Expert Systems With Applications 243 (2024) 122942

Contents lists available at ScienceDirect

Expert Systems With Applications

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

An adaptive differential evolution with opposition-learning based diversity enhancement

Zhenghao Song, Chongle Ren, Zhenyu Meng *

Institute of Artificial Intelligence, Fujian University of Technology, Fuzhou, China

ARTICLE INFO

Keywords:
Differential evolution
Parameter control
Diversity enhancement
Opposition learning

ABSTRACT

Differential Evolution (DE), as a powerful population-based stochastic optimization algorithm, has attracted the attention of researchers from various fields due to its advantages such as simple operation, strong robustness, and few control parameters. However, many existing DE variants often suffer from drawbacks such as premature convergence and stagnation when solving complicated optimization problems. In view of the aforementioned issues, this paper proposes an adaptive DE with opposition learning-based diversity enhancement (OLBADE). The main contributions can be summarized as follows: Firstly, a new adaptive parameter control is proposed with a non-linear weighting strategy incorporating into the framework of parameter adaptation. Secondly, a donor vector perturbation strategy is introduced to complement existing strategy for increasing population diversity. Thirdly, a novel stagnation indicator is proposed, and then opposition learning strategy is employed to renew stagnated individuals in the population when stagnation occurs. OLBADE is compared with five excellent DE variants under a large test-bed containing CEC2013, CEC2014, CEC2017 and CEC2022 test suites to verify its effectiveness. In addition, OLBADE is applied in parameter identification problem of photovoltaic model to verify its feasibility. Experimental results demonstrate that OLBADE achieves higher solution accuracy, faster convergence speed and better stability.

1. Introduction

Meta-heuristic algorithms are a class of optimization algorithms that are inspired by natural and social phenomena, such as evolution, swarm behavior, physics, and so on Meng et al. (2016, 2022) and Yang (2010). Evolutionary algorithms (EAs), such as Genetic Algorithm (GA) (Holland, 1992b), Evolution Strategy (ES) (Back, 1996) and Differential Evolution (DE) (Storn & Price, 1997), are a subset of meta-heuristic algorithms, which mimic the process of natural selection, reproduction, and mutation to iteratively search for optimal solutions of various optimization problems.

Among those EAs, DE, proposed by Storn and Price in 1995, has attracted considerable attention owing to its advantages such as few parameters, easy implementation and strong global search capacity (Meng et al., 2023). Since its inception, DE and its variants have been widely applied in various fields such as complex scheduling (Zhang et al., 2019), neural networks (Son et al., 2021), feature selection (Zhang et al., 2020), network planning (Okagbue et al., 2019), etc. However, similar to other EAs, classic DE struggles to balance global exploration capability and local exploitation capabilities for complicated optimization problems. Consequently, it may suffer from premature convergence or stagnation as the evolution proceeds. In order to achieve exploration and exploitation balance, numerous researchers have made improvements to the framework of DE from different perspectives.

In DE, control parameters including scale factor, crossover rate and population size attach great importance to its overall performance (Deng et al., 2021). Finding an appropriate parameter setting for a certain problem often requires prior knowledge and time-consuming trials. Furthermore, the parameter setting suitable for addressing one issue might yield poor performance when applied to a different problem. Therefore, researchers have developed numerous parameter control mechanisms to solve the above problems (Meng & Pan, 2019; Meng et al., 2019). The parameter control strategies can be categorized into three types: deterministic parameter control, adaptive parameter control and self-adaptive parameter control (Dragoi & Dafinescu, 2016). For deterministic parameter control, parameters are adjusted based on a deterministic rule. Holland (1992a) developed a dynamic strategy for changing the value of scale factor. In Elsayed et al. (2013), a DE variant named DE-APS was proposed, in which control parameters are randomly selected from a pre-defined parameter set. However, deterministic parameter control can avoid parameter tuning to some extend, but it fails to utilize the feedback information of evolution. To mitigate the deficiency, the adaptive parameter control is developed

* Corresponding author.
E-mail address: mzy1314@gmail.com (Z. Meng).

https://doi.org/10.1016/j.eswa.2023.122942
Received 12 July 2023; Received in revised form 7 December 2023; Accepted 11 December 2023
Available online 14 December 2023

based on the feedback information. Qin et al. (2008) proposed a popular DE variant named SaDE, where offspring generation strategies and control parameters are adjusted by learning from information of population. As for self-adaptive parameter control, the parameters are assigned to each individual and evolved along with the population. Success-history based parameter adaptation, as one of the most popular self-adaptive control strategies, was proposed in Tanabe and Fukunaga (2013), in which information of successful individuals are utilized to adapt parameters. Awad et al. (2017) proposed a DE variant named LSHADE-cnEpSin, in which ensemble of sinusoidal waves and covariance matrix learning are adopted. jSO, an improved DE variant, was proposed in Brest et al. (2017) for single objective real-parameter optimization. Zeng et al. (2022) incorporated a weighting strategy based squared Euclidean distance to update control parameters. Although parameter adaptation strategies can greatly improve the performance of DE, but they fails to resolve the problems of premature convergence and population stagnation resulted from reduced population diversity.

Population diversity is a crucial metric in population-based optimization algorithms (Osuna-Enciso et al., 2022). Although population diversity inevitably decreases as population converges to optimal points, excessively low population diversity will result in premature convergence and stagnation. In order to prevent problems resulted by population diversity, measuring the diversity is the first step. Akhila et al. (2016) analyzed different schemes for measuring the diversity of population which utilized three types of information: changes in generic features, fitness information and population distribution. In Yang et al. (2014), a diversity measurement method based on distribution was proposed to detecting stagnation and premature convergence. After determining the population diversity, some actions are required to take when the population diversity is lower than a certain threshold. Song and Meng (2023) proposed a dimension crossing mechanism, which uses two different sets of dimension crossing mechanisms for the stagnant individuals and the dimensions of stagnant individuals to jump out of the local optimum. In Zeng and Zhang (2022), a probability model is based on evolutionary state to improve the greedy selection operation with aim to enhance population diversity. Li et al. (2023) developed two intervention operations including dispersion and aggregation to tackle premature convergence and population stagnation, respectively. In Meng (2023) and Meng and Yang (2022), diversity enhancement mechanism were proposed to update stagnant individuals detected by stagnation indicator. To avoid premature convergence, a stochastic mixed mutation strategy was proposed in Tian and Gao (2019) with a cosine perturbation strategy.

Based on the above analyses, the search capability is greatly enhanced by modifications proposed in those DE variants, but there still exist shortcomings: (1) improper adaptation for scale factor F, which often results in unbalanced exploration and exploitation; (2) over-dependence on a single trial vector generation strategy, which usually leads to low population diversity; and (3) ineffective stagnation intervention mechanism, which usually fails to tackle the stagnation problem. Therefore, an adaptive DE with opposition learning-based diversity enhancement (OLBADE) is proposed in the paper, and the highlights of OLBADE are presented as follows:

1. A parameter adaptation strategy is proposed, which is separated into two stages based on the evolutionary states. In addition, a weighting strategy based on non-linear fitness increment is proposed to adapt control parameters.
2. A donor vector perturbation strategy is introduced to complement existing trial vector generation strategy to enhance population diversity.
3. A diversity enhancement mechanism based on opposition learning is proposed to update stagnant individuals which are detected by stagnation indicator.
4. A large test suite containing 100 benchmark functions from CEC2013, CEC2014, CEC2017 and CEC2022 test suites is employed to verify the performance of OLBADE.

The remaining parts of this paper are organized as follows: Section 2 presents the introduction of the basic DE algorithm. Section 3 elaborates the proposed OLBADE algorithm. In Section 4, performance comparison between OLBADE and five state-of-the-art DE variants are provided. Section 5 provides the experiment results on PV model. Finally, Section 6 concludes the paper.

2. The basic operations of DE

Classic DE comprises two basic operations including initialization and evolution. The population is created in the stage of initialization. After generating the initial population, individuals are underwent three evolutionary operations including mutation, crossover and selection, which are repeated until a certain termination criterion is satisfied.

Initialization: During the stage of initialization, individuals are generated based on uniform distribution within decision space, in which the lower and upper bounds are predefined to limit the distribution of individuals. The ith individual from the initial population can be described as Eq (1).

$$X_i = [X_{i,1}, X_{i,2}, \ldots, X_{i,D}], \qquad i = 1, 2, \ldots, PS \tag{1}$$

where D denotes the dimension of the individual. The jth dimension of ith individual is generated as follows:

$$X_{i,j} = l_j + rand(0,1) \cdot (u_j - l_j) \tag{2}$$

where u_j and l_j are the upper and lower bounds of the jth dimension of the individual; $rand(0,1)$ is a random number within the interval [0,1].

Mutation: During mutation operation, a base vector $X_{i,G}$ is combined with one difference vector or several differential vectors to produce a mutant vector $V_{i,G}$. The popular mutation strategy DE/current-to-pbest/1 proposed in Zhang and Sanderson (2009) is adopted in OLBADE owing to its balanced exploration and exploitation capacities, as shown below:

$$V_{i,G} = X_{i,G} + F \cdot (X^p_{pbest,G} - X_{i,G}) + F \cdot (X_{r_1,G} - \widetilde{X}_{r_2,G}) \tag{3}$$

where X^p_{pbest} is randomly selected from top $p \cdot 100\%$ individuals and p is used to control greediness of the strategy. $\widetilde{X}_{r_2}$ is selected from the union $\mathbf{P} \cup \mathbf{A}$, where $\mathbf{P}$ denotes the current population and $\mathbf{A}$ denotes the archived population.

After mutation, some components of donor vector may exceed the predefined boundary constraints. A simple method which resets violating components is used since the focus of this paper is not on constrained problems. The method is shown as follows:

$$\begin{aligned} V_{i,j,G} &= (l_j + X_{i,j,G})/2, \ if \ V_{i,j,G} < l_j \\ V_{i,j,G} &= (u_j + X_{i,j,G})/2, \ if \ V_{i,j,G} > u_j \end{aligned} \tag{4}$$

Crossover: By crossover operation, a trial vector $U_{i,G}$ is generated by exchanging dimension between the donor vector $V_{i,G}$ and the target vector $X_{i,G}$.

$$U_{j,i,G} = \begin{cases} V_{j,i,G}, & if \ rand(0,1) \le CR_i \ or \ j = j_{rand} \\ X_{j,i,G}, & otherwise \end{cases} \tag{5}$$

where CR is the crossover rate, and j_{rand} is a random integer within [1, D] to ensure that at least one dimension of donor vector is utilized (Meng & Zhang, 2023).

Selection: The selection in DE is used to determine which vector will survive to the next generation, as shown below:

$$X_{i,G+1} = \begin{cases} U_{i,G}, & if \ f(U_{i,G}) < f(X_{i,G}) \\ X_{i,G}, & otherwise \end{cases} \tag{6}$$

3. The proposed OLBADE algorithm

In this section, detailed description of OLBADE is presented into three parts: the first part introduces the new trial vector generation strategy based on donor vector perturbation; the second part presents the adaptive parameter control strategy; and the third part provides the opposition learning-based diversity enhancement mechanism.

3.1. Donor vector perturbation strategy

It is widely acknowledged that the selection of trial vector generation strategy determines the overall performance of DE. Typically, the generation of trial vector relies on mutation operation and crossover operation (Meng & Chen, 2023). Although trial vector generation strategy "DE/target-to-pbest/1/bin" has shown excellent performance in many DE-based variants, its search capacity will deteriorate when premature convergence and stagnation occur.

The premature convergence phenomenon can occur during the iteration process of DE, resulting in a decrease in the search ability of the mutation operation. In such cases, traditional trial vector generation strategy fail to promote individuals to escape from local optima. Perturbation strategy, as an effective way to assist individuals in jumping out of local optima, can be used to complement current trial vector generation strategy to enhance population diversity (Cheng et al., 2019).

In the proposed perturbation strategy, we employ a simple competition mechanism to control the frequency of perturbation and a dynamic parameter β to adjust the degree of perturbation. The value of β is calculated via following equation:

$$\beta = 2 - \frac{nfes}{nfes_{max}} \cdot cos(2 \cdot \pi \cdot rand) \tag{7}$$

where $nfes$ and $nfes_{max}$ denote the current number and maximum number of fitness evaluations, respectively. $cos(\cdot)$ is used to control the scope of search.

The higher value of β during the early stages forces strong perturbation, and thus slows down the convergence and promote population diversity. As evolution proceeds, the value of β decreases leading to more information of original donor vector being maintained.

It is evident that both trial vector and donor vector will move towards a common objective. Simultaneously, trial vector needs to update their positions relative to this objective. As donor vector moves towards its trial vector, the trial vector must continually approach the optimal point. Occasionally, trial vector must tolerate sub-optimal position in order to discover better solution. This phenomenon gives rise to two distinct methods of trial vector generation, as presented in Algorithm 1.

Algorithm 1 Donor vector perturbation strategy

```
Input: Number of dimensions D, boundary constraints [l_i, u_i],
  Control parameters F and CR;
Output: Trial vector U_{j,i,G};
Execute Eq. (3)
Generate j_rand = randint (1, D);
for j = 1 to D do
    if j = j_rand or rand(0, 1) < CR_i then
        U_{j,i,G} = V_{j,i,G};
    else
        if rand < τ_1 then
            U_{j,i,G} = X_{j,i,G};
        else
            if rand > τ_2 then
                U_{j,i,G} = β · (X_{j,i,G} − U_{j,i,G}) + X_{j,i,G};
            else
                U_{j,i,G} = β · (X_{j,i,G} − U_{j,i,G}) − X_{j,i,G};
            end if
        end if
    end if
end for
```

In Algorithm 1, two parameters τ_1 and τ_2 control the frequency of perturbation and the way of perturbation, respectively. τ_1 is set to 0.8 to maintain a high level of perturbation and τ_2 is set to 0.5.

3.2. Adaptive parameter control

Success-history based parameter adaptation proposed in Tanabe and Fukunaga (2013) utilizes the information of successful parameters to update parameters and employs a memory pool to improve robustness. The parameter adaptation strategy has been incorporated into many DE-based variants and yielded satisfactory performance in real-world problems and IEEE CEC competition series. Therefore, our parameter adaptation strategy is proposed based on the success-history based parameter adaptation.

In original success-history based parameter adaptation, the generation of F obeys Cauchy distribution during evolutionary process. However, the generation scheme of F based on Cauchy distribution is highly fluctuating, as shown in Fig. 1. As F increases, population diversity can be enhanced, but the convergence speed will decrease. Conversely, with a smaller F, the convergence speed will increase, albeit at the expense of reduced population diversity. Based on the above considerations, the value of F is supposed to remain stable during the early stage of evolution and fluctuate rapidly during the later stage. To simulate this adaptive behavior, a hybrid method combining the logistic and sine functions is proposed to adjust F at the initial stages. Subsequently, the Cauchy distribution is employed to adjust F. The generation of F and CR is presented as follows:

$$\varepsilon = 0.2 \cdot sin(\pi \cdot rand_i - 0.8) \tag{8}$$

$$F_i = \begin{cases} \dfrac{\mu_F}{min(\mu_F) + \mu_F \cdot (e^{-\mu_F \cdot G} + 1)} - \varepsilon, & if\ nfes < \perp \\ randc_i(\mu_{F,r_i}, 0.1), & otherwise \end{cases} \tag{9}$$

$$CR_i = \begin{cases} 0, & if\ \mu_{CR,r_i} = \emptyset \\ randn_i(\mu_{CR,r_i}, 0.1), & otherwise \end{cases} \tag{10}$$

$$CR_i = \begin{cases} max(CR_i, 0.6), min(CR_i, 1) & if\ nfes < \perp \\ max(CR_i, 0), min(CR_i, 1), & otherwise \end{cases} \tag{11}$$

where $\langle \mu_F, \mu_{CR} \rangle$ is a parameter pair randomly selected from memory pool. $randc_i$ and $randn_i$ denote Cauchy distribution and Gaussian distribution, respectively. $\perp$ denotes the threshold between two stages of evolution for generating F.

The parameter pair of memory pool will update at the end of each generation. In the success history-based parameter adaptation strategy, both μ_F and μ_{CR} are updated using weighted Lehmer mean, as shown in Eq. (13). The weight was calculated based on the fitness improvement between trial vector and its target vector. However, if the calculation of weight is solely dependent on fitness improvement, it may lead to prematrue convergence, especially in high-dimensional search space (Stanovov et al., 2021; Viktorin et al., 2019). To mitigate the problem, we propose a novel weighting strategy, where two non-linear methods based on sine function and logarithmic spiral are used. The first calculation mode involves finding the current optimal fitness value by narrowing a circle while exploring along a spiral path. To simulate this situation, we assume a 50% probability of choosing the mechanism of shrinking circles or the spiral model to record fitness value improvements during the optimization process. Subsequently, through the corresponding weighting strategy, it can be used to guide the search direction during the iteration process.

$$w_k = \begin{cases} f(X_i) - rand_i \cdot (1 - sin(2 \cdot \frac{G}{G_{max}})) \cdot \Delta f, & if\ rand < 0.5 \\ f(X_i) + \Delta f \cdot e^{rand} \cdot (cos(2 \cdot \pi \cdot rand)), & otherwise \end{cases} \tag{12}$$

where G and G_{max} denote the current number and maximum number of generations, respectively. $rand$ denotes a random number between [0, 1].

$$\begin{cases} mean_{WL}(S_F) = \dfrac{\sum_{k=1}^{|S_F|} w_k \cdot S_F^2(k)}{\sum_{k=1}^{|S_F|} w_k \cdot S_F(k)} \\ \mu_{F,k,G+1} = \begin{cases} mean_{WL}(S_F), & if\ S_F \neq \emptyset \\ \mu_{F,k,G}, & otherwise \end{cases} \end{cases} \tag{13}$$

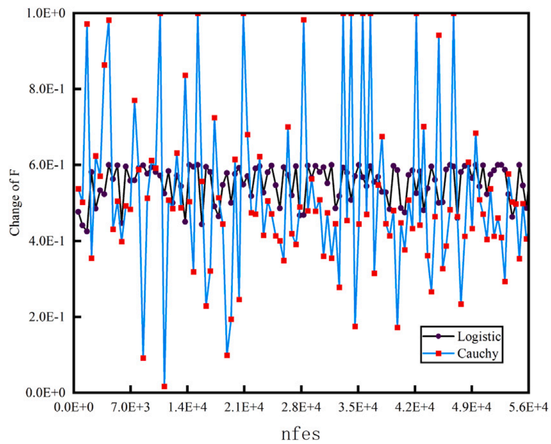


Fig. 1. The generation of F based on Logistic function and Cauchy distribution during the evolution.

$$\begin{cases} mean_{WL}(S_{CR}) = \dfrac{\sum_{k=1}^{|S_{CR}|} w_k \cdot S_{CR}^2(k)}{\sum_{k=1}^{|S_{CR}|} w_k \cdot S_{CR}(k)} \\ \mu_{CR,k,G+1} = \begin{cases} mean_{WL}(S_{CR}), & if\ S_{CR} \neq \emptyset \\ \mu_{CR,k,G}, & otherwise \end{cases} \end{cases} \tag{14}$$

where $f(X_i)$ denotes the fitness value of the ith individual and Δf denotes the fitness difference between the ith trial vector and its target vector.

The linear population size reduction strategy is also employed in OLBADE to enhance the exploitation capacity during the later stage of evolution. The population size is adjusted as follows:

$$PS_{G+1} = round[\frac{PS_{min} - PS_{max}}{nfes_{max}} \cdot nfes + PS_{max}] \tag{15}$$

where PS_{min} and PS_{max} denote the minimum and maximum population size, respectively.

3.3. Opposition learning-based population diversity enhancement strategy

Population diversity, as a main metric for evaluating population state of DE, has been widely analyzed by researchers. The population diversity inevitably decrease as the evolution proceeds when individuals are coalescing into several local optima or global optimum. However, if the diversity is much lower than a certain level, the whole population will suffer from stagnation, thus hindering individuals moving to promising regions. To prevent the stagnation in DE, two main aspects should be taken into full consideration: (1) designing indicator to measure population diversity, and (2) employing intervention methods to renew stagnant individuals. The measurement of diversity can be classified into two types: fitness-based diversity measurement and distribution-based diversity measurement (Osuna-Enciso et al., 2022). To combine the advantages of distribution-based diversity measurement and fitness-based diversity measurement, we propose a novel diversity indicator to examine the population diversity, in which distribution of individuals are calculated using surrogate hypervolumes and fitness improvement of each individuals are recorded by a *counter*. Our approach involves two key calculations: one is to calculate the search space's boundaries, and the other is to calculate the spatial distribution of the population during iterations. The volume associated with the search space boundaries is determined by computing the absolute difference between the lower and upper bounds:

$$V_{lim} = \sqrt{\prod_{j=1}^{D} |u_j - l_j|} \tag{16}$$

The volume obtained from Eq. (16) is initially calculated at the beginning of each iteration. The second super-volume describes the evolution of the population. In this context, we determine the edge vectors by considering the dynamic upper and lower boundaries for each dimension of the entire population as coordinates for the edge vectors:

$$V_{pop} = \sqrt{\prod_{j=1}^{D} |(u_{x_j} - l_{x_j})/2|} \tag{17}$$

where u_{x_j} and l_{x_j} are upper and lower bounds of current population. Finally, the diversity is determined by the ratio between the supervolume of candidate solutions and the supervolume of the search space, as shown below:

$$VOL = \sqrt{V_{pop}/V_{lim}} \tag{18}$$

For the fitness-based measurement, we employed a *counter* to record the number of unsuccessful fitness improvement during evolution, as presented in Algorithm 2.

Algorithm 2 Calculate *counter*

```
Input: fitness value of trial vector f(U_{i,G}), fitness values of target vector f(X_{i,G});
Output: variable counter;
for i = 1 to PS do
    if f(X_{i,G}) > f(U_{i,G}) then
        counter(i) = counter(i) + 1
    else
        counter(i) = 0;
    end if
end for
```

Once the threshold of two diversity measurement methods is satisfied, the individual is considered as stagnant individuals and required to intervene. Opposition-based learning is an strategy firstly proposed in Rahnamayan et al. (2008) to improve search capacity of optimization algorithms. It forces individuals jumping between decision space and improve the population diversity. In addition, it also increases the likelihood of identifying a potential solution closer to the global optimum. In the later stages of evolution, population may suffer from stagnation owing to reduced population diversity. The original opposition-based learning is to generate opposite position based on the current position. When individuals in the population reach a stagnant state, the opportunities for the evolution of individuals in the population are limited. To prevent the opposite position from being far away the optimum, we choose the current best solution as base position, which can increase the possibility of locating the optimal solution. Therefore, the modified opposition-based learning strategy can be used to renew stagnant individual, which is shown as follows:

$$X_{i,j} = rand \cdot (u_{x_j} - l_{x_j}) + (1 - rand) \cdot X_{best,j} \tag{19}$$

The opposition learning-based diversity enhancement strategy is provided in Algorithm 3. The pseudocode of OLBADE is presented in Algorithm 4.

4. Experiment analysis

According to "No Free Lunch Theorem" (Wolpert & Macready, 1997), there is no optimization algorithm which performs well for all types of problems. Some optimization algorithms may yield satisfactory results for one test suite but unsatisfactory results for another. Based on this, more test suites are necessary to fully evaluate the performance of a certain optimization algorithm. Therefore, a large test suite, which contains IEEE CEC2013 (Liang, Qu, Suganthan and Hernández-Díaz, 2013), CEC2014 (Liang, Qu and Suganthan, 2013), CEC2017 (Wu et al.,

Algorithm 3 Opposition learning-based diversity enhancement strategy

Input: Current population X, Global best individual X_{best}, Variables $counter$ and $Vpop$;
Output: Updated population X;
for $i = 1$ **to** PS **do**
    Calculate V_{pop} according to Eq. (16) and Eq. (18);
    $\xi = 0.01$
    if $counter(i) > N$ && $VOL < \xi$ **then**
        Update X_i according to Eq. (19);
        Calculate $f(X_i)$;
        $nfes = nfes + 1$;
        $counter(i) = 0$;
    end if
end for

Algorithm 4 Pseudocode of OLBADE

Input: Number of dimensions D, constraints $[l_i, u_i]$, objective function $f(X)$, maximum number of fitness evaluations $nfes_{max}$;
Output: Global best individual X_{best}, global best fitness value $f(X_{best})$;
while $nfes < nfes_{\max}$ **do**
    Generate F and CR of all individuals;
    for $i = 1$ **to** PS **do**
        Execute Algorithm 1 to generate trial vector $U_{i,G}$;
        Calculate fitness value $f(U_{i,G})$;
    end for
    for $i = 1$ **to** PS **do**
        if $f(U_{i,G}) \leq f(X_{i,G})$ **then**
            $X_{i,G+1} = U_{i,G}$;
        else
            $X_{i,G+1} = X_{i,G}$;
        end if
        if $f(U_{i,G}) < f(X_{i,G})$ **then**
            $X_{i,G} \rightarrow A$;
            $CR_{i,G} \rightarrow S_{CR}$, $F_{i,G} \rightarrow S_F$;
        end if
    end for
    Update external archive A;
    Update memories μ_{CR} and μ_F using Eq. (13), Eq. (14);
    Execute Algorithm 3;
    Calculate PS_{G+1} according to Eq. (15);
    if $PS_G < PS_{G+1}$ **then**
        Individuals are sorted depending on their fitness;
        The worse PS_G - PS_{G+1} individuals are removed from the population;
    end if
    $G = G + 1$;
end while

2017) and CEC2022 (Biedrzycki et al., 2022), is employed in this paper to verify the overall performance of OLBADE. A total of 100 benchmark functions are included in our test suite, in which benchmark functions from CEC2013, CEC2014, CEC2017 and CEC2022 are labeled as f_{a_1}-$f_{a_{28}}$, f_{b_1}-$f_{b_{30}}$, f_{c_1}-$f_{c_{30}}$ and f_{d_1}-$f_{d_{12}}$, respectively. f_{a_1}-f_{a_5}, f_{b_1}-f_{b_3}, f_{c_1}-f_{c_2} and f_{d_1} belong to unimodal function group. f_{a_6}-$f_{a_{20}}$, f_{b_4}-$f_{b_{16}}$, f_{c_3}-f_{c_9} and f_{d_2}-f_{d_5} are simple multimodal functions. $f_{b_{17}}$-$f_{b_{22}}$, $f_{c_{10}}$-$f_{c_{19}}$ and f_{d_6}-f_{d_8} are hybrid functions. $f_{a_{21}}$-$f_{a_{28}}$, $f_{b_{23}}$-$f_{b_{30}}$, $f_{c_{20}}$-$f_{c_{30}}$ and f_{d_9}-$f_{d_{12}}$ are composition functions.

Five DE-based variants including LSHADE-cnEpsin (Awad et al., 2017), jSO (Brest et al., 2017), PaDE-pet (Meng, 2023), ISDE (Tian & Gao, 2019) and TDE (Meng & Yang, 2022), of which LSAHDE ranked the first in IEEE CEC2014 competition and jSO was proposed based on LSAHDE. PaDE-pet, ISDE and TDE are recently proposed variants with excellent performance. It is noteworthy that functions from CEC2013 to CEC2017 are tested on 10D, 30D and 50D optimization and functions from CEC2022 are tested on 10D and 20D optimization.

For CEC2013, CEC2014 and CEC2017 test suites, the maximum number of function evaluations $nfes_{max}$ is set to $10\,000 \cdot D$, where D represents the dimensionality of the objective function. For CEC2022, $nfes_{max}$ is set to 2×10^5 on 10D and 1×10^6 on 20D. Each algorithm is run 51 times for every benchmark function, and then the mean value and standard deviation of the fitness value errors are obtained for comparison. Fitness value error that is smaller than 2.2204E−16 is considered as 0. All experiments are performed on a PC with i7-12700k CPU @2.9 GHz in Matlab 2021a version.

4.1. Optimization accuracy

In this subsection, the performance of OLBADE is verified from the perspective of optimization accuracy by comparing with five state-of-the-art DE variants including LSHADE-cnEpsin, jSO, PaDE-pet, ISDE and TDE, whose parameter settings are in accordance with their original papers, as shown in Table 1. Wilcoxon rank sum test with the significance level $\alpha = 0.05$ is used for performance comparison, in which ">","≈" and "<" denote a certain algorithm obtains "better performance", "similar performance" and "worse performance" by comparing with OLBADE and the optimal result is highlighted in bold.

The experiment results obtained by all algorithm under CEC2013, CEC2014, CEC2017 and CEC2022 across different dimensions are shown in Tables 2 to 12, and summarized in Table 13. For the unimodal functions f_{c_1}-f_{c_2} and f_{d_1}, OLBADE outperforms other DE variants except for f_{c_2}. For the basic multimodal functions, OLBADE obtains the best results in f_{c_3}-f_{c_4} and f_{c_7}-f_{c_9}, while LSHADE-cnEpsin and PaDE-pet achieve the best results in f_{c_5} and f_{c_6}, respectively; OLBADE obtains the best results for f_{d_2} and f_{d_5}, while LSHADE-cnEpsin achieves the optimal result in f_{d_3}. For the hybrid functions ($f_{c_{10}}$-$f_{c_{19}}$, f_{d_6}-f_{d_8}), OLBADE obtains the best results in $f_{c_{12}}$, $f_{c_{14}}$, $f_{c_{15}}$, $f_{c_{19}}$, f_{d_6} and f_{d_7}, while LSHADE-cnEpsin achieves the best results in $f_{c_{10}}$ and $f_{c_{16}}$, jSO and PaDE-pet obtain the best results in $f_{c_{11}}$ and $f_{c_{13}}$ respectively, and TDE achieves the best results in $f_{c_{14}}$, $f_{c_{18}}$, and f_{d_8}. For the composition functions ($f_{c_{20}}$-$f_{c_{30}}$, f_{d_9}-$f_{d_{12}}$), OLBADE obtains the best results in $f_{c_{20}}$, $f_{c_{25}}$, $f_{c_{27}}$-$f_{c_{30}}$, f_{d_9}, and $f_{d_{12}}$. It is obvious that OLBADE achieves improved search capacity and better balance between the global exploration and local exploitation capabilities in the decision space. The reason may be that OLBADE can adjust the search characteristics of the algorithm during the evolution process and avoid stagnation through opposition learning-based diversity enhancement mechanism.

As one of the reviewers suggested, we also employed Wilcoxon rank sum test to show the difference between OLBADE and other algorithms, as shown in Table 13. From Table 13, it is evident that OLBADE outperforms LSHADE-cnEpsin, jSO, ISDE, PaDE-pet, and TDE in a number of benchmark functions from the CEC2013, CEC2014, CEC2017, and CEC2022 test suites. Specifically, OLBADE performs better in 139, 151, 200, 108 and 124 cases, obtains similar performance in 88, 99, 57, 105 and 98 cases, performs worse in 61, 38, 31, 75 and 66 cases, comparing with LSHADE-cnEpsin, jSO, ISDE, PaDE-pet and TDE, respectively.

To provide a clear illustration of the comparison results, statistical results under CEC2017 on 30D optimization and CEC2022 on 20D optimization are provided in Figs. 2 and 3, respectively. The red bar represents the average ranking and the blue one represents the number of best results. From Fig. 2, we can observe that OLBADE ranks the first and obtains 16 best results out of 30 cases. Fig. 3 also demonstrates the superiority of OLBADE under CEC2022 on 20D optimization.

Table 1
Recommended parameter settings of all algorithms.

Algorithms	Parameter settings
OLBADE	$\mu_F = 0.5,\ \mu_{CR} = 0.8,\ p = 0.11,\ r^{rac} = 1.4,\ PS = 25 \cdot ln(D) \cdot \sqrt{D} \sim 4,\ H = 5$
LSHADE-cnEpsin (Awad et al., 2017)	$\mu_{F_k} = 0.6,\ \mu_{CR_k} = 0.5,\ pc = 0.4,\ k = 6,\ PS = 18 \cdot D \sim 4,\ r^{rac} = 5,\ freq = 0.5$
jSO (Brest et al., 2017)	F and CR, r^{rac} same as iLSHADE, $\mu_F = 0.3,\ \mu_{CR} = 0.8,\ PS = 25 \cdot ln(D) \cdot \sqrt{D} \sim 4,\ p = 0.25 \sim 0.125,\ H = 5$
ISDE (Tian & Gao, 2019)	$PS = 50,\ K = 100,\ \alpha = 0.6,\ \beta = \gamma = 0.5,\ freq = 0.01$
PaDE-pet (Meng, 2023)	$\mu_F = 0.5,\ \mu_{CR} = 0.5$, F and CR same as LSHADE, $p = 0.25 \sim 0.05,\ H = 5,\ r^{rac} = 1.6,\ T_0 = 70,\ PS = 25 \cdot ln(D) \cdot \sqrt{D} \sim 6$
TDE (Meng & Yang, 2022)	$\mu_F = 0.3,\ \mu_{CR} = 0.5$, F and CR same as JADE, $PS = 25 \cdot ln(D) \cdot \sqrt{D} \sim 4,\ n = 2 \cdot D$

Table 2
The results of LSHADE-cnEpsin, jSO, ISDE, PaDE-pet and TDE algorithms are compared with the proposed algorithms in test suite 10D using Wilcoxon rank sum test.

DE variants	LSHADE-cnEpsin	jSO	ISDE	PaDE-pet	TDE	OLBADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{a_1}	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0
f_{a_2}	0/0(≈)	0/0(≈)	2.1036E−01/5.8741E+01(<)	4.4583E−15/3.1839E−14(≈)	0/0(≈)	0/0
f_{a_3}	2.8013E−03/1.3988E−02(<)	3.5832E−09/1.4064E−08(<)	6.3192E−01/1.8928E+00(<)	1.1193E−02/2.6209E−02(<)	**3.1208E−14/7.9022E−14(≈)**	5.7790E−03/1.9355E−02
f_{a_4}	0/0(≈)	0/0(≈)	3.5856E−04/1.7847E−03(<)	0/0(≈)	0/0(≈)	0/0
f_{a_5}	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0
f_{a_6}	**4.9598E−03/2.7294E−02(≈)**	7.6960E−01/2.6643E+00(≈)	7.6960E+00/4.0760E+00(<)	7.6960E−01/2.6643E+00(≈)	1.9240E−01/1.3740E+00(≈)	2.0327E−01/7.0564E−01
f_{a_7}	**2.9911E−06/9.8628E−06**(>)	8.5429E−06/1.8093E−05(<)	1.0064E−02/3.4233E−02(<)	4.0128E−05/6.3840E−05(<)	1.9653E−05/3.6826E−05(<)	4.1873E−06/1.6870E−05
f_{a_8}	2.0252E+01/1.4516E−01(≈)	2.0356E+01/6.7083E−02(<)	2.0309E+01/6.7289E−02(≈)	2.0139E+01/1.3093E−01(>)	**2.0073E+01/1.2930E−01**(>)	2.0293E+01/1.1295E−01
f_{a_9}	1.1914E+00/1.3040E+00(<)	5.7642E−01/9.5119E−01(<)	1.0959E+00/7.9954E−01(<)	1.1439E+00/1.5256E+00(<)	4.8432E−01/7.7919E−01(<)	**3.4859E−01/5.4004E−01**
$f_{a_{10}}$	2.7579E−03/4.9658E−03(>)	**7.7336E−04/2.3894E−03**(>)	4.6349E−02/3.0733E−02(<)	1.1634E−02/1.5622E−02(≈)	5.6971E−03/9.5390E−03(>)	1.2746E−02/1.3673E−02
$f_{a_{11}}$	0/0(≈)	3.9018E−02/1.9505E−01(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0
$f_{a_{12}}$	**1.1587E+00/7.2700E−01**(>)	2.3021E+00/7.5748E−01(>)	4.8382E+00/1.8131E+00(<)	2.3864E+00/1.4464E+00(≈)	3.5940E+00/1.8092E+00(<)	2.6813E+00/1.4259E+00
$f_{a_{13}}$	1.9108E+00/1.0070E+00(<)	2.1423E+00/1.4255E+00(≈)	7.1875E+00/3.8761E+00(<)	1.6464E+00/7.5812E−01(<)	2.6682E+00/2.1226E+00(<)	**1.4693E+00/9.6683E−01**
$f_{a_{14}}$	1.2246E−02/2.7985E−02(>)	7.1442E−02/7.9916E−02(<)	1.1717E−01/4.8511E−01(≈)	2.0818E−02/3.6772E−02(≈)	**9.7968E−03/2.6119E−02**(>)	2.4492E−02/4.1575E−02
$f_{a_{15}}$	**2.3622E+02/1.0612E+02(≈)**	2.7237E+02/1.2418E+02(≈)	4.2464E+02/2.0948E+02(<)	2.9840E+02/1.0732E+02(≈)	4.3166E+02/2.0066E+02(<)	2.7499E+02/1.0458E+02
$f_{a_{16}}$	2.2262E−01/1.4960E−01(>)	1.0406E+00/2.2261E−01(<)	9.4484E−01/2.0190E−01(<)	1.5142E−01/1.3454E−01(>)	**1.0853E−01/1.2930E−01**(>)	4.1880E−01/3.6528E−01
$f_{a_{17}}$	1.0122E+01/1.7940E−15(≈)	1.0127E+01/2.4195E−02(<)	1.0123E+01/2.7901E−03(≈)	1.0122E+01/1.7940E−15(≈)	1.0122E+01/1.7940E−15(≈)	1.0122E+01/7.9877E−15
$f_{a_{18}}$	**1.3168E+01/1.5018E+00**(>)	1.4179E+01/1.7076E+00(≈)	2.0349E+01/5.6586E+00(<)	1.4182E+01/1.9910E+00(>)	1.4377E+01/1.9627E+00(≈)	1.4745E+01/1.7565E+00
$f_{a_{19}}$	2.3381E−01/3.7517E−02(≈)	2.6474E−01/5.0905E−02(≈)	2.5509E−01/5.6901E−02(≈)	**2.3170E−01/4.3875E−02(≈)**	2.3845E−01/7.0802E−02(>)	2.5810E−01/6.8947E−02
$f_{a_{20}}$	1.5616E+00/4.6653E−01(≈)	1.7192E+00/4.0347E−01(<)	1.7342E+00/5.0097E−01(<)	1.5332E+00/3.1738E−01(≈)	1.6232E+00/3.6423E−01(<)	**1.4676E+00/3.1058E−01**
$f_{a_{21}}$	4.0019E+02/0.0000E+00(≈)	4.0019E+02/0.0000E+00(≈)	**3.7664E+02/6.5142E+01(≈)**	4.0019E+02/0.0000E+00(≈)	4.0019E+02/0.0000E+00(≈)	4.0019E+02/0.0000E+00
$f_{a_{22}}$	1.1246E+01/2.6951E+01(≈)	1.2013E+01/1.9897E+01(<)	8.4464E+00/2.0799E+01(>)	1.6448E+00/2.8865E+00(>)	**1.5791E+00/2.4795E+00**(>)	4.7186E+00/5.0011E+00
$f_{a_{23}}$	**1.3737E+02/1.1301E+02(≈)**	2.5189E+02/1.3910E+02(<)	3.4193E+02/2.0025E+02(<)	2.6190E+02/1.3941E+02(<)	3.3529E+02/1.8209E+02(<)	1.5710E+02/1.2898E+02
$f_{a_{24}}$	2.0027E+02/1.3467E+00(≈)	2.0161E+02/2.9314E+00(<)	2.0056E+02/1.2491E+01(<)	2.0000E+02/1.2171E−04(≈)	2.0000E+02/2.3950E−05(≈)	**1.9684E+02/1.3083E+01**
$f_{a_{25}}$	1.9878E+02/8.7356E+00(≈)	1.9839E+02/1.1516E+01(≈)	2.0085E+02/1.9076E+00(<)	**1.9835E+02/1.1763E+01(≈)**	2.0000E+02/1.0821E−05(≈)	1.9894E+02/7.5429E+00
$f_{a_{26}}$	1.2844E+02/4.4481E+01(≈)	1.0722E+02/2.0162E+01(>)	1.3730E+02/5.2712E+01(<)	1.0435E+02/1.3724E+01(>)	**1.0350E+02/4.1858E+00(≈)**	1.1936E+02/3.5639E+01
$f_{a_{27}}$	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0353E+02/2.5215E+01(≈)	3.0000E+02/5.0398E−04(≈)	3.0000E+02/3.3884E−07(≈)	3.0000E+02/0.0000E+00
$f_{a_{28}}$	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	**2.9608E+02/2.8006E+01(≈)**	3.0000E+02/7.8765E−14(≈)	3.0000E+02/0.0000E+00
>/≈/<	6/19/3	3/14/11	1/10/17	5/18/5	6/15/7	–/–/–

Table 3
The results of LSHADE-cnEpsin, jSO, ISDE, PaDE-pet and TDE algorithms are compared with the proposed algorithms in test suite 30D using Wilcoxon rank sum test.

DE variants	LSHADE-cnEpsin	jSO	ISDE	PaDE-pet	TDE	OLBADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{a_1}	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	4.4583E−15/3.1839E−14
f_{a_2}	1.8718E+02/4.5776E+02(<)	3.6273E−10/1.1687E−09(<)	3.0554E+04/1.5182E+04(<)	4.7258E−13/3.6335E−13(<)	2.4075E−13/2.0545E−13(<)	**3.1208E−14/7.9022E−14**
f_{a_3}	1.9494E−02/1.3246E−01(>)	1.3461E−08/9.5379E−08(>)	2.8448E+06/3.9479E+06(<)	7.6976E−10/4.8526E−09(>)	**2.9358E−11/1.8044E−10**(>)	3.1857E−04/1.7061E−03
f_{a_4}	4.4583E−15/3.1839E−14(≈)	1.5559E−12/1.8889E−12(<)	1.0864E−05/2.8676E−05(<)	1.8725E−13/8.7542E−14(<)	4.4583E−14/9.1172E−14(≈)	**0/0**
f_{a_5}	1.0031E−13/9.8185E−14(≈)	1.1369E−13/0.0000E+00(≈)	**0.0000E+00/9.9110E−14**(>)	1.1146E−13/1.5919E−14(≈)	9.1395E−14/4.5586E−14(≈)	1.1369E−13/0.0000E+00
f_{a_6}	1.2101E+00/8.1174E−01(<)	2.9260E−08/2.0368E−07(<)	4.6118E+00/7.2958E+00(<)	2.2287E−11/1.5109E−10(≈)	**1.2037E−13/7.6810E−14**(>)	1.8279E−13/1.6409E−13
f_{a_7}	**5.2640E−03/1.7760E−02**(>)	1.8364E−02/3.1066E−02(>)	1.3991E+01/7.2779E+00(<)	1.9140E−02/4.8360E−02(>)	2.1361E−02/3.8724E−02(>)	2.9954E−02/4.7532E−02
f_{a_8}	2.0869E+01/1.2096E−01(≈)	2.0942E+01/5.1029E−02(<)	2.0897E+01/5.4112E−02(≈)	2.0723E+01/2.0329E−01(>)	**2.0670E+01/2.0865E−01**(>)	2.0864E+01/1.0485E−01
f_{a_9}	2.2039E+01/5.5615E+00(≈)	2.3603E+01/3.2673E+00(≈)	**1.6405E+01/3.3469E+00**(>)	2.5852E+01/1.6621E+00(<)	2.4587E+01/3.4435E+00(<)	1.9205E+01/8.0146E+00
$f_{a_{10}}$	0/0(≈)	0/0(≈)	9.2451E−02/5.2606E−02(<)	0/0(≈)	0/0(≈)	0/0
$f_{a_{11}}$	1.4267E−13/5.1314E−14(≈)	1.6719E−13/5.9522E−14(<)	**0/0**(>)	1.0700E−13/5.1658E−14(>)	4.5698E−14/5.0867E−14(>)	1.3375E−13/6.2085E−14
$f_{a_{12}}$	9.3336E+00/2.2055E+00(<)	8.3023E+00/2.2909E+00(<)	2.8132E+01/1.0194E+01(<)	6.3011E+00/1.7946E+00(<)	7.0093E+00/1.6476E+00(<)	**5.0488E+00/1.7270E+00**
$f_{a_{13}}$	1.5075E+01/5.8347E+00(<)	9.5050E+00/5.6025E+00(<)	5.2882E+01/1.8115E+01(<)	7.3126E+00/3.3460E+00(≈)	9.4046E+00/4.1760E+00(<)	**6.9530E+00/4.3535E+00**
$f_{a_{14}}$	1.4381E−01/5.0780E−02(<)	1.0845E+01/4.7694E+00(<)	2.8285E+00/1.6577E+01(≈)	3.3584E−02/2.8167E−02(>)	**5.3069E−03/1.0892E−02**(>)	8.8658E−02/4.1574E−02
$f_{a_{15}}$	**2.4308E+03/3.1234E+02**(>)	2.7617E+03/3.6509E+02(≈)	2.8543E+03/5.2913E+02(≈)	2.7048E+03/2.5978E+02(≈)	2.8449E+03/2.9576E+02(<)	2.7144E+03/2.7255E+02
$f_{a_{16}}$	8.4854E−01/2.6478E−01(≈)	2.3235E+00/2.4246E−01(<)	1.8701E+00/6.2359E−01(<)	6.0819E−01/3.3337E−01(>)	**3.4036E−01/4.1769E−01**(>)	7.7915E−01/2.3213E−01
$f_{a_{17}}$	3.0434E+01/1.2974E−04(≈)	3.0724E+01/1.1414E−01(<)	3.0434E+01/9.0463E−05(≈)	3.0434E+01/9.4299E−07(≈)	3.0434E+01/9.4299E−07(≈)	3.0434E+01/6.3811E−04
$f_{a_{18}}$	5.5731E+01/2.9254E+00(≈)	5.4814E+01/5.5915E+00(≈)	6.4659E+01/3.2979E+01(≈)	**5.2648E+01/3.6480E+00**(>)	5.4596E+01/5.8320E+00(≈)	5.4647E+01/4.1158E+00
$f_{a_{19}}$	1.2621E+00/1.1587E−01(≈)	1.2791E+00/1.1715E−01(≈)	**9.7335E−01/1.7092E−01**(>)	1.1563E+00/1.0255E−01(>)	1.1535E+00/1.0281E−01(>)	1.2944E+00/1.2935E−01
$f_{a_{20}}$	9.6932E+00/1.5520E+00(<)	9.5442E+00/4.1135E−01(<)	9.3995E+00/7.7948E−01(<)	1.0583E+01/2.2538E+00(<)	9.1389E+00/4.2868E−01(≈)	**9.0222E+00/3.2473E−01**
$f_{a_{21}}$	**2.8039E+02/4.0098E+01(≈)**	3.0623E+02/5.3112E+01(≈)	2.9813E+02/7.1282E+01(≈)	3.0000E+02/1.1896E−13(≈)	2.9608E+02/1.9604E+01(≈)	3.0171E+02/3.4946E+01
$f_{a_{22}}$	1.0648E+02/8.6880E−01(>)	1.2001E+02/4.6463E+00(<)	1.0609E+02/2.7677E+01(<)	1.0602E+02/4.2983E−01(>)	**1.0578E+02/4.7375E−01**(>)	1.0750E+02/2.1356E+00
$f_{a_{23}}$	**2.0488E+03/2.6033E+02**(>)	2.4854E+03/4.0249E+02(≈)	2.9299E+03/4.8926E+02(<)	2.7509E+03/3.2469E+02(<)	2.7189E+03/3.4822E+02(<)	2.3979E+03/3.5757E+02
$f_{a_{24}}$	2.0000E+02/2.2883E−03(>)	2.0006E+02/7.7050E−02(>)	2.2082E+02/9.1621E+00(<)	2.0000E+02/2.4478E−03(>)	2.0000E+02/1.8140E−02(>)	2.0015E+02/2.0358E−01
$f_{a_{25}}$	2.3611E+02/1.2587E+01(<)	2.3970E+02/5.0534E+00(<)	2.5832E+02/8.5119E+00(<)	2.1532E+02/2.0246E+01(>)	**2.0067E+02/4.7501E+00**(>)	2.1020E+02/2.0198E+01
$f_{a_{26}}$	2.0000E+02/1.9024E−04(≈)	2.0000E+02/4.5475E−14(≈)	2.2974E+02/5.4497E+01(<)	2.0000E+02/0.0000E+00(≈)	2.0000E+02/1.0129E−13(≈)	2.0000E+02/4.5475E−14
$f_{a_{27}}$	3.0004E+02/6.9870E−02(>)	3.0091E+02/1.3260E+00(>)	5.9583E+02/1.1580E+02(<)	**3.0002E+02/3.0231E−02**(>)	3.0007E+02/1.9428E−01(>)	3.0301E+02/3.8121E+00
$f_{a_{28}}$	3.0000E+02/0.0000E+00(≈)	3.0000E+02/9.0949E−14(≈)	3.0000E+02/9.0949E−14(≈)	3.0000E+02/1.8384E−13(≈)	3.0000E+02/1.3713E−13(≈)	3.0000E+02/0.0000E+00
>/≈/<	7/14/7	4/11/13	4/8/16	12/10/6	12/10/6	–/–/–

4.2. Convergence speed

To verify the performance of OLBADE in terms of convergence speed, convergence curves of the five algorithm are drawn using the median value of 51 runs on benchmark functions for CEC2017 with 30 dimensions, as shown in supplementary file.

From the convergence curves of the unimodal functions, it can be observed that OLBADE has the fastest convergence speed and the best stability, both in the early and later stages. This can be attributed to its parameter control strategy which achieves a better exploration and exploitation balance. From the convergence curves of the simple multimodal functions, OLBADE demonstrates the fastest convergence

Table 4
The results of LSHADE-cnEpsin, jSO, ISDE, PaDE-pet and TDE algorithms are compared with the proposed algorithms in test suite 50D using Wilcoxon rank sum test.

DE variants	LSHADE-cnEpsin	jSO	ISDE	PaDE-pet	TDE	OLBADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{a_1}	*0/0*(>)	5.7958E−14/1.0008E−13(>)	*0.0000E+00/1.4380E−13*(>)	*0/0*(>)	*0/0*(>)	1.3375E−13/1.1302E−13
f_{a_2}	1.4234E+04/1.2000E+04(<)	6.3797E+01/1.1475E+02(<)	2.4543E+05/9.3726E+04(<)	2.2195E+01/5.7538E+01(<)	2.3152E+02/4.1686E+02(<)	**1.9442E−05/5.2270E−05**
f_{a_3}	**6.1926E−01/2.6098E+00**(>)	3.7520E+01/1.9431E+02(>)	1.5913E+07/1.7717E+07(<)	7.5474E+01/3.7008E+02(>)	1.3231E+01/5.9201E+01(>)	2.3530E+03/5.9607E+03
f_{a_4}	8.8625E−09/4.4413E−08(<)	1.6896E−08/2.3904E−08(<)	8.7274E−04/1.3521E−03(<)	1.8872E−11/2.4830E−11(<)	3.0786E−10/5.9272E−10(<)	**1.6719E−12/7.4003E−13**
f_{a_5}	2.1623E−13/5.2122E−14(<)	1.4935E−13/5.3276E−14(<)	*1.1369E−13/0.0000E+00(≈)*	*1.1369E−13/0.0000E+00(≈)*	*1.1369E−13/0.0000E+00(≈)*	*1.1369E−13/0.0000E+00*
f_{a_6}	4.3340E+01/2.4456E−02(<)	4.3447E+01/1.6078E−14(<)	4.3447E+01/3.4445E−13(<)	4.3447E+01/0.0000E+00(<)	4.3441E+01/3.2641E−02(<)	**4.2555E+01/2.1803E−01**
f_{a_7}	**2.2638E−02/3.6272E−02**(>)	1.0704E−01/9.5973E−02(>)	4.7132E+01/1.3053E+01(<)	1.3045E−01/1.3251E−01(<)	1.2518E−01/1.0261E−01(>)	1.2694E−01/2.3505E−01
f_{a_8}	2.1090E+01/9.2286E−02(<)	2.1127E+01/4.0791E−02(<)	2.1110E+01/4.9443E−02(<)	2.0876E+01/1.6374E−01(<)	2.0877E+01/1.8995E−01(<)	**2.0847E+01/1.0843E−01**
f_{a_9}	4.8461E+01/4.7084E+00(<)	4.7598E+01/4.9187E+00(<)	3.7985E+01/6.3098E+00(<)	5.0435E+01/4.0077E+00(<)	4.2541E+01/1.3951E+01(<)	**1.7878E+01/2.7748E+00**
$f_{a_{10}}$	**0/0**(>)	2.2705E−03/4.6174E−03(<)	1.2378E−01/5.1460E−02(<)	3.1409E−03/4.6773E−03(<)	5.2185E−03/6.2456E−03(<)	5.1271E−14/1.7072E−14
$f_{a_{11}}$	1.6968E−07/1.4956E−07(>)	4.2131E−08/6.8218E−08(>)	**0.0000E+00/3.7706E−14**(>)	4.0348E−13/5.8253E−14(>)	1.8168E−13/1.2609E−13(>)	1.3813E−06/3.7055E−06
$f_{a_{12}}$	1.9415E+01/3.3007E+00(<)	1.7010E+01/4.0749E+00(<)	6.9706E+01/2.2026E+01(<)	1.3695E+01/1.9881E+00(<)	1.7266E+01/2.8586E+00(<)	**1.1658E+01/2.8374E+00**
$f_{a_{13}}$	3.7232E+01/1.4060E+01(<)	2.4870E+01/1.3180E+01(<)	1.4348E+02/4.0606E+01(<)	2.2666E+01/6.5318E+00(<)	3.4408E+01/1.0068E+01(<)	**1.2467E+01/4.5831E+00**
$f_{a_{14}}$	1.9067E+01/4.4799E+00(<)	7.4691E+01/2.0332E+01(<)	1.6586E+00/1.3275E+00(>)	6.6425E−02/2.7305E−02(>)	**3.4575E−02/2.0565E−02**(>)	4.8003E+00/2.5698E+00
$f_{a_{15}}$	**5.9867E+03/3.5701E+02**(>)	6.4894E+03/4.2066E+02(<)	6.3277E+03/1.1352E+03(<)	6.2125E+03/3.5989E+02(<)	6.5341E+03/4.6691E+02(<)	6.2021E+03/4.0336E+02
$f_{a_{16}}$	1.2083E+00/2.0268E−01(<)	3.0459E+00/5.4253E−01(<)	2.8696E+00/6.4355E−01(<)	1.0793E+00/3.3286E−01(<)	**7.2431E−01/5.6093E−01**(>)	1.0079E+00/2.5083E−01
$f_{a_{17}}$	5.0968E+01/5.9132E−02(<)	5.2655E+01/4.5213E−01(<)	5.0787E+01/2.8982E−03(<)	5.0786E+01/1.4072E−09(<)	5.0786E+01/3.8031E−09(<)	**5.0684E+01/6.1791E−02**
$f_{a_{18}}$	1.1724E+02/6.6303E+00(<)	1.0956E+02/8.9066E+00(<)	1.1527E+02/5.5062E+01(<)	9.7843E+01/5.9279E+00(<)	1.0729E+02/6.8668E+00(<)	**8.8362E+01/6.5682E+00**
$f_{a_{19}}$	2.7383E+00/1.5471E−01(<)	2.6516E+00/1.7252E−01(<)	**1.8449E+00/3.4798E−01**(>)	2.3569E+00/1.6096E−01(<)	2.3968E+00/1.4700E−01(<)	2.2995E+00/1.9567E−01
$f_{a_{20}}$	1.7914E+01/4.0074E−01(<)	1.8597E+01/5.1731E−01(<)	1.8545E+01/1.0636E+00(<)	1.7923E+01/4.2830E−01(<)	1.7775E+01/5.1864E−01(<)	**1.7565E+01/5.0357E−01**
$f_{a_{21}}$	7.0070E+02/4.6002E+02(>)	7.1139E+02/4.1308E+02(>)	**6.5893E+02/4.5780E+02**(>)	7.3686E+02/4.5537E+02(>)	8.0919E+02/4.3686E+02(>)	8.7974E+02/3.5790E+02
$f_{a_{22}}$	2.7422E+01/4.4647E+00(<)	6.7521E+01/1.5679E+01(<)	2.9115E+01/3.9464E+01(<)	1.1689E+01/1.0873E+00(>)	**1.1521E+01/6.9661E−01**(>)	2.1050E+01/3.4013E+00
$f_{a_{23}}$	**4.8765E+03/4.4309E+02**(>)	5.5734E+03/4.9945E+02(>)	6.3473E+03/8.4592E+02(<)	6.2234E+03/4.0360E+02(<)	6.4016E+03/4.0043E+02(<)	5.9548E+03/5.4223E+02
$f_{a_{24}}$	**2.0002E+02/1.8832E−02**(>)	2.0077E+02/1.2156E+00(>)	2.6151E+02/1.4988E+01(<)	2.0005E+02/6.5079E−02(>)	2.0042E+02/7.5047E−01(>)	2.0291E+02/9.7714E−01
$f_{a_{25}}$	**2.7250E+02/5.5474E+00**(>)	2.7543E+02/6.8045E+00(>)	3.2371E+02/1.2417E+01(<)	2.8127E+02/5.9272E+00(>)	2.8318E+02/5.4303E+00(>)	3.0065E+02/1.7366E+01
$f_{a_{26}}$	2.6882E+02/4.6994E+01(<)	2.1016E+02/3.1138E+01(>)	3.3876E+02/7.5367E+01(<)	**2.0794E+02/2.7476E+01**(>)	2.4251E+02/5.1351E+01(>)	2.6686E+02/5.2075E+01
$f_{a_{27}}$	3.0609E+02/3.4346E+01(>)	3.3365E+02/3.8204E+01(>)	1.1385E+03/1.4324E+02(<)	**3.0521E+02/6.9028E+00**(>)	3.1651E+02/1.6904E+01(>)	3.4765E+02/1.5812E+01
$f_{a_{28}}$	*4.0000E+02/3.2155E−14(≈)*	*4.0000E+02/3.2155E−14(≈)*	4.5886E+02/4.2037E+02(<)	*4.0000E+02/3.2155E−14(≈)*	*4.0000E+02/5.5695E−14(≈)*	*4.0000E+02/0.0000E+00*
>/≈/<	11/1/16	10/1/17	5/1/22	10/2/16	12/2/14	−/−/−

Table 5
The results of LSHADE-cnEpsin, jSO, ISDE, PaDE-pet and TDE algorithms are compared with the proposed algorithms in test suite 10D using Wilcoxon rank sum test.

DE variants	LSHADE-cnEpsin	jSO	ISDE	PaDE-pet	TDE	OLBADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{b_1}	*0/0(≈)*	*0/0(≈)*	*0.0000E+00/1.7749E−14(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{b_2}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{b_3}	*0/0(≈)*	*0/0(≈)*	2.0192E−09/1.0192E−08(≈)	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{b_4}	**2.3450E−01/9.4735E−01**(>)	3.2734E+01/8.2650E+00(<)	2.6937E+01/1.4316E+01(<)	1.8498E+01/1.7452E+01(≈)	2.4507E+01/1.4763E+01(<)	5.5408E+00/2.4474E+00
f_{b_5}	**1.3202E+00/4.7400E+00**(>)	1.6786E+01/7.6890E+00(<)	1.7678E+01/6.5193E+00(<)	1.2933E+01/9.1880E+00(≈)	1.5459E+01/8.1619E+00(≈)	1.2936E+01/7.2077E+00
f_{b_6}	1.7540E−02/1.2526E−01(≈)	*0/0(≈)*	1.7540E−02/1.2526E−01(≈)	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{b_7}	**8.7012E−04/2.4066E−03**(>)	1.1114E−03/3.2059E−03(>)	2.4669E−02/2.0962E−02(<)	9.9941E−03/1.5350E−02(≈)	7.4424E−03/1.3883E−02(>)	1.2509E−02/1.2756E−02
f_{b_8}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{b_9}	**1.7255E+00/7.4295E−01(≈)**	1.9314E+00/8.5385E−01(>)	3.5701E+00/1.4647E+00(<)	2.6938E+00/1.1305E+00(≈)	2.2083E+00/1.4384E+00(≈)	2.6928E+00/1.8046E+00
$f_{b_{10}}$	*9.7968E−03/2.2939E−02*(>)	4.6787E−02/5.7407E−02(<)	1.5146E−01/4.9674E−01(<)	*9.7968E−03/2.2939E−02*(>)	1.4695E−02/3.2061E−02(>)	1.8369E−02/3.1337E−02
$f_{b_{11}}$	3.0009E+01/4.0094E+01(>)	4.0028E+01/5.0029E+01(≈)	5.9118E+01/6.6471E+01(≈)	**2.0229E+01/2.1015E+01**(>)	4.2643E+01/7.0387E+01(>)	2.0269E+01/8.0847E+01
$f_{b_{12}}$	1.3215E−01/7.7277E−02(<)	1.9840E−01/1.8980E−01(<)	1.8318E−01/8.6235E−02(<)	6.4394E−02/2.6160E−02(≈)	**6.3070E−02/4.6287E−02(≈)**	6.6226E−02/4.7651E−02
$f_{b_{13}}$	**4.5723E−02/1.1561E−02**(>)	6.9045E−02/1.8246E−02(>)	4.7849E−02/2.9182E−02(>)	5.1484E−02/1.2417E−02(>)	4.7536E−02/1.8684E−02(>)	7.6061E−02/2.0874E−02
$f_{b_{14}}$	1.2237E−01/4.5491E−02(<)	6.2407E−02/2.0556E−02(>)	**5.2437E−02/1.8366E−02**(>)	6.9373E−02/2.3737E−02(≈)	9.6691E−02/3.9625E−02(<)	7.1249E−02/2.9237E−02
$f_{b_{15}}$	**3.7083E−01/6.6334E−02(≈)**	4.1753E−01/8.2374E−02(≈)	4.9611E−01/1.1534E−01(<)	3.8301E−01/7.6392E−02(≈)	4.4506E−01/1.7789E−01(≈)	3.9416E−01/8.2337E−02
$f_{b_{16}}$	8.6394E−01/2.6790E−01(≈)	8.9799E−01/3.8374E−01(≈)	**6.3899E−01/3.5169E−01**(>)	1.1793E+00/2.6807E−01(<)	1.1718E+00/4.8888E−01(<)	9.1422E−01/2.9492E−01
$f_{b_{17}}$	4.4753E+01/5.5016E+01(<)	1.5979E+00/1.8664E+00(<)	1.0976E+01/2.2539E+01(<)	**1.4701E+00/1.4419E+00**(<)	2.6046E+00/6.4775E+00(≈)	2.7636E+00/7.8175E+00
$f_{b_{18}}$	3.6816E−01/4.3845E−01(<)	1.7503E−01/1.6026E−01(<)	5.8371E−01/5.6907E−01(<)	1.5885E−01/2.5778E−01(≈)	**5.6309E−02/1.5479E−01**(>)	1.0610E−01/1.2856E−01
$f_{b_{19}}$	2.9744E−01/4.3631E−01(<)	7.8376E−02/1.9117E−01(≈)	**3.9941E−02/3.1577E−02**(>)	6.5648E−02/3.2022E−02(<)	5.6674E−02/2.9530E−02(≈)	4.8443E−02/3.2054E−02
$f_{b_{20}}$	4.2489E−01/1.7991E−01(<)	2.5098E−01/2.0540E−01(≈)	**3.7383E−02/6.6783E−02**(>)	1.1249E−01/1.1698E−01(>)	1.8087E−01/1.8379E−01(>)	2.5416E−01/1.8721E−01
$f_{b_{21}}$	7.6541E+00/2.4683E+01(<)	5.0087E−01/2.9086E−01(≈)	1.4451E+00/3.9718E+00(≈)	**2.8816E−01/2.9710E−01**(>)	4.0443E−01/3.1934E−01(≈)	7.1927E−01/2.3047E+00
$f_{b_{22}}$	3.6138E+00/7.2820E+00(<)	3.1398E−01/1.9054E−01(<)	4.4843E−01/2.8078E+00(>)	**8.8129E−02/3.7465E−02(≈)**	1.0828E−01/8.5396E−02(≈)	1.0377E−01/4.3877E−02
$f_{b_{23}}$	3.2946E+02/0.0000E+00(<)	3.2946E+02/0.0000E+00(<)	3.2946E+02/4.5927E−13(<)	3.2946E+02/0.0000E+00(<)	3.2946E+02/0.0000E+00(<)	**2.0000E+02/0.0000E+00**
$f_{b_{24}}$	**1.0517E+02/2.5607E+00**(>)	1.0703E+02/2.1806E+00(≈)	1.1042E+02/1.9333E+00(<)	1.0677E+02/3.0283E+00(>)	1.0800E+02/2.6780E+00(≈)	1.0794E+02/2.2319E+00
$f_{b_{25}}$	1.4933E+02/3.9465E+01(<)	1.2600E+02/2.6804E+01(≈)	1.6188E+02/4.1669E+01(<)	**1.1586E+02/2.5608E+01**(>)	1.2326E+02/2.1835E+01(≈)	1.2941E+02/2.7784E+01
$f_{b_{26}}$	*1.0005E+02/1.2699E−02*(>)	1.0007E+02/1.6306E−02(≈)	*1.0005E+02/2.8356E−02*(>)	*1.0005E+02/1.4341E−02*(>)	*1.0005E+02/2.0696E−02*(>)	1.0007E+02/2.3301E−02
$f_{b_{27}}$	1.2620E+02/1.6665E+02(≈)	4.2239E+01/1.0425E+02(>)	1.4230E+02/1.5925E+02(≈)	**2.2837E+01/8.7601E+01(≈)**	3.6666E+01/1.0880E+02(<)	3.0897E+01/8.3493E+01
$f_{b_{28}}$	3.9383E+02/4.4746E+01(<)	3.7932E+02/2.9547E+01(<)	3.7291E+02/2.9555E+01(<)	3.7257E+02/3.7936E+01(<)	3.7976E+02/4.4049E+01(<)	**2.3237E+02/8.4642E+01**
$f_{b_{29}}$	2.2450E+02/2.1196E+00(<)	2.2185E+02/2.9100E−01(<)	2.2171E+02/7.2619E+00(<)	2.2177E+02/1.3707E−01(<)	2.2199E+02/5.2450E−01(<)	**2.0423E+02/1.8942E+01**
$f_{b_{30}}$	4.7224E+02/1.9428E+01(≈)	4.6463E+02/9.1912E+00(≈)	4.7831E+02/2.0699E+01(<)	4.6277E+02/1.5007E+00(≈)	4.6855E+02/2.5524E+00(≈)	**4.6260E+02/4.7167E+01**
>/≈/<	8/10/12	5/15/10	7/8/15	8/16/6	7/16/7	−/−/−

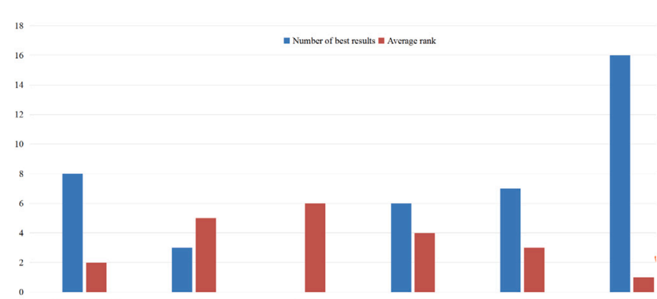


Fig. 2. Statistical results of OLBADE and comparing algorithms under CEC2017 on 30D optimization.

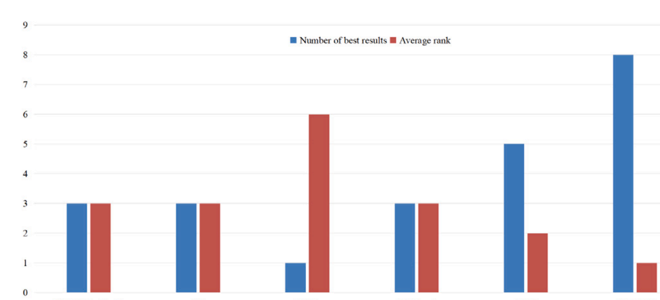


Fig. 3. Statistical results of OLBADE and comparing algorithms under CEC2022 on 20D optimization.

Table 6
The results of LSHADE-cnEpsin, jSO, ISDE, PaDE-pet and TDE algorithms are compared with the proposed algorithms in test suite 30D using Wilcoxon rank sum test.

DE variants	LSHADE-cnEpsin	jSO	ISDE	PaDE-pet	TDE	OLBADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{b_1}	3.0130E−01/1.9386E+00(<)	2.8923E−13/8.6313E−13(<)	2.5826E+04/2.3076E+04(<)	1.7555E−14/7.8293E−15(<)	9.1953E−15/6.8587E−15(<)	**1.1146E−15/3.8586E−15**
f_{b_2}	2.5078E−14/9.2483E−15(<)	*0/0*(≈)	1.4211E−13/6.4894E−13(≈)	*0/0*(≈)	*0/0*(≈)	5.5729E−16/3.9798E−15
f_{b_3}	*0/0*(≈)	*0/0*(≈)	1.2671E−02/7.8579E−02(<)	*0/0*(≈)	*0/0*(≈)	*0/0*
f_{b_4}	2.6774E+01/2.0851E+00(<)	7.6906E−14/4.3771E−14(<)	5.1252E−02/1.6056E−02(<)	3.7896E−14/2.7063E−14(≈)	**7.8020E−15/1.9755E−14**(>)	2.5635E−14/2.8566E−14
f_{b_5}	2.0126E+01/2.2431E−02(>)	2.0922E+01/5.8782E−02(<)	**2.0075E+01/6.6962E−02**(>)	2.0083E+01/4.0174E−02(>)	2.0103E+01/8.4383E−02(>)	2.0148E+01/1.0208E−01
f_{b_6}	3.3345E−02/1.1645E−01(≈)	2.3534E−05/1.2654E−04(≈)	2.2954E+00/1.4787E+00(<)	1.3843E−07/9.8860E−07(>)	**0/0**(>)	2.0370E−02/6.1471E−02
f_{b_7}	*0/0*(≈)	*0/0*(≈)	*0/0*(≈)	*0/0*(≈)	*0/0*(≈)	*0/0*
f_{b_8}	2.8310E−13/7.9978E−14(≈)	3.8119E−13/2.1549E−13(<)	**0/0**(>)	2.0285E−13/9.1837E−14(>)	3.5666E−14/5.3276E−14(>)	3.0094E−13/2.1064E−13
f_{b_9}	1.1233E+01/2.2560E+00(<)	8.6090E+00/2.1322E+00(<)	3.1849E+01/8.2364E+00(<)	8.1051E+00/2.0364E+00(<)	9.4799E+00/2.3285E+00(<)	**6.3942E+00/1.6968E+00**
$f_{b_{10}}$	2.5521E−02/2.4697E−02(<)	1.7187E+00/1.1945E+00(<)	5.1799E−01/6.5877E−01(<)	**1.2247E−03/4.9474E−03**(>)	2.5995E−02/1.8269E−01(>)	2.0004E−02/2.1866E−02
$f_{b_{11}}$	**9.6033E+02/1.7286E+02**(>)	1.3158E+03/2.1718E+02(≈)	1.6013E+03/4.6646E+02(<)	1.2466E+03/1.8513E+02(>)	1.1649E+03/1.7607E+02(>)	1.3720E+03/2.0159E+02
$f_{b_{12}}$	1.7340E−01/3.3640E−02(>)	4.0150E−01/3.6990E−01(<)	**1.0756E−01/5.3199E−02**(>)	1.4339E−01/2.7873E−02(>)	1.8978E−01/9.3477E−02(≈)	1.8919E−01/3.3626E−02
$f_{b_{13}}$	1.2847E−01/1.4295E−02(≈)	1.3636E−01/2.0916E−02(≈)	**9.1149E−02/2.9790E−02**(>)	1.2618E−01/1.5488E−02(>)	1.2584E−01/2.6773E−02(≈)	1.3343E−01/1.7865E−02
$f_{b_{14}}$	2.1353E−01/2.5251E−02(<)	2.3321E−01/3.8834E−02(<)	1.6801E−01/2.5695E−02(≈)	2.0181E−01/2.3324E−02(<)	2.2065E−01/2.8673E−02(<)	**1.6154E−01/2.8485E−02**
$f_{b_{15}}$	2.2888E+00/2.3661E−01(≈)	2.3817E+00/3.5824E−01(≈)	2.7036E+00/6.1622E−01(<)	**2.1387E+00/2.3424E−01**(>)	2.2167E+00/3.1483E−01(≈)	2.2829E+00/2.7528E−01
$f_{b_{16}}$	8.3402E+00/4.6462E−01(≈)	8.7558E+00/6.8644E−01(<)	**7.6715E+00/9.9050E−01**(>)	8.7120E+00/3.8433E−01(<)	8.8364E+00/7.1175E−01(<)	8.4111E+00/4.0774E−01
$f_{b_{17}}$	1.8306E+02/9.7710E+01(<)	6.9148E+01/2.6578E+01(<)	4.5204E+03/3.9187E+03(<)	7.7867E+01/4.3336E+01(<)	5.6265E+01/3.1293E+01(≈)	**5.5367E+01/1.9229E+01**
$f_{b_{18}}$	1.1312E+01/4.8786E+00(<)	2.5739E+00/1.4906E+00(≈)	1.6135E+02/2.7725E+02(<)	2.8635E+00/1.5322E+00(≈)	3.8567E+00/1.7310E+00(<)	**2.3598E+00/1.4016E+00**
$f_{b_{19}}$	2.9108E+00/6.6142E−01(<)	2.1476E+00/6.7918E−01(≈)	2.7467E+00/6.4991E−01(<)	2.5428E+00/5.2637E−01(<)	2.3371E+00/5.2493E−01(<)	**2.0165E+00/5.8821E−01**
$f_{b_{20}}$	2.8342E+00/1.2400E+00(<)	2.1296E+00/8.6850E−01(<)	2.4885E+01/1.4889E+01(<)	2.1891E+00/8.4219E−01(<)	3.0454E+00/1.0889E+00(<)	**1.7114E+00/6.0100E−01**
$f_{b_{21}}$	9.7208E+01/8.6540E+01(<)	1.5517E+01/1.9771E+01(<)	8.9483E+02/1.2650E+03(<)	1.1558E+01/2.1163E+01(≈)	2.9860E+01/5.1172E+01(<)	**8.5355E+00/1.7568E+01**
$f_{b_{22}}$	4.4604E+01/4.3797E+01(≈)	**3.7612E+01/3.5660E+01(≈)**	1.1003E+02/7.9216E+01(<)	4.8950E+01/4.4645E+01(≈)	6.8034E+01/4.9606E+01(<)	4.5322E+01/4.2947E+01
$f_{b_{23}}$	**3.1477E+02/1.0906E−01**(>)	3.1524E+02/0.0000E+00(<)	3.1524E+02/9.1854E−13(<)	3.1524E+02/0.0000E+00(<)	3.1524E+02/1.3628E−02(<)	3.1496E+02/6.2902E−02
$f_{b_{24}}$	2.2198E+02/5.7485E−01(<)	2.0424E+02/8.6695E+00(≈)	2.2508E+02/2.2899E+00(<)	2.2039E+02/6.0452E+00(<)	2.2189E+02/3.2389E+00(<)	**2.0012E+02/8.3426E−01**
$f_{b_{25}}$	2.0259E+02/9.0920E−02(<)	2.0256E+02/2.5379E−02(<)	2.0440E+02/1.3530E+00(<)	2.0261E+02/5.4926E−02(<)	2.0263E+02/8.8661E−02(<)	**2.0137E+02/3.4631E−01**
$f_{b_{26}}$	1.0013E+02/1.7463E−02(≈)	1.0013E+02/2.5376E−02(<)	**1.0010E+02/3.6419E−02**(>)	1.0012E+02/1.7865E−02(≈)	1.0012E+02/2.5718E−02(≈)	1.0012E+02/1.8020E−02
$f_{b_{27}}$	3.0000E+02/1.5814E−13(<)	3.0000E+02/2.1971E−13(<)	3.3801E+02/3.7829E+01(<)	3.0000E+02/2.2964E−13(<)	3.0000E+02/9.0949E−14(<)	**2.3204E+02/5.4498E+01**
$f_{b_{28}}$	8.2091E+02/3.2517E+01(<)	8.2143E+02/1.6614E+01(<)	8.2149E+02/2.6797E+01(<)	8.1548E+02/2.6698E+01(<)	8.3456E+02/2.0818E+01(<)	**6.1955E+02/2.7536E+02**
$f_{b_{29}}$	6.6615E+02/1.4998E+02(<)	7.1521E+02/1.2596E+00(<)	1.0470E+03/2.5748E+02(<)	5.8189E+02/2.4507E+02(<)	5.8989E+02/2.6507E+02(<)	**2.0765E+02/2.6146E+01**
$f_{b_{30}}$	8.0498E+02/2.5113E+02(<)	6.8403E+02/2.6542E+02(<)	1.6525E+03/5.9792E+02(<)	4.4965E+02/8.3665E+01(<)	4.6165E+02/7.5665E+01(<)	**3.0360E+02/2.2153E+01**
>/≈/<	4/9/17	0/11/19	6/3/21	8/8/14	6/8/16	–/–/–

Table 7
The results of LSHADE-cnEpsin, jSO, ISDE, PaDE-pet and TDE algorithms are compared with the proposed algorithms in test suite 50D using Wilcoxon rank sum test.

DE variants	LSHADE-cnEpsin	jSO	ISDE	PaDE-pet	TDE	OLBADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{b_1}	5.4085E+03/4.7512E+03(<)	1.3383E+01/3.5788E+01(<)	4.1385E+05/1.5129E+05(<)	2.5978E+00/6.3449E+00(<)	4.5821E+01/2.1017E+02(<)	**3.9932E−08/9.2374E−08**
f_{b_2}	1.3937E−10/7.2650E−10(<)	6.0187E−14/1.7655E−14(<)	3.7137E+03/4.1310E+03(<)	3.1208E−14/8.5358E−15(>)	**2.7864E−14/3.9798E−15**(>)	4.6812E−14/1.5899E−14
f_{b_3}	3.9010E−14/5.3915E−14(>)	6.1302E−14/1.5434E−14(<)	5.5802E+02/6.7198E+02(<)	3.3437E−14/2.8254E−14(>)	**1.6719E−14/2.6158E−14**(>)	5.5729E−14/7.9597E−15
f_{b_4}	9.4987E+01/3.7837E+00(<)	5.4091E+01/4.9069E+01(<)	5.8559E+01/3.9928E+01(<)	3.9884E+00/1.9204E+01(>)	1.3499E+01/3.4081E+01(>)	**2.7082E+00/9.3152E+00**
f_{b_5}	2.0258E+01/3.3085E−02(>)	2.1065E+01/1.5500E−01(<)	**2.0147E+01/1.7304E−01**(>)	2.0189E+01/8.4550E−02(>)	2.0251E+01/1.1352E−01(≈)	2.0285E+01/9.0555E−02
f_{b_6}	**1.9040E−05/1.7647E−05**(>)	1.9165E−02/8.2473E−02(>)	9.0547E+00/4.0871E+00(<)	2.9413E−02/2.1004E−01(>)	1.4716E−02/1.0501E−01(>)	5.6273E−01/2.7649E−01
f_{b_7}	**0/0**(>)	9.1395E−14/4.5586E−14(≈)	6.2767E−04/3.2231E−03(>)	1.5604E−14/3.9511E−14(>)	4.4583E−15/2.2287E−14(>)	1.0700E−13/2.7016E−14
f_{b_8}	7.6600E−08/4.1222E−08(>)	5.8644E−08/1.1841E−07(>)	**0.0000E+00/7.1902E−14**(>)	8.0918E−13/1.3384E−13(>)	3.4106E−13/2.5217E−13(>)	1.4627E−06/2.9704E−06
f_{b_9}	2.4842E+01/4.8745E+00(<)	1.6417E+01/2.6669E+00(<)	6.5220E+01/1.2821E+01(<)	1.4727E+01/3.4388E+00(<)	1.8509E+01/3.1755E+00(<)	**1.1225E+01/2.2082E+00**
$f_{b_{10}}$	2.8566E+00/1.0212E+00(<)	1.1358E+01/3.5380E+00(<)	5.4568E+00/2.3390E+01(≈)	1.0058E−02/1.2974E−02(>)	**1.6360E−03/4.1755E−03**(>)	5.4871E−01/3.4844E−01
$f_{b_{11}}$	3.6463E+03/3.1248E+02(≈)	3.4360E+03/4.1569E+02(≈)	3.8626E+03/6.9362E+02(<)	3.2152E+03/2.4108E+02(>)	3.3076E+03/2.5332E+02(>)	**3.1132E+03/3.7313E+02**
$f_{b_{12}}$	2.4821E−01/2.9562E−02(≈)	4.6154E−01/5.0656E−01(<)	**1.6064E−01/8.3150E−02**(>)	1.8061E−01/2.3060E−02(>)	2.3051E−01/5.4957E−02(>)	2.1842E−01/2.9851E−02
$f_{b_{13}}$	2.0289E−01/1.9987E−02(<)	1.9066E−01/2.2343E−02(<)	2.0521E−01/3.8843E−02(<)	1.8810E−01/2.0613E−02(<)	2.0422E−01/3.1112E−02(<)	**1.5028E−01/2.1114E−02**
$f_{b_{14}}$	2.5203E−01/2.0646E−02(<)	3.0917E−01/3.5671E−02(<)	3.0241E−01/1.2052E−01(<)	2.6622E−01/2.6210E−02(<)	2.9706E−01/1.6204E−02(<)	**1.6142E−01/1.8037E−02**
$f_{b_{15}}$	5.2576E+00/5.4311E−01(≈)	5.5814E+00/5.2079E−01(<)	5.8856E+00/1.1738E+00(<)	**4.7498E+00/4.5973E−01**(>)	5.2975E+00/4.5665E−01(≈)	5.0260E+00/4.7167E−01
$f_{b_{16}}$	1.6641E+01/5.5396E−01(≈)	1.7085E+01/8.5325E−01(<)	**1.6085E+01/9.1737E−01**(>)	1.6987E+01/3.6172E−01(≈)	1.7033E+01/8.8123E−01(≈)	1.6780E+01/4.7202E−01
$f_{b_{17}}$	4.2449E+02/1.8245E+02(<)	3.5897E+02/2.0699E+02(<)	2.9792E+04/1.5419E+04(<)	5.0208E+02/2.1787E+02(<)	5.2679E+02/2.5685E+02(<)	**1.5527E+02/1.2880E+02**
$f_{b_{18}}$	3.4161E+01/1.5018E+01(<)	**1.2213E+01/4.1102E+00**(>)	5.7557E+02/4.6099E+02(<)	1.8505E+01/5.8623E+00(<)	2.0773E+01/7.0639E+00(<)	1.5366E+01/5.8318E+00
$f_{b_{19}}$	9.1357E+00/1.6440E+00(≈)	9.2427E+00/7.7397E−01(>)	9.2247E+00/1.8153E+00(≈)	8.9186E+00/1.4648E+00(>)	**8.4446E+00/1.4185E+00**(>)	8.5100E+00/1.1849E+00
$f_{b_{20}}$	8.3905E+00/2.4873E+00(<)	6.3164E+00/1.9852E+00(<)	2.4374E+02/1.9508E+02(<)	7.4045E+00/2.1222E+00(<)	9.0401E+00/2.7295E+00(<)	**3.5251E+00/1.4141E+00**
$f_{b_{21}}$	3.6884E+02/1.1005E+02(<)	2.5891E+02/8.5815E+01(≈)	3.4282E+04/2.8723E+04(<)	3.4876E+02/1.0154E+02(<)	3.6554E+02/1.1085E+02(<)	**2.3537E+02/6.1387E+01**
$f_{b_{22}}$	**9.0163E+01/5.9552E+01**(>)	1.4748E+02/9.3225E+01(≈)	4.9516E+02/2.2854E+02(<)	1.9476E+02/8.7409E+01(<)	1.8103E+02/6.8131E+01(<)	1.3220E+02/4.8001E+01
$f_{b_{23}}$	3.4199E+02/5.9112E−01(<)	3.4400E+02/1.8955E−13(<)	3.4400E+02/0.0000E+00(<)	3.4400E+02/1.7667E−13(<)	3.4400E+02/1.9810E−02(<)	**3.4006E+02/6.0651E−01**
$f_{b_{24}}$	2.6967E+02/1.9375E+00(<)	2.7197E+02/1.8322E+00(<)	2.6720E+02/3.1065E+00(<)	2.7284E+02/1.3934E+00(<)	2.7332E+02/1.2400E+00(<)	**2.1465E+02/2.3341E+01**
$f_{b_{25}}$	2.0531E+02/3.0298E−01(<)	2.0497E+02/1.5287E−01(<)	2.1100E+02/4.8531E+00(<)	2.0550E+02/3.0820E−01(<)	2.0560E+02/3.6958E−01(<)	**2.0233E+02/5.9548E−01**
$f_{b_{26}}$	1.0020E+02/4.7284E−02(<)	1.0019E+02/2.9932E−02(≈)	1.0805E+02/2.7113E+01(<)	**1.0018E+02/2.1181E−02**(≈)	1.1347E+02/2.6785E+01(<)	1.0019E+02/2.8024E−02
$f_{b_{27}}$	3.0575E+02/1.4709E+01(>)	3.0988E+02/1.9079E+01(≈)	5.0093E+02/8.3127E+01(<)	3.0599E+02/1.6826E+01(>)	**3.0259E+02/1.3028E+01**(>)	3.0678E+02/8.9706E+00
$f_{b_{28}}$	1.1149E+03/4.7761E+01(>)	**1.0850E+03/2.5238E+01**(>)	1.1787E+03/5.1474E+01(≈)	1.1979E+03/4.1638E+01(≈)	1.2473E+03/6.3371E+01(<)	1.1386E+03/2.6203E+02
$f_{b_{29}}$	7.7203E+02/8.6483E+01(<)	8.0839E+02/4.6602E+01(<)	1.5200E+03/2.0747E+02(<)	**4.7628E+02/1.3270E+02**(≈)	6.4191E+02/1.3082E+02(<)	4.8014E+02/9.8816E+01
$f_{b_{30}}$	8.3017E+03/7.8178E+02(<)	8.4171E+03/4.5903E+02(<)	9.2076E+03/1.0072E+03(<)	8.9717E+03/5.4964E+02(<)	9.6693E+03/6.3711E+02(<)	**1.0191E+03/1.9291E+02**
>/≈/<	8/5/17	5/6/19	5/3/22	13/4/13	11/3/16	–/–/–

speed on f_{c_3}-f_{c_4} and f_{c_6}-f_{c_8}. It is worth noting that on benchmark function f_{c_4}, when other algorithms fail to obtain the global optimal solution and suffer from stagnation, OLBADE continues to exhibit strong exploitation ability after 150,000 iterations, indicating the effectiveness of the opposition learning-based diversity enhancement. As for hybrid functions, it is evident that OLBADE achieves good convergence performance on $f_{c_{12}}$, $f_{c_{13}}$, $f_{c_{15}}$, $f_{c_{18}}$, and $f_{c_{19}}$. LSHADE-cnEpsin performs the best on $f_{c_{10}}$, jSO obtains the best convergence performance on $f_{c_{11}}$, $f_{c_{14}}$ and $f_{c_{16}}$, and ISDE achieves the best convergence performance on $f_{c_{17}}$. From the convergence curves of the composition functions, it can be observed that all algorithms are capable of converging towards the global optimal solution. In the case of $f_{c_{30}}$, it is noteworthy that while other algorithms tend to experience stagnation, OLBADE stands out as the sole algorithm capable of breaking free from stagnation at $nfes$=20 000. This observation underscores the efficacy of OLBADE's diversity enhancement mechanism and perturbation strategy. Moreover, on $f_{c_{30}}$, OLBADE continues to exploit the optimal solution, as clearly seen that the convergence curve still shows a decreasing trend after $nfes$=30 000.

As one of the reviewers suggested, we also examine the convergence speed of OLBADE on benchmark functions for CEC2014 with 50 dimensions and CEC2022 with 20 dimensions, as shown in Figs. 4 to 6.

Table 8
The results of LSHADE-cnEpsin, jSO, ISDE, PaDE-pet and TDE algorithms are compared with the proposed algorithms in test suite 10D using Wilcoxon rank sum test.

DE variants	LSHADE-cnEpsin	jSO	ISDE	PaDE-pet	TDE	OLBADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{c_1}	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0
f_{c_2}	1.8391E−14/8.0954E−14(≈)	0/0(≈)	3.0411E−12/1.3716E−11(<)	0/0(≈)	0/0(≈)	0/0
f_{c_3}	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0
f_{c_4}	2.3450E−01/9.4735E−01(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0
f_{c_5}	**1.7375E+00/9.5088E−01(≈)**	1.7753E+00/7.7871E−01(>)	3.9603E+00/1.5477E+00(<)	2.3625E+00/7.1587E−01(≈)	3.0080E+00/1.6933E+00(<)	2.4786E+00/1.3693E+00
f_{c_6}	0/0(≈)	0/0(≈)	0/0(≈)	6.9728E−09/4.9748E−08(<)	8.4708E−14/5.0038E−14(<)	1.1146E−14/3.4143E−14
f_{c_7}	**1.0668E+01/1.3091E−01**(>)	1.1970E+01/6.4821E−01(>)	1.3606E+01/1.3221E+00(<)	1.2403E+01/9.6962E−01(>)	1.2725E+01/1.8744E+00(>)	1.3125E+01/1.6247E+00
f_{c_8}	**5.4828E−01/4.9837E−01**(>)	2.2241E+00/8.5801E−01(>)	4.5456E+00/2.3143E+00(<)	2.4012E+00/1.0181E+00(≈)	2.5995E+00/1.4607E+00(≈)	2.7715E+00/1.2466E+00
f_{c_9}	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0
$f_{c_{10}}$	6.1425E+01/6.7054E+01(≈)	4.5575E+01/5.9859E+01(≈)	7.7565E+01/7.8718E+01(<)	**2.3959E+01/4.0661E+01(≈)**	5.7707E+01/1.0459E+02(≈)	4.9009E+01/7.1891E+01
$f_{c_{11}}$	4.3193E+00/2.7944E−01(<)	**1.9509E−02/1.3932E−01(≈)**	2.9264E−01/4.9923E−01(<)	3.7665E−01/6.5852E−01(<)	1.4074E−01/4.2960E−01(≈)	2.5524E−02/1.6747E−01
$f_{c_{12}}$	1.0380E+02/1.4235E+02(<)	2.7110E+00/1.6775E+01(<)	1.0313E+02/1.3815E+02(<)	**3.0609E−01/1.6834E−01(≈)**	5.2109E−01/1.5891E+00(≈)	1.1896E+01/3.5791E+01
$f_{c_{13}}$	3.0486E+00/2.6006E+00(≈)	3.2823E+00/2.2204E+00(<)	4.1818E+00/2.5926E+00(<)	**1.0740E+00/1.9062E+00(≈)**	1.4628E+00/2.0983E+00(≈)	1.9078E+00/2.3572E+00
$f_{c_{14}}$	1.7558E−01/3.8307E−01(≈)	1.5607E−01/3.6544E−01(≈)	1.1709E−01/3.2374E−01(≈)	5.7182E−01/7.6495E−01(<)	6.0478E−01/7.7270E−01(<)	**1.0337E−01/3.9107E−01**
$f_{c_{15}}$	2.9634E−01/1.9989E−01(<)	3.4994E−01/1.9239E−01(<)	2.1781E−01/3.2440E−01(<)	**8.0370E−02/1.4248E−01(≈)**	1.3310E−01/1.9436E−01(≈)	1.1142E−01/1.8208E−01
$f_{c_{16}}$	7.8122E−01/3.7709E−01(<)	6.0458E−01/2.7309E−01(<)	3.0473E+00/1.8861E+01(≈)	**3.7026E−01/1.6211E−01(≈)**	4.6565E−01/2.7522E−01(≈)	4.2215E−01/2.1708E−01
$f_{c_{17}}$	1.1525E+00/3.9359E+00(<)	9.4763E−01/2.8235E+00(<)	3.2377E−01/3.2262E−01(≈)	**1.3093E−01/1.4863E−01(≈)**	2.5389E−01/2.4672E−01(≈)	1.7902E−01/2.4995E−01
$f_{c_{18}}$	2.6972E+00/6.4931E+00(<)	2.5521E−01/2.1589E−01(≈)	3.1588E−01/3.8590E−01(≈)	**1.8715E−01/1.7446E−01(≈)**	1.9785E−01/1.9947E−01(≈)	2.0743E−01/2.0219E−01
$f_{c_{19}}$	7.5084E−02/1.9710E−01(<)	1.1902E−02/1.7241E−02(≈)	2.2371E−02/1.3908E−01(>)	1.6001E−02/9.4920E−03(<)	1.1544E−02/9.6595E−03(<)	**3.9341E−03/9.3625E−03**
$f_{c_{20}}$	1.2334E+00/3.9409E+00(<)	8.2791E−01/2.8369E+00(<)	5.5089E−02/1.2019E−01(≈)	0/0(≈)	1.2242E−02/6.1198E−02(≈)	0/0
$f_{c_{21}}$	1.6442E+02/5.0135E+01(≈)	1.4247E+02/5.1274E+01(≈)	1.8516E+02/4.2498E+01(<)	**1.3284E+02/4.8376E+01(≈)**	1.3450E+02/4.9294E+01(≈)	1.4284E+02/5.1722E+01
$f_{c_{22}}$	1.0002E+02/7.7923E−02(≈)	1.0000E+02/0.0000E+00(≈)	1.0013E+02/2.2233E−01(<)	1.0002E+02/9.6731E−02(≈)	**9.8052E+01/1.4005E+01(≈)**	1.0004E+02/1.2690E−01
$f_{c_{23}}$	**3.0078E+02/1.2414E+00(≈)**	3.0109E+02/1.5983E+00(≈)	3.0484E+02/1.4949E+00(<)	3.0181E+02/2.2027E+00(<)	3.0255E+02/2.2182E+00(<)	3.0100E+02/1.7419E+00
$f_{c_{24}}$	3.0440E+02/6.8100E+01(≈)	2.6650E+02/1.0345E+02(≈)	3.2843E+02/3.2700E+01(<)	**2.5174E+02/1.0838E+02(≈)**	3.0093E+02/7.4346E+01(≈)	2.8510E+02/8.6971E+01
$f_{c_{25}}$	4.2567E+02/2.2449E+01(<)	4.0331E+02/1.4773E+01(>)	4.1902E+02/2.3491E+01(<)	**4.0324E+02/1.4796E+01**(>)	4.1215E+02/2.1311E+01(≈)	4.1664E+02/2.2572E+01
$f_{c_{26}}$	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/2.2437E−13(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00
$f_{c_{27}}$	3.9193E+02/2.1733E+00(<)	3.8942E+02/2.0549E−01(≈)	3.9061E+02/2.3229E+00(<)	3.9114E+02/2.4031E+00(<)	3.9225E+02/2.0879E+00(<)	**3.8646E+02/6.6124E+00**
$f_{c_{28}}$	3.8858E+02/1.3921E+02(<)	3.2568E+02/8.1963E+01(≈)	4.3090E+02/1.4882E+02(<)	3.0556E+02/3.9732E+01(≈)	**3.0393E+02/2.8050E+01(≈)**	3.0877E+02/3.6964E+01
$f_{c_{29}}$	**2.2815E+02/1.7224E+00**(>)	2.3458E+02/3.1148E+00(<)	2.3506E+02/5.5574E+00(<)	2.3332E+02/3.2760E+00(≈)	2.3879E+02/5.1255E+00(<)	2.3250E+02/3.5279E+00
$f_{c_{30}}$	4.9674E+04/2.0668E+05(<)	**3.9454E+02/6.2039E−02**(<)	4.6053E+02/6.2108E+01(<)	3.9476E+02/1.8110E+00(≈)	3.9553E+02/6.4511E−02(<)	4.7232E+02/5.2978E+02
>/≈/<	3/15/12	4/18/8	1/11/18	2/22/6	1/21/8	–/–/–

Table 9
The results of LSHADE-cnEpsin, jSO, ISDE, PaDE-pet and TDE algorithms are compared with the proposed algorithms in test suite 30D using Wilcoxon rank sum test.

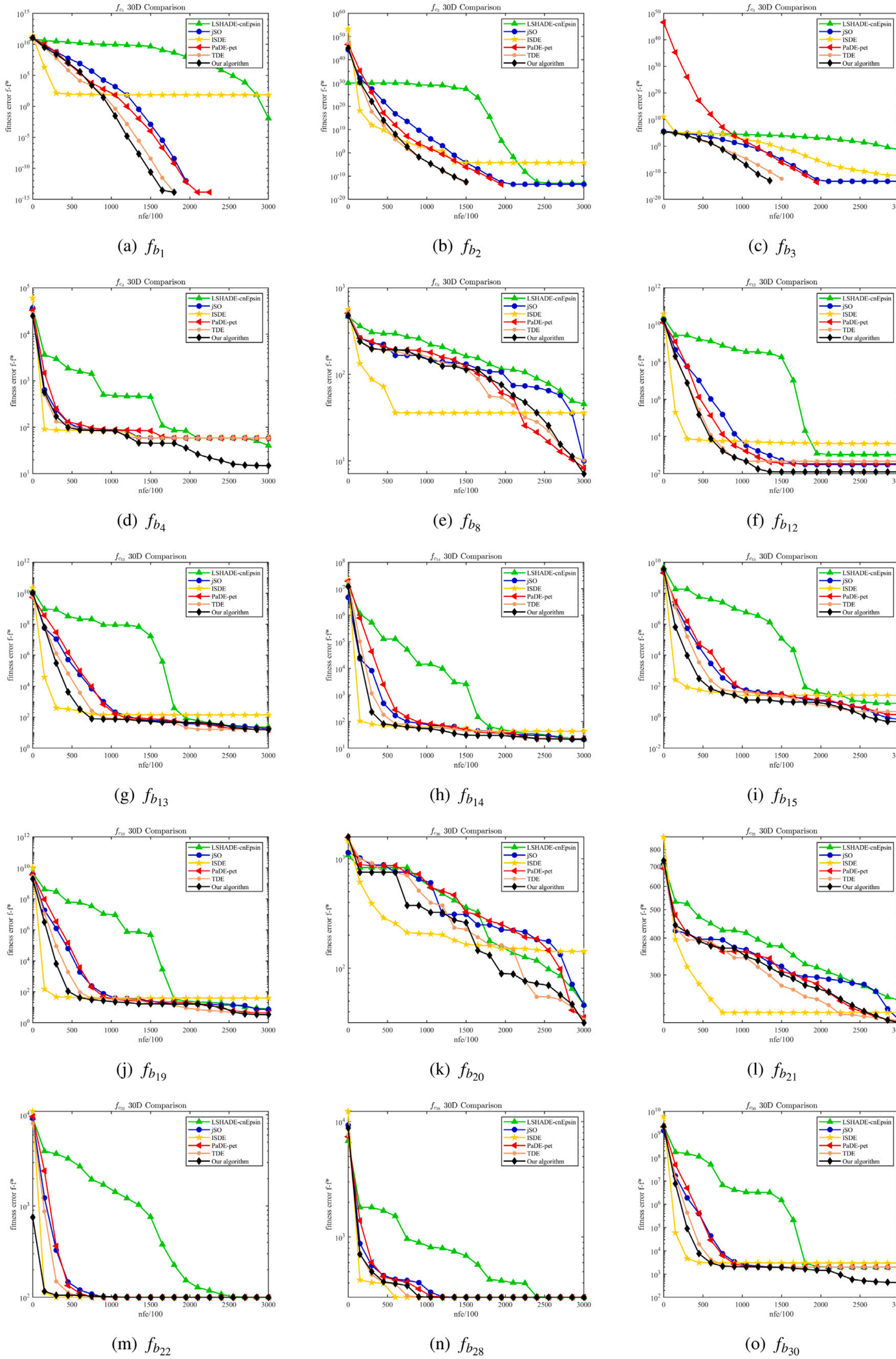
DE variants	LSHADE-cnEpsin	jSO	ISDE	PaDE-pet	TDE	OLBADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{c_1}	1.3654E−14/5.6564E−15(<)	2.2292E−15/5.2195E−15(≈)	3.1033E+02/7.8284E+02(<)	1.6719E−15/4.6242E−15(≈)	0/0(≈)	0/0
f_{c_2}	5.7401E−14/7.8660E−14(<)	**0/0**(>)	2.4135E−04/5.8817E−04(<)	5.5729E−16/3.9798E−15(>)	5.5729E−16/3.9798E−15(>)	8.3593E−15/1.3079E−14
f_{c_3}	0/0(≈)	4.3468E−14/2.6875E−14(<)	5.6843E−14/6.7258E−14(<)	5.5729E−15/1.7072E−14(≈)	0/0(≈)	0/0
f_{c_4}	4.0525E+01/2.5722E+00(<)	5.8670E+01/7.7797E−01(<)	4.8000E+01/2.3625E+01(<)	5.7522E+01/8.2525E+00(<)	5.6324E+01/1.1525E+01(<)	**1.5091E+01/8.0275E−01**
f_{c_5}	1.0734E+01/2.0822E+00(<)	8.6048E+00/1.9887E+00(<)	3.0372E+01/8.2212E+00(<)	7.8510E+00/1.8149E+00(<)	1.0105E+01/2.2351E+00(<)	**6.7708E+00/1.5040E+00**
f_{c_6}	1.2414E−11/8.7843E−11(>)	3.2808E−08/1.5573E−07(>)	2.2880E−05/1.5675E−04(≈)	**1.0254E−13/4.1021E−14**(>)	1.1369E−13/0.0000E+00(>)	1.3033E−06/3.2691E−06
f_{c_7}	4.1241E+01/1.9022E+00(<)	3.8443E+01/2.0003E+00(<)	6.5355E+01/8.7166E+00(<)	3.7833E+01/1.7855E+00(<)	3.9249E+01/2.4724E+00(<)	**3.6989E+01/1.1864E+00**
f_{c_8}	1.1998E+01/1.9942E+00(<)	8.9693E+00/2.1989E+00(<)	3.2248E+01/8.6422E+00(<)	9.0059E+00/2.2149E+00(<)	1.0391E+01/2.2392E+00(<)	**6.3777E+00/1.3448E+00**
f_{c_9}	0/0(≈)	0/0(≈)	9.9092E−02/2.3527E−01(<)	0/0(≈)	0/0(≈)	0/0
$f_{c_{10}}$	**1.3481E+03/2.0362E+02**(>)	1.5725E+03/2.3732E+02(≈)	1.7245E+03/4.4774E+02(<)	1.4962E+03/2.0184E+02(>)	1.5785E+03/2.9953E+02(≈)	1.5897E+03/2.1073E+02
$f_{c_{11}}$	5.6204E+00/1.3399E+00(<)	**4.2944E+00/8.2863E+00(≈)**	2.7788E+01/2.2895E+01(<)	9.8310E+00/1.9130E+01(≈)	9.0786E+00/1.6677E+01(<)	5.8304E+00/1.1828E+01
$f_{c_{12}}$	7.5392E+02/2.5071E+02(<)	1.9033E+02/1.3062E+02(<)	1.1709E+04/7.0957E+03(<)	4.0452E+02/2.2049E+02(<)	3.6085E+02/2.3124E+02(<)	**1.0205E+02/7.7462E+01**
$f_{c_{13}}$	2.0668E+01/1.5294E+01(<)	1.5355E+01/5.8308E+00(<)	4.7128E+03/7.0994E+03(<)	**1.3105E+01/6.5884E+00(≈)**	1.4582E+01/5.8020E+00(<)	1.3727E+01/5.3334E+00
$f_{c_{14}}$	2.1820E+01/3.9929E+00(<)	2.1319E+01/4.0986E+00(<)	4.0269E+01/1.5800E+01(<)	2.2078E+01/4.9272E+00(<)	**1.9239E+01/8.6782E+00**(<)	2.1179E+01/3.0949E+00
$f_{c_{15}}$	5.6760E+00/3.0113E+00(<)	1.2194E+00/7.1566E−01(<)	3.6694E+01/3.5871E+01(<)	1.4809E+00/9.4431E−01(<)	2.1701E+00/1.4019E+00(<)	**8.1972E−01/6.1372E−01**
$f_{c_{16}}$	**1.5536E+01/1.7362E+01**(>)	7.2027E+01/8.4558E+01(≈)	3.0296E+02/1.8168E+02(<)	1.2654E+02/9.1668E+01(<)	1.4452E+02/9.5752E+01(<)	7.8815E+01/7.4252E+01
$f_{c_{17}}$	**2.5537E+01/6.7229E+00**(>)	3.2809E+01/9.0650E+00(<)	3.5230E+01/3.1291E+01(≈)	3.3053E+01/8.2070E+00(<)	3.3623E+01/7.7080E+00(<)	2.8744E+01/7.0387E+00
$f_{c_{18}}$	2.3574E+01/2.0583E+00(<)	2.0855E+01/4.2168E−01(<)	1.1372E+03/2.3157E+03(<)	2.0716E+01/4.0094E−01(≈)	2.0678E+01/4.2064E+00(<)	**1.9782E+01/3.9284E+00**
$f_{c_{19}}$	4.8492E+00/1.2405E+00(<)	4.4298E+00/1.9901E+00(≈)	1.7853E+01/1.1931E+01(<)	4.5440E+00/1.4861E+00(≈)	**3.9009E+00/1.3319E+00(≈)**	3.9629E+00/1.1232E+00
$f_{c_{20}}$	**2.9120E+01/4.4923E+00**(>)	3.0435E+01/5.0635E+00(≈)	7.3778E+01/6.2155E+01(≈)	3.9516E+01/2.4573E+01(<)	3.5909E+01/1.9218E+01(<)	3.1167E+01/6.6696E+00
$f_{c_{21}}$	2.1061E+02/2.6415E+00(<)	2.0936E+02/2.4318E+00(<)	2.3192E+02/8.6853E+00(<)	2.0848E+02/1.6566E+00(<)	2.0923E+02/2.5085E+00(<)	**2.0670E+02/1.4760E+00**
$f_{c_{22}}$	1.0000E+02/1.4352E−14(≈)	1.0000E+02/1.4352E−14(≈)	3.0698E+02/6.3779E+02(≈)	1.0000E+02/8.9223E−14(≈)	1.0000E+02/1.4352E−14(≈)	1.0000E+02/1.4352E−14
$f_{c_{23}}$	3.4933E+02/3.3386E+00(<)	3.5104E+02/2.6622E+00(<)	3.8325E+02/8.9163E+00(<)	**3.4337E+02/3.3192E+00**(>)	3.4407E+02/4.7870E+00(>)	3.4565E+02/3.4896E+00
$f_{c_{24}}$	4.2381E+02/2.7869E+00(<)	4.2705E+02/2.0143E+00(<)	4.4868E+02/8.2440E+00(<)	**4.2035E+02/2.3610E+00**(>)	4.2046E+02/3.1771E+00(>)	4.2216E+02/2.2418E+00
$f_{c_{25}}$	3.8670E+02/2.3545E−02(<)	3.8670E+02/6.3724E−03(<)	3.8705E+02/5.5833E−01(<)	3.8673E+02/1.7575E−02(<)	3.8624E+02/1.3709E+00(<)	**3.7858E+02/1.0537E−01**
$f_{c_{26}}$	8.7488E+02/5.4702E+01(≈)	9.3945E+02/4.8108E+01(<)	1.3101E+03/1.1271E+02(<)	8.6160E+02/3.5079E+01(>)	**8.6030E+02/5.4212E+01(≈)**	8.7575E+02/3.3590E+01
$f_{c_{27}}$	5.0000E+02/5.9041E+00(<)	4.9451E+02/5.9813E+00(<)	5.0214E+02/5.3206E+00(<)	4.9946E+02/7.6782E+00(<)	4.9360E+02/5.4948E+00(<)	**4.7267E+02/6.7395E+00**
$f_{c_{28}}$	3.1055E+02/3.2346E+01(≈)	3.0670E+02/2.7084E+01(≈)	3.3890E+02/5.6790E+01(<)	3.1149E+02/3.5663E+01(≈)	3.3450E+02/5.1615E+01(<)	**3.0223E+02/1.5897E+01**
$f_{c_{29}}$	**4.2831E+02/1.6071E+01(≈)**	4.3694E+02/1.2304E+01(<)	4.3404E+02/3.7383E+01(≈)	4.3463E+02/1.0376E+01(≈)	4.3355E+02/9.5656E+00(≈)	4.3233E+02/1.2197E+01
$f_{c_{30}}$	2.1158E+03/8.3564E+01(<)	1.9682E+03/9.5995E+00(<)	2.7775E+03/1.0215E+03(<)	1.9953E+03/2.5576E+01(<)	1.9986E+03/2.4588E+01(<)	**4.2212E+02/4.1592E+01**
>/≈/<	5/6/19	2/9/19	0/5/25	6/10/14	4/8/18	–/–/–

From Figs. 4 to 6, we can observe that OLBADE is able obtain the best convergence performance on most benchmark functions from the three test suites. It can be concluded that OLBADE is highly competitive with other powerful DE variants in terms of convergence speed.

4.3. Parameter sensitivity analysis

In this part, we examine the fitness-based threshold N used in diversity enhancement strategy. Six cases of the N including 5D, 10D, 15D, 20D, 25D, 30D, and 35D are tested under CEC2022 with 20 dimensions. Table 14 presents the experiment results of OLBADE with different values of N and results of the Friedman test.

From Table 14, it is evident that the value of parameter N is closely related to the characteristics of the functions. We can observe that when N equals to 25D OLBADE will obtains the best performance. It can be also validated by the fact that when N is set as 5D, 10D, 15D, 20D, 25D, 30D, and 35D average rankings are 4.58, 3.51, 3.71, 3.33, 1.58, 3.91 and 3.54, respectively. Therefore, the recommended value N is 25D to maintain a better balance between convergence speed and diversity degree.



(a) f_{b_1} (b) f_{b_2} (c) f_{b_3}

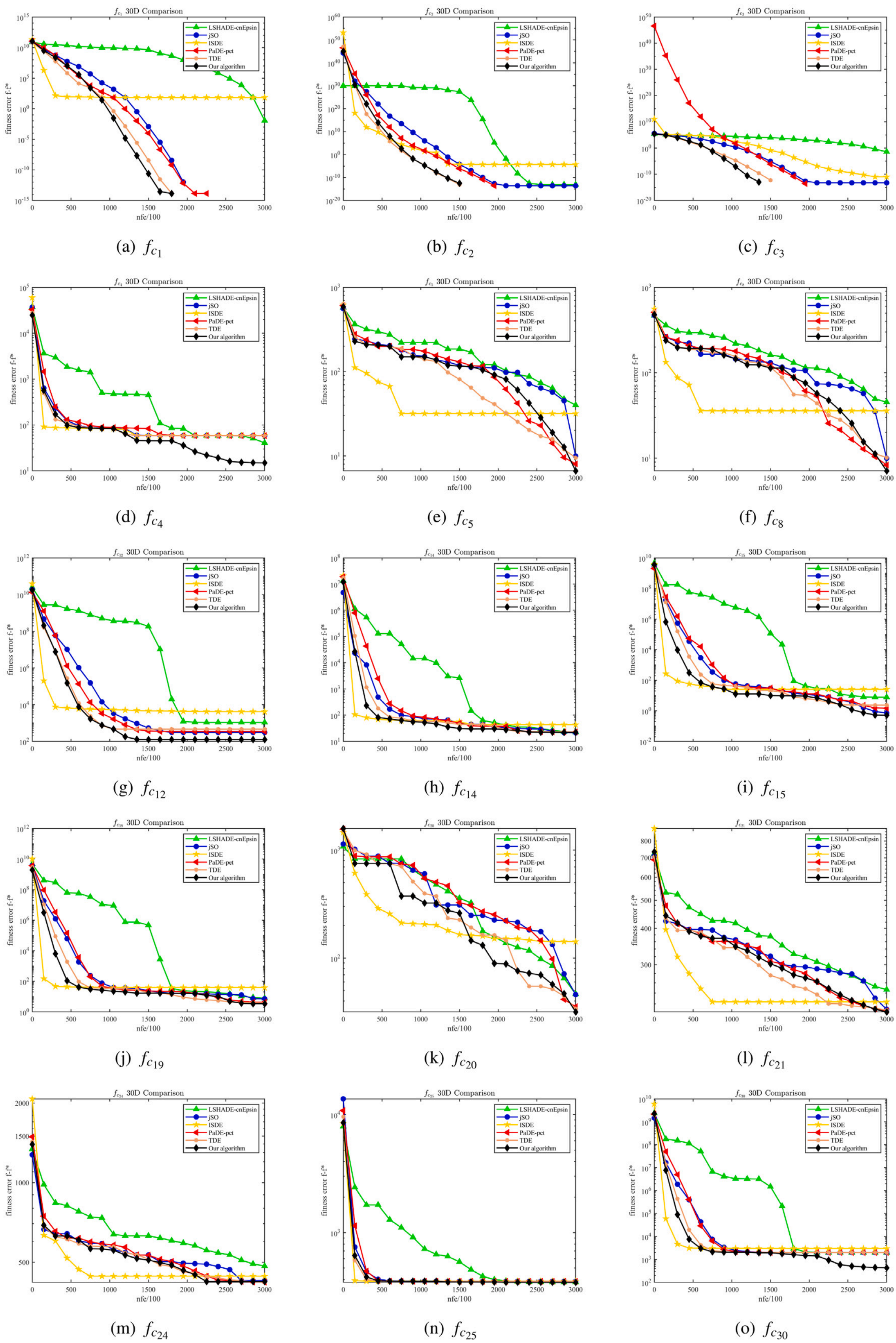
(d) f_{b_4} (e) f_{b_8} (f) $f_{b_{12}}$

(g) $f_{b_{13}}$ (h) $f_{b_{14}}$ (i) $f_{b_{15}}$

(j) $f_{b_{19}}$ (k) $f_{b_{20}}$ (l) $f_{b_{21}}$

(m) $f_{b_{22}}$ (n) $f_{b_{28}}$ (o) $f_{b_{30}}$

Fig. 4. Here we present the convergence curves by using the median value of 51 runs under CEC2014 with 50 dimensions.



(a) f_{c_1} (b) f_{c_2} (c) f_{c_3}

(d) f_{c_4} (e) f_{c_5} (f) f_{c_8}

(g) $f_{c_{12}}$ (h) $f_{c_{14}}$ (i) $f_{c_{15}}$

(j) $f_{c_{19}}$ (k) $f_{c_{20}}$ (l) $f_{c_{21}}$

(m) $f_{c_{24}}$ (n) $f_{c_{25}}$ (o) $f_{c_{30}}$

Fig. 5. Here we present the convergence curves by using the median value of 51 runs under CEC2017 with 30 dimensions.

Table 10
The results of LSHADE-cnEpsin, jSO, ISDE, PaDE-pet and TDE algorithms are compared with the proposed algorithms in test suite 50D using Wilcoxon rank sum test.

DE variants	LSHADE-cnEpsin	jSO	ISDE	PaDE-pet	TDE	OLBADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{c_1}	9.5849E−11/3.2674E−10(<)	3.0930E−14/7.8897E−15(<)	3.7241E+03/4.7808E+03(<)	1.5883E−14/4.6242E−15(>)	**1.3932E−14/1.9899E−15**(>)	2.4521E−14/7.0072E−15
f_{c_2}	3.2512E−12/1.4288E−11(<)	6.2416E−14/6.8689E−14(<)	2.2377E+00/1.5973E+01(<)	6.0744E−14/8.3353E−14(<)	**1.9505E−14/1.7511E−14(≈)**	2.2849E−14/1.7993E−14
f_{c_3}	4.3580E−13/5.4790E−13(<)	2.8310E−13/8.9148E−14(<)	3.4106E−12/5.6065E−12(<)	1.2818E−13/3.5668E−14(≈)	**1.0923E−13/4.0742E−14(≈)**	1.2037E−13/3.5311E−14
f_{c_4}	7.7701E+01/3.9611E+00(<)	5.6418E+01/4.7040E+01(<)	7.6234E+01/4.2179E+01(<)	8.1938E+01/4.7422E+01(<)	8.9316E+01/4.2289E+01(<)	**4.9473E+01/4.1021E+01**
f_{c_5}	2.4501E+01/5.2308E+00(<)	1.6199E+01/2.4788E+00(<)	7.5377E+01/1.5733E+01(<)	1.5232E+01/2.2404E+00(<)	1.9370E+01/3.1963E+00(<)	**1.3457E+01/2.0566E+00**
f_{c_6}	7.7560E−09/1.7586E−08(>)	4.7353E−07/6.6856E−07(>)	5.7571E−07/2.5059E−06(>)	**2.8203E−09/1.1393E−08**(>)	4.7004E−09/1.4399E−08(>)	2.5978E−03/2.3964E−03
f_{c_7}	7.4126E+01/5.1152E+00(<)	6.6238E+01/3.2214E+00(<)	1.1979E+02/1.5192E+01(<)	6.4081E+01/2.2254E+00(<)	6.6881E+01/3.4933E+00(<)	**6.2497E+01/1.7170E+00**
f_{c_8}	2.4821E+01/5.7810E+00(<)	1.6906E+01/2.6186E+00(<)	6.9287E+01/1.5826E+01(<)	1.6037E+01/2.7941E+00(<)	2.0667E+01/4.0021E+00(<)	**1.3807E+01/2.0687E+00**
f_{c_9}	0/0(>)	8.6937E−14/4.8704E−14(≈)	7.9781E−01/1.6032E+00(<)	6.6875E−15/2.7016E−14(>)	0/0(>)	1.4044E−02/3.2883E−02
$f_{c_{10}}$	3.2484E+03/3.9974E+02(≈)	3.1902E+03/3.3435E+02(≈)	3.5280E+03/7.4439E+02(<)	**3.1230E+03/3.2389E+02**(>)	3.1723E+03/3.7679E+02(≈)	3.3028E+03/3.0719E+02
$f_{c_{11}}$	3.8484E+01/2.3131E+01(<)	2.7776E+01/3.1155E+00(<)	9.2134E+01/2.7463E+01(<)	3.3166E+01/5.0982E+00(<)	3.8667E+01/6.3774E+00(<)	**1.5385E+01/2.5994E+00**
$f_{c_{12}}$	**1.7269E+03/4.5286E+02**(>)	1.7865E+03/4.8589E+02(≈)	9.3641E+04/5.4037E+04(<)	1.9798E+03/5.0899E+02(≈)	1.7806E+03/5.4930E+02(≈)	1.8929E+03/3.9192E+02
$f_{c_{13}}$	1.0570E+02/5.8202E+01(<)	**2.8161E+01/2.4834E+01**(>)	4.9044E+03/5.8484E+03(<)	4.1374E+01/2.3374E+01(≈)	4.8555E+01/2.2430E+01(<)	3.2748E+01/2.0964E+01
$f_{c_{14}}$	2.9853E+01/3.2617E+00(<)	2.4916E+01/1.9607E+00(≈)	1.1488E+03/1.2459E+03(<)	2.6486E+01/2.4367E+00(<)	2.8551E+01/2.5149E+00(<)	**2.4544E+01/1.7777E+00**
$f_{c_{15}}$	3.6313E+01/8.2889E+00(<)	2.2984E+01/2.5989E+00(<)	2.0729E+03/2.9338E+03(<)	2.5549E+01/3.6365E+00(<)	2.8251E+01/4.9601E+00(<)	**2.1107E+01/2.7983E+00**
$f_{c_{16}}$	**2.3044E+02/8.8438E+01**(>)	4.2207E+02/1.2573E+02(<)	8.7435E+02/2.5067E+02(<)	3.7239E+02/1.1526E+02(≈)	3.6731E+02/1.2526E+02(≈)	3.3090E+02/1.1737E+02
$f_{c_{17}}$	**2.1132E+02/6.2962E+01**(>)	2.8670E+02/1.0996E+02(≈)	5.3395E+02/1.9793E+02(<)	3.0324E+02/8.4017E+01(≈)	3.2762E+02/8.0603E+01(<)	2.7099E+02/6.6353E+01
$f_{c_{18}}$	3.2669E+01/8.1842E+00(<)	2.4906E+01/2.2626E+00(<)	7.0704E+03/4.1689E+03(<)	2.6212E+01/2.6218E+00(<)	2.7269E+01/2.7494E+00(<)	**2.2213E+01/1.2390E+00**
$f_{c_{19}}$	1.8736E+01/3.1538E+00(<)	1.3194E+01/2.5710E+00(<)	4.9997E+03/6.3772E+03(<)	1.5038E+01/2.6039E+00(<)	1.4341E+01/2.3967E+00(<)	**1.1798E+01/2.2113E+00**
$f_{c_{20}}$	**1.0412E+02/2.6077E+01**(>)	1.4218E+02/8.1732E+01(≈)	2.7681E+02/1.4946E+02(<)	1.6767E+02/6.6374E+01(<)	2.0107E+02/8.6758E+01(<)	1.3871E+02/5.0088E+01
$f_{c_{21}}$	2.2444E+02/6.8095E+00(<)	2.1644E+02/3.0197E+00(<)	2.7175E+02/1.5911E+01(<)	2.1653E+02/2.3030E+00(<)	2.2030E+02/2.5959E+00(<)	**2.1437E+02/2.4019E+00**
$f_{c_{22}}$	1.3606E+03/1.6394E+03(<)	1.4131E+03/1.7393E+03(≈)	4.1522E+03/8.8582E+02(<)	3.7546E+02/9.4627E+02(≈)	**1.0224E+02/1.1348E+01(≈)**	4.0966E+02/1.0503E+03
$f_{c_{23}}$	4.3146E+02/8.1403E+00(<)	4.3220E+02/6.6582E+00(<)	5.0039E+02/1.8377E+01(<)	**4.2077E+02/5.6655E+00**(>)	4.2385E+02/8.5303E+00(≈)	4.2524E+02/8.3491E+00
$f_{c_{24}}$	5.0290E+02/6.0811E+00(<)	5.0801E+02/3.9418E+00(<)	5.6176E+02/1.7988E+01(<)	**4.9872E+02/4.7650E+00(≈)**	5.0056E+02/6.1967E+00(≈)	4.9954E+02/5.6056E+00
$f_{c_{25}}$	4.8981E+02/1.7236E+01(<)	4.8173E+02/4.1139E+00(<)	5.0161E+02/3.4103E+01(<)	4.8257E+02/7.5561E+00(<)	4.8549E+02/1.5055E+01(<)	**4.7727E+02/4.4610E+00**
$f_{c_{26}}$	1.0976E+03/9.8700E+01(<)	1.1565E+03/6.2196E+01(<)	1.8996E+03/2.1401E+02(<)	**1.0440E+03/7.1293E+01(≈)**	1.1056E+03/7.0658E+01(<)	1.0630E+03/7.6714E+01
$f_{c_{27}}$	5.1841E+02/1.2835E+01(<)	5.1181E+02/9.6220E+00(<)	5.4707E+02/2.4362E+01(<)	5.2692E+02/7.4058E+00(<)	5.1578E+02/1.4343E+01(<)	**4.8855E+02/1.8572E+01**
$f_{c_{28}}$	4.7419E+02/2.3370E+01(<)	4.5981E+02/6.8398E+00(<)	4.7703E+02/2.2988E+01(<)	4.6555E+02/1.6976E+01(<)	4.8073E+02/2.0898E+01(<)	**4.5017E+02/1.0705E+00**
$f_{c_{29}}$	3.5774E+02/1.0794E+01(≈)	3.6272E+02/1.4434E+01(<)	4.7489E+02/1.2466E+02(<)	3.4975E+02/9.6040E+00(≈)	3.5690E+02/1.6565E+01(≈)	**3.3923E+02/3.6303E+01**
$f_{c_{30}}$	6.2815E+05/6.2035E+04(<)	6.1103E+05/4.1635E+04(<)	6.4838E+05/8.1502E+04(<)	6.2293E+05/3.4154E+04(<)	6.0657E+05/3.2568E+04(<)	**7.7030E+02/2.3309E+02**
>/≈/<	6/2/22	2/7/21	1/0/29	5/9/16	3/9/18	–/–/–

Table 11
The results of LSHADE-cnEpsin, jSO, ISDE, PaDE-pet and TDE algorithms are compared with the proposed algorithms in test suite 10D using Wilcoxon rank sum test.

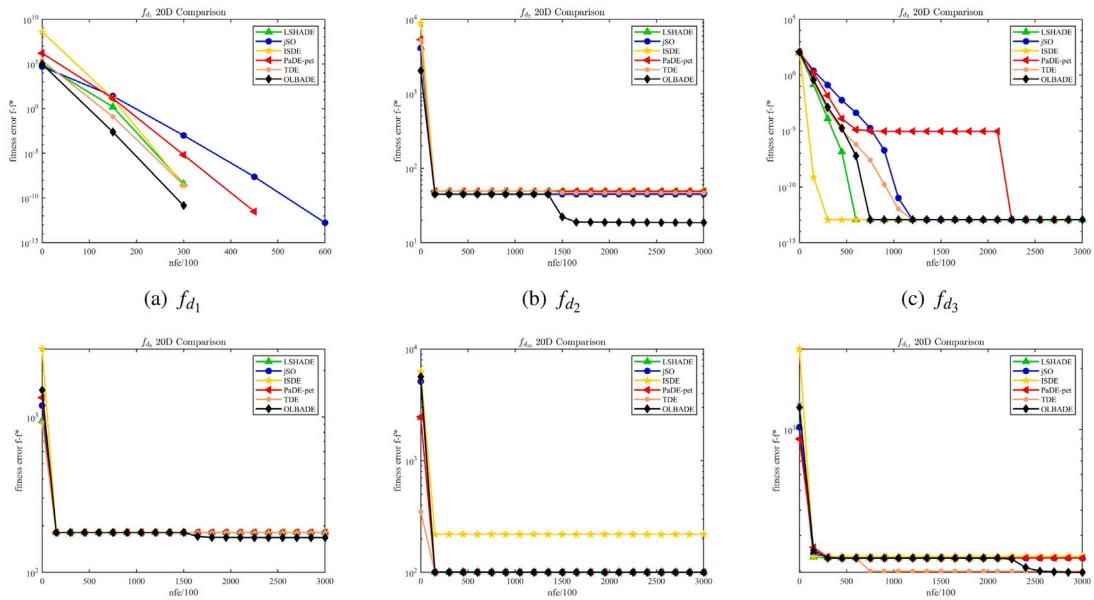
DE variants	LSHADE-cnEpsin	jSO	ISDE	PaDE-pet	TDE	OLBADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{d_1}	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0
f_{d_2}	4.0746E+00/3.3365E+00(<)	1.9130E+00/2.4092E+00(<)	5.7819E+00/2.8057E+00(<)	**5.6567E−01/1.6890E+00**(>)	1.0347E+00/1.9986E+00(>)	1.3355E+00/1.8356E+00
f_{d_3}	0/0(≈)	0/0(≈)	0/0(≈)	7.5791E−14/5.4126E−14(<)	6.6875E−15/2.7016E−14(<)	0/0
f_{d_4}	**1.3073E+00/6.4447E−01**(>)	2.7118E+00/7.7270E−01(>)	4.3895E+00/1.7368E+00(<)	2.7508E+00/1.2520E+00(>)	3.0253E+00/1.9697E+00(<)	2.9654E+00/1.6104E+00
f_{d_5}	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0
f_{d_6}	4.3238E−01/6.8451E−01(<)	2.6070E−01/1.4855E−01(<)	8.2219E−01/9.2749E−01(<)	**1.7340E−01/1.6065E−01**(>)	2.7931E−01/1.8861E−01(<)	2.5810E−01/1.5306E−01
f_{d_7}	1.4188E−01/3.1938E−01(<)	1.1309E−01/4.0320E−01(<)	4.3235E−01/2.8894E+00(<)	1.0767E−05/6.4578E−05(>)	**6.8504E−07/1.7849E−06**(>)	9.1625E−02/2.8015E−01
f_{d_8}	2.8736E+00/6.9327E+00(<)	6.2029E−01/2.2245E+00(<)	2.1901E+00/6.0208E+00(<)	5.6302E−01/5.9407E−01(<)	6.0004E−01/7.5693E−01(<)	**2.6933E−01/1.9084E−01**
f_{d_9}	2.2434E+02/3.2586E+00(<)	2.2928E+02/0.0000E+00(<)	2.2928E+02/0.0000E+00(<)	2.2928E+02/0.0000E+00(<)	2.2632E+02/4.2880E+00(<)	**1.8585E+02/4.5609E−01**
$f_{d_{10}}$	1.0407E+02/1.9575E+01(<)	1.0018E+02/2.1690E−02(<)	1.1050E+02/3.1668E+01(<)	1.0019E+02/2.0813E−02(<)	1.0227E+02/1.4780E+01(<)	**1.0016E+02/1.4737E+01**
$f_{d_{11}}$	0/0(≈)	0/0(≈)	0/0(≈)	3.4775E−13/1.9482E−13(<)	1.7833E−14/8.9148E−14(<)	0/0
$f_{d_{12}}$	1.6379E+02/1.3527E+00(<)	1.6468E+02/6.8770E−01(<)	1.6400E+02/9.8838E−01(<)	1.6480E+02/5.2906E−01(<)	1.6365E+02/1.6042E+00(<)	**1.4631E+02/3.1470E−01**
>/≈/<	1/4/7	1/4/7	0/4/8	4/2/6	2/2/8	–/–/–

Table 12
The results of LSHADE-cnEpsin, jSO, ISDE, PaDE-pet and TDE algorithms are compared with the proposed algorithms in test suite 20D using Wilcoxon rank sum test.

DE variants	LSHADE-cnEpsin	jSO	ISDE	PaDE-pet	TDE	OLBADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{d_1}	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0(≈)	0/0
f_{d_2}	2.5114E+01/8.9149E+00(<)	4.5470E+01/1.4558E+00(<)	4.8838E+01/9.9546E−01(<)	4.8017E+01/1.8436E+00(<)	4.4455E+01/5.5827E+00(<)	**1.8591E+01/1.4826E−01**
f_{d_3}	**4.4583E−15/2.2287E−14**(>)	6.4645E−14/5.6866E−14(>)	4.0585E−07/2.0292E−06(≈)	5.1271E−14/5.7132E−14(>)	6.9104E−14/5.6058E−14(≈)	2.1395E−14/4.5586E−14
f_{d_4}	5.2480E+00/1.2279E+00(>)	7.7841E+00/1.5247E+00(<)	1.4573E+01/5.1539E+00(<)	5.1699E+00/8.9035E−01(>)	**4.6237E+00/1.2238E+00**(>)	6.6086E+00/3.1765E+00
f_{d_5}	0/0(≈)	0/0(≈)	1.7555E−03/1.2536E−02(≈)	0/0(≈)	0/0(≈)	0/0
f_{d_6}	3.7142E+00/2.4471E+00(<)	4.7230E−01/4.3372E−02(<)	1.9718E+02/5.6180E+02(<)	4.9789E−01/1.1035E−02(<)	5.1753E−01/2.1000E−01(<)	**4.6284E−01/8.9812E−02**
f_{d_7}	6.9960E+00/9.4526E+00(≈)	1.5121E+01/8.3229E+00(<)	9.6320E+00/8.9730E+00(<)	3.7787E+00/1.9651E+00(<)	4.0084E+00/6.7939E+00(<)	**2.7680E−01/7.3006E+00**
f_{d_8}	1.6404E+01/7.9341E+00(<)	2.0214E+01/2.1658E+00(<)	1.9076E+01/4.9069E+00(<)	1.8122E+01/3.0640E+00(≈)	**9.1292E+00/9.8106E+00**(>)	1.2082E+01/9.8736E+00
f_{d_9}	1.7870E+02/9.7128E−01(<)	1.8078E+02/0.0000E+00(<)	1.8078E+02/4.5927E−13(<)	1.8078E+02/0.0000E+00(<)	1.8027E+02/9.0919E−01(<)	**1.6732E+02/2.9630E−01**
$f_{d_{10}}$	1.0240E+02/1.5491E+01(<)	**1.0021E+02/1.3894E−02**(>)	1.1376E+02/4.9855E+01(>)	1.0028E+02/2.6575E−02(<)	1.0028E+02/3.6285E−02(<)	1.0023E+02/3.0553E−02
$f_{d_{11}}$	3.0196E+02/1.4003E+01(<)	3.0000E+02/9.0949E−14(≈)	3.0980E+02/3.0033E+01(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/2.2964E−13(≈)	3.0000E+02/0.0000E+00
$f_{d_{12}}$	2.3328E+02/2.9675E+00(<)	2.3194E+02/1.3436E+00(<)	2.3351E+02/2.8883E+00(<)	2.3269E+02/1.7362E+00(<)	2.0399E+02/7.7487E+00(<)	**1.9987E+02/9.7106E+00**
>/≈/<	2/3/7	2/3/7	1/4/7	2/4/6	2/4/6	–/–/–

Table 13
Summary of comparison results between OLBADE and other algorithms under Wilcoxon rank sum test.

Test suit:	CEC2013			CEC2014			CEC2017			CEC2022		All
>/≈/<	D = 10	D = 30	D = 50	D = 10	D = 30	D = 50	D = 10	D = 30	D = 50	D = 10	D = 20	∑
LSHADE-cnEpsin	6/19/3	7/14/7	11/1/16	8/10/12	4/9/17	8/5/17	3/15/12	5/6/19	6/2/22	1/4/7	2/3/7	61/88/139
jSO	3/14/11	4/11/13	10/1/17	5/15/10	0/11/19	5/6/19	4/18/8	2/9/19	2/7/21	1/4/7	2/3/7	38/99/151
ISDE	1/10/17	4/8/16	5/1/22	7/8/15	6/3/21	5/3/22	1/11/18	0/5/25	1/0/29	0/4/8	1/4/7	31/57/200
PaDE-pet	5/18/5	12/10/6	10/2/16	8/16/6	8/8/14	13/4/13	2/22/6	6/10/14	5/9/16	4/2/6	2/4/6	75/105/108
TDE	6/15/7	12/10/6	12/2/14	7/16/7	6/8/16	11/3/16	1/21/8	4/8/18	3/9/18	2/2/8	2/4/6	66/98/124



(a) f_{d_1} (b) f_{d_2} (c) f_{d_3}

(d) f_{d_9} (e) $f_{d_{10}}$ (f) $f_{d_{12}}$

Fig. 6. Here we present the convergence curves by using the median value of 51 runs under CEC2022 with 20 dimensions.

Table 14
The sensitivity analysis of N in diversity enhancement strategy under CEC2022 with 20 dimensions.

N	N = 5D	N = 10D	N = 15D	N = 20D	N = 30D	N = 35D	N = 25D
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{d_1}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{d_2}	1.9901E+01/6.4594E−01(<)	1.9560E+01/7.0699E−01(<)	1.8604E+01/4.0032E−01(<)	1.8627E+01/2.6059E−01(<)	1.8640E+01/5.5741E−01(<)	**1.8447E+01/5.0707E−01**(>)	1.8591E+01/1.4826E−01
f_{d_3}	1.0700E−13/2.7016E−14(<)	9.8083E−14/3.9511E−14(<)	1.0477E−13/3.0869E−14(<)	1.0031E−13/3.6993E−14(<)	1.0254E−13/3.4143E−14(<)	9.5854E−14/4.1756E−14(<)	**2.1395E−14/4.5586E−14**
f_{d_4}	7.8267E+00/2.7499E+00(<)	6.6149E+00/2.9762E+00(<)	7.9598E+00/3.0439E+00(<)	6.9104E+00/2.2513E+00(<)	7.4618E+00/2.2717E+00(<)	8.0574E+00/2.8370E+00(<)	**6.6086E+00/3.1765E+00**
f_{d_5}	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
f_{d_6}	**4.4990E−01/7.2673E−02**(>)	4.6945E−01/4.6903E−02(<)	4.7269E−01/6.1135E−02(<)	4.7292E−01/8.6962E−02(<)	4.7858E−01/3.7701E−02(<)	4.7354E−01/7.4409E−02(<)	4.6284E−01/8.9812E−02
f_{d_7}	1.9412E+01/6.6213E+00(<)	4.9088E+00/4.8118E+00(<)	5.1111E+00/7.9848E+00(<)	3.5333E+00/7.0011E+00(<)	6.6826E+00/9.0818E+00(<)	4.1884E+00/7.6888E+00(<)	**2.7680E−01/7.3006E+00**
f_{d_8}	1.6975E+01/7.5871E+00(<)	1.5423E+01/8.3372E+00(<)	1.6049E+01/7.0531E+00(<)	1.3175E+01/9.6692E+00(<)	1.3676E+01/9.3891E+00(<)	1.2118E+01/9.8311E+00(<)	**1.2082E+01/9.8736E+00**
f_{d_9}	**1.6697E+02/4.3484E−01**(>)	1.6741E+02/1.8697E−01(<)	1.6735E+02/1.1374E−01(<)	1.6736E+02/2.1300E−01(<)	1.6734E+02/2.5366E−01(<)	1.6738E+02/1.5557E−01(<)	1.6732E+02/2.9630E−01
$f_{d_{10}}$	1.0028E+02/3.4949E−02(<)	1.0026E+02/3.5780E−02(<)	1.0024E+02/2.6634E−02(<)	*1.0023E+02/2.4838E−02(≈)*	*1.0023E+02/3.0253E−02(≈)*	1.0024E+02/2.5203E−02(<)	*1.0023E+02/3.0553E−02*
$f_{d_{11}}$	3.0196E+02/1.4003E+01(<)	3.0392E+02/1.9604E+01(<)	*3.0000E+02/1.8524E−13(≈)*	3.0392E+02/1.9604E+01(<)	3.0196E+02/1.4003E+01(<)	3.0196E+02/1.4003E+01(<)	*3.0000E+02/0*
$f_{d_{12}}$	2.0021E+02/4.7546E+00(<)	**1.9886E+02/5.4616E+00**(>)	1.9897E+02/6.8742E+00(>)	1.9947E+02/7.9267E+00(>)	1.9993E+02/3.4532E+00(<)	1.9978E+02/9.5587E+00(>)	1.9987E+02/9.7106E+00
>/≈/<	2/2/8	1/2/9	1/3/8	1/3/8	0/3/9	2/2/8	−/−/−
Rank	4.58	3.51	3.71	3.33	3.91	3.54	1.58

4.4. Ablation experiments

To demonstrate the effectiveness of three components proposed in OBLADE, ablation experiments were conducted. Based on the original OLBADE, we develop three variants: the first is OLBADE with Cauchy distribution used exclusively for parameter adaptation, named as OLBADE-1; the second is OLBADE without perturbation strategy, named as OLBADE-2 and the third is OLBADE without opposition learning-based diversity enhancement, named as OLBADE-3. All algorithms were examined under CEC2022 with 20 dimensions. From Table 15, we can observe that OLBADE obtains better or similar performance on 9 out of 12 benchmark functions comparing with OLBADE-1, thus indicating the proposed parameter control strategy is able to generate more appropriate scale factor F according to different stages of evolution. From Tables 16 and 17, it can be seen that OLBADE performs better for complicated benchmark functions compared with OLBADE-2 and OLBADE-3. The reason may lie in that the stagnation problem in complicated benchmark functions can be mitigated by perturbation strategy and diversity enhancement mechanism.

Table 15
Comparison between OLBADE-1 and OLBADE.

Variants	OLBADE-1	OLBADE
f_{d_1}	*0/0(≈)*	*0/0*
f_{d_2}	1.8687E+01/5.5557E−02(<)	**1.8591E+01/1.4826E−01**
f_{d_3}	9.5854E−14/4.1756E−14(<)	**2.1395E−14/4.5586E−14**
f_{d_4}	**6.4575E+00/2.6201E+00**(>)	6.6086E+00/3.1765E+00
f_{d_5}	*0/0(≈)*	*0/0*
f_{d_6}	4.8436E−01/2.2668E−01(<)	**4.6284E−01/8.9812E−02**
f_{d_7}	7.6458E+00/9.3363E+00(<)	**2.7680E−01/7.3006E+00**
f_{d_8}	1.3286E+01/9.5682E+00(<)	**1.2082E+01/9.8736E+00**
f_{d_9}	**1.6729E+02/2.8951E−01**(>)	1.6732E+02/2.9630E−01
$f_{d_{10}}$	1.0026E+02/3.5008E−02(<)	**1.0023E+02/3.0553E−02**
$f_{d_{11}}$	*3.0000E+02/1.3713E−13(≈)*	*3.0000E+02/0*
$f_{d_{12}}$	**1.9675E+02/2.9804E+00**(>)	1.9987E+02/9.7106E+00
>/≈/<	3/3/6	−/−/−

4.5. Time complexity analysis

In this section, time complexity of OLBADE is analyzed by comparing with other powerful DE variants. According to Liang, Qu, Suganthan and Hernández-Díaz (2013), the time complexity of a certain algorithm can be calculated based on T_0, T_1, $\widehat{T}_2$ and $\frac{\widehat{T}_2 - T_1}{T_0}$. T_0 denotes the time spent on basic arithmetic expressions, T_1 denotes the time for 200,000 function evaluations for $f_{a_{14}}$ on 30D optimization and $\widehat{T}_2$ represents the overall cost of optimizing $f_{a_{14}}$, which is calculated by executing 5 times. $\frac{\widehat{T}_2 - T_1}{T_0}$ is used to determine the time complexity. Based on the

Table 16
Comparison between OLBADE-2 and OLBADE.

Variants	OLBADE-2	OLBADE
f_{d_1}	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00*
f_{d_2}	3.7483E+01/1.0173E+01(<)	**1.8591E+01/1.4826E−01**
f_{d_3}	1.0254E−13/3.4143E−14(<)	**2.1395E−14/4.5586E−14**
f_{d_4}	**5.4660E+00/2.2752E+00**(>)	6.6086E+00/3.1765E+00
f_{d_5}	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00*
f_{d_6}	**4.6184E−01/5.6463E−02**(>)	4.6284E−01/8.9812E−02
f_{d_7}	5.2532E+00/5.0570E+00(<)	**2.7680E−01/7.3006E+00**
f_{d_8}	1.5840E+01/7.9380E+00(<)	**1.2082E+01/9.8736E+00**
f_{d_9}	1.8000E+02/2.0055E+00(<)	**1.6732E+02/2.9630E−01**
$f_{d_{10}}$	1.0026E+02/2.8358E−02(<)	**1.0023E+02/3.0553E−02**
$f_{d_{11}}$	3.0196E+02/1.4003E+01(<)	**3.0000E+02/0.0000E+00**
$f_{d_{12}}$	2.3067E+02/8.4603E+00(<)	**1.9987E+02/9.7106E+00**
>/≈/<	2/2/8	−/−/−

Table 17
Comparison between OLBADE-3 and OLBADE.

Variants	OLBADE-3	OLBADE
f_{d_1}	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00*
f_{d_2}	**1.8541E+01/3.7529E−01**(>)	1.8591E+01/1.4826E−01
f_{d_3}	8.0250E−14/5.2316E−14(<)	**2.1395E−14/4.5586E−14**
f_{d_4}	8.9741E+00/3.2727E+00(<)	**6.6086E+00/3.1765E+00**
f_{d_5}	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00*
f_{d_6}	4.7157E−01/6.9117E−02(<)	**4.6284E−01/8.9812E−02**
f_{d_7}	4.4843E+00/7.7967E+00(<)	**2.7680E−01/7.3006E+00**
f_{d_8}	1.4684E+01/9.0981E+00(<)	**1.2082E+01/9.8736E+00**
f_{d_9}	1.7962E+02/3.6328E+00(<)	**1.6732E+02/2.9630E−01**
$f_{d_{10}}$	**1.0022E+02/2.3394E−02**(>)	1.0023E+02/3.0553E−02
$f_{d_{11}}$	*3.0000E+02/0.0000E+00(≈)*	*3.0000E+02/0.0000E+00*
$f_{d_{12}}$	2.2759E+02/1.2576E+01(<)	**1.9987E+02/9.7106E+00**
>/≈/<	2/3/7	−/−/−

Table 18
The comparison of time complexity on benchmark $f_{a_{14}}$.

Algorithms	T_0	T_1	$\hat{T}_2$	$\frac{\hat{T}_2-T_1}{T_0}$
LSHADE-cnEpsin	0.0583	0.6117	3.3821	47.5197
jSO			1.5076	15.3671
ISDE			3.7290	53.4670
PaDE-pet			3.6541	52.1852
TDE			1.7714	19.8919
OLBADE			1.6519	17.8421

experimental results, it can be observed that the OLBADE algorithm possesses a lower time complexity in comparison with ISDE, PaDE-pet, and TDE, implying that the performance enhancements do not require sacrificing increased time complexity (see Table 18).

5. Engineering application

Precise modeling of photovoltaic (PV) modules plays a critical role in the design, simulation, and control of PV systems (Abd El-Mageed et al., 2023). PV models can be constructed based on nonlinear current–voltage (I–V) characteristic curves, which involve numerous unknown parameters. Therefore, accurately determining these parameters is of great importance to ensure the reliability of PV models. The parameter identification of PV model is nonlinear, multimodal and multivariate in nature, making it challenging for traditional methods to address (Yang et al., 2020). To verify the feasibility of OBLADE, we apply it to identify the unknown parameters of triple diode model (TDM), a complicated PV model with nine parameters. The circuit model of TDM is shown in

Table 19
Parameters ranges of TDM.

Parameters	Upper bound	Lower bound
I_{ph} (A)	1	0
(μ_A) I_{sd3}, I_{sd2}, I_{sd1}, I_{sd3}	1	0
$a3$, $a2$, $a1$	2	1
R_s (Ω)	0.5	0
R_{sh} (Ω)	100	0

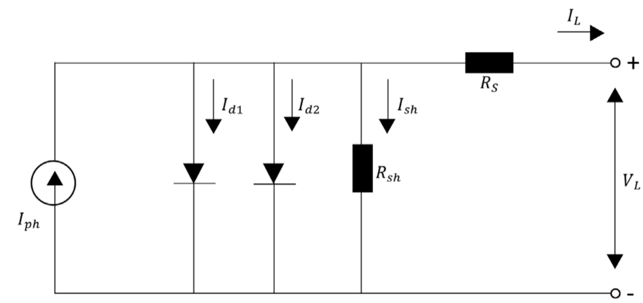


Fig. 7. Circuit model for the TDM.

Fig. 7, whose output current is calculated as follows:

$$I = I_{ph} - I_{o1}[\exp\frac{q(V+R_sI)}{a_1kT} - 1] - I_{o2}[\exp(\frac{q(V+R_sI)}{a_2kT}) - 1] - I_{o3}[\exp(\frac{q(V_L+R_sI)}{a_3kT}) - 1] - \frac{V+R_sI}{R_{sh}} \tag{20}$$

where I denotes the overall output current, which is determined by photocurrent I_{ph}, shunt resistor current I_{sh} and saturation current I_o. V stands for the terminal voltage, R_s represents the series resistance, R_{sh} represents the shunt resistance, a is the ideal diode factor, q represents the charge of an electron which is 1.6×10^{-18} C. k is the symbol for the Boltzmann constant, which has a value of 1.38×10^{-23} J/K. T denotes the temperature in kelvin. There are three ideal diode factors (a_1, a_2 and a_3) and three saturation currents (I_{o1}, I_{o2} and I_{o3}), which enabling TDM to accurately capture the characteristics of PV system. The measured I-V datasets of PV cell are obtained on RTC France solar cell of 57 mm diameter at 33 °C and at an irradiance of 1000 W/m^2. The boundaries of parameters of TDM are shown in Table 19.

The root mean squared error (RMSE) is used to measure the degree of difference between the measured and simulated current, as shown below:

$$RMSE(X) = \sqrt{\frac{1}{L}\sum_{i=1}^{L} f^2(V_i, I_i, X)} \tag{21}$$

Therefore, the parameter identification problem of TDM can be formulated as follows:

$$\begin{cases} f_i(V, I, X_{TDM}) = I_{ph} - I_{o1}[\exp\frac{q(V+IR_s)}{a_1kT} - 1] - \\ \quad I_{o2}[\exp(\frac{q(V+R_sI)}{a_2kT}) - 1] - \\ \quad I_{o3}[\exp(\frac{q(V+IR_s)}{a_3kT}) - 1] - \\ \quad \frac{V+IR_s}{R_{sh}} - I \\ X = \{I_{ph}, I_{o1}, I_{o2}, I_{o3}, R_s, R_{sh}, a_1, a_2, a_3\} \end{cases} \tag{22}$$

To verify the performance of OLBADE in parameter identification problem, we select four recently proposed EAs, including SEDE (Liang et al., 2020), IJAYA (Yu et al., 2017), MLBSA (Yu et al., 2018) and ATLDE (Li et al., 2020), which were specially designed for parameter identification of PV models. All algorithms conducted 30 independent runs to avoid errors, and the minimum RMSE value of 30 runs and

Table 20
The RMSE and parameters of TDM identified by different algorithms.

Algorithm	$I_{ph}(A)$	$I_{sd1}(\mu_A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	$a1$	$I_{sd2}(\mu_A)$	$a2$	$I_{sd3}(\mu_A)$	$a3$	RMSE
Our algorithm	0.76078109	0.74933885	0.03674042	55.48542928	2.00000000	0.22597534	1.45101716	6.64797744	1.82076481	**9.82484851785155E−04**
SEDE	0.76078108	0.22597384	0.03674043	55.48546468	1.45101661	0.19142884	2.00000000	0.55792219	2.00000000	9.82484851787748E−04
IJAYA	0.76068875	0.25362808	0.03652780	56.25756354	1.46115453	0.58429302	2.00000000	0.00117188	1.85723227	9.85756777793189E−04
MLBSA	0.76077737	0.00160179	0.03676652	55.59698426	1.68152905	0.78136611	1.99952788	0.22108617	1.44922356	9.82521011980619E−04
ATLDE	0.76047458	0.27001385	0.03607668	58.78678666	1.47868676	0.00786867	1.61042886	0.16291756	1.71807215	9.82484851788530E−04

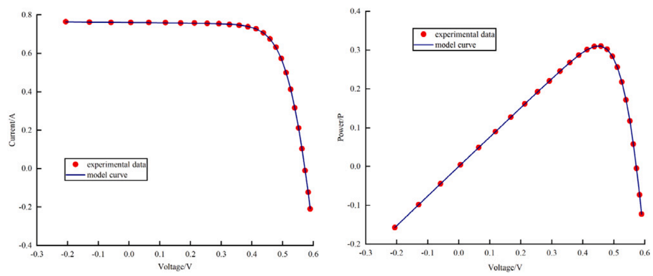


Fig. 8. Comparison between the measured and simulated data for TDM: I-V characteristics and P-V characteristics.

its corresponding parameters of TDM are listed on the Table 20. To confirm the accuracy of parameter identified by OLBADE, the extracted parameters are employed to depict the I–V and P–V characteristics between the measured and simulated data, as shown in Fig. 8. The comparison results demonstrate that OLBADE achieves the best RMSE value on TDM comparing with other algorithms.

From Fig. 8, one can observe a high agreement between the measured and simulated I-V and P-V characteristics. As a result, OLBADE is a promising optimization algorithm for real-world applications.

6. Conclusion

In this paper, an adaptive DE with opposition learning-based diversity enhancement (OLBADE) is proposed to mitigate the problems of inefficient parameter control and stagnation during evolution. By utilizing the novel adaptive parameter control, a better exploration and exploitation balance can be achieved. A donor vector perturbation strategy is proposed to complement existing trial vector generation strategy, thus assisting individuals in escaping from local optima. In addition, a novel stagnation indicator based on fitness information and distribution information is designed to locate stagnant individuals, which are then renewed by opposition learning technique. To avoid the problem of overfitting occurred in single test suite, a large test suite containing CEC2013, CEC2014, CEC2017 and CEC2022 test suites is employed to comprehensively verify the performance of OLBADE. The extensive experiment results demonstrate that OLBADE exhibits highly competitive performance in terms of both convergence speed and optimization accuracy. Moreover, experiments results on parameter identification of TDM verify the feasibility of OLBADE. It is noteworthy that the optimization accuracy of OLBADE remain stable as the dimensionality increases.

Though the multiple strategies and algorithms proposed in this paper have made significant progress compared to previous ones, they still fail to accurately assess the search characteristics of individuals and the evolution state of the population. Therefore, future research should focus on developing more effective evaluation mechanism for search characteristics and evolution state.

CRediT authorship contribution statement

Zhenghao Song: Methodology, Software, Writing – original draft. **Chongle Ren:** Writing – review & editing. **Zhenyu Meng:** Conceptualization, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the Natural Science Foundation of Fujian Province (Grant No. 2021J05227).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.eswa.2023.122942.

References

Abd El-Mageed, A. A., Abohany, A. A., Saad, H. M., & Sallam, K. M. (2023). Parameter extraction of solar photovoltaic models using queuing search optimization and differential evolution. *Applied Soft Computing*, *134*, Article 110032.

Akhila, M., Vidhya, C., & Jeyakumar, G. (2016). Population diversity measurement methods to analyze the behavior of differential evolution algorithm. *International Journal of Control Theory and Applications*, *8*(5), 1709–1717.

Awad, N. H., Ali, M. Z., & Suganthan, P. N. (2017). Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving CEC2017 benchmark problems. In *2017 IEEE congress on evolutionary computation (CEC)* (pp. 372–379). IEEE.

Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press.

Biedrzycki, R., Arabas, J., & Warchulski, E. (2022). A version of NL-SHADE-RSP algorithm with midpoint for CEC 2022 single objective bound constrained problems. In *2022 IEEE congress on evolutionary computation (CEC)* (pp. 1–8). IEEE.

Brest, J., Maučec, M. S., & Bošković, B. (2017). Single objective real-parameter optimization: algorithm jSO. In *2017 IEEE congress on evolutionary computation (CEC)* (pp. 1311–1318). IEEE.

Cheng, C.-Y., Li, S.-F., & Lin, Y.-C. (2019). Self-adaptive parameters in differential evolution based on fitness performance with a perturbation strategy. *Soft Computing*, *23*, 3113–3128.

Deng, W., Xu, J., Song, Y., & Zhao, H. (2021). Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem. *Applied Soft Computing*, *100*, Article 106724.

Dragoi, E.-N., & Dafinescu, V. (2016). Parameter control and hybridization techniques in differential evolution: a survey. *Artificial Intelligence Review*, *45*, 447–470.

Elsayed, S. M., Sarker, R. A., & Ray, T. (2013). Differential evolution with automatic parameter configuration for solving the CEC2013 competition on real-parameter optimization. In *2013 IEEE congress on evolutionary computation* (pp. 1932–1937). IEEE.

Holland, J. H. (1992a). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press.

Holland, J. H. (1992b). Genetic algorithms. *Scientific American*, *267*(1), 66–73.

Li, S., Gong, W., Wang, L., Yan, X., & Hu, C. (2020). A hybrid adaptive teaching–learning-based optimization and differential evolution for parameter identification of photovoltaic models. *Energy Conversion and Management*, *225*, Article 113474.

Li, C., Sun, G., Deng, L., Qiao, L., & Yang, G. (2023). A population state evaluation-based improvement framework for differential evolution. *Information Sciences*, *629*, 15–38.

Liang, J., Qiao, K., Yu, K., Ge, S., Qu, B., Xu, R., & Li, K. (2020). Parameters estimation of solar photovoltaic models via a self-adaptive ensemble-based differential evolution. *Solar Energy*, *207*, 336–346.

Liang, J., Qu, B., & Suganthan, P. (2013). *Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization*: *Computational intelligence laboratory, Zhengzhou University, Zhengzhou China and Technical Report 635*, Singapore: Nanyang Technological University.

Liang, J., Qu, B., Suganthan, P., & Hernández-Díaz, A. G. (2013). *Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization*: *Technical report 201212 (34)*, (pp. 281–295). Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore.

Meng, Z. (2023). Dimension improvements based adaptation of control parameters in Differential Evolution: A fitness-value-independent approach. *Expert Systems with Applications*, *223*, Article 119848.

Meng, Z., & Chen, Y. (2023). Differential Evolution with exponential crossover can be also competitive on numerical optimization. *Applied Soft Computing*, *146*, Article 110750.

Meng, Z., & Pan, J.-S. (2019). HARD-DE: Hierarchical ARchive based mutation strategy with depth information of evolution for the enhancement of Differential Evolution on numerical optimization. *IEEE Access*, *7*, 12832–12854.

Meng, Z., Pan, J.-S., & Tseng, K.-K. (2019). PaDE: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization. *Knowledge-Based Systems*, *168*, 80–99.

Meng, Z., Pan, J.-S., & Xu, H. (2016). QUasi-Affine TRansformation evolutionary (QUATRE) algorithm: a cooperative swarm based algorithm for global optimization. *Knowledge-Based Systems*, *109*, 104–121.

Meng, Z., Song, Z., Shao, X., Zhang, J., & Xu, H. (2023). FD-DE: Differential Evolution with fitness deviation based adaptation in parameter control. *ISA Transactions*.

Meng, Z., & Yang, C. (2022). Two-stage differential evolution with novel parameter control. *Information Sciences*, *596*, 321–342.

Meng, Z., & Zhang, J. (2023). QUATRE-EMS: QUATRE algorithm with novel adaptation of evolution matrix and selection operation for numerical optimization. *Information Sciences*, *651*, Article 119714.

Meng, Z., Zhong, Y., Mao, G., & Liang, Y. (2022). PSO-sono: A novel PSO variant for single-objective numerical optimization. *Information Sciences*, *586*, 176–191.

Okagbue, H. I., Adamu, M. O., & Anake, T. A. (2019). Differential evolution in wireless communications: A review. *iJOE*, *15*(11), 29–51.

Osuna-Enciso, V., Cuevas, E., & Castañeda, B. M. (2022). A diversity metric for population-based metaheuristic algorithms. *Information Sciences*, *586*, 192–208.

Qin, A. K., Huang, V. L., & Suganthan, P. N. (2008). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, *13*(2), 398–417.

Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. (2008). Opposition versus randomness in soft computing techniques. *Applied Soft Computing*, *8*(2), 906–918.

Son, N. N., Van Kien, C., & Anh, H. P. (2021). Hysteresis compensation and adaptive control based evolutionary neural networks for piezoelectric actuator. *International Journal of Intelligent Systems*, *36*(10), 5472–5492.

Song, Z., & Meng, Z. (2023). Differential Evolution with wavelet basis function based parameter control and dimensional interchange for diversity enhancement. *Applied Soft Computing*, Article 110492.

Stanovov, V., Akhmedova, S., & Semenkin, E. (2021). Biased parameter adaptation in differential evolution. *Information Sciences*, *566*, 215–238.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359.

Tanabe, R., & Fukunaga, A. (2013). Success-history based parameter adaptation for differential evolution. In *2013 IEEE congress on evolutionary computation* (pp. 71–78). IEEE.

Tian, M., & Gao, X. (2019). An improved differential evolution with information intercrossing and sharing mechanism for numerical optimization. *Swarm and Evolutionary Computation*, *50*, Article 100341.

Viktorin, A., Senkerik, R., Pluhacek, M., Kadavy, T., & Zamuda, A. (2019). Distance based parameter adaptation for success-history based differential evolution. *Swarm and Evolutionary Computation*, *50*, Article 100462.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82.

Wu, G., Mallipeddi, R., & Suganthan, P. N. (2017). *Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization*: *Technical Report*, Singapore: National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University.

Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver Press.

Yang, M., Li, C., Cai, Z., & Guan, J. (2014). Differential evolution with auto-enhanced population diversity. *IEEE Transactions on Cybernetics*, *45*(2), 302–315.

Yang, B., Wang, J., Zhang, X., Yu, T., Yao, W., Shu, H., Zeng, F., & Sun, L. (2020). Comprehensive overview of meta-heuristic algorithm applications on PV cell parameter identification. *Energy Conversion and Management*, *208*, Article 112595.

Yu, K., Liang, J., Qu, B., Chen, X., & Wang, H. (2017). Parameters identification of photovoltaic models using an improved JAYA optimization algorithm. *Energy Conversion and Management*, *150*, 742–753.

Yu, K., Liang, J., Qu, B., Cheng, Z., & Wang, H. (2018). Multiple learning backtracking search algorithm for estimating parameters of photovoltaic models. *Applied Energy*, *226*, 408–422.

Zeng, Z., & Zhang, H. (2022). An evolutionary-state-based selection strategy for enhancing differential evolution algorithm. *Information Sciences*, *617*, 373–394.

Zeng, Z., Zhang, M., Zhang, H., & Hong, Z. (2022). Improved differential evolution algorithm based on the sawtooth-linear population size adaptive method. *Information Sciences*, *608*, 1045–1071.

Zhang, Y., Gong, D.-w., Gao, X.-z., Tian, T., & Sun, X.-y. (2020). Binary differential evolution with self-learning for multi-objective feature selection. *Information Sciences*, *507*, 67–85.

Zhang, J., & Sanderson, A. C. (2009). JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, *13*(5), 945–958.

Zhang, Q., Zou, D., Duan, N., & Shen, X. (2019). An adaptive differential evolutionary algorithm incorporating multiple mutation strategies for the economic load dispatch problem. *Applied Soft Computing*, *78*, 641–669.