity analysis

Let the relevant parameters be T, total number of iterations, N, population size, and D, problem dimension. Four primary components contribute to the overall time complexity of the RLDN-AOO algorithm:

a) Individual fitness calculation: Each iteration needs to calculate the fitness of N individuals; let the computational complexity of each fitness be COF. In T iterations, the time complexity of this part is $O\left(T \cdot N \cdot COF\right)$.

b) Local optimality determination: This step involves calculating the diversity of each niche, traversing the individuals within each niche, with a time complexity of $O\left(T \cdot N \cdot D\right)$ due to the total number of individuals in all niches being N.

c) Niche reclassification: A reclassification is triggered when the global optimum is invariant for consecutive G generations. The worst-case scenario is assumed here, where one repartition is

performed every G times. Each reclassification involves a distance calculation $O\left(N^2 \cdot D\right)$, with a total of $\lfloor T/G \rfloor$ reclassifications and a time complexity of $O\left(T \cdot N^2 \cdot D/G\right)$.

d) Multi-armed bandit strategy selection: Each individual update involves operations such as calculating the UCB value, which has a complexity of $O(D)$ per individual. In T iterations, the time complexity of this part is $O\left(T \cdot N \cdot D\right)$.

In summary, in the worst case, the time complexity of the proposed algorithm is:

$$O\left(T \cdot N \cdot (D + COF)\right) + O\left(T \cdot \frac{N^2 \cdot D}{G}\right). \tag{36}$$

The first term corresponds to the cost of fitness evaluation and scales similarly to conventional population-based metaheuristics such as PSO or DE. The second term arises from niche reclassification, which introduces an additional quadratic component with respect to the population size N. Therefore, the computational overhead of RLDN-AOO is higher than that of standard PSO/DE, especially when both N and D are large.

Nevertheless, compared with reinforcement learning–based algorithms such as Q-learning or policy-gradient methods, which often require repeated updates of value functions or deep neural networks with a complexity of at least $O\left(T \cdot |S| \cdot |A|\right)$ or even $O\left(T \cdot N \cdot D^2\right)$ in high-dimensional settings, the proposed RLDN-AOO remains significantly more efficient. In other words, our method introduces only lightweight reinforcement-guided components while avoiding the heavy computational burden typically associated with full reinforcement learning frameworks. When N is large, $O(T \cdot N^2 \cdot D/G)$ is the dominant term. In the best case, i.e., the number of global optimal stalls never exceeds G, there is no niche reclassification, i.e., $O\left(T \cdot N \cdot (D + COF)\right)$.

Overall, the algorithm trades slightly higher complexity than classical population-based metaheuristics for better performance on multimodal and high-dimensional problems but maintains substantially lower computational requirements than traditional reinforcement learning approaches. This balance ensures that the algorithm is both practical and scalable in high-dimensional search spaces.

4. Experimental results and discussion

To conduct a quantitative assessment of the RLDN-AOO algorithm's competitiveness and confirm its advantages, this research carried out comprehensive tests using the CEC2017 benchmark suite. Beyond analyzing the convergence, population diversity, and balance between exploration and exploitation pertaining to the RLDN-AOO algorithm, a systematic comparison of the algorithm was made with several others—either those excelling in addressing HDO problems or those adopting similar strategies. The average fitness value, standard deviation, convergence speed, and other indicators were monitored for performance measurement, and the Wilcoxon signed-rank test was used for significant difference analysis. The superiority of the RLDN-AOO algorithm in the core performance indicators, such as solution accuracy and convergence efficiency, is effectively revealed.

4.1. Experimental parameter configurations and experimental setup

All algorithms set the same experimental parameters, in which the maximum number of iterations

was 1000, and the problem dimension was 100. All parameters of the RLDN-AOO algorithm included the ecological-niche number bounds (lower/upper), the stagnation iteration threshold, the diversity threshold, the precision threshold, the maximum continuous stagnation iteration threshold, the number of strategy pools, the UCB exploration coefficient, the beta adjustment range, and the weight decay rate. Table 3 displays the key parameters of other algorithms. In addition, to reduce the impact of random factors on the experiment, each algorithm was executed independently 30 times, with the average and optimal values during operation recorded for follow-up research and analysis.

Table 3. Configuration of parameters for each associated algorithm.

Algorithm	Parameter
AOO	Weight 0.5
	Gravitational acceleration 9.8 divide by dimensions.
	Lévy index beta = 1.5
	Random scale factor $k \in [0.5, 1]$.
RLNNA	Correction factors beta $\in [0, 1]$.
	Dimensional perturbations $N \in [1, dim]$.
RLTLBO	Q-learning rate lambda $\in [0, 1]$.
	Discount factor sigma $\in [0, 1]$.
NCOA	Shrinkage factor 0.5
	Convergence accuracy threshold 1e-3
	Number of segments 50
NSSA	Stagnation threshold 10
	Diversity threshold 0.05
	Permitted stagnation algebra G = 5
	Fitness accuracy threshold 10
	Producer ratio P = 0.2
	Investigators ratio 0.2
	Step size random factor beta = 1.5
MMPA	Control the probability of Eddy formation and FADs effect 0.2
	Step length scaling factor P = 0.5
	Step size random factor beta = 1.5
SBOA	Lévy index beta = 1.5
LRA-CMA-ES	LRA learning rate m = 0.1, sigma = 0.03, gamma = 0.1
mLSHADE-SPACMA	Learning rate attenuation coefficient 0.8
	Elite selection ratio p = 0.11
	Archive size ratio 1.4
	Parameter memory size 5

The simulation experiments in this study were performed within the same computational

environment, specifically a computer configured with Windows 11 64-bit, an Intel(R) Core (TM) i9-14900HX CPU running at 2.20 GHz, 32.0 GB of RAM, and MATLAB R2024b.

To ensure fairness across the board, we standardized the population size for all algorithms to 100 and set the search space to span from -100 to 100.

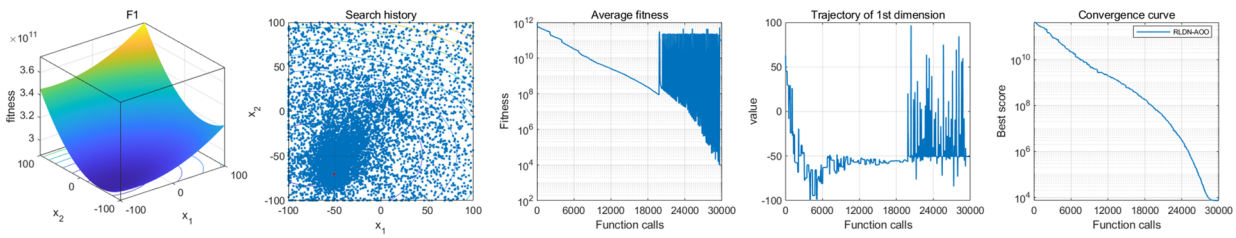
Benchmark functions: The CEC2017 benchmark set was used for experimentation in this paper. The ability of the algorithm to achieve convergence accuracy when determining the optimal was evaluated by examining the unimodal functions (F1–F3) that have a single global minimum and no local minimum. Functions F4–F10, being multimodal in nature, provide a useful means to test global exploration effectiveness and resilience against local optima entrapment. The integration of unimodal and multimodal functions gives rise to mixed (F11–F20) and ensemble (F21–F30) functions, which form more challenging benchmarks for examining an algorithm's balance between exploitation and exploration in seeking optimal solutions.

In this experiment, we selected the unimodal function F1, the multimodal functions F7 and F9, the unimodal and multimodal mixed functions F16, F17, and F20, and the integral functions F23, F26, and F29 to demonstrate the optimization effect of the RLDN-AOO algorithm. In the 200-dimensional and 500-dimensional tests, we extended the functions F1–F7 and F21–F23 in the CEC2017 test set, resulting in a higher-dimensional test set.

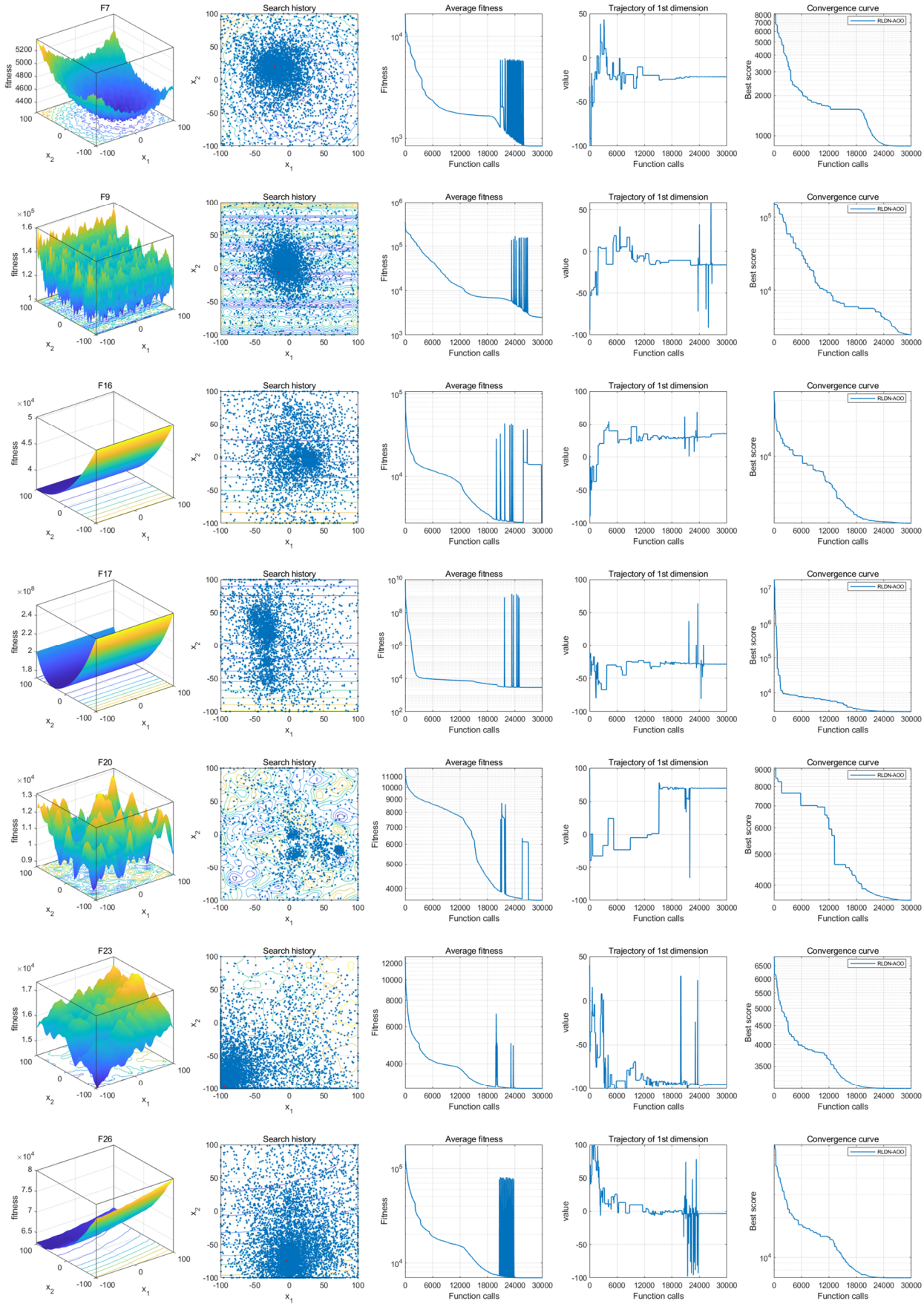
4.2. Experimental results and analysis of the RLDN-AOO algorithm

In this experiment, the results of the RLDN-AOO algorithm on the 100 dimensions of the CEC2017 benchmark set are demonstrated, and the RLDN-AOO algorithm is analyzed in terms of convergence, population diversity, and the balance between exploration and exploitation.

The search history graph in Figure 9 illustrates that the RLDN-AOO algorithm has powerful scanning capabilities for global exploration and local exploitation. The trajectory of the first dimension demonstrates that throughout iterations, individual niches progressively locate the primary optimal region. The frequency and magnitude of their positional adjustments diminish over time. However, during later iterations, habitat reinitialization occurs as some niches become trapped in local optima. This leads to a significant resurgence in the frequency and scale of location changes toward the end of the optimization process. The average fitness curve indicates that the collective population consistently converges toward the optimal solution with each algorithmic iteration. These findings collectively demonstrate that the proposed RLDN-AOO algorithm offers significant algorithmic advantages for solving HDO problems, effectively balancing and mediating the interplay between exploration and exploitation.



Continued on next page



Continued on next page

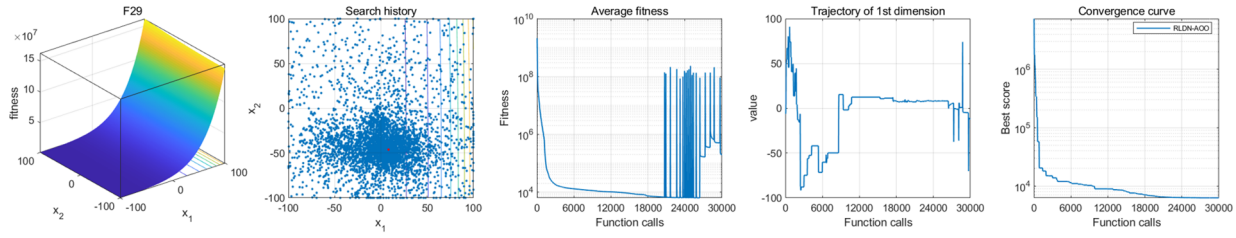
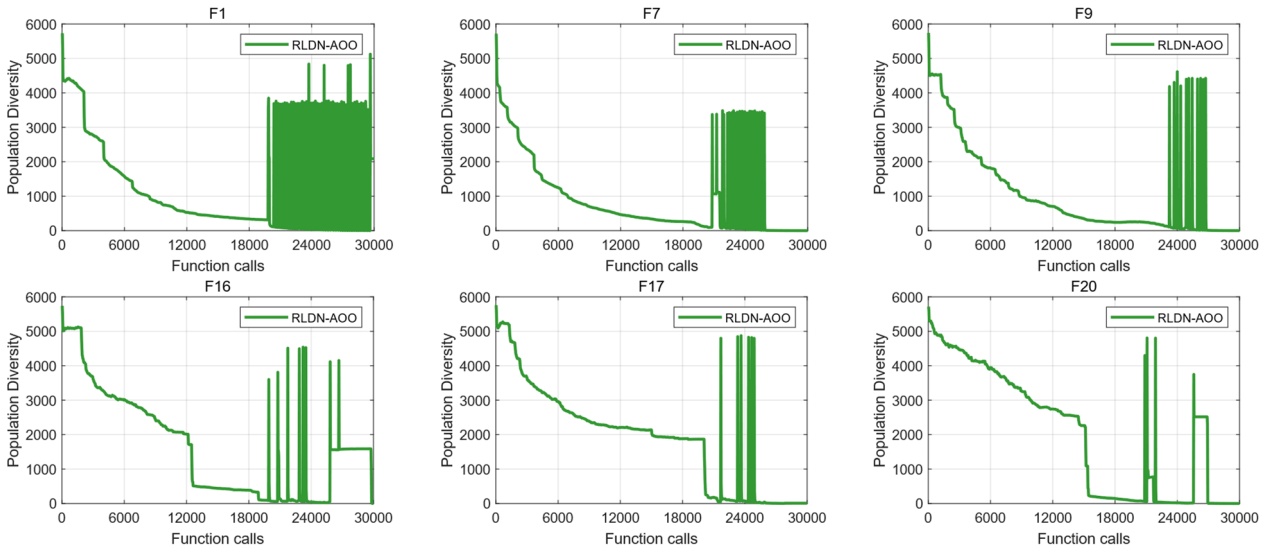


Figure 9. Qualitative performance of RLDN-AOO.

Figure 10 shows the change in population diversity during the 100-dimensional operation of the RLDN-AOO algorithm in the CEC2017 benchmark set. As shown in the figure, during the iteration, the population diversity gradually decreased, and in the later iterations, as the search area under the taboo area mechanism continued to shrink, some niches stagnated or fell into local optimums. Reinitialization of the niche and the repeated behavior led to strong fluctuations in population diversity in some test functions in the later stage. This algorithm makes it possible to find the optimal region more accurately in the later stage in a finite number of iterations when dealing with HDO problems.

Figure 11 shows the change in the exploration and exploitation ratio of the RLDN-AOO algorithm in the 100-dimensional operation of the CEC2017 benchmark set. As illustrated in the figure, the RLDN-AOO algorithm demonstrates a strong exploration capability and efficient solution space search during the early stages of iteration. With the progression of iterations, the emphasis gradually shifts toward exploitation, which in turn accelerates the convergence process. In the late iteration stage, due to the niche mechanism, some habitats reinitialize the niche in some test functions, which makes the ratio of exploration and exploitation at this stage fluctuate greatly.

It can be observed from the results that the RLDN-AOO algorithm possesses a strong capability to avoid local optima through maintaining population diversity. As a consequence, improvements are observed in the algorithm's performance when addressing HDO problems.



Continued on next page

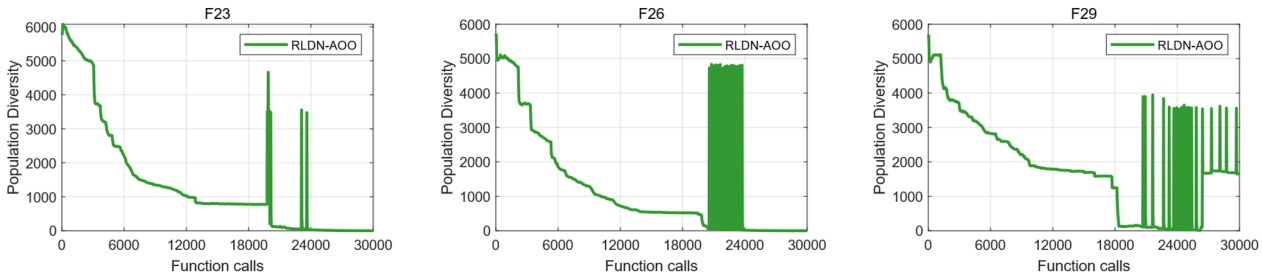


Figure 10. Population diversity of RLDN-AOO.

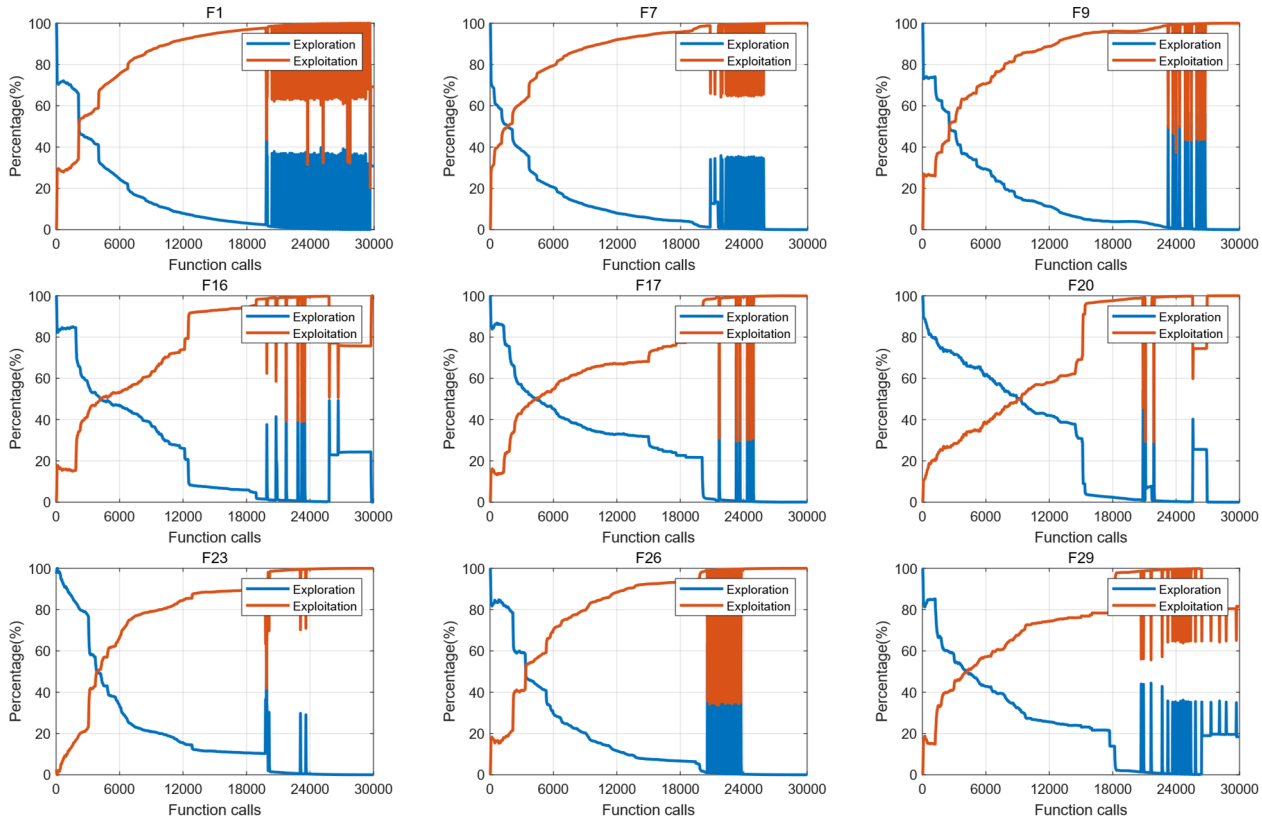


Figure 11. Exploration and exploitation of RLDN-AOO.

As can be seen from Figure 12, in the optimization process, the RLDN-AOO algorithm with MAB dynamic adaptive mechanism is obviously different from the AOO algorithm relying on a static parameter selection strategy, which reflects the difference of the MAB mechanism-based algorithm in the selection of optimization strategies.

Specifically, compared with the AOO algorithm, which selects strategies according to static parameters when dealing with the optimization of different functions, the RLDN-AOO algorithm has different strategy selection tendencies when dealing with the unimodal function (F1), the composite function (F18), and the integration function (F26), reflecting the characteristics of dynamic adaptation.

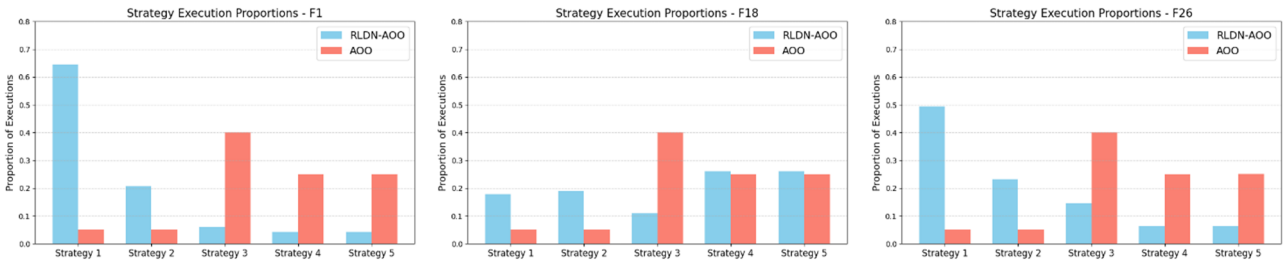


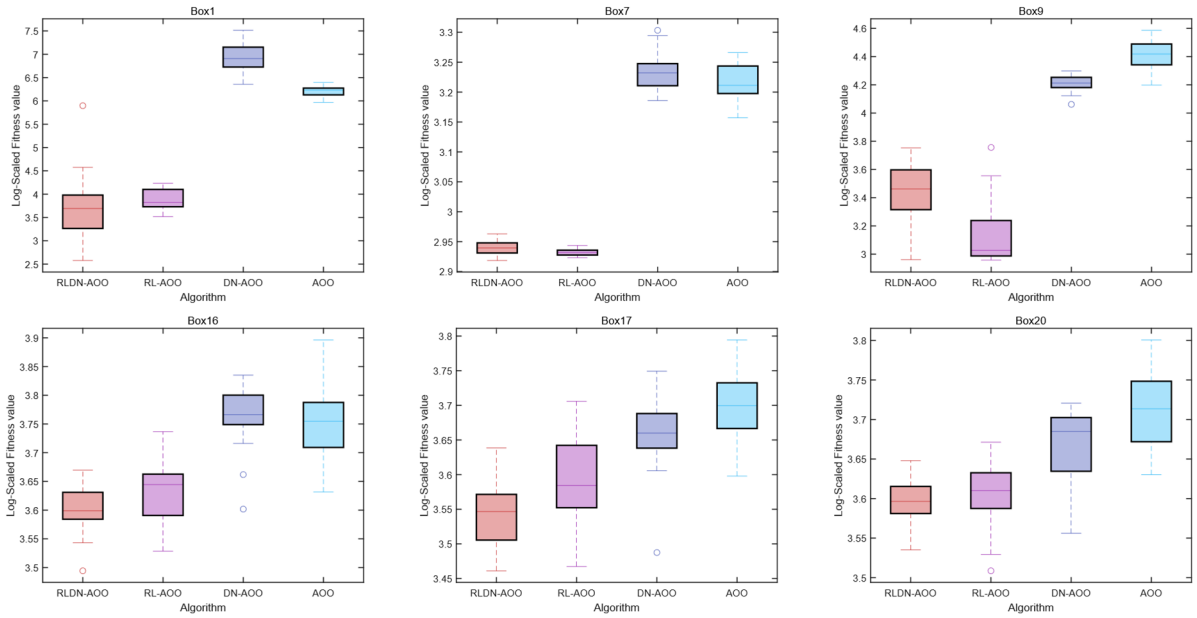
Figure 12. Selection of optimization strategies.

These experiments prove that the RLDN-AOO algorithm shows a strong and reasonable ability to dynamically explore and exploit, avoiding falling into local optimization in the later stage with its unique and excellent DN and MAB strategy; as such, it has an excellent ability to deal with HDO problems.

4.3. Ablation experiment of RLDN-AOO

In this section, we disassemble the RLDN-AOO algorithm and decompose it into the RL-AOO, DN-AOO, and AOO algorithms. Thus, the effectiveness of each strategy in the RLDN-AOO algorithm is proved.

Figure 13 shows the fitness values of each algorithm on a 100-dimensional test function of the CEC2017. As shown in the figure, the MAB strategy based on dynamic niching has significant advantages in dealing with multimodal problems and composite problems, and the AOO algorithm greatly increases the optimization efficiency with its strong global exploration and local exploitation strategies in dealing with mixed problems. The RLDN-AOO algorithm combines the advantages of the two to the greatest extent, allowing the algorithm to maintain a balanced search behavior in high-dimensional settings and improve its adaptability to intricate problem environments.



Continued on next page

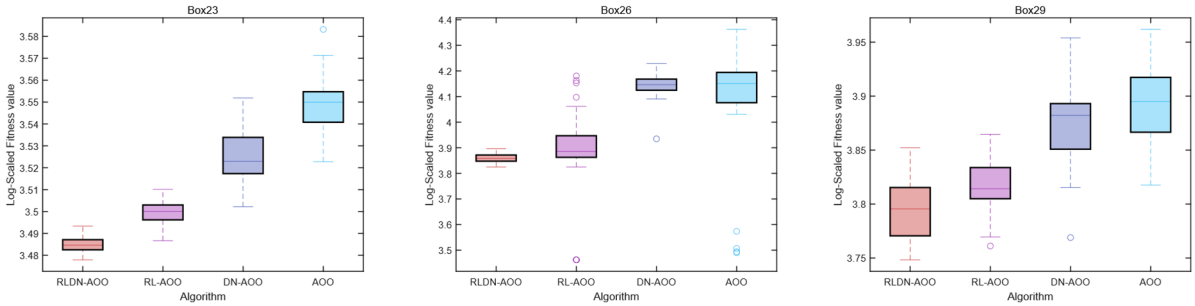


Figure 13. Fitness values of each dismantling algorithm.

4.4. Comparison of RLDN-AOO with other algorithms

In this section, we compare the RLDN-AOO algorithm with the Animated Oat Optimization Algorithm (AOO), Modified Marine Predator Algorithm (MMPA), Reinforcement Learning Neural Network Algorithm (RLNNA), Reinforcement Learning Teaching-Learning-Based Optimization (RLTLBO) Algorithm, Niching Chaos Optimization Algorithm (NCOA), Niching Sparrow Search Algorithm (NSSA), Secretary Bird Optimization Algorithm (SBOA), Covariance Matrix Adaptation Evolution Strategy with Learning Rate Adaptation (LRA-CMA-ES), and Modified L-SHADE with Semi-Parameter Adaptation and Covariance Matrix Adaptation (mLSHADE-SPACMA). Among them, the NSSA is an algorithm with a niche mechanism added to the Sparrow Optimization Algorithm (SSA). Thus, this proves the superiority of the RLDN-AOO algorithm in dealing with high-dimensional single-objective optimization problems.

Figures 14–17 show the fitness convergence curves of algorithms similar to the RLDN-AOO algorithm on the 50, 100, 200, and 500-dimensional CEC2017. The figure illustrates that the RLDN-AOO algorithm achieves clear advantages over comparable algorithms when optimizing unimodal, multimodal, hybrid, and composite functions, thereby confirming the effectiveness of the MAB strategy and DN in handling HDO problems.

The experimental results are shown in Tables 4–7. The RLDN-AOO algorithm has excellent optimization ability in various test functions: F5–F10, F16, F17, F20–F24, F26, F4–F10, F16, F17, F20–F26, and F28 in 100 dimensions, F1, F2, F4–F6, and F21–F22 in 200 dimensions, and F1–F5 and F21–F23 in 500 dimensions; the average fitness value of the RLDN-AOO algorithm ranks first among all algorithms and in 100 and 200 dimensions. The total average fitness value on 500 dimensions is ranked 1. In addition, in terms of optimal values, the RLDN-AOO algorithm ranks 1 in the total average of 200 and 500 dimensions in the test functions of the four dimensions, and is only slightly worse than the mLSHADE-SPACMA in the 100-dimension standard CEC2017 test set. This reflects the effectiveness and superiority of the RLDN-AOO algorithm.

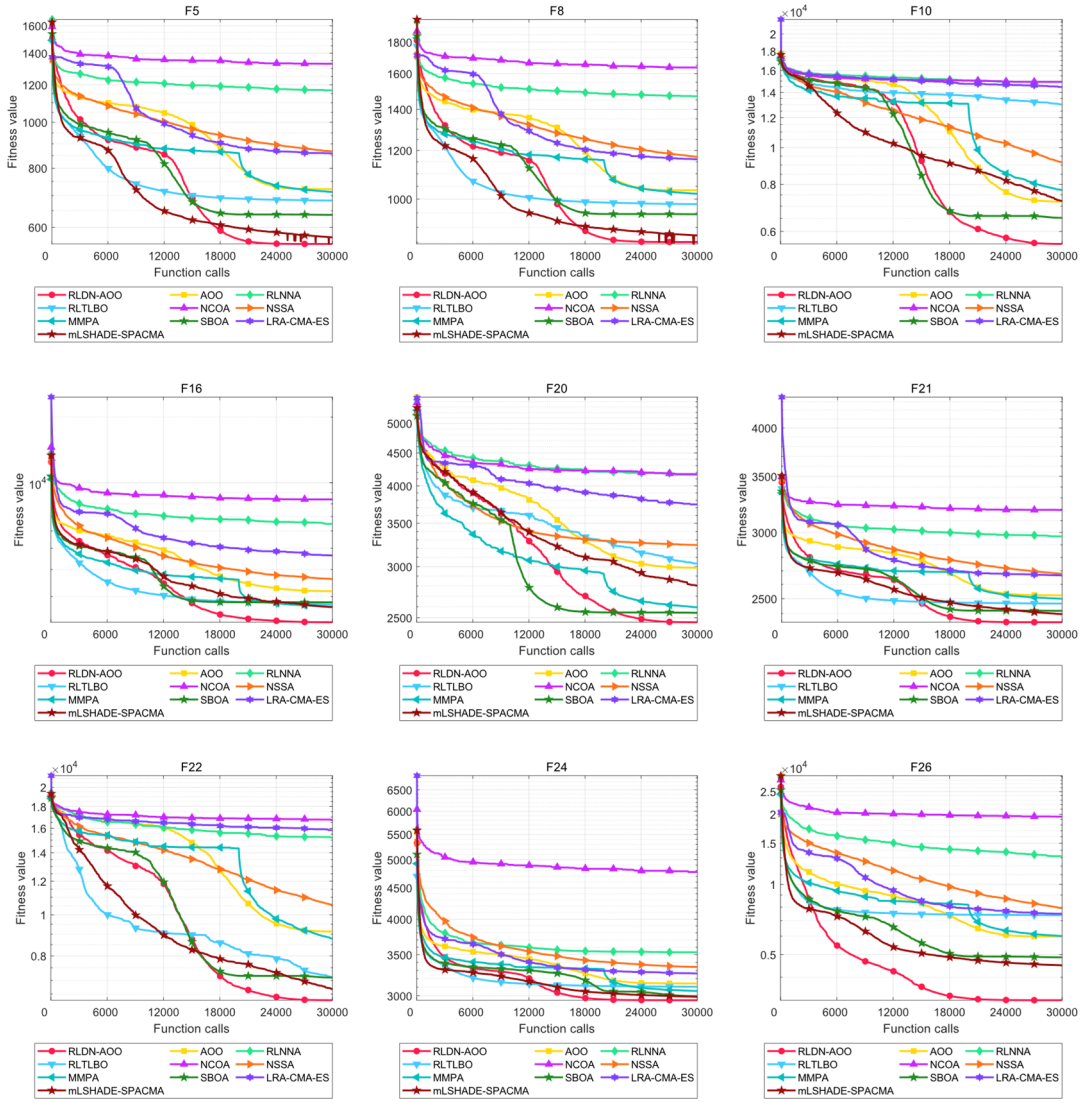
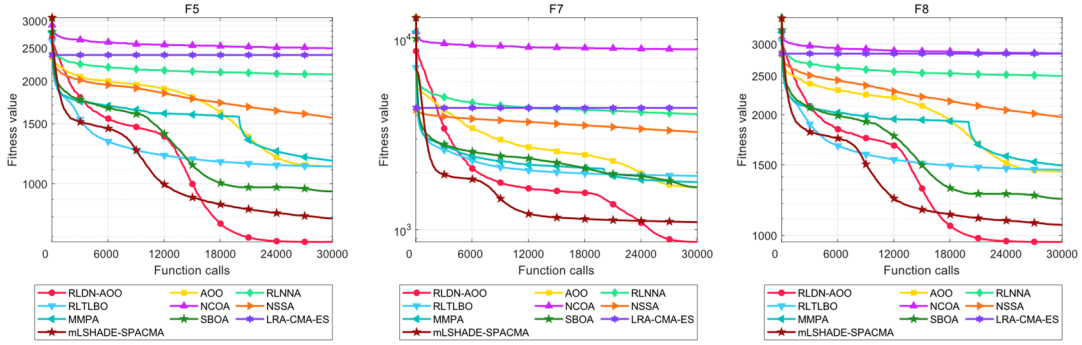


Figure 14. Convergence curves of RLDN-AOO and other algorithms in 50 dimensions.



Continued on next page

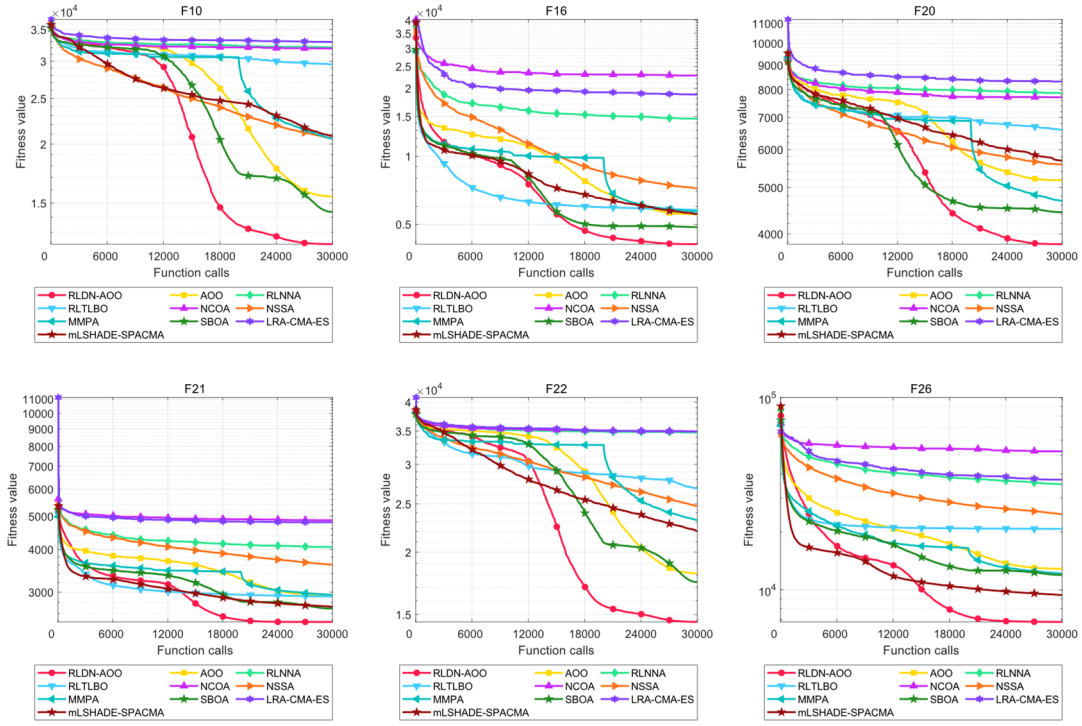


Figure 15. Convergence curves of RLDN-AOO and other algorithms in 100 dimensions.

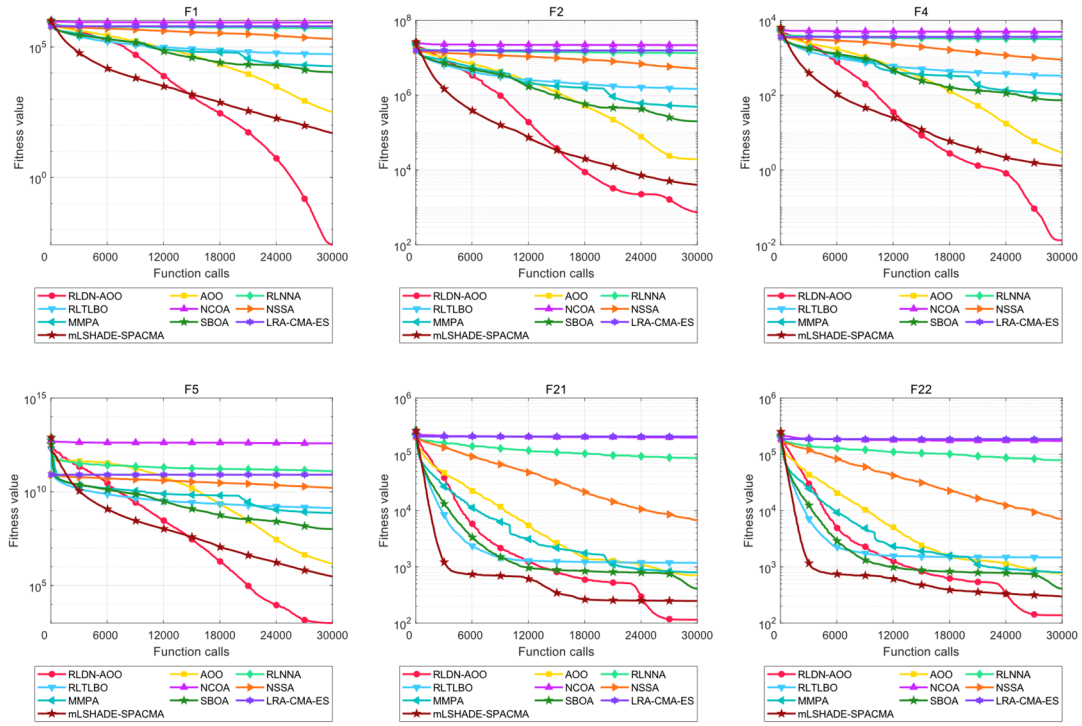


Figure 16. Convergence curves of RLDN-AOO and other algorithms in 200 dimensions.

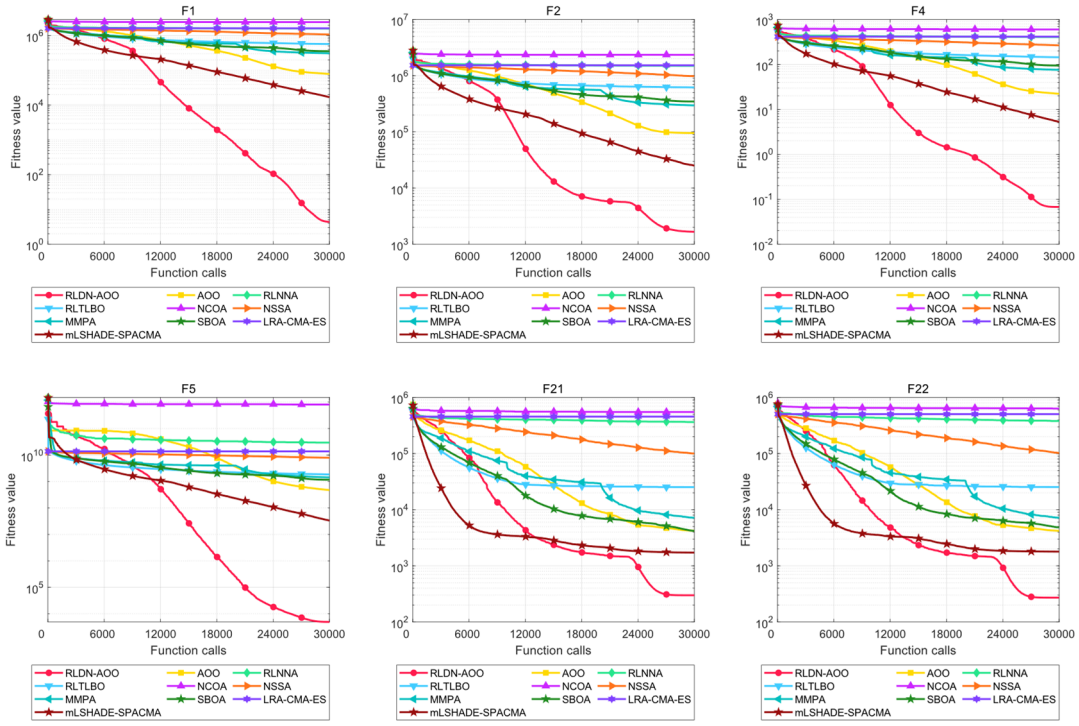


Figure 17. Convergence curves of RLDN-AOO and other algorithms in 500 dimensions.

Tables 4–7 clearly show that the RLDN-AOO algorithm achieves lower standard deviation in various functions in CEC2017 benchmarks compared with other algorithms, indicating that it can avoid local optimal solutions. Notably, it excels in multimodal and compound functions, with significantly lower deviations than competing methods. These results show that the algorithm has high stability and reliability in dealing with complex optimization problems, as well as its strong ability to evade local optimization and converge to a global optimum. It reflects the dynamic adaptive strategy selection mechanism of the RLDN-AOO algorithm and the flexibility and strong responsiveness of MAB and DN.

Table 4. Experimental results of RLDN-AOO with other algorithms in 50 dimensions.

		RLDN-AOO	AOO	RLNNA	RLTLBO	NCOA	NSSA	MMPA	SBOA	LRA-CMA-ES	mLSHADE-SPACMA
F1	**avg**	5.16E+03	4.14E+04	5.33E+10	1.52E+06	1.39E+11	2.07E+09	4.08E+07	4.61E+03	4.09E+08	**1.01E+02**
	best	1.26E+02	1.90E+04	3.85E+10	3.96E+03	1.09E+11	5.10E+07	2.21E+07	1.54E+02	2.90E+08	**1.00E+02**
	std	9.46E+03	1.19E+04	8.21E+09	4.03E+06	1.51E+10	3.07E+09	1.66E+07	4.16E+03	7.80E+07	1.40E+00
F2	**avg**	2.99E+22	7.14E+24	5.00E+70	1.02E+29	2.31E+77	8.06E+53	3.39E+31	4.01E+29	1.33E+60	**1.68E+13**
	best	4.66E+10	1.13E+17	2.67E+63	5.26E+19	4.32E+67	8.40E+40	1.67E+24	9.06E+17	4.34E+52	**2.99E+02**
	std	1.64E+23	3.66E+25	1.48E+71	3.24E+29	6.55E+77	4.27E+54	1.10E+32	1.52E+30	5.48E+60	5.97E+13
F3	**avg**	2.26E+04	1.15E+04	1.94E+05	5.15E+04	2.51E+05	1.89E+05	**2.40E+03**	2.07E+04	1.93E+05	5.74E+04
	best	6.87E+03	6.30E+03	1.45E+05	3.60E+04	1.84E+05	1.48E+05	1.30E+03	1.03E+04	1.46E+05	**3.14E+02**
	std	1.17E+04	3.06E+03	2.50E+04	9.27E+03	3.19E+04	1.88E+04	7.68E+02	6.51E+03	2.08E+04	9.71E+04
F4	**avg**	5.27E+02	5.96E+02	1.00E+04	5.77E+02	4.17E+04	7.59E+02	5.40E+02	5.36E+02	9.20E+02	**5.06E+02**
	best	4.98E+02	4.95E+02	7.38E+03	4.89E+02	2.92E+04	5.07E+02	4.41E+02	4.59E+02	8.53E+02	**4.29E+02**
	std	3.08E+01	4.66E+01	2.28E+03	5.26E+01	8.54E+03	9.10E+01	4.59E+01	5.34E+01	3.06E+01	4.93E+01
F5	**avg**	**5.53E+02**	7.23E+02	1.17E+03	6.83E+02	1.33E+03	8.68E+02	7.12E+02	6.37E+02	8.58E+02	5.71E+02
	best	5.36E+02	6.57E+02	1.08E+03	6.46E+02	1.25E+03	8.10E+02	6.74E+02	5.71E+02	8.21E+02	**5.26E+02**
	std	7.94E+00	4.07E+01	3.55E+01	2.18E+01	4.25E+01	3.00E+01	2.59E+01	3.75E+01	1.72E+01	2.61E+01
F6	**avg**	**6.00E+02**	6.34E+02	6.92E+02	6.16E+02	7.10E+02	6.59E+02	6.13E+02	6.01E+02	6.11E+02	6.00E+02
	best	**6.00E+02**	6.21E+02	6.79E+02	6.08E+02	6.98E+02	6.42E+02	6.04E+02	6.00E+02	6.09E+02	6.00E+02
	std	2.58E-02	9.73E+00	7.74E+00	5.17E+00	5.61E+00	6.25E+00	4.30E+00	9.78E-01	1.09E+00	4.99E-02
F7	**avg**	**7.77E+02**	9.91E+02	1.96E+03	9.85E+02	3.78E+03	1.64E+03	1.03E+03	9.50E+02	1.11E+03	8.05E+02
	best	**7.60E+02**	9.31E+02	1.71E+03	9.09E+02	3.35E+03	1.53E+03	9.59E+02	8.78E+02	1.08E+03	7.80E+02
	std	1.25E+01	3.58E+01	1.02E+02	5.15E+01	1.88E+02	7.58E+01	4.06E+01	5.32E+01	1.76E+01	1.97E+01
F8	**avg**	**8.52E+02**	1.03E+03	1.47E+03	9.81E+02	1.64E+03	1.17E+03	1.02E+03	9.45E+02	1.16E+03	8.73E+02
	best	8.41E+02	9.51E+02	1.37E+03	9.20E+02	1.50E+03	1.11E+03	9.62E+02	8.97E+02	1.13E+03	**8.24E+02**
	std	7.07E+00	5.06E+01	3.67E+01	3.04E+01	5.07E+01	3.59E+01	3.36E+01	2.69E+01	1.55E+01	2.72E+01

Continued on next page

F9	**avg**	**9.00E+02**	8.37E+03	3.54E+04	3.55E+03	5.01E+04	1.85E+04	2.24E+03	1.56E+03	1.85E+03	9.17E+02
	best	**9.00E+02**	4.89E+03	2.47E+04	1.37E+03	4.06E+04	1.35E+04	9.71E+02	9.32E+02	1.52E+03	9.03E+02
	std	5.45E-05	2.84E+03	4.50E+03	1.67E+03	6.01E+03	4.07E+03	1.09E+03	6.41E+02	2.17E+02	1.21E+01
F10	**avg**	**5.56E+03**	7.19E+03	1.49E+04	1.30E+04	1.49E+04	9.14E+03	7.72E+03	6.52E+03	1.45E+04	7.22E+03
	best	**3.78E+03**	5.38E+03	1.41E+04	8.16E+03	1.41E+04	6.95E+03	5.65E+03	4.50E+03	1.32E+04	6.34E+03
	std	8.40E+02	9.41E+02	3.52E+02	1.58E+03	3.05E+02	1.27E+03	8.59E+02	9.24E+02	4.21E+02	6.53E+02
F11	**avg**	1.28E+03	1.36E+03	1.65E+04	1.32E+03	2.55E+04	3.12E+03	**1.22E+03**	1.25E+03	2.11E+04	1.26E+03
	best	1.21E+03	1.20E+03	7.63E+03	1.23E+03	1.99E+04	2.23E+03	1.18E+03	**1.16E+03**	1.68E+04	1.20E+03
	std	3.97E+01	6.57E+01	4.23E+03	5.60E+01	3.43E+03	5.30E+02	2.04E+01	4.82E+01	2.89E+03	4.38E+01
F12	**avg**	7.30E+06	3.54E+07	1.11E+10	1.94E+06	5.75E+10	6.51E+07	1.11E+06	2.96E+06	2.92E+08	**9.46E+04**
	best	2.07E+06	5.53E+06	6.64E+09	1.79E+05	4.23E+10	8.48E+06	4.56E+05	9.58E+05	2.08E+08	**1.49E+04**
	std	3.40E+06	2.15E+07	2.29E+09	1.38E+06	1.11E+10	1.50E+08	5.88E+05	2.08E+06	5.09E+07	7.97E+04
F13	**avg**	4.03E+04	1.06E+05	3.87E+09	8.46E+03	2.26E+10	2.49E+06	6.12E+03	9.94E+03	1.78E+07	**4.84E+03**
	best	2.29E+04	3.43E+04	1.97E+09	3.11E+03	1.10E+10	9.47E+03	3.92E+03	**1.47E+03**	4.43E+06	2.23E+03
	std	1.27E+04	6.41E+04	9.12E+08	4.00E+03	6.80E+09	9.22E+06	1.25E+03	1.04E+04	1.08E+07	2.20E+03
F14	**avg**	2.03E+04	1.65E+05	6.84E+06	4.86E+04	2.85E+07	5.54E+05	**1.59E+03**	9.79E+04	2.78E+06	1.63E+03
	best	6.56E+03	3.69E+04	2.65E+06	9.67E+03	9.65E+06	1.32E+05	1.55E+03	1.72E+04	1.22E+06	**1.52E+03**
	std	8.05E+03	9.55E+04	4.07E+06	3.46E+04	1.92E+07	4.69E+05	1.31E+01	7.10E+04	9.96E+05	6.98E+01
F15	**avg**	2.40E+04	5.55E+04	9.18E+08	9.09E+03	4.53E+09	1.95E+04	1.98E+03	1.12E+04	3.43E+07	**1.97E+03**
	best	9.68E+03	1.19E+04	2.48E+08	1.97E+03	1.73E+09	5.51E+03	1.85E+03	**1.65E+03**	6.47E+06	1.72E+03
	std	1.91E+04	2.97E+04	4.07E+08	5.37E+03	1.46E+09	5.55E+03	6.36E+01	6.54E+03	2.66E+07	1.78E+02
F16	**avg**	**2.29E+03**	3.18E+03	6.47E+03	2.76E+03	8.38E+03	3.61E+03	2.70E+03	2.82E+03	4.64E+03	2.68E+03
	best	1.98E+03	2.30E+03	4.54E+03	2.37E+03	6.81E+03	2.49E+03	1.99E+03	2.16E+03	3.83E+03	**1.79E+03**
	std	1.91E+02	3.68E+02	5.26E+02	3.35E+02	1.12E+03	5.62E+02	3.18E+02	3.11E+02	2.86E+02	3.92E+02
F17	**avg**	**2.36E+03**	2.91E+03	5.20E+03	2.66E+03	7.06E+03	3.15E+03	2.51E+03	2.52E+03	4.11E+03	2.51E+03
	best	2.09E+03	2.41E+03	4.10E+03	2.32E+03	5.35E+03	2.37E+03	2.12E+03	2.02E+03	3.74E+03	**1.98E+03**
	std	1.87E+02	2.85E+02	4.02E+02	2.01E+02	8.66E+02	3.19E+02	1.97E+02	2.65E+02	1.95E+02	2.86E+02

Continued on next page

F18	**avg**	3.53E+05	1.45E+06	4.51E+07	4.96E+05	7.46E+07	2.79E+06	**2.52E+03**	1.00E+06	2.92E+07	5.46E+03
	best	6.77E+04	2.05E+05	1.91E+07	1.25E+05	3.14E+07	7.66E+05	**2.25E+03**	2.08E+05	1.43E+07	2.36E+03
	std	2.16E+05	1.16E+06	1.47E+07	3.60E+05	2.13E+07	1.94E+06	1.43E+02	5.36E+05	9.50E+06	3.12E+03
F19	**avg**	3.20E+05	5.33E+05	4.41E+08	1.60E+04	1.87E+09	2.42E+04	**2.05E+03**	1.44E+04	1.39E+07	2.07E+03
	best	8.01E+04	7.28E+04	1.50E+08	2.09E+03	1.10E+09	3.55E+03	2.02E+03	**1.95E+03**	1.63E+06	1.98E+03
	std	1.83E+05	3.66E+05	2.15E+08	8.34E+03	5.10E+08	3.57E+04	2.40E+01	1.16E+04	1.04E+07	4.08E+01
F20	**avg**	**2.46E+03**	2.99E+03	4.16E+03	3.03E+03	4.17E+03	3.24E+03	2.60E+03	2.54E+03	3.75E+03	2.80E+03
	best	2.28E+03	2.32E+03	3.61E+03	2.27E+03	3.63E+03	2.74E+03	2.32E+03	**2.05E+03**	3.24E+03	2.28E+03
	std	1.15E+02	3.01E+02	1.94E+02	4.41E+02	1.65E+02	2.44E+02	1.71E+02	2.62E+02	1.45E+02	2.34E+02
F21	**avg**	**2.34E+03**	2.52E+03	2.97E+03	2.47E+03	3.19E+03	2.68E+03	2.50E+03	2.42E+03	2.66E+03	2.40E+03
	best	**2.33E+03**	2.46E+03	2.86E+03	2.42E+03	3.04E+03	2.54E+03	2.44E+03	2.37E+03	2.64E+03	2.34E+03
	std	7.06E+00	4.06E+01	5.27E+01	2.65E+01	6.23E+01	5.61E+01	2.86E+01	2.31E+01	1.37E+01	2.06E+01
F22	**avg**	**6.29E+03**	9.13E+03	1.52E+04	7.12E+03	1.67E+04	1.05E+04	8.79E+03	7.11E+03	1.58E+04	6.68E+03
	best	2.30E+03	7.30E+03	6.79E+03	2.31E+03	1.59E+04	2.97E+03	2.36E+03	2.30E+03	1.44E+04	**2.30E+03**
	std	2.21E+03	1.00E+03	2.95E+03	5.86E+03	3.50E+02	2.59E+03	1.49E+03	2.31E+03	5.14E+02	3.10E+03
F23	**avg**	**2.78E+03**	3.01E+03	3.50E+03	2.96E+03	4.31E+03	3.19E+03	2.92E+03	2.84E+03	3.11E+03	2.82E+03
	best	**2.65E+03**	2.91E+03	3.35E+03	2.90E+03	3.99E+03	3.03E+03	2.87E+03	2.79E+03	3.08E+03	2.80E+03
	std	2.78E+01	5.93E+01	6.98E+01	4.00E+01	1.99E+02	7.05E+01	3.23E+01	2.68E+01	1.50E+01	1.48E+01
F24	**avg**	**2.95E+03**	3.14E+03	3.53E+03	3.10E+03	4.78E+03	3.34E+03	3.05E+03	3.00E+03	3.26E+03	2.99E+03
	best	**2.93E+03**	3.02E+03	3.43E+03	3.03E+03	4.40E+03	3.14E+03	3.01E+03	2.95E+03	3.23E+03	2.94E+03
	std	1.17E+01	6.04E+01	6.08E+01	4.55E+01	2.49E+02	1.05E+02	2.90E+01	2.76E+01	1.72E+01	2.20E+01
F25	**avg**	3.03E+03	3.06E+03	9.43E+03	3.11E+03	2.25E+04	3.28E+03	3.08E+03	3.08E+03	3.55E+03	**3.02E+03**
	best	3.02E+03	3.02E+03	7.71E+03	3.04E+03	1.55E+04	3.14E+03	2.99E+03	3.03E+03	3.44E+03	**2.96E+03**
	std	1.15E+01	2.71E+01	1.03E+03	3.32E+01	3.40E+03	1.15E+02	3.34E+01	2.91E+01	4.91E+01	3.26E+01
F26	**avg**	**3.19E+03**	5.98E+03	1.32E+04	7.38E+03	1.95E+04	7.90E+03	6.01E+03	4.87E+03	7.48E+03	4.50E+03
	best	**2.90E+03**	2.90E+03	1.02E+04	3.18E+03	1.75E+04	4.04E+03	5.10E+03	2.90E+03	6.88E+03	4.09E+03
	std	5.35E+02	1.45E+03	1.68E+03	1.76E+03	1.32E+03	2.35E+03	6.34E+02	1.23E+03	1.94E+02	2.32E+02

Continued on next page

F27	**avg**	3.32E+03	3.52E+03	4.06E+03	3.56E+03	6.26E+03	3.63E+03	3.29E+03	**3.29E+03**	3.94E+03	3.30E+03
	best	3.21E+03	3.35E+03	3.72E+03	3.37E+03	5.27E+03	3.42E+03	3.23E+03	3.21E+03	3.87E+03	**3.21E+03**
	std	3.45E+01	9.89E+01	2.78E+02	1.33E+02	4.29E+02	9.15E+01	4.19E+01	3.87E+01	3.69E+01	5.39E+01
F28	**avg**	3.29E+03	3.32E+03	7.93E+03	3.40E+03	1.39E+04	3.74E+03	3.32E+03	3.33E+03	5.62E+03	**3.28E+03**
	best	3.26E+03	3.26E+03	5.80E+03	3.32E+03	1.10E+04	3.52E+03	3.28E+03	3.28E+03	5.21E+03	**3.26E+03**
	std	2.02E+01	2.54E+01	7.68E+02	3.98E+01	1.28E+03	2.22E+02	2.90E+01	3.72E+01	1.28E+02	2.34E+01
F29	**avg**	3.82E+03	4.60E+03	8.87E+03	4.40E+03	1.86E+04	5.02E+03	3.84E+03	3.70E+03	7.76E+03	**3.52E+03**
	best	3.62E+03	4.00E+03	6.93E+03	3.91E+03	1.03E+04	4.60E+03	3.50E+03	**3.30E+03**	7.18E+03	3.38E+03
	std	1.35E+02	3.47E+02	1.10E+03	2.75E+02	1.04E+04	3.18E+02	1.80E+02	2.33E+02	3.60E+02	1.06E+02
F30	**avg**	2.46E+07	2.39E+07	1.11E+09	9.15E+05	3.34E+09	8.16E+06	3.60E+06	9.94E+05	3.05E+08	**7.91E+05**
	best	1.23E+07	1.38E+07	6.21E+08	6.35E+05	1.20E+09	9.36E+05	2.21E+06	7.14E+05	2.36E+08	**5.86E+05**
	std	1.08E+07	5.31E+06	2.79E+08	1.92E+05	9.97E+08	1.39E+07	9.29E+05	2.90E+05	2.59E+07	1.63E+05
+/≈/-			**28/1/1**	**30/0/0**	**23/2/5**	**30/0/0**	**26/2/2**	**18/2/10**	**18/4/8**	**30/0/0**	**13/4/13**
Rank Avg		**2**	**6**	**9**	**5**	**10**	**7**	**4**	**3**	**8**	**1**
Rank Best		**3**	**6**	**9**	**5**	**10**	**7**	**4**	**2**	**8**	**1**

Table 5. Experimental results of RLDN-AOO with other algorithms in 100 dimensions.

		RLDN-AOO	**AOO**	**RLNNA**	**RLTLBO**	**NCOA**	**NSSA**	**MMPA**	**SBOA**	**LRA-CMA-ES**	**mLSHADE-SPACMA**
F1	**avg**	8.51E+03	1.84E+06	1.97E+11	3.28E+09	4.02E+11	2.93E+10	1.83E+09	5.32E+07	2.98E+11	**5.11E+03**
	best	3.17E+02	9.96E+05	1.54E+11	3.89E+08	3.27E+11	7.61E+09	8.85E+08	1.74E+06	2.98E+11	**2.39E+02**
	std	1.04E+04	5.76E+05	1.90E+10	1.98E+09	2.43E+10	1.08E+10	7.31E+08	1.70E+08	1.24E-04	5.18E+03
F2	**avg**	1.88E+64	1.23E+86	9.42E+162	5.88E+97	1.25E+172	2.27E+139	1.33E+97	1.90E+88	4.87E+176	**1.00E+30**
	best	3.13E+41	2.36E+73	2.54E+147	3.81E+81	3.35E+159	4.08E+119	9.71E+82	2.38E+73	3.21E+163	**1.00E+30**
	std	7.28E+64	4.78E+86	6.55E+04	2.53E+98	6.55E+04	9.85E+139	7.25E+97	7.59E+88	6.55E+04	1.43E+14
F3	**avg**	2.05E+05	2.14E+05	4.86E+05	2.92E+05	6.16E+05	3.43E+05	**8.26E+04**	1.73E+05	1.06E+06	2.00E+05
	best	1.65E+05	1.39E+05	3.89E+05	2.55E+05	5.06E+05	2.93E+05	6.01E+04	1.40E+05	8.43E+05	**4.89E+04**
	std	2.69E+04	2.97E+04	4.77E+04	2.54E+04	5.48E+04	1.23E+04	1.28E+04	2.19E+04	1.12E+05	2.21E+05

Continued on next page

F4	**avg**	**6.69E+02**	7.82E+02	4.39E+04	1.29E+03	1.46E+05	3.89E+03	1.07E+03	8.94E+02	1.60E+05	6.79E+02
	best	6.12E+02	6.46E+02	3.25E+04	1.06E+03	1.10E+05	1.93E+03	9.31E+02	7.18E+02	1.60E+05	**5.50E+02**
	std	2.55E+01	5.89E+01	5.43E+03	1.76E+02	1.80E+04	1.60E+03	8.72E+01	7.25E+01	8.88E-11	5.01E+01
F5	**avg**	**6.75E+02**	1.13E+03	2.09E+03	1.12E+03	2.50E+03	1.56E+03	1.17E+03	9.50E+02	2.38E+03	7.91E+02
	best	6.47E+02	9.57E+02	1.92E+03	1.04E+03	2.25E+03	1.35E+03	1.06E+03	8.28E+02	2.38E+03	**6.10E+02**
	std	1.87E+01	1.15E+02	6.90E+01	6.44E+01	9.16E+01	7.24E+01	5.50E+01	6.51E+01	1.39E-12	1.18E+02
F6	**avg**	**6.01E+02**	6.52E+02	7.08E+02	6.36E+02	7.31E+02	6.72E+02	6.41E+02	6.17E+02	7.41E+02	6.03E+02
	best	**6.00E+02**	6.38E+02	6.97E+02	6.25E+02	7.23E+02	6.67E+02	6.25E+02	6.06E+02	7.41E+02	6.01E+02
	std	1.05E+00	5.55E+00	5.16E+00	6.14E+00	4.20E+00	2.96E+00	7.60E+00	4.64E+00	1.16E-13	1.20E+00
F7	**avg**	**8.63E+02**	1.67E+03	4.05E+03	1.92E+03	8.88E+03	3.26E+03	1.78E+03	1.67E+03	4.37E+03	1.10E+03
	best	**8.25E+02**	1.43E+03	3.62E+03	1.65E+03	8.22E+03	2.70E+03	1.63E+03	1.38E+03	4.37E+03	9.82E+02
	std	2.25E+01	1.39E+02	1.88E+02	1.50E+02	3.45E+02	1.72E+02	8.84E+01	1.73E+02	1.85E-12	8.71E+01
F8	**avg**	**9.63E+02**	1.44E+03	2.50E+03	1.46E+03	2.85E+03	1.97E+03	1.49E+03	1.23E+03	2.84E+03	1.06E+03
	best	9.34E+02	1.30E+03	2.28E+03	1.31E+03	2.62E+03	1.80E+03	1.37E+03	1.14E+03	2.84E+03	**9.24E+02**
	std	1.98E+01	8.81E+01	8.24E+01	8.01E+01	9.69E+01	7.54E+01	7.82E+01	5.22E+01	1.39E-12	1.08E+02
F9	**avg**	**2.53E+03**	2.47E+04	9.46E+04	3.52E+04	1.43E+05	4.02E+04	3.06E+04	1.43E+04	1.18E+05	2.63E+03
	best	**9.06E+02**	1.82E+04	7.80E+04	1.77E+04	1.19E+05	3.24E+04	1.64E+04	6.73E+03	1.18E+05	1.28E+03
	std	9.79E+02	4.63E+03	7.24E+03	1.02E+04	9.83E+03	7.80E+03	6.01E+03	4.00E+03	0.00E+00	8.32E+02
F10	**avg**	**1.23E+04**	1.55E+04	3.21E+04	2.95E+04	3.19E+04	2.06E+04	2.05E+04	1.44E+04	3.29E+04	2.08E+04
	best	**8.79E+03**	1.36E+04	3.13E+04	1.71E+04	3.02E+04	1.75E+04	1.78E+04	1.20E+04	3.15E+04	1.65E+04
	std	1.68E+03	1.15E+03	5.53E+02	3.16E+03	6.54E+02	1.84E+03	1.64E+03	1.30E+03	6.09E+02	2.71E+03
F11	**avg**	3.95E+03	3.33E+03	1.85E+05	3.40E+03	2.30E+05	2.19E+05	**3.29E+03**	7.85E+03	4.09E+05	1.50E+04
	best	2.36E+03	2.70E+03	1.14E+05	2.60E+03	1.65E+05	1.56E+05	2.67E+03	4.31E+03	2.94E+05	**1.92E+03**
	std	2.28E+03	4.41E+02	2.74E+04	7.16E+02	3.68E+04	3.61E+04	3.45E+02	2.58E+03	5.73E+04	2.92E+04
F12	**avg**	1.02E+08	2.62E+08	6.61E+10	1.14E+08	2.08E+11	2.41E+09	2.20E+08	3.36E+07	2.58E+11	**1.74E+06**
	best	5.84E+07	9.53E+07	4.65E+10	3.36E+07	1.75E+11	4.15E+08	9.81E+07	1.40E+07	2.18E+11	**4.02E+05**
	std	2.18E+07	1.06E+08	1.21E+10	1.15E+08	1.75E+10	2.64E+09	1.01E+08	1.36E+07	9.31E+09	8.69E+05

Continued on next page

F13	**avg**	3.04E+04	5.99E+04	1.05E+10	1.40E+04	4.43E+10	1.43E+07	2.24E+05	1.37E+04	6.24E+10	**9.34E+03**
	best	2.15E+04	2.78E+04	6.05E+09	7.41E+03	2.99E+10	1.08E+05	1.12E+05	**2.48E+03**	3.91E+10	3.34E+03
	std	4.51E+03	2.24E+04	2.68E+09	4.22E+03	5.67E+09	6.42E+07	7.71E+04	2.08E+04	7.33E+09	4.10E+03
F14	**avg**	6.01E+05	1.59E+06	4.58E+07	6.34E+05	8.81E+07	7.39E+06	1.19E+05	1.43E+06	1.55E+08	**1.47E+04**
	best	1.93E+05	5.64E+05	2.43E+07	2.33E+05	5.27E+07	3.85E+06	**2.14E+03**	3.17E+05	8.34E+07	2.21E+03
	std	2.07E+05	7.49E+05	1.27E+07	2.61E+05	2.53E+07	2.26E+06	3.67E+05	7.57E+05	4.17E+07	1.12E+04
F15	**avg**	2.41E+04	5.70E+04	3.49E+09	4.72E+03	1.82E+10	3.13E+06	1.22E+04	6.89E+03	1.93E+10	**4.35E+03**
	best	1.65E+04	2.65E+04	1.76E+09	2.18E+03	1.24E+10	1.35E+04	7.94E+03	1.96E+03	1.16E+10	**1.96E+03**
	std	4.78E+03	2.39E+04	1.19E+09	2.89E+03	3.71E+09	1.18E+07	3.06E+03	7.56E+03	3.23E+09	4.01E+03
F16	**avg**	**4.08E+03**	5.53E+03	1.46E+04	5.79E+03	2.28E+04	7.20E+03	5.67E+03	4.85E+03	1.87E+04	5.54E+03
	best	**3.17E+03**	4.12E+03	1.18E+04	3.87E+03	1.95E+04	5.52E+03	4.40E+03	3.32E+03	1.64E+04	3.45E+03
	std	4.86E+02	6.44E+02	1.15E+03	7.74E+02	2.14E+03	7.88E+02	6.51E+02	6.19E+02	1.13E+03	1.14E+03
F17	**avg**	**3.53E+03**	5.04E+03	2.02E+04	4.84E+03	1.39E+06	5.67E+03	4.68E+03	4.33E+03	1.16E+06	4.71E+03
	best	**2.62E+03**	3.97E+03	9.12E+03	3.88E+03	1.71E+05	4.10E+03	3.82E+03	3.47E+03	2.20E+05	2.94E+03
	std	3.43E+02	5.59E+02	9.54E+03	4.72E+02	8.71E+05	6.33E+02	4.76E+02	6.03E+02	5.00E+05	7.52E+02
F18	**avg**	7.61E+05	2.17E+06	7.72E+07	7.94E+05	1.74E+08	6.89E+06	**4.83E+04**	2.36E+06	2.95E+08	8.40E+04
	best	2.77E+05	4.32E+05	3.00E+07	3.71E+05	1.04E+08	1.94E+06	3.26E+04	5.77E+05	1.66E+08	**2.85E+04**
	std	2.94E+05	1.30E+06	2.79E+07	3.42E+05	3.72E+07	3.53E+06	9.58E+03	1.19E+06	7.37E+07	3.77E+04
F19	**avg**	1.01E+06	1.67E+06	3.82E+09	4.95E+03	1.76E+10	2.49E+06	1.79E+04	7.57E+03	1.88E+10	**4.23E+03**
	best	4.56E+05	1.37E+05	2.47E+09	2.21E+03	1.15E+10	3.56E+04	7.95E+03	**2.14E+03**	1.05E+10	2.23E+03
	std	4.19E+05	1.23E+06	6.63E+08	4.25E+03	3.21E+09	1.17E+07	7.59E+03	7.14E+03	3.33E+09	2.48E+03
F20	**avg**	**3.81E+03**	5.18E+03	7.87E+03	6.60E+03	7.71E+03	5.58E+03	4.69E+03	4.44E+03	8.32E+03	5.69E+03
	best	**3.13E+03**	4.48E+03	7.28E+03	4.09E+03	7.13E+03	4.51E+03	3.87E+03	3.29E+03	7.78E+03	3.84E+03
	std	3.51E+02	5.03E+02	2.07E+02	7.65E+02	2.19E+02	4.94E+02	3.69E+02	6.06E+02	2.57E+02	4.54E+02
F21	**avg**	**2.45E+03**	2.94E+03	4.05E+03	2.91E+03	4.87E+03	3.61E+03	2.94E+03	2.69E+03	4.80E+03	2.72E+03
	best	**2.43E+03**	2.78E+03	3.91E+03	2.75E+03	4.70E+03	3.35E+03	2.82E+03	2.56E+03	4.64E+03	2.48E+03
	std	1.35E+01	9.11E+01	9.11E+01	5.89E+01	1.09E+02	1.35E+02	6.74E+01	5.88E+01	8.87E+01	5.36E+01
F22	**avg**	**1.45E+04**	1.81E+04	3.48E+04	2.68E+04	3.49E+04	2.47E+04	2.32E+04	1.74E+04	3.49E+04	2.20E+04

Continued on next page

	best	**2.30E+03**	1.52E+04	3.37E+04	3.38E+03	3.42E+04	1.90E+04	2.04E+04	1.54E+04	3.28E+04	2.06E+04
	std	2.70E+03	1.32E+03	3.97E+02	1.07E+04	4.30E+02	2.03E+03	1.36E+03	1.34E+03	5.71E+02	1.19E+03
F23	**avg**	**3.05E+03**	3.57E+03	4.60E+03	3.59E+03	6.64E+03	4.13E+03	3.48E+03	3.16E+03	4.93E+03	3.15E+03
	best	**3.01E+03**	3.38E+03	4.37E+03	3.39E+03	6.08E+03	3.85E+03	3.32E+03	3.05E+03	4.76E+03	3.10E+03
	std	2.59E+01	1.18E+02	1.40E+02	9.81E+01	3.28E+02	1.55E+02	1.69E+02	6.57E+01	7.41E+01	2.51E+01
F24	**avg**	**3.44E+03**	4.19E+03	5.33E+03	4.25E+03	1.09E+04	4.97E+03	3.90E+03	3.68E+03	6.27E+03	3.64E+03
	best	**3.38E+03**	3.90E+03	5.10E+03	4.00E+03	9.71E+03	4.59E+03	3.74E+03	3.55E+03	5.98E+03	3.48E+03
	std	2.84E+01	1.40E+02	2.06E+02	1.46E+02	8.00E+02	2.80E+02	7.60E+01	8.73E+01	1.56E+02	7.25E+01
F25	**avg**	**3.32E+03**	3.47E+03	2.11E+04	3.89E+03	5.99E+04	6.09E+03	3.80E+03	3.51E+03	3.59E+04	3.32E+03
	best	**3.18E+03**	3.35E+03	1.65E+04	3.67E+03	5.21E+04	4.49E+03	3.63E+03	3.35E+03	3.59E+04	3.19E+03
	std	5.67E+01	6.62E+01	2.01E+03	1.55E+02	4.19E+03	8.72E+02	1.02E+02	8.47E+01	2.22E-11	6.27E+01
F26	**avg**	**6.81E+03**	1.29E+04	3.53E+04	2.07E+04	5.25E+04	2.47E+04	1.21E+04	1.20E+04	3.74E+04	9.39E+03
	best	**2.90E+03**	2.97E+03	2.43E+04	1.54E+04	4.24E+04	2.08E+04	5.05E+03	8.99E+03	3.37E+04	7.19E+03
	std	1.17E+03	4.49E+03	7.66E+03	3.42E+03	4.25E+03	2.19E+03	1.45E+03	2.39E+03	1.59E+03	8.81E+02
F27	**avg**	3.50E+03	3.83E+03	5.35E+03	4.10E+03	1.16E+04	4.00E+03	3.56E+03	3.52E+03	7.46E+03	**3.48E+03**
	best	3.42E+03	3.64E+03	4.29E+03	3.68E+03	9.03E+03	3.64E+03	3.46E+03	3.41E+03	6.40E+03	**3.39E+03**
	std	4.04E+01	1.48E+02	6.32E+02	2.20E+02	1.02E+03	2.22E+02	5.96E+01	6.16E+01	5.08E+02	5.32E+01
F28	**avg**	**3.43E+03**	3.56E+03	2.41E+04	4.35E+03	4.22E+04	6.90E+03	4.14E+03	3.62E+03	4.37E+04	3.43E+03
	best	3.39E+03	3.47E+03	1.81E+04	3.95E+03	3.78E+04	5.02E+03	3.75E+03	3.51E+03	4.37E+04	**3.36E+03**
	std	2.98E+01	4.39E+01	3.11E+03	2.36E+02	2.87E+03	1.19E+03	1.08E+03	4.87E+01	1.48E-11	4.38E+01
F29	**avg**	6.44E+03	7.70E+03	2.64E+04	7.82E+03	3.92E+05	9.08E+03	6.83E+03	5.96E+03	9.02E+05	**5.85E+03**
	best	5.62E+03	6.05E+03	1.70E+04	6.71E+03	5.70E+04	7.40E+03	5.87E+03	4.80E+03	2.07E+05	**4.34E+03**
	std	4.18E+02	6.65E+02	6.08E+03	6.69E+02	2.08E+05	1.13E+03	4.38E+02	5.09E+02	4.54E+05	5.85E+02
F30	**avg**	1.28E+07	3.43E+07	6.84E+09	9.52E+04	3.42E+10	6.68E+07	2.13E+06	3.20E+04	2.61E+10	**1.04E+04**
	best	6.43E+06	1.10E+07	3.94E+09	2.70E+04	2.24E+10	2.23E+06	8.00E+05	1.20E+04	1.89E+10	**6.50E+03**
	std	3.81E+06	1.57E+07	1.50E+09	6.53E+04	5.68E+09	1.68E+08	1.07E+06	1.39E+04	3.50E+09	4.06E+03
+/≈/-			**28/2/0**	**30/0/0**	**22/4/4**	**30/0/0**	**29/1/0**	**23/1/6**	**22/1/7**	**30/0/0**	**14/6/10**
Rank Avg		**1**	**5**	**8**	**6**	**9**	**7**	**4**	**3**	**10**	**2**
Rank Best		**2**	**5**	**8**	**6**	**9**	**7**	**4**	**3**	**10**	**1**

Table 6. Experimental results of RLDN-AOO with other algorithms in 200 dimensions.

		RLDN-AOO	AOO	RLNNA	RLTLBO	NCOA	NSSA	MMPA	SBOA	LRA-CMA-ES	mLSHADE-SPACMA
F1	**avg**	**2.62E-03**	3.17E+02	5.30E+05	5.33E+04	8.71E+05	2.04E+05	1.83E+04	1.11E+04	6.26E+05	4.95E+01
	best	**1.06E-03**	2.49E+02	4.98E+05	3.03E+04	8.33E+05	1.25E+05	1.32E+04	3.77E+03	6.26E+05	3.47E+01
	std	2.48E-03	6.49E+01	2.35E+04	1.55E+04	2.43E+04	5.69E+04	3.07E+03	5.46E+03	0.00E+00	2.35E+01
F2	**avg**	**7.41E+02**	1.94E+04	1.33E+07	1.47E+06	2.18E+07	5.17E+06	4.91E+05	2.01E+05	1.57E+07	3.98E+03
	best	**2.66E+02**	1.38E+04	1.27E+07	9.88E+05	2.10E+07	3.38E+06	3.66E+05	3.57E+04	1.57E+07	3.09E+03
	std	3.64E+02	4.09E+03	4.90E+05	3.33E+05	5.99E+05	1.80E+06	9.26E+04	8.59E+04	0.00E+00	6.75E+02
F3	**avg**	2.08E+01	**2.07E+01**	2.15E+01	2.15E+01	2.15E+01	2.09E+01	2.15E+01	2.10E+01	2.15E+01	2.14E+01
	best	**2.05E+01**	2.07E+01	2.14E+01	2.15E+01	2.14E+01	2.07E+01	2.14E+01	2.09E+01	2.15E+01	2.13E+01
	std	3.65E-01	3.18E-02	2.03E-02	5.77E-03	2.33E-02	1.56E-01	1.17E-02	4.41E-02	1.34E-02	7.49E-02
F4	**avg**	**1.33E-02**	2.88E+00	3.13E+03	3.31E+02	4.94E+03	8.86E+02	1.06E+02	7.34E+01	3.63E+03	1.30E+00
	best	**8.34E-03**	2.21E+00	2.79E+03	2.33E+02	4.64E+03	6.31E+02	8.94E+01	2.59E+01	3.63E+03	1.15E+00
	std	4.36E-03	6.61E-01	2.14E+02	8.93E+01	1.65E+02	2.54E+02	2.05E+01	5.57E+01	9.72E-13	1.13E-01
F5	**avg**	**9.85E+02**	1.44E+06	1.28E+11	1.36E+09	3.84E+12	1.62E+10	7.40E+08	1.04E+08	8.12E+10	2.99E+05
	best	**1.99E+02**	5.90E+05	9.95E+10	8.88E+08	3.06E+12	9.69E+09	4.89E+08	3.16E+07	8.12E+10	1.11E+05
	std	1.58E+03	7.34E+05	1.84E+10	5.18E+08	3.60E+11	4.61E+09	2.57E+08	6.06E+07	0.00E+00	1.43E+05
F6	**avg**	**2.66E+02**	2.93E+02	3.50E+02	3.47E+02	3.49E+02	3.27E+02	3.46E+02	2.70E+02	3.51E+02	3.27E+02
	best	**1.89E+02**	2.72E+02	3.48E+02	3.40E+02	3.41E+02	3.10E+02	3.34E+02	2.56E+02	3.49E+02	2.91E+02
	std	5.11E+01	1.37E+01	1.97E+00	3.23E+00	4.05E+00	1.09E+01	6.19E+00	1.15E+01	1.93E+00	2.37E+01
F7	**avg**	3.47E+04	3.65E+04	6.23E+04	3.86E+04	7.14E+04	4.78E+04	3.77E+04	3.63E+04	6.92E+04	**3.36E+04**
	best	3.41E+04	3.51E+04	5.78E+04	3.64E+04	6.95E+04	4.54E+04	3.66E+04	3.51E+04	6.87E+04	**3.19E+04**
	std	5.15E+02	9.82E+02	3.86E+03	1.16E+03	9.12E+02	1.97E+03	7.92E+02	8.99E+02	3.71E+02	7.99E+02
F21	**avg**	**1.15E+02**	7.09E+02	8.63E+04	1.18E+03	1.98E+05	6.71E+03	7.99E+02	4.13E+02	2.08E+05	2.48E+02
	best	**1.06E+02**	5.61E+02	6.81E+04	8.14E+02	1.87E+05	3.48E+03	7.01E+02	2.84E+02	2.08E+05	1.82E+02
	std	4.83E+00	1.07E+02	1.61E+04	2.36E+02	6.75E+03	3.75E+03	4.66E+01	7.60E+01	0.00E+00	4.52E+01

Continued on next page

F22	**avg**	**1.38E+02**	7.71E+02	7.88E+04	1.47E+03	1.73E+05	7.13E+03	8.02E+02	4.13E+02	1.87E+05	2.97E+02
	best	**1.20E+02**	4.58E+02	5.30E+04	1.00E+03	1.68E+05	3.85E+03	6.90E+02	3.35E+02	1.87E+05	2.02E+02
	std	1.18E+01	1.69E+02	1.41E+04	5.40E+02	7.11E+03	3.26E+03	5.14E+01	6.95E+01	0.00E+00	7.70E+01
F23	**avg**	6.08E+04	1.12E+05	1.14E+10	3.32E+06	5.21E+10	2.65E+08	4.26E+05	1.52E+05	8.00E+10	**7.93E+03**
	best	5.51E+04	7.30E+04	7.16E+09	6.00E+05	4.44E+10	1.38E+07	2.61E+05	1.81E+04	8.00E+10	**1.20E+03**
	std	2.43E+03	2.18E+04	3.09E+09	4.21E+06	4.56E+09	6.16E+08	1.10E+05	1.21E+05	0.00E+00	6.90E+03
+/≈/-			**8/2/0**	**10/0/0**	**10/0/0**	**10/0/0**	**9/1/0**	**10/0/0**	**7/3/0**	**10/0/0**	**8/0/2**
Rank Avg		**1**	**3**	**8**	**6**	**9**	**7**	**5**	**4**	**10**	**2**
Rank Best		**1**	**3**	**8**	**6**	**9**	**6**	**5**	**4**	**10**	**2**

Table 7. Experimental results of RLDN-AOO with other algorithms in 500 dimensions.

		RLDN-AOO	**AOO**	**RLNNA**	**RLTLBO**	**NCOA**	**NSSA**	**MMPA**	**SBOA**	**LRA-CMA-ES**	**mLSHADE-SPACMA**
F1	**avg**	**4.35E+00**	7.88E+04	1.59E+06	5.71E+05	2.44E+06	1.08E+06	3.09E+05	3.56E+05	1.62E+06	1.70E+04
	best	**1.37E-01**	6.83E+04	1.57E+06	5.05E+05	2.38E+06	9.80E+05	2.87E+05	2.79E+05	1.62E+06	1.43E+04
	std	4.47E+00	7.09E+03	1.55E+04	3.16E+04	3.03E+04	5.83E+04	1.67E+04	4.14E+04	2.49E-10	2.22E+03
F2	**avg**	**1.66E+03**	9.51E+04	1.50E+06	6.18E+05	2.35E+06	9.80E+05	2.95E+05	3.47E+05	1.53E+06	2.50E+04
	best	**1.16E+03**	8.15E+04	1.46E+06	5.22E+05	2.30E+06	8.17E+05	2.54E+05	2.63E+05	1.53E+06	2.10E+04
	std	3.94E+02	7.22E+03	2.08E+04	4.50E+04	2.95E+04	8.38E+04	2.66E+04	4.59E+04	0.00E+00	3.43E+03
F3	**avg**	**1.42E+01**	2.09E+01	2.16E+01	2.10E+01	2.16E+01	2.07E+01	2.07E+01	2.04E+01	2.12E+01	2.13E+01
	best	**1.94E+00**	2.09E+01	2.16E+01	2.09E+01	2.16E+01	2.06E+01	2.05E+01	2.02E+01	2.12E+01	2.06E+01
	std	6.52E+00	3.38E-02	9.70E-03	3.65E-02	5.98E-03	7.76E-02	1.14E-01	2.67E-01	0.00E+00	2.92E-01
F4	**avg**	**6.75E-02**	2.23E+01	4.07E+02	1.44E+02	6.00E+02	2.66E+02	7.59E+01	9.46E+01	4.16E+02	5.29E+00
	best	**3.97E-02**	1.83E+01	3.95E+02	1.37E+02	5.71E+02	2.42E+02	6.71E+01	7.98E+01	4.16E+02	4.58E+00
	std	2.51E-02	4.15E+00	7.65E+00	5.86E+00	1.65E+01	1.50E+01	4.92E+00	1.05E+01	6.08E-14	4.85E-01
F5	**avg**	**4.92E+03**	4.73E+08	2.95E+10	1.85E+09	8.11E+11	7.91E+09	1.43E+09	1.17E+09	1.37E+10	3.34E+07
	best	**1.78E+03**	3.11E+08	2.63E+10	1.65E+09	7.79E+11	7.24E+09	1.25E+09	7.60E+08	1.37E+10	2.39E+07
	std	4.74E+03	1.03E+08	3.45E+09	1.48E+08	2.91E+10	6.06E+08	1.70E+08	3.24E+08	2.04E-06	5.51E+06

Continued on next page

F6	**avg**	7.99E+02	8.31E+02	9.20E+02	9.14E+02	9.21E+02	8.87E+02	9.15E+02	**7.90E+02**	9.24E+02	8.40E+02
	best	**7.00E+02**	7.90E+02	9.14E+02	9.07E+02	9.17E+02	8.41E+02	8.98E+02	7.58E+02	9.19E+02	8.20E+02
	std	5.81E+01	2.32E+01	4.64E+00	4.64E+00	3.99E+00	2.23E+01	7.77E+00	2.01E+01	2.80E+00	2.58E+01
F7	**avg**	1.04E+05	1.13E+05	1.64E+05	1.24E+05	1.94E+05	1.46E+05	1.23E+05	1.08E+05	1.89E+05	**1.03E+05**
	best	1.01E+05	1.09E+05	1.57E+05	1.17E+05	1.91E+05	1.44E+05	1.17E+05	1.06E+05	1.87E+05	**9.92E+04**
	std	2.04E+03	2.32E+03	5.42E+03	4.19E+03	1.24E+03	2.65E+03	4.51E+03	1.56E+03	1.32E+03	1.89E+03
F21	**avg**	**2.99E+02**	4.23E+03	3.64E+05	2.53E+04	5.51E+05	1.00E+05	7.15E+03	4.21E+03	4.57E+05	1.72E+03
	best	**2.80E+02**	3.60E+03	3.49E+05	2.05E+04	5.22E+05	5.59E+04	5.61E+03	2.57E+03	4.57E+05	9.31E+02
	std	1.37E+01	5.04E+02	1.21E+04	5.42E+03	1.86E+04	2.33E+04	1.09E+03	8.82E+02	0.00E+00	5.90E+02
F22	**avg**	**2.71E+02**	4.21E+03	3.81E+05	2.54E+04	6.34E+05	1.02E+05	7.15E+03	4.86E+03	5.05E+05	1.79E+03
	best	**2.43E+02**	3.48E+03	3.59E+05	1.70E+04	6.00E+05	7.94E+04	6.61E+03	3.05E+03	5.05E+05	1.24E+03
	std	3.18E+01	6.11E+02	2.07E+04	6.09E+03	1.91E+04	2.46E+04	4.36E+02	1.37E+03	6.22E-11	3.50E+02
F23	**avg**	**1.21E+02**	1.21E+02	1.22E+02	1.22E+02	1.22E+02	1.21E+02	1.22E+02	1.21E+02	1.22E+02	1.21E+02
	best	**1.20E+02**	1.21E+02	1.22E+02	1.22E+02	1.22E+02	1.21E+02	1.21E+02	1.21E+02	1.22E+02	1.21E+02
	std	3.64E-01	2.26E-02	7.34E-03	1.08E-02	8.54E-03	8.08E-02	8.71E-02	3.86E-02	1.17E-02	2.20E-02
+/≈/-			**8/2/0**	**10/0/0**	**10/0/0**	**10/0/0**	**9/1/0**	**10/0/0**	**8/2/0**	**10/0/0**	**7/2/1**
Rank Avg		**1**	**3**	**8**	**6**	**10**	**7**	**5**	**4**	**9**	**2**
Rank Best		**1**	**4**	**8**	**6**	**10**	**6**	**5**	**3**	**9**	**2**

The RLDN-AOO algorithm exhibits outstanding optimization performance across diverse test functions. Owing to the strengths inherent in its strategic design and operational mechanisms, the proposed algorithm achieves notably superior results in locating optimal values compared to other comparable algorithms, demonstrating remarkable competitiveness.

To further test the time performance of the RLDN-AOO algorithm, we compare the running time of the RLDN-AOO algorithm with the algorithm with the same strategy or with advantages in dealing with high-dimensional complex problems. Tables 8–11 show the test results in CEC2017 in 50, 100, 200, and 500 dimensions. Overall, the RLDN-AOO algorithm ranks 9th, 7th, 8th, and 6th, respectively. Specifically, the RLDN-AOO algorithm has certain advantages over the RLNNA and RLTLBO algorithms using reinforcement learning strategies in some dimensions; using the MAB strategy, the RLDN-AOO algorithm is about 0.56 times and 0.64 times that of the RLNNA algorithm, and about 0.98 times and 0.79 times that of RLTLBO in lower dimensions (50D, 100D). At higher dimensions (200D, 500D), it is about 1.48 and 1.02 times that of the RLNNA algorithm, and about 0.44 and 0.21 times that of RLTLBO. Compared with NCOA and NSSA, which also use niche measurement mechanisms, the RLDN-AOO algorithm is relatively time-consuming. Compared with SBOA, LRA-CMA-ES, and mLSHADE-SPACMA, which have certain advantages in dealing with high-dimensional complex optimization problems, the RLDN-AOO algorithm is relatively time-consuming.

The addition of a reinforcement learning strategy obviously greatly increases the time cost, but we introduced the AOO algorithm as a selection arm of the MAB strategy through the fine design of the optimization process and the specific optimization implementation. To ensure that the optimization performance reduction is minimized, the time bottleneck of the RLDN-AOO algorithm is weakened as much as possible, so that the RLDN-AOO algorithm not only has an advantage in the best adaptability but also has a certain advantage in time overhead compared with similar algorithms that use a reinforcement learning strategy.

Table 8. Time results of RLDN-AOO with other algorithms in 50 dimensions.

	RLDN-AOO	AOO	RLNNA	RLTLBO	NCOA	NSSA	MMPA	SBOA	LRA-CMA-ES	mLSHADE-SPACMA
F1	9.44	1.77	11.92	3.25	3.09	1.96	6.41	2.12	3.01	**1.51**
F2	7.14	1.78	13.74	3.71	3.23	1.83	6.26	2.64	3.90	**1.55**
F3	6.08	1.32	13.39	3.09	3.19	1.78	6.05	2.07	2.39	**1.11**
F4	5.78	1.30	13.18	2.93	2.59	1.52	5.62	2.09	2.45	**1.06**
F5	6.44	1.51	13.60	4.18	3.11	3.58	6.12	2.54	2.85	**1.38**
F6	7.33	2.30	14.36	8.40	3.70	2.79	7.73	4.31	3.66	**1.85**
F7	6.60	1.52	13.48	4.19	3.00	3.53	6.05	2.62	2.85	**1.40**
F8	6.42	1.52	13.57	4.20	2.92	2.55	6.00	2.57	2.99	**1.33**
F9	5.88	1.49	13.56	4.12	2.84	3.05	5.98	2.51	2.96	**1.33**
F10	6.71	1.69	13.78	5.16	3.16	2.08	6.50	3.00	3.25	**1.60**
F11	6.23	1.45	13.67	3.72	3.05	1.63	5.99	2.36	3.21	**1.29**
F12	6.74	1.67	14.10	4.62	2.99	1.95	6.30	2.76	3.03	**1.39**
F13	6.40	1.44	13.74	3.69	2.72	1.70	5.98	2.42	2.59	**1.13**
F14	6.94	1.78	14.11	5.27	3.15	2.03	6.54	3.00	3.17	**1.35**
F15	6.40	1.41	13.80	3.36	2.82	1.58	5.66	2.17	2.85	**1.14**

Continued on next page

F16	6.56	1.52	13.88	3.96	2.93	1.68	6.00	2.52	3.27	**1.31**
F17	7.38	2.16	14.66	7.23	3.64	2.53	7.32	3.86	3.57	**1.78**
F18	6.59	1.51	13.90	4.00	3.06	1.75	5.91	2.52	3.00	**1.19**
F19	11.26	5.52	18.06	23.96	7.53	6.90	14.05	10.34	6.93	**4.02**
F20	10.49	2.34	14.93	8.04	3.80	2.74	7.66	4.15	3.87	**1.96**
F21	7.68	2.50	14.79	9.14	4.31	3.01	8.01	4.50	4.18	**2.11**
F22	7.67	2.66	15.03	10.11	4.28	3.24	8.37	4.91	4.30	**2.21**
F23	8.30	3.00	15.11	11.62	4.82	3.69	9.11	5.56	4.89	**2.57**
F24	8.83	3.36	15.39	13.47	5.37	4.27	10.30	6.36	5.24	**2.76**
F25	7.90	2.73	15.27	10.37	4.45	3.47	8.52	4.99	4.28	**2.17**
F26	8.54	3.43	15.96	14.50	5.69	5.12	11.51	7.07	5.36	**2.71**
F27	8.99	3.92	15.72	16.65	5.63	4.88	11.14	7.45	5.56	**2.97**
F28	7.74	3.11	15.17	12.68	4.80	3.97	9.41	5.83	4.58	**2.32**
F29	7.87	2.90	15.06	11.34	4.50	3.58	8.97	5.33	4.25	**2.17**
F30	11.55	6.24	18.43	28.12	8.14	7.73	15.55	11.83	7.28	**4.28**
Rank	**9**	**2**	**10**	**7**	**6**	**3**	**8**	**5**	**4**	**1**

Table 9. Time results of RLDN-AOO with other algorithms in 100 dimensions.

	RLDN-AOO	**AOO**	**RLNNA**	**RLTLBO**	**NCOA**	**NSSA**	**MMPA**	**SBOA**	**LRA-CMA-ES**	**mLSHADE-SPACMA**
F1	11.36	**2.89**	15.56	7.12	6.06	3.48	12.34	4.02	8.13	8.40
F2	10.97	3.40	19.19	8.45	5.87	**3.29**	12.59	4.63	8.99	8.55
F3	10.72	**2.82**	17.99	6.97	5.11	3.13	11.77	3.92	7.82	6.88
F4	9.22	2.76	17.63	6.91	4.90	**2.65**	11.52	3.84	6.71	6.97
F5	10.75	**3.20**	17.92	9.01	5.76	5.28	12.21	4.97	7.03	7.90
F6	13.11	**4.59**	19.13	17.50	6.71	5.55	15.60	8.46	9.00	8.18
F7	11.50	**3.22**	17.99	9.45	5.57	5.44	12.36	4.84	7.46	7.56
F8	10.53	**3.05**	18.20	9.25	5.65	5.13	12.24	4.93	7.41	7.30
F9	11.34	**3.29**	18.10	9.34	5.47	5.69	12.27	4.87	7.02	7.38
F10	11.31	**3.40**	18.15	11.37	5.70	3.85	13.25	5.74	7.88	7.52
F11	10.65	3.04	17.95	7.96	5.70	**3.01**	12.18	4.30	6.80	7.24
F12	10.49	**3.18**	17.87	9.91	5.50	3.38	12.62	5.18	7.52	7.29
F13	10.97	3.00	17.94	8.05	4.97	**2.92**	11.98	4.31	6.80	7.40
F14	12.21	**3.70**	17.97	11.49	5.72	3.77	13.20	5.85	7.19	7.29
F15	10.58	3.02	17.78	7.61	5.20	**2.74**	11.81	4.20	6.64	6.45
F16	11.11	**2.94**	17.83	8.92	5.08	3.12	12.19	4.76	6.97	7.21
F17	13.37	**4.20**	18.72	14.72	6.61	4.56	14.59	7.36	8.59	7.91
F18	11.33	**3.09**	17.95	8.94	5.41	3.21	12.17	4.66	7.91	6.98
F19	21.32	11.02	24.03	47.03	15.21	13.45	27.19	19.75	15.46	**10.83**
F20	23.04	**4.39**	18.90	16.32	6.84	5.21	15.16	7.88	9.05	8.17
F21	13.78	**5.72**	19.68	22.97	8.93	6.74	17.67	10.53	10.19	8.66

Continued on next page

F22	15.07	**6.20**	20.03	24.59	9.55	7.39	18.35	11.04	10.81	9.01
F23	15.26	**7.11**	19.89	28.87	10.41	8.40	19.89	12.77	11.29	9.25
F24	15.98	**7.70**	19.73	31.89	11.29	9.08	21.18	13.80	12.08	9.19
F25	14.78	**6.97**	20.79	28.47	10.37	8.43	19.98	12.58	11.71	9.16
F26	17.62	**8.42**	21.18	35.71	11.83	10.27	22.74	15.43	12.74	9.67
F27	19.47	10.00	22.09	42.47	14.38	11.93	25.56	17.97	14.52	**9.98**
F28	16.65	**8.32**	21.81	35.42	13.02	10.18	22.72	15.34	13.05	9.74
F29	14.68	**6.46**	20.39	26.40	9.64	7.86	19.07	11.81	10.91	8.77
F30	23.45	13.25	25.97	59.09	17.72	16.62	31.92	24.62	17.64	**11.91**
Rank	**7**	**1**	**10**	**8**	**3**	**2**	**9**	**4**	**6**	**5**

Table 10. Time results of RLDN-AOO with other algorithms in 200 dimensions.

	RLDN-AOO	**AOO**	**RLNNA**	**RLTLBO**	**NCOA**	**NSSA**	**MMPA**	**SBOA**	**LRA-CMA-ES**	**mLSHADE-SPACMA**
F1	339.93	**5.60**	306.66	1143.99	426.65	295.36	507.04	481.10	16.54	18.31
F2	395.83	**5.87**	366.90	1135.46	450.03	305.86	507.89	482.24	13.51	13.00
F3	432.41	**5.80**	265.44	1113.33	557.09	257.99	447.27	426.66	18.92	23.35
F4	519.53	**5.51**	365.29	1149.67	453.59	292.80	513.83	489.90	21.23	20.98
F5	464.68	**5.78**	310.04	1085.85	409.15	294.74	498.84	487.07	17.50	20.79
F6	326.69	**18.05**	290.25	1321.86	366.71	299.01	517.06	523.69	38.75	42.32
F7	446.05	**5.96**	388.22	1057.63	542.23	259.38	441.20	423.52	15.93	18.25
F21	683.69	**6.63**	444.21	1338.65	657.84	334.31	554.22	540.62	15.43	19.23
F22	681.43	**6.82**	434.34	1342.31	656.64	334.05	553.45	548.75	12.65	12.66
F23	673.81	**6.89**	443.03	1410.95	657.93	363.02	585.20	580.99	11.39	9.79
Rank	**8**	**1**	**5**	**10**	**7**	**4**	**9**	**6**	**2**	**3**

Table 11. Time results of RLDN-AOO with other algorithms in 500 dimensions.

	RLDN-AOO	**AOO**	**RLNNA**	**RLTLBO**	**NCOA**	**NSSA**	**MMPA**	**SBOA**	**LRA-CMA-ES**	**mLSHADE-SPACMA**
F1	531.83	**11.11**	508.39	2459.38	516.76	623.72	1015.50	997.99	140.41	53.23
F2	615.84	**12.87**	541.20	2845.72	538.73	656.00	1053.92	1030.55	129.66	44.02
F3	566.41	**11.54**	551.79	2636.21	516.16	629.77	1210.01	1174.39	148.56	67.60
F4	575.54	**12.87**	601.67	2793.62	551.11	701.04	1149.37	1074.73	134.21	48.26
F5	546.56	**11.72**	585.27	2784.94	606.48	744.40	1059.97	1182.54	144.16	47.29
F6	714.68	**70.65**	717.40	3330.25	700.24	780.36	1265.54	1229.90	160.68	67.82
F7	525.56	**11.35**	515.36	2796.88	526.26	687.32	1087.36	995.29	124.56	39.05
F21	557.22	**11.81**	544.05	2629.96	549.93	655.33	1071.92	1050.09	122.63	39.26
F22	549.88	**11.86**	533.37	2603.58	548.90	652.31	1073.78	1051.55	124.04	40.51
F23	548.82	**13.08**	539.27	2640.47	548.65	638.94	1086.37	1061.76	126.89	49.88
Rank	**6**	**1**	**4**	**10**	**5**	**7**	**9**	**8**	**3**	**2**

4.5. Wilcoxon signed-rank test

In order to verify the credibility of the experimental results and confirm the initial hypothesis, this section compares the RLDN-AOO algorithm with other algorithms using the Wilcoxon signed-rank test, thus determining the uniqueness and superiority of the algorithm's results.

Widely applied in statistics, the Wilcoxon signed-rank test originates from the sign test for paired observations and serves to assess whether two independent samples differ in their distributional location. It determines whether the median of the two sets of data is significantly different, rather than directly comparing the means. The Wilcoxon signed-rank test is nonparametric because it does not require assumptions about the form of the measured distribution.

Tables 12–15 show the p-value in different dimensions, with the nonparametric Wilcoxon signed-rank sum test at a significance level of 0.05. The evidence presented by these findings strongly indicates that the RLDN-AOO algorithm has a statistically significant advantage in most test scenarios. In addition, in the comparative analysis involving RLNNA, NCOA, and LRA-CMA-ES in 50 and 100 dimensions, RLNNA, RLTLBO, NCOA, MMPA, LRA-CMA-ES, and mLSHADE-SPACMA in 200 dimensions, and RLNNA, RLTLBO, NCOA, MMPA, and mLSHADE-SPACMA in 500 dimensions, the p-values obtained were all less than 0.05. This strongly proves the innovation of the RLDN-AOO algorithm architecture. These findings prove that the RLDN-AOO algorithm has good generalization ability and robustness and show its innovation in the field of optimization algorithms.

Table 12. CEC2017 (50D) Wilcoxon signed-rank test results of the compared algorithms.

	AOO	RLNNA	RLTLBO	NCOA	NSSA	MMPA	SBOA	LRA-CMA-ES	mLSHADE-SPACMA
F1	2.15E-10	3.02E-11	5.57E-10	3.02E-11	3.02E-11	3.02E-11	**7.98E-02**	3.02E-11	3.02E-11
F2	9.76E-10	3.02E-11	8.99E-11	3.02E-11	3.02E-11	3.02E-11	8.99E-11	3.02E-11	1.20E-08
F3	3.37E-05	3.02E-11	2.15E-10	3.02E-11	3.02E-11	3.02E-11	**9.00E-01**	3.02E-11	1.95E-03
F4	4.11E-07	3.02E-11	8.15E-05	3.02E-11	2.61E-10	**2.01E-01**	**7.28E-01**	3.02E-11	**1.71E-01**
F5	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	5.08E-03
F6	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.69E-11	3.02E-11	2.00E-05
F7	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	5.53E-08
F8	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.08E-02
F9	3.02E-11	3.02E-11	3.02E-11	3.01E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F10	5.53E-08	3.02E-11	3.02E-11	3.02E-11	4.50E-11	2.44E-09	1.17E-04	3.02E-11	1.07E-09
F11	7.04E-07	3.02E-11	1.84E-02	3.02E-11	3.02E-11	9.53E-07	1.33E-02	3.02E-11	**9.05E-02**
F12	4.18E-09	3.02E-11	2.44E-09	3.02E-11	5.07E-10	5.49E-11	3.26E-07	3.02E-11	3.02E-11
F13	4.69E-08	3.02E-11	3.02E-11	3.02E-11	**9.35E-01**	3.02E-11	8.10E-10	3.02E-11	3.02E-11
F14	3.02E-11	3.02E-11	1.89E-04	3.02E-11	3.02E-11	3.02E-11	1.85E-08	3.02E-11	3.02E-11
F15	7.60E-07	3.02E-11	6.53E-07	3.02E-11	**8.53E-01**	3.02E-11	4.71E-04	3.02E-11	3.02E-11
F16	1.96E-10	3.02E-11	2.02E-08	3.02E-11	4.50E-11	9.53E-07	1.10E-08	3.02E-11	2.60E-05
F17	2.03E-09	3.02E-11	6.53E-07	3.02E-11	2.15E-10	1.52E-03	7.29E-03	3.02E-11	3.51E-02
F18	2.38E-07	3.02E-11	**1.19E-01**	3.02E-11	3.34E-11	3.02E-11	9.83E-08	3.02E-11	3.02E-11
F19	2.92E-02	3.02E-11	3.02E-11	3.02E-11	8.15E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F20	1.01E-08	3.02E-11	5.60E-07	3.02E-11	3.02E-11	8.12E-04	**1.26E-01**	3.02E-11	2.57E-07

Continued on next page

F21	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	8.89E-10
F22	1.01E-08	2.87E-10	**8.07E-01**	3.02E-11	2.39E-08	7.09E-08	9.47E-03	3.02E-11	**7.24E-02**
F23	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	9.92E-11	3.02E-11	1.78E-10
F24	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	8.10E-10	3.02E-11	2.92E-09
F25	1.09E-05	3.02E-11	6.07E-11	3.02E-11	3.02E-11	3.52E-07	2.23E-09	3.02E-11	**5.30E-01**
F26	8.89E-10	3.02E-11	2.37E-10	3.02E-11	5.49E-11	3.02E-11	1.69E-09	3.02E-11	8.99E-11
F27	4.50E-11	3.02E-11	3.34E-11	3.02E-11	3.02E-11	3.18E-04	1.41E-04	3.02E-11	1.33E-02
F28	9.06E-08	3.02E-11	3.02E-11	3.02E-11	3.02E-11	2.01E-04	2.68E-06	3.02E-11	1.78E-04
F29	6.70E-11	3.02E-11	2.15E-10	3.01E-11	3.02E-11	**4.73E-01**	9.88E-03	3.02E-11	1.29E-09
F30	**4.38E-01**	3.02E-11	3.02E-11	3.02E-11	1.01E-08	3.02E-11	3.02E-11	3.02E-11	3.02E-11
Count	**1**	**0**	**2**	**0**	**2**	**2**	**4**	**0**	**4**

Table 13. CEC2017 (100D) Wilcoxon signed-rank test results of the compared algorithms.

	AOO	**RLNNA**	**RLTLBO**	**NCOA**	**NSSA**	**MMPA**	**SBOA**	**LRA-CMA-ES**	**mLSHADE-SPACMA**
F1	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.21E-12	**2.17E-01**
F2	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.21E-12
F3	**1.22E-01**	3.02E-11	1.78E-10	3.02E-11	3.02E-11	3.02E-11	8.88E-06	3.02E-11	1.95E-03
F4	7.38E-10	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.21E-12	**6.10E-01**
F5	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.21E-12	2.89E-03
F6	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.21E-12	4.12E-06
F7	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.21E-12	3.02E-11
F8	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.21E-12	2.50E-03
F9	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.21E-12	**9.82E-01**
F10	8.48E-09	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.09E-05	3.02E-11	3.02E-11
F11	**2.17E-01**	3.02E-11	**2.97E-01**	3.02E-11	3.02E-11	**2.34E-01**	2.57E-07	3.02E-11	1.03E-02
F12	5.57E-10	3.02E-11	**1.05E-01**	3.02E-11	3.02E-11	4.20E-10	4.50E-11	3.16E-12	3.02E-11
F13	2.44E-09	3.02E-11	5.49E-11	3.02E-11	3.02E-11	3.02E-11	9.76E-10	7.88E-12	3.02E-11
F14	1.20E-08	3.02E-11	**7.28E-01**	3.02E-11	3.02E-11	8.35E-08	2.78E-07	3.02E-11	3.02E-11
F15	1.46E-10	3.02E-11	3.02E-11	3.02E-11	5.97E-09	7.39E-11	5.46E-09	3.02E-11	3.34E-11
F16	2.03E-09	3.02E-11	2.03E-09	3.02E-11	3.02E-11	1.78E-10	5.46E-06	3.02E-11	5.46E-06
F17	3.69E-11	3.02E-11	6.07E-11	3.02E-11	3.34E-11	1.21E-10	2.78E-07	3.02E-11	6.05E-07
F18	1.11E-06	3.02E-11	**8.19E-01**	3.02E-11	3.02E-11	3.02E-11	1.29E-09	3.02E-11	3.02E-11
F19	1.27E-02	3.02E-11	3.02E-11	3.02E-11	1.03E-06	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F20	4.50E-11	3.02E-11	5.49E-11	3.02E-11	3.34E-11	1.07E-09	7.66E-05	3.02E-11	1.09E-10
F21	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.69E-11
F22	1.41E-09	3.02E-11	2.43E-05	3.02E-11	3.02E-11	3.02E-11	1.56E-08	3.02E-11	3.02E-11
F23	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	4.62E-10	3.02E-11	3.34E-11
F24	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.34E-11
F25	2.37E-10	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.61E-10	1.21E-12	**9.12E-01**
F26	4.74E-06	3.02E-11	3.02E-11	3.02E-11	3.02E-11	4.20E-10	3.02E-11	3.02E-11	1.21E-10
F27	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.56E-04	**2.17E-01**	3.02E-11	**7.24E-02**

Continued on next page

F28	5.49E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.21E-12	**9.23E-01**
F29	2.44E-09	3.02E-11	2.37E-10	3.02E-11	3.34E-11	5.26E-04	1.58E-04	3.02E-11	9.21E-05
F30	9.26E-09	3.02E-11	3.02E-11	3.02E-11	**8.53E-01**	3.02E-11	3.02E-11	3.02E-11	3.02E-11
Count	**2**	**0**	**4**	**0**	**1**	**1**	**1**	**0**	**6**

Table 14. CEC2017 (200D) Wilcoxon signed-rank test results of the compared algorithms.

	AOO	**RLNNA**	**RLTLBO**	**NCOA**	**NSSA**	**MMPA**	**SBOA**	**LRA-CMA-ES**	**mLSHADE-SPACMA**
F1	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F2	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F3	**4.42E-01**	1.55E-04	1.55E-04	1.55E-04	**4.42E-01**	1.55E-04	**4.42E-01**	1.55E-04	1.55E-04
F4	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F5	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F6	**5.74E-01**	1.55E-04	1.55E-04	1.55E-04	6.22E-04	1.55E-04	**9.59E-01**	1.55E-04	6.99E-03
F7	6.22E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	6.22E-04	1.55E-04	1.86E-03
F21	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F22	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F23	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	**4.42E-01**	1.55E-04	1.55E-04
Count	**2**	**0**	**0**	**0**	**1**	**0**	**3**	**0**	**0**

Table 15. CEC2017 (500D) Wilcoxon signed-rank test results of the compared algorithms.

	AOO	**RLNNA**	**RLTLBO**	**NCOA**	**NSSA**	**MMPA**	**SBOA**	**LRA-CMA-ES**	**mLSHADE-SPACMA**
F1	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F2	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F3	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F4	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F5	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F6	**5.74E-01**	1.55E-04	1.55E-04	1.55E-04	1.09E-03	1.55E-04	**9.59E-01**	1.55E-04	**2.79E-01**
F7	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	3.11E-04	1.55E-04	**1.30E-01**
F21	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F22	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04	1.55E-04
F23	**1.05E-01**	1.55E-04	1.55E-04	1.55E-04	**1.05E-01**	6.22E-04	**1.05E-01**	1.55E-04	2.81E-02
Count	**2**	**0**	**0**	**0**	**1**	**0**	**2**	**0**	**2**

5. BP neural network optimization

In the field of machine learning, optimizing backpropagation (BP) neural networks remains a critical challenge due to their widespread use in regression and prediction tasks across various domains such as energy forecasting, financial modeling, and industrial control. While BP networks are capable of approximating complex nonlinear relationships, their performance heavily relies on the appropriate setting of weights and biases. Traditional gradient-based training methods are often prone to falling

into local minima and present sensitivity to initial parameters and slow convergence, especially in networks with large architectures. These limitations necessitate the use of metaheuristic optimization algorithms to efficiently navigate the high-dimensional parameter space, achieve better generalization, and enhance the reliability of the trained models. This study addresses these inherent difficulties by employing and comparing multiple advanced optimization strategies to improve the learning process and output accuracy of BP networks.

5.1. Mathematical model of BP neural network optimization

5.1.1. Neural network forward propagation

The output of the feedforward neural network is computed through the following layer-wise transformations:

$$a = \sigma(W_1 x + b_1) . \tag{37}$$

where x represents the input feature vector, W_1 denotes the weight matrix connecting the input layer to the hidden layer, and b_1 is the bias vector of the hidden layer. The function σ refers to the activation function.

$$y = W_2 a + b_2 \tag{38}$$

where the hidden layer output a is then propagated through the output layer using weight matrix W_2 and bias b_2, yielding the final prediction y.

5.1.2. Optimization objective function

The goal of training is to minimize the total absolute error between predictions and true values:

$$error = \sum_{i=1}^{N} | y_i - \widehat{y_i} | \tag{39}$$

where the loss function computes the sum of absolute differences across all N training samples. Here, y_i is the predicted value from the neural network for the i_{th} sample, and $\widehat{y_i}$ is the corresponding ground truth value. Minimizing this error encourages the model to produce predictions that are as close as possible to the actual targets.

5.1.3. Search space formulation

The optimization algorithm searches within a bounded continuous space:

$$x = [vec(W_1), b_1, vec(W_2), b_2] \in [-5, 5]^D , \tag{40}$$

where the dimension D is determined by:

$$D = n_i \cdot n_h + n_h + n_h \cdot n_o + n_o , \tag{41}$$

where the solution vector x encodes all trainable parameters of the network: flattened weights and biases from both hidden and output layers. The dimension D depends on the network structure, specifically the number of input n_i, hidden n_h, and output n_o neurons. Each variable in x is constrained to the interval [−5,5] to limit the search space and avoid extreme parameter values.

5.2. Parameter settings

Table 16. BP neural network optimization of relevant parameters.

Parameter	Value
Input nodes	2
Output nodes	1
Training set	190
Testing set	10
Population size	100
Search space	[-5,5]
Training epochs	20
Learning rate	0.01
Target error	0.00001
Display frequency	100

In order to fully demonstrate the algorithm's ability to solve high-dimensional problems, we choose an hidden layer equal to 25, 50, 125, and 250, that is, the dimension is approximately equal to 100, 200, 500, and 1000.

5.3. Results and analysis

In this experiment, the performance of different algorithms is compared by monitoring their optimum, average value, and running time in optimizing BP neural networks.

In the selection of the comparison algorithm, we choose the basic algorithm AOO, RLNNA representing RL optimization, and NCOA using DN strategy. The LSHADE-SPACMA variant mLSHADE-SPACMA and the CMA-ES variant LRA-CMA-ES are two algorithms that can represent the latest state-of-the-art algorithms. The four algorithms that performed most poorly in the CEC test and did not adapt to the high-dimensional optimization problem were omitted.

Table 17. The best value of the algorithms in different dimensions.

	100D	200D	500D	1000D
RLDN-AOO	**1.63E-02**	**2.02E-02**	2.08E-02	**1.30E-02**
AOOv4	2.50E-02	1.20E-01	2.81E-01	1.24E-01
RLNNA	1.40E-01	2.09E-01	5.30E-01	4.86E-01
NCOA	3.31E-01	3.98E-01	1.11E+00	1.79E+00
mLSHADE-SPACMA	1.82E-02	2.79E-02	**1.47E-02**	1.32E-02
LRA-CMA-ES	2.01E-01	2.01E+00	2.28E+00	2.66E+00

From Table 17, it can be seen that in terms of optimal values, the RLDN-AOO algorithm ranks first in 100, 200, and 1000 dimensions, and ranks second after mLSHADE-SPACMA in 500 dimensions. It shows the excellent ability of algorithms in optimizing high-dimensional engineering.

Table 18. Average of the algorithms in different dimensions.

	100D	**200D**	**500D**	**1000D**
RLDN-AOO	**1.87E-02**	**2.28E-02**	2.17E-02	2.88E-02
AOOv4	5.91E-02	4.15E-01	4.32E-01	2.11E-01
RLNNA	1.64E-01	2.92E-01	5.51E-01	5.30E-01
NCOA	3.31E-01	4.57E-01	1.25E+00	1.98E+00
mLSHADE-SPACMA	2.67E-02	3.03E-02	**1.75E-02**	**1.43E-02**
LRA-CMA-ES	2.53E-01	2.12E+00	2.30E+00	2.82E+00

Table 18 shows the average value of algorithm optimization, which shows the excellent ability and stability of the RLDN-AOO algorithm in optimizing high-dimensional problems. The adaptability and stability of high-dimensional optimization engineering make the proposed algorithm competitive among similar RL optimization algorithms.

Table 19. Time result of the algorithms in different dimensions.

	100D	**200D**	**500D**	**1000D**
RLDN-AOO	3324.57	3388.09	4576.23	6227.13
AOOv4	3289.88	3361.73	4649.66	6458.58
RLNNA	3301.28	3364.53	4680.43	6522.26
NCOA	3287.73	3364.49	4752.48	6704.48
mLSHADE-SPACMA	**2095.56**	**2274.29**	**3018.87**	**4095.75**
LRA-CMA-ES	3284.90	3525.40	5829.55	9451.95

Table 19 shows the runtime of the algorithm. Although the RLDN-AOO algorithm performs well in terms of optimal and average values, there is still a gap between RLDN-AOO and the champion algorithm, mLSHADE-SPACMA, in terms of time cost, which is also a potential problem of RLDN-AOO.

6. Conclusions

This study proposes RLDN-AOO, an enhanced variant of the original AOO algorithm, specifically designed to address its limitations in HDO problems. The RLDN-AOO algorithm combines the strategy selection mechanism of the MAB with the adaptive division mechanism of dynamic niching, and proposes an RLDN collaborative optimization framework, which endows particles with high mobility and maintains population diversity in the later stages of iteration. At the same time, the optimization strategy in the AOO algorithm is retained as the decision-making arm of the MAB mechanism, which forms a co-evolution of spatial subdivision guidance strategy focus and strategy optimization feeding structure adjustment, and greatly enhances the adaptability in strategy switching. In addition, dynamic parameter adjustment across multiple niche stages enables more

strategic position updates of particles.

In order to comprehensively and quantitatively assess the RLDN-AOO algorithm's performance, we employed the CEC2017 benchmarking suite. The findings indicate that RLDN-AOO demonstrates strong capabilities within high-dimensional, complex optimization landscapes, exhibiting superior convergence rates and solution precision compared to numerous established algorithms. Moreover, despite its notable strengths in solution quality and robustness, the long running time of the RLDN-AOO algorithm is one of the bottlenecks that may occur in practical applications. Specifically, the RLDN-AOO algorithm adopts a more reasonable dynamic selection mechanism, the RLDN collaborative optimization framework, in the optimization process, which increases the computational burden. In addition, efforts will be made to adapt and deploy the algorithm in real-world applications, where practical constraints and dynamic environments pose additional challenges.

Finally, we apply RLDN-AOO to BP neural network optimization problems, and in this high-dimensional optimization project, RLDN-AOO fully demonstrates its exploration and exploitation capabilities, further confirming the excellent performance of the algorithm.

Furthermore, an interesting avenue for future research lies in integrating optimization frameworks with fractional-order models and numerical schemes, which can preserve important physical properties and structural characteristics of dynamical systems. Recent progress in this direction includes robust orthogonal Gauss collocation methods exhibiting superconvergence for fourth-order sub-diffusion problems [40]. Such approaches could provide valuable insights into enhancing the applicability of RLDN-AOO to optimization tasks constrained by physical laws.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by Beijing College Students Innovation and Entrepreneurship Training Project 2025.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. R. Wang, W. Wang, L. Xu, J. S. Pan, S. C. Chu, An adaptive parallel arithmetic optimization algorithm for robot path planning, *J. Adv. Transp.*, **2021** (2021), 1–22. https://doi.org/10.1155/2021/3606895
2. J. Zhang, H. Tran, G. Zhang, Accelerating reinforcement learning with a Directional-Gaussian-Smoothing evolution strategy, *Electron. Res. Arch.*, **29** (2021), 4119–4135. https://doi.org/10.3934/era.2021075

3. H. Xu, An adaptive initialization and reproduction-based evolutionary algorithm for tackling Bi-objective feature selection in classification, *Symmetry*, **17** (2025), 671. https://doi.org/10.3390/sym17050671
4. Z. Zhang, Z. Wei, B. Nie, Y. Li, Discontinuous maneuver trajectory prediction based on HOA-GRU method for the UAVs, *Electron. Res. Arch.*, **30** (2022), 3111–3129. https://doi.org/10.3934/era.2022158
5. K. Liu, Z. He, H. Zhang, X. Yang, A Crank–Nicolson ADI compact difference scheme for the three-dimensional nonlocal evolution problem with a weakly singular kernel, *Comput. Appl. Math.*, **44** (2025), 164. https://doi.org/10.1007/s40314-025-03125-x
6. F. Caraffini, F. Neri, G. Iacca, Large scale problems in practice: the effect of dimensionality on the interaction among variables, in *Applications of Evolutionary Computation*, (2017), 636–652. https://doi.org/10.1007/978-3-319-55849-3_58
7. S. Mahdavi, M. E. Shiri, S. Rahnamayan, Metaheuristics in large-scale global continues optimization: A survey, *Inf. Sci.*, **295** (2015), 407–428. https://doi.org/10.1016/j.ins.2014.10.042
8. Q. Yang, W. N. Chen, Z. Yu, T. Gu, Y. Li, H. Zhang, Adaptive multimodal continuous ant colony optimization, *IEEE Trans. Evol. Comput.*, **21** (2017), 191–205. https://doi.org/10.1109/TEVC.2016.2591064
9. W. Zhang, J. Zhao, H. Liu, L. Tu, Cleaner fish optimization algorithm: a new bio-inspired meta-heuristic optimization algorithm, *J. Supercomput.*, **80** (2024), 17338–17376. https://doi.org/10.1007/s11227-024-06105-w
10. J. Bai, H. Nguyen-Xuan, E. Atroshchenko, G. Kosec, L. Wang, M. Abdel Wahab, Blood-sucking leech optimizer, *Adv. Eng. Software*, **195** (2024), 103696. https://doi.org/10.1016/j.advengsoft.2024.103696
11. Y. Fu, D. Liu, J. Chen, L. He, Secretary bird optimization algorithm: a new metaheuristic for solving global optimization problems, *Artif. Intell. Rev.*, **57** (2024), 123. https://doi.org/10.1007/s10462-024-10729-y
12. J. Jian, Z. Zhan, J. Zhang, Large-scale evolutionary optimization: a survey and experimental comparative study, *Int. J. Mach. Learn. Cybern.*, **11** (2020), 729–745. https://doi.org/10.1007/s13042-019-01030-4
13. R. Wang, R. Hu, F. Geng, L. Xu, S. C. Chu, J. S. Pan, et al., The Animated Oat Optimization Algorithm: A nature-inspired metaheuristic for engineering optimization and a case study on Wireless Sensor Networks, *Knowledge-Based Syst.*, **318** (2025), 113589. https://doi.org/10.1016/j.knosys.2025.113589
14. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95 - International Conference on Neural Networks*, (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
15. S. Fu, C. Ma, K. Li, C. Xie, Q. Fan, H. Huang, et al., Modified LSHADE-SPACMA with new mutation strategy and external archive mechanism for numerical optimization and point cloud registration, *Artif. Intell. Rev.*, **58** (2025), 72. https://doi.org/10.1007/s10462-024-11053-1
16. H. Nakagawa, Y. Yamada, K. Uchida, S. Shirakawa, Learning rate adaptation CMA-ES for multimodal and noisy problems with low effective dimensionality, in *GECCO '25 Companion: Proceedings of the Genetic and Evolutionary Computation Conference Companion*, (2025), 491–494. https://doi.org/10.1145/3712255.3726725
17. M. I. Jordan, T. M. Mitchell, Machine learning: Trends, perspectives, and prospects, *Science*, **349** (2015), 255–260. https://doi.org/10.1126/science.aaa8415

18. Y. Zhang, Neural network algorithm with reinforcement learning for parameters extraction of photovoltaic models, *IEEE Trans. Neural Networks Learn. Syst.*, **34** (2023), 2806–2816. https://doi.org/10.1109/TNNLS.2021.3109565
19. D. Wu, S. Wang, Q. Liu, L. Abualigah, H. Jia, An improved teaching-learning-based optimization algorithm with reinforcement learning strategy for solving optimization problems, *Comput. Intell. Neurosci.*, **2022** (2022), 1535957. https://doi.org/10.1155/2022/1535957
20. R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, *IEEE Trans. Neural Networks*, **9** (1998), 1054–1054. https://doi.org/10.1109/TNN.1998.712192
21. S. Ding, W. Du, X. Zhao, L. Wang, W. Jia, A new asynchronous reinforcement learning algorithm based on improved parallel PSO, *Appl. Intell.*, **49** (2019), 4211–4222. https://doi.org/10.1007/s10489-019-01487-4
22. K. Jiao, J. Chen, B. Xin, L. Li, A reference vector based multiobjective evolutionary algorithm with Q-learning for operator adaptation, *Swarm Evol. Comput.*, **76** (2023), 101225. https://doi.org/10.1016/j.swevo.2022.101225
23. H. P. Chan, The multi-armed bandit problem: An efficient nonparametric solution, *Ann. Stat.*, **48** (2020), 346–373. https://doi.org/10.1214/19-AOS1809
24. H. Flynn, D. Reeb, M. Kandemir, J. Peters, PAC-Bayesian lifelong learning for multi-armed bandits, *Data Min. Knowl. Discovery*, **36** (2022), 841–876. https://doi.org/10.1007/s10618-022-00825-4
25. X. Zhang, Q. Kang, Q. Tu, J. Cheng, X. Wang, Efficient and merged biogeography-based optimization algorithm for global optimization problems, *Soft Comput.*, **23** (2019), 4483–4502. https://doi.org/10.1007/s00500-018-3113-1
26. L. Lyu, F. Yang, MMPA: A modified marine predator algorithm for 3D UAV path planning in complex environments with multiple threats, *Expert Syst. Appl.*, **257** (2024), 124955. https://doi.org/10.1016/j.eswa.2024.124955
27. A. Faramarzi, M. Heidarinejad, S. Mirjalili, A. H. Gandomi, Marine Predators Algorithm: A nature-inspired metaheuristic, *Expert Syst. Appl.*, **152** (2020), 113377. https://doi.org/10.1016/j.eswa.2020.113377
28. S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
29. X. Zhang, S. Wen, Hybrid whale optimization algorithm with gathering strategies for high-dimensional problems, *Expert Syst. Appl.*, **179** (2021), 115032. https://doi.org/10.1016/j.eswa.2021.115032
30. R. Storn, K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.*, **11** (1997), 341–359. https://doi.org/10.1023/A:1008202821328
31. C. Li, Y. Zhai, V. Palade, W. Fang, H. Lu, L. Mao, et al., Diversity-based adaptive differential evolution algorithm for multimodal optimization problems, *Swarm Evol. Comput.*, **93** (2025), 101869. https://doi.org/10.1016/j.swevo.2025.101869
32. C. Rim, S. Piao, G. Li, U. Pak, A niching chaos optimization algorithm for multimodal optimization, *Soft Comput.*, **22** (2018), 621–633. https://doi.org/10.1007/s00500-016-2360-2
33. X. Li, M. G. Epitropakis, K. Deb, A. Engelbrecht, Seeking multiple solutions: an updated survey on niching methods and their applications, *IEEE Trans. Evol. Comput.*, **21** (2017), 518–538. https://doi.org/10.1109/TEVC.2016.2638437

34. K. A. Luong, M. A. Wahab, J. H. Lee, Simultaneous imposition of initial and boundary conditions via decoupled physics-informed neural networks for solving initial-boundary value problems, *Appl. Math. Mech.*, **46** (2025), 763–780. https://doi.org/10.1007/s10483-025-3240-7
35. Y. Guo, C. Wang, S. Han, G. Kosec, Y. Zhou, L. Wang, et al., A deep neural network model for parameter identification in deep drawing metal forming process, *J. Manuf. Processes*, **133** (2025), 380–394. https://doi.org/10.1016/j.jmapro.2024.11.067
36. D. Q. Nguyen, K. Q. Tran, T. D. Le, M. A. Wahab, H. Nguyen-Xuan, A data-driven uncertainty quantification framework in probabilistic bio-inspired porous materials (Material-UQ): An investigation for RotTMPS plates, *Comput. Methods Appl. Mech. Eng.*, **435** (2025), 117603. https://doi.org/10.1016/j.cma.2024.117603
37. A. C. Kocabıçak, C. Wang, S. Han, H. Nguyen-Xuan, G. Kosec, L. Wang, et al., A deep neural network approach to predict dimensional accuracy of thin-walled tubes in backward flow forming plasticity process, *J. Manuf. Processes*, **141** (2025), 59–80. https://doi.org/10.1016/j.jmapro.2025.02.044
38. L. Nguyen-Ngoc, Q. Nguyen-Huu, G. De Roeck, T. Bui-Tien, M. Abdel-Wahab, Deep neural network and evolved optimization algorithm for damage assessment in a truss bridge, *Mathematics*, **12** (2024), 2300. https://doi.org/10.3390/math12152300
39. Z. Wang, X. Huang, D. Zhu, S. Yan, Q. Li, W. Guo, Learning sparrow search algorithm of hybrids boundary processing mechanisms, *J. Beijing Univ. Aeronaut. Astronaut.*, **50** (2024), 286–298. https://doi.org/10.13700/j.bh.1001-5965.2022.0195
40. X. Yang, Z. Zhang, Superconvergence analysis of a robust orthogonal gauss collocation method for 2D fourth-order subdiffusion equations, *J. Sci. Comput.*, **100** (2024), 62. https://doi.org/10.1007/s10915-024-02616-z

AIMS Press