Knowledge-Based Systems 168 (2019) 80–99

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

PaDE: An enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization

Zhenyu Meng [a,*], Jeng-Shyang Pan [a], Kuo-Kun Tseng [b]

[a] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, China
[b] Department of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

ARTICLE INFO

Article history:
Received 15 July 2018
Received in revised form 31 December 2018
Accepted 3 January 2019
Available online 11 January 2019

Keywords:
Control parameters
Differential evolution
Numerical optimization
Population size reduction

ABSTRACT

Differential Evolution (DE) variants have been proven to be excellent algorithms in tackling real-parameter single objective numerical optimization because they have secured the front ranks of these competitions for many years. Nevertheless, there are still some weaknesses, e.g. (1) improper control parameter adaptation schemes; and (2) defect in a given mutation strategy., existing in some state-of-the-art DE variants, which may result in slow convergence and worse optimization performance. Therefore, in this paper, a novel Parameter adaptive DE (PaDE) is proposed to tackle the above mentioned weaknesses and the PaDE algorithm has three advantages: (1) A grouping strategy with novel adaptation scheme for *Cr* is proposed to tackle the improper adaptation schemes of *Cr* in some state-of-the-art DE variants; (2) A novel parabolic population size reduction scheme is proposed to tackle the weakness in linear population size reduction scheme; (3) An enhanced time stamp based mutation strategy is proposed to tackle the weakness in a former mutation strategy. The novel PaDE algorithm is verified under 58 benchmarks from two Congress on Evolutionary Computation (CEC) Competition test suites on real-parameter single objective numerical optimization, and experiment results show that the proposed PaDE algorithm is competitive with the other state-of-the-art DE variants.

1. Introduction

There are many optimization demands for tough optimization problems in our daily lives nowadays, and the common approach to tackle such a complex optimization problem usually begins by designing the objectives that can model the problem or the task, sometimes it may also incorporate some constraints [1]. Generally, these objectives depend on certain characteristic of the problem or the task, called parameter or variable, and our goal is to find the values of these parameters that can optimize these objectives [2,3]. Single objective optimization is the foundation of a more complex multi-objective optimization problems, and research on single objective optimization will definitely influence the development of these more complex optimization problems [4]. For a single objective $f = f(X)$, the real-parameter single objective minimizing optimization can be formally represented as finding a proper solution X^* of the following set:

$$\Omega^* \equiv \arg\min_{X \in \Omega} f(X) = \{X^* \in \Omega : f(X^*) \leq f(X), \forall X \in \Omega\} \quad (1)$$

where X denotes a D-dimensional vector of parameters while $\Omega \subseteq \mathbb{R}^D$ denotes the whole solution space [5]. Usually, a search domain of vector X is restricted by a lower bound $X_{min} = (x_{min,1}, x_{min,2}, \ldots, x_{min,D})$ and an upper bound $X_{max} = (x_{max,1}, x_{max,2}, \ldots, x_{max,D})$ of each parameters, therefore, the parameters of the vector $X = (x_1, x_2, \ldots, x_j, \ldots, x_D)$ should always satisfy $x_{min,j} \leq x_j \leq x_{max,j}, j \in \{1, 2, \ldots, D\}$ during the whole search, and this kind of optimization can be further named as bound constrained optimization. Differential Evolution (DE) is an excellent population-based stochastic optimization branch tackling such optimization problems, and the canonical DE algorithm [6] obtained excellent ranks in the first and second international contest on evolution computation after its inception [7] in 1995. Furthermore, some newly proposed DE variants, e.g. JADE [8], SHADE [9], LSHADE [10], iLSHADE [11] and jSO [12], also secured front ranks in recent competitions.

The canonical DE algorithm originated with Genetic Annealing Algorithm [7] which hybridized Genetic Algorithm (GA) [13] and Simulated Annealing (SA) [14]. Therefore, operations such as selection, mutation and crossover invented in GA were directly inherited into DE though the sequences of these operations were different [15,16]. Accordingly, there are three control parameters involved in the three operations, *ps* denotes the population size which defines the number of selection operations in each generation, *F* denotes the scale factor which restricts the differential mutation operation, and *Cr* denotes the crossover rate which determines how many parameters in the target vector are changed during the crossover operation [17]. The recommended values of

* Corresponding author at: Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, China.
E-mail address: mzy1314@gmail.com (Z. Meng).

https://doi.org/10.1016/j.knosys.2019.01.006
0950-7051/

the three control parameters by DE inventors Storn and Price [7] are given as follows: $ps \in [5D, 10D]$, D is the dimension number, usually $ps = 100$ is a good initial value; $F \in [0.4, 1]$, and $F = 0.5$ is a good initial value; $Cr \in [0, 1]$, and $Cr = 0.1$ is a good initial value for unimodal separable functions while $Cr = 0.9$ is a good initial value for multi-modal and nonseparable functions. Nevertheless, there are also some other recommendations for the control parameters mentioned in the literature [7,18,19]. Due to many claims and counter claims concerning the rules for the choice of proper control parameters, DE variants with adaptive control parameters became much more popular both for scientific researchers and for engineers [7,17,20–22]. Usually, these adaptive parameter mechanisms can be divided into three categories: the first category is called "Deterministic parameter control", which means that parameters are renewed according to a certain deterministic rule, the time dependent mutation rate r in [23], the function evaluation dependent inertia weight iw in [24] and the top super individual rate p in [11,12] etc., can be classified into this category; the second category is called "adaptive parameter control", which means some feedbacks from the evolution process are employed in the guidance of parameters adaptation, the adaptive scheme of Cr in [21] and the adaptive schemes of F and Cr in [8–12] etc., can be classified into this category; the third category is called "Self-adaptive parameter control", which means evolution of the evolution, parameters are encoded into each individual and they also undergo the operations of evolution, the self-adaptive mechanisms of F and Cr in [20,25–27] and the self-adaptive mechanism of population size ps in [28] etc., can be classified into this category.

Beyond the control parameters, there were several different mutation strategies, e.g. DE/rand/1/bin, DE/best/1/bin, DE/target-to-best/1/bin, DE/rand-to-best/1/bin, DE/rand/2/bin, DE/best/2/bin and DE/target-to-pbest/1/bin etc., introduced in the literature [1,5,7,21,29–31], which also played very important roles in the overall optimization performance. A general convention DE/x/y/z [7] is used to differ these mutation strategies, x denotes the base of the mutant vector, y denotes the number of difference pairs, z denotes the scheme of crossover operations. Usually, DE/rand/1/bin has a good exploration characteristic while DE/best/1/bin has a good exploitation characteristic, DE/target-to-pbest/1/bin makes a balance between the exploration of DE/rand/1/bin and exploitation of DE/best/1/bin. Empirically, powerful DE variant was a combination of both a specific mutation strategy and the associated control parameter adaptation scheme. For example, Zhang and Sanderson introduced novel control parameter adaptation schemes which employed Lehmer mean and arithmetic mean of the current success F and Cr in the production of next generation control parameters in JADE algorithm [8]. Moreover, a novel mutation strategy DE/target-to-pbest/1/bin was also proposed in JADE, and all of these helped JADE win the competition on scale-invariant optimization at WCCI2008. Tanabe and Fukunaga proposed a success history based control parameter pool to enhance the diversity of control parameters in SHADE algorithm [10] meanwhile a novel weighted Lehmer mean and arithmetic mean based adaptation schemes for F and Cr as well as a dynamic linear population size reduction scheme were also incorporated into this algorithm. All these adaptations of parameters with the powerful mutation strategy DE/target-to-pbest/1/bin helped LSHADE win the competition on real-parameter single objective optimization at CEC2014.

There were also other approaches mentioned in the literature enhancing the overall performance of the canonical DE algorithm. Rahnamayan et al. proposed an opposition based learning DE algorithm [32], opposition vector was employed in the diversity enhancement both in the initialization stage and some generations of the evolution stage. The implementation of this DE variant was simple and easy coding, but the overall performance was not good enough especially on some unimodal separable function and some complex multi-modal functions because of the weaknesses in DE/rand/1/bin mutation strategy with fixed control parameters. Das et al. proposed a neighbourhood based DE/target-to-best/1/bin mutation operator to make a balance between exploration and exploitation abilities of DE [33], and it obtained better performance on the tested benchmarks, however, the premature convergence of DE/target-to-best/1/bin still existed in this DE variant. Wang et al. proposed a DE variant [22] with composite trial vectors and control parameters, three control parameter pairs and three trial vector generation strategies were combined randomly in this DE variant to tackle optimization problems. As this control parameter settings were based on some previous knowledge or experience of CEC2005 benchmarks, these settings may be not well fitted for other benchmarks according to the No Free Lunch Theorem [34]. Gong and Cai proposed a ranking-based mutation structure [35] to enhance several former proposed DE variants, such as jDE [20], SaDE [21] and CoDE [22], etc. In this structure, some of the parents in the mutation operator were proportionally selected according to their ranks in the population, and better performances were obtained by DE variants incorporating such a structure than the corresponding original ones, nevertheless, the inborn weaknesses of these DE variants were still left unsolved. Pan et al. proposed a QUATRE structure in [15,16,36,37] to tackle the high dimensional inborn representative bias of DE, this structure achieved an overall better optimization performance, however, the parameter control mechanisms in these QUATRE algorithms still need to be further polished. By examining the mis-interaction between control parameters of JADE [8], SHADE [9] and LSHADE [10], Meng et al. proposed a parameter with adaptive learning mechanism [5] for the enhancement of the canonical DE algorithm. This new DE variant tackled the mis-interaction between control parameter F and Cr by separating these parameters into different groups and only one variable was allowed in each group. Moreover, each of the control parameter was updated independently, and a time stamp based mutation strategy was also introduced into the new variant to avoid inferior solutions being archive residents during the whole evolution. However, the update scheme of Cr in this DE variant was heavily dependent on the number of individuals in each group, which might fall into a bad adaptation of Cr when population size became relative small in a linear population size reduction scheme.

Besides the above mentioned weaknesses of these DE variants, we found that the linear population size reduction scheme and the time stamp based mutation strategy still have some weaknesses. Therefore, we summarize all these weaknesses into two aspects: (1) improper control parameter adaptation schemes; and (2) defect in a given mutation strategy., and then we proposed the novel PaDE algorithm in the paper. The main contributions of the paper are listed as follows:

1. A grouping strategy with a novel adaptation scheme for Cr is proposed to tackle the improper adaptation schemes of Cr in some state-of-the-art DE variants.
2. A novel parabolic population size reduction scheme is proposed to tackle the weakness in linear population size reduction scheme as we see that the quick reduction of population size at the beginning of the evolution leads to a bad perception of the landscape of the objective functions.
3. An enhanced mutation strategy with time stamp scheme is proposed both to avoid too old and harmful inferior solutions being archive-residents (in comparison with mutation strategy in LSHADE, iLSHADE and jSO) and to enhance the diversity of external individuals (in comparison with mutation strategy in LPLAMDE).
4. An initiative to use more benchmarks or more than one Congress on Evolutionary Computation (CEC) Competition test suites for algorithm evaluation is presented as some state-of-the-art DE variants may be over-tuned for a number of benchmarks or a certain CEC Competition test suite.

The rest of the paper is organized as follows. The related work presenting reviews of several powerful DE variants is given in Sections 2 and 3 illustrates the new proposed PaDE algorithm in detail. In Section 4, a deep analysis of the novel PaDE algorithm under CEC2013 and CEC2017 test suites for real-parameter single-objective numerical optimization is presented. Finally, conclusion is given in Section 5.

2. Related works

By following some recently proposed powerful DE variants, e.g. SHADE [9], LSHADE [10], iLSHADE [11] and jSO [12], we can see that they all originated with JADE algorithm [8]. Moreover, the new proposed algorithm in this paper was also an extension of JADE algorithm, therefore, in this part, a brief review of JADE, SHADE, LSHADE, iLSHADE and jSO was conducted first to get a better understanding of the development in the DE branch of evolutionary algorithms.

2.1. JADE

A new external archive based mutation strategy DE/target-to-pbest/1/bin was proposed in JADE algorithm [8] with its equation shown in Eq. (2):

$$V_{i,G} = X_{i,G} + F \cdot (X^{p}_{best,G} - X_{i,G}) + F \cdot (X_{r_1,G} - \widetilde{X}_{r_2,G}) \tag{2}$$

$X_{i,G}$ denotes the target vector, $V_{i,G}$ denotes the mutant/donor vector, $X^{p}_{best,G}$ denotes a certain vector selected from the top 100p% individuals of the population, p denotes the percentage of top superior individuals. As we know, the mutation strategies DE/rand/1/bin and DE/target-to-rand/1/bin usually have good exploration capacity and converge slowly in many optimizations while some greedy strategies such as DE/best/1/bin and DE/target-to-best/1/bin have good exploitation capacity and usually converge prematurely [18]. The new proposed mutation strategy in JADE made a good balance between the two characteristics by introducing a handful top superior individuals in the mutation strategy. Moreover, an optional external archive was also maintained to diversify donor vectors, a symbol A was used for denoting the external archive which recorded infer solutions during the evolution, and symbol P denoted the current population of individuals. $X_{r_1,G}$ in Eq. (2) is a randomly selected vector from P while $\widetilde{X}_{r_2,G}$ is a randomly selected vector from the union $P \cup A$, and the indices of i, r_1 and r_2 always satisfy $i \neq r_1 \neq r_2$. The external archive A is firstly initialized empty, $A = \varnothing$, and then inferior solutions are gradually inserted into it during evolution before the archive is full. When A is full, the same number of inferior solutions are firstly discarded from A, and then new ones are allowed to be inserted into it.

Besides the mutation strategy, a well designed parameter adaptation scheme can also enhance the robustness of a certain DE variant [8]. The JADE algorithm also introduced well-designed adaptation schemes of control parameters, Eqs. (3) and (4) present the adaptation schemes of control parameter F and Cr respectively:

$$\begin{cases} \mu_F = (1-c)\cdot \mu_F + c \cdot mean_L(S_F) \\ mean_L(S_F) = \frac{\sum_{F\in S_F} F^2}{\sum_{F\in S_F} F} \end{cases} \tag{3}$$

$$\begin{cases} \mu_{Cr} = (1-c)\cdot \mu_{Cr} + c \cdot mean_A(S_{Cr}) \\ mean_A(S_{Cr}) = \dfrac{\sum_{Cr\in S_{Cr}} Cr}{|S_{Cr}|} \end{cases} \tag{4}$$

Control parameter F in JADE obeys Cauchy distribution, $F \sim C(\mu_F, 0.1)$, μ_F is the location parameter. Control parameter Cr obeys Normal distribution, $Cr \sim N(\mu_{Cr}, 0.1)$, μ_{Cr} denotes the mean value. The scale parameter of F and standard deviation of Cr are all set to a constant value 0.1. $mean_L(S_F)$ and $mean_A(S_{Cr})$ denote the Lehmer mean of set S_F and arithmetic mean of set S_{Cr} respectively. Accordingly, S_F and S_{Cr} denote the success sets of control parameters F and Cr, and "success" means that a certain individual employing these control parameters produces an offspring with better fitness value. $|S_{Cr}|$ denotes the size of set S_{Cr}, parameter c is used for balancing the old μ_F (or μ_{Cr}) and the corresponding success mean in the update scheme. Usually, c is recommended to be set a value that satisfies $1/c \in [5, 20]$, and $c = 0.1$ is the default value in JADE algorithm.

2.2. SHADE

SHADE algorithm [9] is an enhanced JADE algorithm by incorporating both a diversity entry pool of control parameters and fitness value based weighted mean proposed by Peng et al. [38] in parameter adaptation scheme. There are H entries in the pool and each entry records a μ_F and μ_{Cr} pair within it. All the control parameter pairs in the H entries are assigned equal values, $\mu_F = \mu_{Cr} = 0.5$, in the initialization stage. The control parameters that a certain individual employed in each generation are randomly selected from the H entries. Moreover, only one entry is updated in each generation, the update sequence of H entries are from the 1st to the last and then back to the 1st as a circle. As we all know that algorithms in DE branch are all population-based stochastic optimization algorithms, due to the probabilistic nature, parameters adaptation scheme may lead to some undesirable values which may degrade the optimization performance. Therefore, the success history based H-entry pool employed in SHADE maintains a good diversity of better control parameters and turn down the risks of potential bad values.

For the fitness value based renewing schemes of control parameters μ_F and μ_{Cr}, they are presented in Eqs. (5) and (6) respectively:

$$\begin{cases} w_k = \dfrac{\Delta f_k}{\sum_{k=1}^{|S_F|} \Delta f_k} \\ \Delta f_j = f(X_{j,G}) - f(U_{j,G}) \\ mean_{WL}(S_F) = \dfrac{\sum_{k=1}^{|S_F|} w_k \cdot S_{F,k}^2}{\sum_{k=1}^{|S_F|} w_k \cdot S_{F,k}} \\ \mu_{F,G+1} = \begin{cases} mean_{WL}(S_F), & if\ S_F \neq \varnothing \\ \mu_{F,G}, & otherwise \end{cases} \end{cases} \tag{5}$$

$$\begin{cases} w_k = \dfrac{\Delta f_k}{\sum_{k=1}^{|S_{Cr}|} \Delta f_k} \\ \Delta f_j = f(X_{j,G}) - f(U_{j,G}) \\ mean_{WA}(S_{Cr}) = \sum_{k=1}^{|S_{Cr}|} w_k \cdot S_{Cr,k} \\ \mu_{Cr,k,G+1} = \begin{cases} mean_{WA}(S_{Cr}), & if\ S_{Cr} \neq \varnothing \\ \mu_{Cr,k,G}, & otherwise \end{cases} \end{cases} \tag{6}$$

where $|S_F|$ denotes the size of set S_F, $|S_{Cr}|$ denotes the size of set S_{Cr}. Both S_F and the S_{Cr} are the corresponding parameter set of S which denotes the success set of individuals. Δf_k denotes the fitness difference of the kth individual in S, and the kth individual in S is also with a jth index of the whole population in the Gth generation. We can see from the above equations that when there are better trial vectors produced, both μ_F and μ_{Cr} are renewed by the corresponding weighted means calculated in Eqs. (5) and (6) respectively, otherwise, μ_F and μ_{Cr} keep the same as the ones in the previous generation.

2.3. LSHADE

LSHADE algorithm [10] improved the overall optimization performance of SHADE by introducing both a linear population size reduction mechanism and modifications of parameters, e.g. H, $\boldsymbol{A}$ and Cr. All these changes helped LSHADE obtain the first place in competition of CEC2014. For the population size reduction mechanism, Eq. (7) presents the detailed change of ps_G during evolution:

$$ps_G = \begin{cases} round[\frac{ps_{min} - ps_{ini}}{nfe_{\max}} \cdot nfe + ps_{ini}], & if \ G > 1 \\ ps_{ini}, & otherwise \end{cases} \tag{7}$$

where ps_{ini} denotes the initial population size, ps_{min} denotes the minimum population size, $nfe_{\max}$ denotes the maximum number of function evaluations, and nfe denotes the current number of function evaluations, $round[\cdot]$ means "round to the nearest integer". For the modifications of parameters H, $\boldsymbol{A}$ and Cr, a much smaller value of H is used in LSHADE, $H = 6$; The size of external archive $\boldsymbol{A}$ is changed adaptively according to population size ps, $|\boldsymbol{A}| = r^{arc} \cdot ps_G$, r^{arc} denotes a constant ratio of the external archive size to population size; Control parameter Cr is renewed according to Eq. (8):

$$\begin{cases} w_k = \dfrac{\Delta f_k}{\sum_{k=1}^{|S_{Cr}|} \Delta f_k} \\ \Delta f_j = f(X_{j,G}) - f(U_{j,G}) \\ mean_{WL}(S_{Cr}) = \dfrac{\sum_{k=1}^{|S_{Cr}|} w_k \cdot S_{Cr,k}^2}{\sum_{k=1}^{|S_{Cr}|} w_k \cdot S_{Cr,k}} \\ \mu_{Cr,k,G+1} = \begin{cases} mean_{WL}(S_{Cr}), & if \ S_{Cr} \neq \varnothing \\ \mu_{Cr,k,G}, & otherwise \end{cases} \\ Cr_i = \begin{cases} 0, & if \ \mu_{Cr,r_i} = 0; \\ randn_i(\mu_{Cr,r_i}, 0.1) & otherwise \end{cases} \end{cases} \tag{8}$$

where r_i is a random index in $[1, H]$. The change is used for locking Cr_i to 0, which means forcing the ith individual searching toward one coordinate direction a time, and this scheme usually performs better especially on some multi-modal functions.

2.4. iLSHADE

The iLSHADE algorithm [11] can be considered as an extension of LSHADE with five small modifications, which are beneficial to tackle multi-modal optimization problems. These five modifications are listed as follows. First, a larger value of μ_{Cr} and smaller value of ps, $\mu_{Cr} = 0.8$, $ps = 12 \cdot D$ instead of $\mu_{Cr} = 0.5$, $ps = 18 \cdot D$ in LSHADE, are used in the initial stage; Second, the last entry of the H-entry pool in iLSHADE is set as a constant entry, $\mu_F = \mu_{Cr} = 0.9$, during the whole evolution. Third, if a terminal value or a negative value of μ_{Cr} is selected from a certain entry of the H-entry pool, then the corresponding Cr value is set to zero; Fourth, the adaptive parameter control mechanisms of μ_F and μ_{Cr} are changed to the following rules shown in Eqs. (9), (10) and (11) respectively.

$$\begin{cases} w_k = \dfrac{\Delta f_k}{\sum_{k=1}^{|S_F|} \Delta f_k} \\ \Delta f_j = f(X_{j,G}) - f(U_{j,G}) \\ mean_{WL}(S_F) = \dfrac{\sum_{k=1}^{|S_F|} w_k \cdot S_{F,k}^2}{\sum_{k=1}^{|S_F|} w_k \cdot S_{F,k}} \\ \mu_{F,k,G+1} = \begin{cases} (mean_{WL}(S_F) + \mu_{F,k,G+1})/2, & if \ S_F \neq \varnothing \\ \mu_{F,k,G}, & otherwise \end{cases} \end{cases} \tag{9}$$

$$\begin{cases} w_k = \dfrac{\Delta f_k}{\sum_{k=1}^{|S_{Cr}|} \Delta f_k} \\ \Delta f_j = f(X_{j,G}) - f(U_{j,G}) \\ mean_{WL}(S_{Cr}) = \dfrac{\sum_{k=1}^{|S_{Cr}|} w_k \cdot S_{Cr,k}^2}{\sum_{k=1}^{|S_{Cr}|} w_k \cdot S_{Cr,k}} \\ \mu_{Cr,k,G+1} = \begin{cases} (mean_{WL}(S_{Cr}) + \mu_{Cr,k,G+1})/2, & if \ S_{Cr} \neq \varnothing \\ \mu_{Cr,k,G}, & otherwise \end{cases} \end{cases} \tag{10}$$

$$Cr_i = \begin{cases} 0, & if \ \mu_{Cr,r_i} = 0; \\ randn_i(\mu_{Cr,r_i}, 0.1) & otherwise \end{cases} \tag{11}$$

Moreover, the generated control parameters Cr and F of each individual of the population are also conducted a readjustment according to Eqs. (12) and (13) respectively:

$$F_{i,G} = \begin{cases} \min(F_{i,G}, 0.7), & if \ nfe < 0.25 \cdot nfe_{\max} \\ \min(F_{i,G}, 0.8), & if \ nfe < 0.5 \cdot nfe_{\max} \\ \min(F_{i,G}, 0.9), & if \ nfe < 0.75 \cdot nfe_{\max} \end{cases} \tag{12}$$

$$Cr_{i,G} = \begin{cases} \max(Cr_{i,G}, 0.5), & if \ nfe < 0.25 \cdot nfe_{\max} \\ \max(Cr_{i,G}, 0.25), & if \ nfe < 0.5 \cdot nfe_{\max} \end{cases} \tag{13}$$

where nfe denotes the current number of function evaluations while $nfe_{\max}$ denotes the maximum number of function evaluations allowed during the optimization. The last but also very important one, the parameter p that specifies 100p% top superior individuals is changed dynamically during the whole evolution, and this is totally different from JADE, SHADE and LSHADE employing a constant p value. The renewing scheme of p in iLSHADE is given in Eq. (14):

$$p = \frac{p_{\max} - p_{\min}}{nfe_{\max}} \cdot nfe + p_{\min} \tag{14}$$

where $p_{\max}$ is the initial value of parameter p, $p_{\max} = 0.2$, and $p_{\min}$ is the terminal value of parameter p, $p_{\min} = 0.1$.

2.5. jSO

The jSO algorithm [12] can be considered as an improved algorithm of iLSHADE by incorporating a new inertia weight into the mutation strategy. Eq. (15) presents the detailed equation of the mutation strategy:

$$V_{i,G} = X_{i,G} + F_w \cdot F \cdot (X_{best,G}^p - X_{i,G}) + F \cdot (X_{r_1,G} - \widetilde{X}_{r_2,G}) \tag{15}$$

where F_w is the new incorporated inertial weight and it satisfies:

$$Fw = \begin{cases} 0.7, & if \ nfe < 0.2 \cdot nfe_{\max} \\ 0.8, & if \ 0.2 \leq nfe < 0.4 \cdot nfe_{\max} \\ 1.2. & otherwise \end{cases} \tag{16}$$

Other symbols in Eq. (15) has the same meanings as the ones in Eq. (2). nfe and $nfe_{\max}$ in Eq. (16) have the same meanings as Eq. (12). Furthermore, the generated control parameters F and Cr of each individual in the population are also conducted a readjustment according to Eqs. (17) and (18) respectively.

$$F = \begin{cases} F, & if \ F < 0.7 \& nfe < 0.6 nfe_{\max} \\ 0.7 & otherwise \end{cases} \tag{17}$$

Index	1	2	$\cdots$	j	$\cdots$	k
μ_{Cr}	μ_{Cr_1}	μ_{Cr_2}	$\cdots$	μ_{Cr_j}	$\cdots$	μ_{Cr_k}
$P(\bullet)$	$P(1)$	$P(2)$	$\cdots$	$P(j)$	$\cdots$	$P(k)$

Fig. 1. Grouping strategy of the PaDE algorithm.

$$Cr = \begin{cases} 0.7. \; if \;\; Cr < 0.7\&nfe < 0.25nfe_{\max} \\ 0.6, \;\; if \;\; Cr < 0.6\&0.25nfe_{\max} \le nfe < 0.5nfe_{\max} \\ Cr, \;\; otherwise \end{cases} \tag{18}$$

Besides the differences in control parameters and mutation strategy, there is an extra difference lying in the initialization stage. The initial population size ps, the initial control parameter μ_F, $p_{\max}$ and $p_{\min}$ values that specify the ratio of top superior individuals are all different from iLSHADE, and the recommended values of them are listed as follows: $ps = 25log(D)\sqrt{D}$, $\mu_F = 0.3$, $p_{\max} = 0.25$ and $p_{\min} = p_{\max}/2$.

To summarize, the exact relationship of these DE variants can be described as follows: SHADE was produced from JADE by incorporating both a diversity entry pool of control parameters and fitness value based parameter adaptation scheme. LSHADE was produced from SHADE by incorporating a linear population size reduction scheme as well as some slight modifications in crossover rate Cr and the external archive $\boldsymbol{A}$. iLSHADE was produced from LSHADE by changing parameter adaptation scheme and some assignments of the control parameters. jSO was produced from iLSHADE by incorporating a inertia weight into the former mutation strategy as well as some modifications of the control parameters.

3. The proposed PaDE algorithm

In this part, the description of the novel PaDE algorithm is separated into three parts: the first part presents the novel grouping strategy with adaptation schemes for F and Cr; the second part presents the novel parabolic population size reduction scheme; the last part presents the enhanced mutation strategy with time stamp mechanism.

3.1. Grouping strategy with adaptation schemes for F and Cr

In PaDE algorithm, there are k groups maintained during the whole evolution and all individuals are classified into the k groups for evolution. Fig. 1 presents the grouping strategy in PaDE algorithm, and we can see that each group is associated with two unique parameters, control parameters μ_{Cr} and selection probability $P(\cdot)$. Control parameter μ_{Cr} is the mean value of a Normal distribution, $N(\mu_{Cr}, 0.1)$, that all control parameter Cr of each individual in the group should obey. The selection probability $P(\cdot)$ denotes the probability that a certain individual in the population is classified into the group. The initial values of them are $\mu_{Cr_1} = \mu_{Cr_2} = \cdots = \mu_{Cr_j} = \cdots = \mu_{Cr_k} = 0.6$, $P(1) = P(2) = \cdots = P(j) = \cdots = P(k) = \frac{1}{k}$. In the initialization stage, all the individuals are separated into these k groups by employing the stochastic universal selection [5,39] presented in Algorithm 1.

Control parameter F for each individual in PaDE algorithm obeys Cauchy distribution with the location parameters equalling to μ_F and scale parameter equalling to μ_F, $F \sim C(\mu_F, \sigma_F)$, and the initial value of μ_F equals to 0.8. Unlike control parameter F, control parameter Cr for each individual in the same group obeys the same

Algorithm 1: Pseudo code of stochastic universal selection

Input: Selection probability $P(j)$ of the j^{th}, $j \in \{1, 2, \ldots, k\}$ group and population size ps;
Output: The group indices that the ps individuals are classified into;
Initialization $space = \frac{1}{ps}$, $rnd = rand()$,
$rndn = (rnd : 1 : ps) \times space$, $sumP = 0$, $label = zeros(ps, 1)$,
$index = zeros(ps, 1)$;
for $j = 1; j \le k; j++$ **do**
  $sumP = sumP + P(j)$;
  $nlabel$ records binary inverted $label$ value.
  $label = nlabel\&(rndn < sumP)'$;
  $index = index + label \times j$;
  $label = label|index$;
$index = index(randperm(ps))$;
return $index$;

Normal distribution, e.g. control parameter Cr in the jth group obeys $Cr \sim N(\mu_{Cr_j}, \sigma_{Cr})$. Both σ_F and σ_{Cr} are set constant values, $\sigma_F = \sigma_{Cr} = 0.1$, during the whole evolution. For the generated control parameter F, a new value of F should be regenerated while $F \le 0$, and if the generated F is greater than 1, then it is truncated to $F = 1$. For the control parameter Cr, it is generated according to Eq. (19) during evolution and if the value of Cr is out of the range [0,1], then it is truncated to 0 or 1, the closest one to the generated value.

$$Cr_i = \begin{cases} randn_i(\mu_{Cr_j}, 0.1). & if \;\; \mu_{Cr_j} > 0 \\ 0. & otherwise \end{cases} \tag{19}$$

After initialization, the trial vector of each individual can be generated according to the corresponding control parameters and mutation strategy. When a better solution is found by the trial vector, a success sign 's' is labelled on the individual, otherwise a failure sign 'f' is labelled. Then the control parameter μ_F can be renewed according to the following Eq. (20):

$$\begin{cases} w_s = \dfrac{\Delta f_s}{\sum_{s=1}^{|S_F|} \Delta f_s} \\ \Delta f_i = f(X_{i,G}) - f(U_{i,G}) \\ mean_{WL}(S_F) = \dfrac{\sum_{s=1}^{|S_F|} w_s \cdot S_F^2(s)}{\sum_{s=1}^{|S_F|} w_s \cdot S_F(s)} \\ \mu_{F,G+1} = \begin{cases} mean_{WL}(S_F), \;\; if \;\; S_F \ne \varnothing \\ \mu_{F,G}, \;\; otherwise \end{cases} \end{cases} \tag{20}$$

where s denotes the index of 's' individuals in the success individual set and i is the corresponding index of the individual in the population; Δf denotes the fitness error, and S_F denotes the set of control parameter F values of the 's' individuals; $mean_{WL}(S_F)$ denotes the weighted Lehmer mean of set S_F. Furthermore, the selection probability of each group is updated according to Eq. (21):

$$\begin{cases} r_j = \begin{cases} \dfrac{ns_j^2}{ns \cdot (ns_j + nf_j)}, & if \;\; ns_j > 0, \\ \epsilon, \;\; otherwise. \end{cases} \\ ns = \sum_{j=1}^{k} ns_j, \\ P(j) = \dfrac{r_j}{\sum_{j=1}^{k}(r_j)}. \end{cases} \tag{21}$$

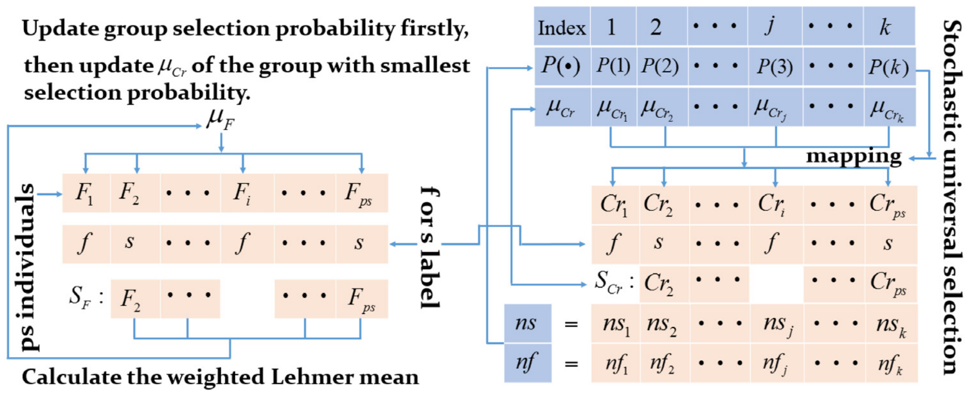


Fig. 2. Illustration of the adaptation schemes for F, Cr and $P(\cdot)$ in PaDE algorithm.

where ns_j and nf_j denote the number of 's' individuals and 'f' individuals in the jth group respectively while ns denotes the number of 's' individuals in the population. ϵ is assigned a small value, e.g. $\epsilon = 0.01$, which is used to avoid possible null values of probability. After the selection probability of all groups are renewed, we can find the index idx of the group with smallest selection probability. If there are more than one index, idx will be assigned a random index from these indices, then the parameter μ_{Cr} in the idx-group is to renewed according to Eq. (22):

$$\begin{cases} w_s = \dfrac{\Delta f_s}{\sum_{s=1}^{|S_{Cr}|} \Delta f_s} \\ \Delta f_i = f(X_{i,G}) - f(U_{i,G}) \\ mean_{WL}(S_{Cr}) = \dfrac{\sum_{s=1}^{|S_{Cr}|} w_s \cdot S_{Cr}^2(s)}{\sum_{s=1}^{|S_{Cr}|} w_s \cdot S_{Cr}(s)} \\ \mu_{Cr_{idx},G+1} = \begin{cases} mean_{WL}(S_{Cr}), & if\ S_{Cr} \neq \varnothing \& \max\{S_{Cr}\} > 0 \\ 0, & if\ S_{Cr} \neq \varnothing \& \mu_{Cr_{idx},G} = 0 \\ \mu_{Cr_{idx},G}. & otherwise \end{cases} \\ Cr_i = \begin{cases} randn_i(\mu_{Cr_{idx}}, 0.1), & if\ \ \mu_{Cr_{idx}} > 0; \\ 0. & otherwise \end{cases} \end{cases} \tag{22}$$

where S_{Cr} denotes the set of control parameter Cr values of the 's' individuals and $mean_{WL}(S_{Cr})$ denotes the weighted Lehmer mean of set S_{Cr}. Fig. 2 illustrates the grouping strategy with adaptation schemes for F, Cr and $P(\cdot)$ in PaDE algorithm in detail.

3.2. Parabolic population size reduction scheme in PaDE algorithm

In this part a novel parabolic population size reduction scheme is proposed for the dynamic adaptation of population size during evolution. As we know that the linear population size reduction scheme proposed in LSHADE was proven to be excellent scheme for population size adaptation, however, we see that the quick reduction of population size at the beginning of the evolution usually leads to a bad perception of the landscape of some objective functions. That is why we proposed a parabolic population size reduction scheme in the novel PaDE algorithm. The detailed parabolic population size reduction scheme is given in Eq. (23):

$$ps_{G+1} = round[\frac{ps_{\min} - ps_{ini}}{(nfe_{\max} - ps_{ini})^2} \cdot (nfe - ps_{ini})^2 + ps_{ini}] \tag{23}$$

where $ps_{\min}$ and ps_{ini} denote the minimum and initial value of population size, nfe and $nfe_{\max}$ denote the current number of function evaluation and maximum number of function evaluation respectively, $round[\cdot]$ denotes rounding to the nearest value operation.

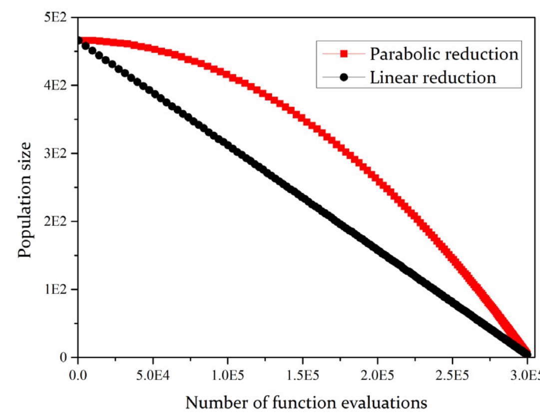


Fig. 3. Illustration of the population size reduction schemes between the novel parabolic approach and linear approach.

Here we give an example to illustrate the difference between the novel parabolic population size reduction scheme and the commonly used linear population size reduction scheme (shown in Eq. (24)) on two aspects: one is population size, and the other is number of generations.

$$ps_{G+1} = round[\frac{ps_{\min} - ps_{ini}}{nfe_{\max}} \cdot nfe + ps_{ini}] \tag{24}$$

In the example, the dimension number is set to $D = 30$, initial population size is set to $ps_{ini} = 25 \cdot \sqrt{(D)} \cdot \log(D)$, the minimum population size (also the terminal population size) is set to $ps_{\min} = 4$, and the maximum number of function evaluations is set to $nfe_{\max} = 10000 \cdot D$. Figs. 3 and 4 present the comparison between the novel parabolic approach and the linear approach for the adaptation of population size on the above mentioned two aspects respectively. We can see from Fig. 3 that the population size reduction of the parabolic approach is much slower than the linear approach at the beginning of the evolution, and this slow reduction of population size at the beginning is beneficial to get a relative full perception of the landscape of objective functions. We also can see from Fig. 4 that the parabolic approach consumes less generations during the evolution.

A pivot $pt = (x, y)$ can be introduced into the parabolic approach and then balance can be achieved between population size and number of generations. x denotes the current number of function evaluations and y denotes the current population size, then

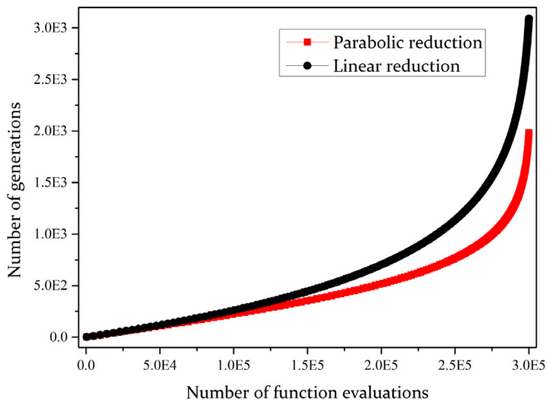


Fig. 4. Illustration of the number of generations between the novel parabolic approach and linear approach.

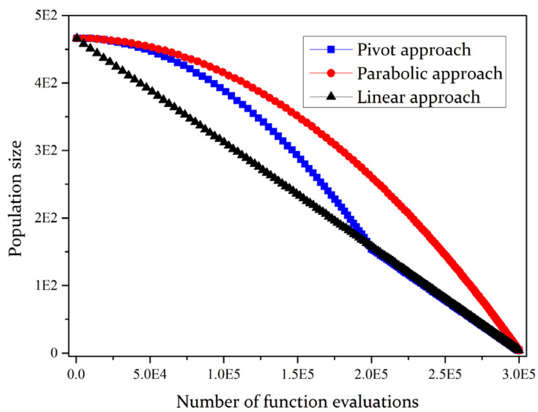


Fig. 5. Illustration of the number of generations between the novel parabolic approach and linear approach.

the population size reduction scheme can be changed to Eq. (25):

$$ps_{G+1} = \begin{cases} ceil[\frac{y - ps_{ini}}{(x - ps_{ini})^2} \cdot (nfe - ps_{ini})^2 + ps_{ini}], \ \ if \ nfe < x \\ floor[\frac{y - ps_{ini}}{(x - ps_{\min})} \cdot (nfe - nfe_{\max}) + ps_{min}]. \ otherwise \end{cases} \tag{25}$$

Fig. 5 illustrates the comparison among parabolic approach, linear approach and pivot approach of population size reduction with the pivot $pt = round[(2/3 \cdot nfe_{\max}, 1/3 \cdot ps_{ini})]$. In the novel PaDE algorithm, the pivot approach of population size reduction scheme is the default setting for the adaptation of population size and it can be denoted as PaDE-Pivot, moreover, the linear population size reduction scheme and the parabolic population size reduction scheme can be considered as special cases of the default PaDE algorithm by setting the pivot to $pt = (ps_{ini}, ps_{ini})$ and $pt = (nfe_{\max}, ps_{min})$ respectively. These two variants are denoted as PaDE-linear and PaDE-Para respectively herein this paper, and deeper analysis of the overall optimization performances of these three PaDE variants will be discussed in Section 4.4.

3.3. Time stamp based mutation strategy in PaDE algorithm

As is mentioned in Section 1, mutation strategy plays a key role in optimization performance of DE variants. Recent state-of-the-art DE variants, e.g. JADE, LSHADE, iLSHADE and jSO etc., all focused on the DE/target-to-pbest/1/bin mutation strategy. Moreover, by introducing an enhanced DE/target-to-pbest/1/bin mutation strategy with a time stamp scheme, LPALMDE [5] and QUATRE-EAR [37] obtained an overall better performance on CEC2013 [4] and CEC2014 [40] benchmarks. However, further experiments revealed that the diversity of external individuals can be increased by introducing a decay rate of the time stamp scheme. Therefore, in this part, an enhanced DE/target-to-pbest/1/bin mutation strategy with time stamp scheme is presented, and all the inferior individuals are firstly initialized by the same time stamp T_0 and a decay rate r^d, and then inserted into the external archive. The detailed equation of the mutation strategy in PaDE is given in Eq. (26):

$$V_{i,G} = X_{i,G} + F \cdot (X^p_{best,G} - X_{i,G}) + F \cdot (X_{r_1,G} - \widehat{X}_{r_2,G}) \tag{26}$$

where all symbols have the same meanings as Eq. (2) in JADE except for $\widehat{X}_{r_2,G}$. Here $\widehat{X}_{r_2,G}$ denotes a randomly selected vector from the union $P \cup \widehat{A}$, P denotes the current population of individuals, and $\widehat{A}$ denotes the external archived inferior solution set with time stamp scheme.

The external archive is initiated to be empty, and then new inferior solutions with time stamp t equalling to T_0 are inserted into it generation by generation. For each individual in the external archive, its associated time stamp t in the nth generation can be calculated by Eq. (27):

$$t = T_0 - r^d \cdot (n - m) \tag{27}$$

if it was inserted into the external archive in the mth generation ($n \geq m$). In each generation of the evolution, the time stamp t of each individual is checked first whether it is larger than 0 or not at the beginning. If the time stamp satisfies $t < 0$, the individual should be erased from the external archive first and then mutation operation is conducted by employing the renewed external archive. Moreover, inferior solutions of the current population are also added into the external archive at the end of each generation, and if the size of the external archive exceeds a fixed maximum, $r^{arc} \cdot ps$, then a number of solutions are randomly picked and erased from the archive keeping the archive size equalling to $r^{arc} \cdot ps$. The decay rate can also be dynamically decreased according to the ratio of better solutions found by the trial vectors, and the default setting of decay rate r^d is a fixed constant value $r^d = 0.04$ in the PaDE algorithm.

Actually, the time stamp based external archive can be considered as a balance between the mutation strategies with archive and without archive in JADE algorithm, which will be further discussed in Section 4.5. The pseudo code of the PaDE is presented in Algorithm 2.

4. Experiment analysis of PaDE algorithm

4.1. Experiment environment description

As is different from the deterministic algorithms, the verification that an evolutionary algorithm outperforms another is difficult because of limited knowledge. Therefore, benchmark functions come to take effect on these performance evaluations. In real-parameter single optimization, there are several commonly used benchmarks of test suites at CEC Competitions, and they are CEC2005 [41] test suite, CEC2013 test suite, CEC2014 test suite and CEC2017 [42] test suite. CEC2013 test suite was based on the comments of CEC2005, and CEC2017 test suite was based on the comments of CEC2014. Improvements were also made as well as new benchmarks added into the later test suite. Though CEC2014 test suites was also asserted that they were improved from CEC2013 test suite, they were actually different because they employed different rotation matrices. Moreover, each benchmark in CEC2014 test suite had a shift data, which was different from CEC2013 and same as CEC2017.

Algorithm 2: Pseudo code of PaDE algorithm

Input: Bound constraints $[R^D_{min}, R^D_{max}]$, the fixed maximum number of function evaluations $nfe_{\max}$, benchmark functions $f(\boldsymbol{X})$;
Output: Best fitness value $f(X_{gbest})$, best individual X_{gbest}, number of function evaluations nfe;
Initialize the population size $ps = ps_{ini}$, locations of all individuals $\boldsymbol{X} = \{X_1, X_2, ..., X_{ps}\}$, $k = 4$, $\boldsymbol{A} = \varnothing$, $\mu_F = 0.8$, $\mu_{Cr_1} = \mu_{Cr_2} ... = \mu_{Cr_k} = \mu_{Cr} = 0.6$, $p = 0.11$, $r^{arc} = 1.6$, $P(1) = P(2) = ... = P(k) = \frac{1}{k}$, $T_0 = 70$, $r^d = 0.04$, $G = 1$;

```
for i = 1; i ≤ ps; i++ do
    X_{i,G} = X_i;
    Calculate the fitness value f(X_{i,G});
nfe = ps;
Find the global best X_{gbest,G} and fitness value f(X_{gbest,G});
while nfe ≤ nfe_max or Δf > eps do
    Generate the top p superior individuals of the target
    individuals, same as LSHADE algorithm;
    Categorize ps individuals into k-groups by stochastic universal
    selection in Algorithm 1;
    Generate F of all individuals according to F ~ C(μ_F, 0.1);
    for j = 1; j ≤ k; j++ do
        Generating Cr of all individuals in the j^th group according
        to the distribution Cr ~ N(μ_{Cr_j}, 0.1);
    Readjust F and Cr into the bound constraints if necessary;
    for i = 1; i ≤ ps; i++ do
        Generating trial vectors U_{i,G} according to Eq. (26);
        Calculate fitness value f(U_{i,G});
    nfe = nfe + ps;
    for i = 1; i ≤ ps; i++ do
        if f(U_{i,G}) ≤ f(X_{i,G}) then
            X_{i,G+1} = U_{i,G};
        else
            X_{i,G+1} = X_{i,G};
    if S_F ≠ ∅ then
        Update μ_F according to Eq. (20);
        Update P(1), P(2), ..., P(k) according to Eq. (21);
        Update μ_{Cr_idx} according to Eq. (22);
    G++;
    Adjust archive A;
    Find X_{gbest,G} and the associate f(X_{gbest,G});
    Adjust population size according to Eq. (24);
    Adjust the population of individuals;
f(X_gbest) = f(X_{gbest,G}), X_gbest = X_{gbest,G};
return f(X_gbest), X_gbest, and nfe;
```

Here in the paper, CEC2013 and CEC2017 test suites containing 58 benchmarks are employed in the comparisons, and the reason why two test suites are employed in our comparison is that the state-of-the-art DE variant jSO algorithm [12] may be over-fitting on the benchmarks of CEC2017 test suite because of the brutal change of control parameters. Therefore, JADE [8], SHADE [9], LSHADE [10], iLSHADE [11], jSO [12], LPALMDE [5] and the new proposed PaDE algorithm are firstly contrasted on CEC2013 test suite and then the proposed PaDE and jSO are also compared under CEC2017 test suite. The second benchmark in CEC2017 test suite is also employed in the comparison because jSO algorithm in paper [12] is verified on this benchmark as well.

Usually, there are two ways to collect data (samples) for algorithm evaluation on each benchmark, one is fixed-cost and the other is fixed-target. Fixed-cost means the number of function evaluations is fixed and the obtained optimization results are taken into consideration for algorithm comparison while fixed-target means the target optimization results are fixed and the consumed number of function evaluations is taken into consideration for algorithm comparison. In some real-world optimization applications, the final optimization results should be returned in a tolerable cost, e.g. time or run-length, etc., therefore, we employ the fixed-cost criterion in this paper for algorithm evaluation.

For the benchmarks from CEC2013, they can be categorized into three groups, f_1–f_5 are uni-modal functions, f_6–f_{20} are basic multi-modal functions, and f_{21}–f_{28} are composition functions. All these benchmarks are shifted to the same minimum $O = (o_1, o_2, \ldots, o_d)$, and they are also mapped into our test suite and labelled as fa_1–fa_{28}. For the benchmarks from CEC2017, they can be categorized into four groups, f_1–f_3 are uni-modal functions, f_4–f_{10} are simple multi-modal functions, f_{11}–f_{20} are hybrid functions and f_{21}–f_{30} are composition functions. It is different from the CEC2013 benchmarks that benchmarks from CEC2017 are shifted to different minima, and all these benchmarks are also mapped into our test suite and labelled as fb_1–fb_{30}.

Experiment results in this paper were obtained on RedHat Linux Enterprise Edition 5.5 x64 Operating System of a PC with Intel(R) Core(TM) $i5 - 3470$ CPU @ 3.2 Hz, and algorithms were all implemented in Matlab 2011b Unix version. Fitness error $\Delta f = f - f^*$ (f was the optimization result obtained by the corresponding algorithm on a certain benchmark and f^* was the minimum of the benchmark) that smaller than eps, $eps = 2.220446e{-}016$, was considered as zero herein.

4.2. Parameter settings of the contrasted algorithms

As is mentioned above, there are seven DE variants contrasted in this paper, and they are JADE, SHADE, LSHADE, iLSHADE, jSO, LPALMDE and the new proposed PaDE algorithm. All these algorithms employ the recommended parameter settings in our conducted experiments, and they are summarized in Table 1.

In JADE, control parameter F obeys semi-fixed Cauchy distribution, $F \sim C(\mu_F, 0.1)$, μ_F is the location parameter with its initial value equalling to 0.5. Control parameter Cr obeys semi-fixed Normal distribution, $Cr \sim N(\mu_{Cr}, 0.1)$, μ_{Cr} denotes the mean value, and the initial value of μ_{Cr} is set to $\mu_{Cr} = 0.5$. Population size in JADE is set to $ps = 100$, the ratio of top superior individuals is set to $p = 0.05$, and balance parameter c is set to $c = 0.1$. In SHADE, the control parameter settings of F, Cr and ps are the same as JADE. Moreover, historical success values of F and Cr are recorded in a H-entry pool in SHADE, $H = 100$, and a bigger ratio is used for the restriction of top superior individuals, $p = 0.2$. LSHADE employs the same initial values and distributions of control parameters F and Cr as SHADE, and a linear population size reduction scheme is also employed in LSHADE with the initial population size equalling to $ps = 18D$ and the minimum population size equalling to 4. Furthermore, the parameter r^{arc} defining the factor of external archive size, the parameter H defining entry number in the memory pool and the parameter p defining the ratio of top superior individuals are also different from the ones in SHADE, these parameters are set to $r^{arc} = 2.6, H = 6, p = 0.11$ respectively, as they are tuned under the linear population size reduction scheme. iLSHADE employs the same distributions of control parameters F and Cr but different initial values of μ_F and μ_{Cr} are used in this algorithm, the initial values of μ_F and μ_{Cr} are set to $\mu_F = 0.8$ and $\mu_{Cr} = 0.5$ respectively. Moreover, the control parameters in the last entry is set fixed control parameter values, $F_H = Cr_H = 0.9$, during the whole evolution. The population size reduction scheme is also the same as LSHADE but with different initial value, $ps = 12D$. There is still another difference in iLSHADE, the parameter p defining the ratio of top superior individuals is dynamically changed from 0.2 to 0.1, which is set a constant value in LSHADE algorithm. For jSO, the

Table 1
Recommended parameter settings of all these contrasted algorithms.

Algorithms.	Parameters initial settings
JADE	$\mu_F = 0.5$, $F \sim C(\mu_F, 0.1)$, $\mu_{Cr} = 0.5$, $Cr \sim N(\mu_{Cr}, 0.1)$, $ps = 100$, $p = 0.05$, $c = 0.1$
SHADE	$\mu_F = 0.5$, $F \sim C(\mu_F, 0.1)$, $\mu_{Cr} = 0.5$, $Cr \sim N(\mu_{Cr}, 0.1)$, $ps = 100$, $p = 0.2$, $H = 100$
LSHADE	F&Cr same as SHADE, $ps = 18D \sim 4$, $r^{arc} = 2.6$, $p = 0.11$, $H = 6$
iLSHADE	$\mu_F = 0.8$, $\mu_{Cr} = 0.5$, F, Cr&r^{arc} same as LSHADE,$H = 6$, $F_H = Cr_H = 0.9$, $ps = 12D \sim 4$, $p = 0.2 \sim 0.1$
jSO	$\mu_F = 0.3$, $\mu_{Cr} = 0.8$, F, Cr&r^{arc} same as iLSHADE,$H = 5$, $ps = 25log(D)\sqrt{D} \sim 4$, $p = 0.25 \sim 0.125$
LPALMDE	$F_j = 0.5$, $F_{ji} \sim C(F_j, 0.2)$, $\mu_{Cr} = 0.5$, $Cr \sim N(\mu_{Cr}, 0.1)$, $k = 8$, $ps = 23D \sim k$, $p = 0.11$, $r^{arc} = 1.6$, $T_0 = 70$
PaDE	$\mu_F = 0.8$, $\mu_{Cr} = 0.6$, F&Cr same as LSHADE,$k = 4$, $p = 0.11$, $ps = 25log(D)\sqrt{D} \sim 4$, $r^{arc} = 1.6$, $T_0 = 70$, $r^d = 0.04$

distribution of control parameters F and Cr, the dynamic change of parameter r^{arc} and population size ps are all the same as the ones in iLSHADE, however, there also exists some differences, for example, the initial value of μ_F is set to 0.3, μ_{Cr} is set to 0.8, entry number H is set to 5, the initial population size ps is set to $25log(D)\sqrt{D}$, and parameter p is set a dynamic value in [0.25, 0.125] during evolution. For LPALMDE algorithm, control parameters F and Cr also obey semi-fixed distributions, $F \sim C(\mu_F, 0.2)$, $Cr \sim N(\mu_{Cr}, 0.1)$ and the initial values of μ_F and μ_{Cr} are set to the same value, $\mu_F = \mu_{Cr} = 0.5$. All individuals in LPALMDE algorithm are categories into k groups, and k is set a constant value, $k = 8$. Population size is dynamically changed according to the population size reduction scheme in LSHADE, and the initial value of population is set to $ps = 23D$. There should be at least one individual in each group, therefore, the minimum value of ps is set to k. Moreover, a time stamp scheme is employed in the external archive, and the time stamp threshold is set to $T_0 = 70$. Parameters r^{arc} and p are also set tuned values under the time stamp scheme, $r^{arc} = 1.6$ and $p = 0.11$. For the proposed PaDE algorithm, control parameters F and Cr obey Cauchy distribution $C(\mu_F, 0.1)$ and Normal distribution $N(\mu_{Cr}, 0.1)$ respectively, and the initial values of μ_F and μ_{Cr} are set to $\mu_F = 0.8$ and $\mu_{Cr} = 0.6$. The initial population size is set a same value as jSO, $ps = 25log(D)\sqrt{D}$. Parameter p and time stamp threshold T_0 are the same values as LPALMDE. The number of groups in PaDE is different from that in LPALMDE, a smaller group number, $H = 4$ is employed within it. Moreover, the code of JADE can be found through Prof. Zhang's homepage[1] or Prof. Wang's homepage,[2] codes of SHADE, LSHADE, iLSHADE and jSO can be obtained from the shared files on Prof. Suganthan's homepage,[3] and the codes of PaDE variants can also be obtained through the first author's homepage.[4]

4.3. Optimization performance comparison with some state-of-the-art DE variants

Here we mainly discuss the overall optimization performance among JADE, SHADE, LSHADE, iLSHADE, jSO, LPALMDE and the proposed PaDE algorithm. The experiments are conducted on fa_1–fa_{28} benchmarks of our test suite on 10-D and 30-D optimization respectively. The maximum number of function evaluations is set to $nfe_{\max} = 10000D$ for all DE variants, and 51 runs are conducted on each benchmarks. Tables 2 and 3 present the mean/std (mean value and the corresponding standard deviation) of fitness error over the 51 runs on 10D and 30D optimization respectively. Symbols ">", "=" and "<" in the parentheses behind the mean/std pair denote "Better Performance", "Similar Performance" and "Worse Performance" respectively, all of which are measured under Wilcoxon's signed rank test with a level of significant $\alpha = 0.05$.

From Table 2 on 10D optimization, we can see that all contrasted algorithms perform equally well on benchmarks fa_1, fa_2, fa_4 and fa_5, and they all can find the global optima every time during 51 runs on these four benchmarks. Furthermore, the proposed PaDE algorithm reveals 27 better or similar performances out of 28 benchmarks in comparison with JADE algorithm, it also reveals 25 better or similar performances out of 28 benchmarks in comparison with SHADE algorithm, 20 better or similar performances out of 28 benchmarks in comparison with LSHADE algorithm, 20 better or similar performances in comparison with iLSHADE algorithm, 18 better or similar performances in comparison with jSO algorithm, 21 better or similar performances in comparison with LPALMDE algorithm. In addition, the proposed PaDE algorithm performs the best or tier best performance on benchmarks $fa_1, fa_2, fa_4, fa_5, fa_{11}, fa_{17}, f_{19}, fa_{20}, fa_{22}, fa_{27}$ in comparison with all other contrasted algorithms. In a word, the new proposed PaDE secures an overall better performance on 10D optimization under the tested fa_1–fa_{28} benchmark functions.

From Table 3 on 30D optimization, we can see that all contrasted algorithms perform equally well on benchmarks fa_1 and fa_{28}, and all algorithms can only find the global optima on benchmark fa_1. Moreover, the proposed PaDE algorithm obtains 26 better or similar performances out of 28 benchmarks in comparison with JADE algorithm, obtains 23 better or similar performances out of 28 benchmarks in comparison with SHADE algorithm, obtains 20 better or similar performances in comparison with LSHADE algorithm, 21 better or similar performances in comparison with iLSHADE algorithm, 20 better or similar performances in comparison with jSO algorithm and 20 better or similar performances in comparison with LPALMDE algorithm. Furthermore, the proposed PaDE algorithm performs the best or tier best performance on benchmarks $fa_1, fa_6, fa_{17}, fa_{24}, fa_{26}$ and fa_{28} in comparison with all other contrasted algorithms. As a result, the new proposed PaDE is still competitive with the other state-of-the-art DE variants on 30D optimization under these benchmark functions.

From Table 4 on 100D optimization, we can see that all contrasted algorithms perform worse under our test suite except for benchmark f_1 and f_5, in other words, all these algorithms were not good at tackling higher dimensional ($D > 100$) optimization problems especially for that multi-modal and rotated benchmarks in our test suite. Moreover, the novel proposed PaDE algorithm obtains 24 better or similar performances out of 28 benchmarks in comparison with JADE algorithm, obtains 24 better or similar performances out of 28 benchmarks in comparison with SHADE algorithm, obtains 22 better or similar performances in comparison with LSHADE algorithm, 18 better or similar performances in comparison with iLSHADE algorithm, 18 better or similar performances in comparison with jSO algorithm and 17 better or similar performances in comparison with LPALMDE algorithm. To summarize, the new proposed PaDE is still competitive with the contrasted state-of-the-art DE variants on 100D optimization under our test suite.

The convergence speed comparisons are also given here in Figs. 6–10 for algorithm evaluation. Median value of the 51-run

1 http://dces.essex.ac.uk/staff/qzhang/.
2 http://ist.csu.edu.cn/YongWang.htm.
3 http://www3.ntu.edu.sg/home/epnsugan/.
4 https://sites.google.com/view/zhenyumeng/.

Table 2
Mean/Std fitness error $\Delta f = f - f*$ comparison on 10D optimization among JADE, SHADE, LSHADE, iLSHADE, jSO, LPALMDE and PaDE is presented here. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations $nfe_{\max}$ equalling to $10000D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant level $\alpha = 0.05$ in comparison with the new proposed PaDE.

D = 10	JADE	SHADE	LSHADE	iLSHADE	jSO	LPALMDE	PaDE
fa_{1}	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0
fa_{2}	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0
fa_{3}	3.7651E+001/7.7285E+001(<)	1.0974E−001/7.6340E−001(<)	1.9091E−001/9.7404E−001(<)	1.1193E−002/2.6209E−002(<)	1.3992E−003/9.9919E−003(=)	4.1975E−003/1.6957E−002(<)	1.3992E−003/9.9919E−003
fa_{4}	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0
fa_{5}	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0
fa_{6}	4.4252E+000/4.9312E+000(<)	8.6580E+000/3.1929E+000(<)	5.1948E+000/4.9465E+000(<)	6.1568E+000/4.7914E+000(<)	1.3468E+000/3.4102E+000(>)	1.1544E+000/3.1929E+000(>)	1.5392E+000/3.6040E+000
fa_{7}	1.0759E−001/1.6301E−001(<)	4.6560E−003/4.8715E−003(<)	1.4181E−005/2.3149E−005(>)	1.4712E−005/4.1995E−005(>)	3.4153E−005/1.0479E−004(<)	2.0074E−005/3.7632E−005(<)	2.2893E−005/4.3456E−005
fa_{8}	2.0329E+001/6.4992E−002(<)	2.0367E+001/7.1822E−002(<)	2.0230E+001/1.5088E−001(<)	2.0338E+001/9.3426E−002(<)	2.0358E+001/8.3752E−002(<)	2.0079E+001/1.4596E−001(>)	2.0217E+001/1.7095E−001
fa_{9}	3.7562E+000/6.2099E−001(<)	3.2557E+000/9.9109E−001(<)	2.3294E+000/1.6767E+000(<)	5.8850E−001/8.7988E−001(>)	7.0117E−001/8.6243E−001(>)	4.0449E−001/6.3400E−001(>)	2.1341E+000/1.4587E+000
fa_{10}	1.8488E−002/8.3409E−003(<)	1.1660E−002/7.5163E−003(<)	1.0334E−002/1.2181E−002(<)	6.0844E−003/8.2879E−003(>)	1.7401E−003/3.8597E−003(>)	1.1630E−002/1.6007E−002(<)	6.8103E−003/1.2698E−002
fa_{11}	0/0(=)	0/0(=)	0/0(=)	1.9509E−002/1.3932E−001(<)	0/0(=)	0/0(=)	0/0
fa_{12}	4.2590E+000/1.3539E+000(<)	3.6701E+000/9.9258E−001(<)	2.1393E+000/7.9528E−001(>)	2.0906E+000/7.7796E−001(>)	2.3801E+000/8.2236E−001(>)	3.6872E+000/1.5579E+000(<)	3.0146E+000/1.9179E+000
fa_{13}	5.4677E+000/2.2796E+000(<)	3.5892E+000/1.4849E+000(<)	2.0873E+000/1.1803E+000(>)	1.7814E+000/8.5561E−001(>)	2.1201E+000/1.0770E+000(>)	3.1493E+000/1.9315E+000(<)	2.4506E+000/1.5657E+000
fa_{14}	1.2246E−002/2.7985E−002(>)	2.4492E−003/1.2244E−002(>)	2.8166E−002/5.0451E−002(<)	2.8030E−001/6.4621E−001(<)	3.6075E−002/4.3136E−002(<)	1.3327E+000/2.7736E+000(<)	1.3471E−002/2.8793E−002
fa_{15}	4.8334E+002/1.2449E+002(<)	4.3274E+002/1.2858E+002(<)	3.0419E+002/1.1924E+002(>)	2.5327E+002/1.1714E+002(>)	2.8348E+002/1.1125E+002(>)	4.7538E+002/1.7888E+002(<)	3.6068E+002/1.5552E+002
fa_{16}	1.1555E+000/2.1311E−001(<)	6.7928E−001/1.9768E−001(<)	2.9314E−001/1.6262E−001(<)	8.3311E−001/3.0579E−001(<)	1.0942E+000/2.0374E−001(<)	1.1155E−001/1.1431E−001(>)	2.4235E−001/1.7421E−001
fa_{17}	1.0122E+001/7.9877E−015(=)	1.0122E+001/4.4466E−014(=)	1.0122E+001/7.9877E−015(=)	1.0126E+001/6.6650E−003(<)	1.0123E+001/8.5962E−004(<)	1.0147E+001/9.5300E−002(<)	1.0122E+001/1.3510E−014
fa_{18}	1.8796E+001/1.6743E+000(<)	1.7192E+001/1.5687E+000(<)	1.3860E+001/1.2415E+000(>)	1.3379E+001/1.2681E+000(>)	1.6434E+001/1.9716E+000(<)	1.5273E+001/2.6917E+000(>)	1.5493E+001/2.6075E+000
fa_{19}	3.3622E−001/4.2728E−002(<)	3.2097E−001/5.2957E−002(<)	2.2556E−001/3.2446E−002(<)	3.1047E−001/5.9523E−002(<)	2.7388E−001/4.9580E−002(<)	5.3609E−001/1.5695E−001(<)	2.1061E−001/3.2362E−002
fa_{20}	2.2740E+000/4.6883E−001(<)	2.2493E+000/4.1469E−001(<)	1.9943E+000/3.8533E−001(<)	1.8281E+000/5.2566E−001(<)	1.7248E+000/3.1470E−001(<)	1.7195E+000/4.0531E−001(<)	1.7078E+000/4.4343E−001
fa_{21}	4.0019E+002/0.0000E+000(=)	3.9627E+002/2.8033E+001(>)	4.0019E+002/0.0000E+000(=)	4.0019E+002/0.0000E+000(=)	3.9627E+002/2.8033E+001(>)	4.0019E+002/0.0000E+000(=)	4.0019E+002/0.0000E+000
fa_{22}	3.2721E+000/4.2043E+000(<)	4.8362E+000/5.3147E+000(<)	1.1607E+001/2.3918E+001(<)	2.2267E+001/3.2619E+001(<)	6.5426E+000/4.8560E+000(<)	2.3400E+001/2.3195E+001(<)	2.7887E+000/3.3285E+000
fa_{23}	5.2172E+002/1.7317E+002(<)	4.3764E+002/1.6590E+002(<)	2.8705E+002/1.5536E+002(>)	2.2308E+002/1.1810E+002(>)	2.2153E+002/1.1252E+002(>)	4.1630E+002/2.0770E+002(<)	3.2387E+002/1.7130E+002
fa_{24}	1.9932E+002/9.7873E+000(<)	2.0002E+002/4.7469E+000(<)	2.0074E+002/1.4231E+001(<)	2.0109E+002/1.1946E+001(<)	1.9992E+002/1.1073E+001(<)	1.9812E+002/1.3405E+001(>)	1.9807E+002/1.3771E+001
fa_{25}	2.0033E+002/5.1683E+000(<)	2.0038E+002/1.8329E+000(<)	1.9963E+002/1.3963E+001(>)	2.0054E+002/1.7517E+000(<)	2.0009E+002/6.3574E−001(<)	1.9813E+002/1.3359E+001(>)	2.0000E+002/2.6788E−005
fa_{26}	1.4147E+002/4.5620E+001(<)	1.2359E+002/3.8210E+001(<)	1.6054E+002/4.7715E+001(<)	1.2384E+002/4.0550E+001(<)	1.0258E+002/1.8040E+000(>)	1.1345E+002/2.8903E+001(>)	1.1843E+002/3.5574E+001
fa_{27}	3.0230E+002/1.5159E+001(<)	3.0000E+002/5.3249E−005(=)	3.0000E+002/0.0000E+000(=)	3.1415E+002/4.9002E+001(<)	3.0000E+002/0.0000E+000(=)	3.0000E+002/0.0000E+000(=)	3.0000E+002/0.0000E+000
fa_{28}	2.9608E+002/2.8006E+001(=)	2.9216E+002/3.9208E+001(>)	2.9216E+002/3.9208E+001(>)	3.0000E+002/0.0000E+000(<)	3.0000E+002/0.0000E+000(<)	2.9608E+002/2.8006E+001(=)	2.9608E+002/2.8006E+001
w/l/d	1/8/19	3/7/18	8/8/12	8/5/15	9/7/12	7/8/13	–

Table 3
Mean/Std fitness error $\Delta f = f - f*$ comparison on 30D optimization among JADE, SHADE, LSHADE, iLSHADE, jSO, LPALMDE and PaDE is presented here. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations $nfe_{\max}$ equalling to $10000D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant level $\alpha = 0.05$ in comparison with the new proposed PaDE.

D = 30	JADE	SHADE	LSHADE	iLSHADE	jSO	LPALMDE	PaDE
fa_{1}	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0(=)	0/0
fa_{2}	7.0462E+003/5.8745E+003(<)	6.8659E+003/5.1832E+003(<)	7.8288E−012/3.8820E−011(>)	2.0823E−009/1.0525E−008(<)	3.7115E−010/7.7174E−010(<)	3.5191E−010/1.8040E−009(<)	4.2688E−011/2.1931E−010
fa_{3}	3.1464E+005/1.0965E+006(<)	3.4573E+001/8.9998E+001(<)	9.7559E−001/2.7597E+000(<)	5.3028E−001/2.7556E−002(<)	2.3793E−009/1.5886E−008(>)	7.3958E−003/3.6215E−002(<)	2.5361E−004/1.2253E−003
fa_{4}	3.5640E+003/1.0960E+004(<)	1.6040E−005/6.0575E−005(<)	6.6875E−014/1.0463E−013(>)	3.9679E−013/2.5255E−013(<)	2.5457E−012/1.7191E−012(<)	3.2991E−013/2.1481E−013(<)	2.1846E−013/1.0129E−013
fa_{5}	1.1146E−013/1.5919E−014(=)	1.0923E−013/2.2287E−014(>)	1.1369E−013/0.0000E+000(<)	1.1369E−013/0.0000E+000(<)	1.1369E−013/0.0000E+000(<)	1.1369E−013/0.0000E+000(<)	1.1146E−013/1.5919E−014
fa_{6}	2.0712E+000/7.1703E+000(<)	5.1779E−001/3.6978E+000(<)	1.8417E−005/9.6326E−005(<)	1.0356E+000/5.1769E+000(<)	1.0356E+000/5.1769E+000(<)	3.6853E−004/1.9134E−003(<)	1.3097E−006/9.2577E−006
fa_{7}	2.6358E+000/2.3447E+000(<)	2.8994E+000/2.9797E+000(<)	9.6534E−001/8.0675E−001(<)	2.8039E−001/2.2748E−001(<)	4.9068E−002/1.5007E−001(>)	1.0292E−001/1.2628E−001(>)	1.2669E−001/1.4861E−001
fa_{8}	2.0942E+001/5.3025E−002(<)	2.0905E+001/1.1924E−001(<)	2.0860E+001/1.2012E−001(<)	2.0833E+001/1.0683E−001(<)	2.0957E+001/4.8471E−002(<)	2.0717E+001/1.2151E−001(>)	2.0818E+001/1.4319E−001
fa_{9}	2.6891E+001/1.8036E+000(<)	2.7693E+001/1.5686E+000(<)	2.6537E+001/1.6240E+000(<)	2.0158E+001/4.9504E+000(>)	2.3704E+001/2.6676E+000(>)	1.5636E+001/3.7837E+000(>)	2.5498E+001/1.4712E+000
fa_{10}	4.2114E−002/2.4790E−002(<)	6.3724E−002/4.2076E−002(<)	0/0(>)	0/0(>)	0/0(>)	9.6611E−004/3.1950E−003(>)	5.0749E−003/5.0128E−003
fa_{11}	0/0(>)	0/0(>)	8.2479E−014/3.0745E−014(>)	6.5760E−014/2.0878E−014(<)	1.6496E−013/7.3021E−014(<)	3.8060E+000/2.2312E+000(<)	3.5666E−014/2.7756E−014
fa_{12}	2.4090E+001/4.8150E+000(<)	2.1005E+001/3.3392E+000(<)	6.4414E+000/1.6541E+000(>)	7.0237E+000/2.0623E+000(>)	9.0538E+000/2.4992E+000(<)	1.1296E+001/3.2566E+000(<)	8.1114E+000/1.4309E+000
fa_{13}	4.8920E+001/1.1661E+001(<)	4.1887E+001/1.0769E+001(<)	7.9206E+000/2.8862E+000(>)	1.0136E+001/5.7137E+000(<)	1.0715E+001/5.2616E+000(<)	2.2353E+001/1.0266E+001(<)	9.8167E+000/4.3167E+000
fa_{14}	2.7759E−002/2.3431E−002(<)	1.6329E−002/1.8767E−002(>)	2.8984E−002/2.2836E−002(<)	4.6537E−002/2.9907E−002(<)	8.4381E+000/4.4500E+000(<)	2.1041E+002/1.1899E+002(<)	2.5718E−002/2.1063E−002
fa_{15}	3.2918E+003/3.0778E+002(<)	3.2371E+003/2.6852E+002(<)	2.7202E+003/2.3811E+002(>)	2.5137E+003/2.7530E+002(>)	2.6816E+003/3.3684E+002(>)	3.1328E+003/4.4921E+002(<)	2.8145E+003/3.1207E+002
fa_{16}	1.9783E+000/6.2751E−001(<)	9.6293E−001/1.3840E−001(<)	8.1803E−001/1.3294E−001(<)	9.2240E−001/4.3183E−001(<)	2.3393E+000/3.0407E−001(<)	3.4416E−001/2.2456E−001(>)	6.3143E−001/3.0285E−001
fa_{17}	3.0434E+001/8.0389E−015(=)	3.0434E+001/8.0389E−015(=)	3.0434E+001/9.8684E−013(=)	3.0434E+001/2.1913E−006(=)	3.0669E+001/1.0979E−001(<)	3.7692E+001/2.4068E+000(<)	3.0434E+001/9.4299E−007
fa_{18}	7.5750E+001/7.6650E+000(<)	7.3036E+001/6.5639E+000(<)	5.4372E+001/3.0490E+000(>)	4.4237E+001/4.0750E+000(>)	6.0103E+001/5.7123E+000(<)	4.6128E+001/5.9005E+000(>)	5.5348E+001/5.0973E+000
fa_{19}	1.4356E+000/1.2019E−001(<)	1.3692E+000/1.0134E−001(<)	1.1862E+000/9.8409E−002(<)	1.0496E+000/1.3757E−001(>)	1.2898E+000/1.0311E−001(<)	2.3760E+000/6.0265E−001(<)	1.1203E+000/8.7538E−002
fa_{20}	1.0161E+001/4.9428E−001(<)	1.0669E+001/4.7842E−001(<)	1.1187E+001/1.8136E+000(<)	1.0761E+001/1.5341E+000(<)	9.7472E+000/3.8117E−001(>)	9.2988E+000/5.5861E−001(>)	1.0359E+001/1.8654E+000
fa_{21}	3.1161E+002/7.0509E+001(<)	2.9080E+002/5.4313E+001(>)	2.9693E+002/3.1662E+001(<)	3.0316E+002/6.2148E+001(<)	3.0708E+002/5.8493E+001(<)	3.0281E+002/2.0100E+001(<)	2.9608E+002/1.9604E+001
fa_{22}	1.0324E+002/1.8223E+001(>)	9.5458E+001/2.5459E+001(>)	1.0840E+002/2.3645E+000(<)	1.0676E+002/1.1981E+000(<)	1.1976E+002/3.4337E+000(<)	2.1300E+002/6.6635E+001(<)	1.0632E+002/1.0848E+000
fa_{23}	3.4268E+003/3.8445E+002(<)	3.4876E+003/4.1432E+002(<)	2.4144E+003/3.0693E+002(>)	2.1826E+003/4.0411E+002(>)	2.3659E+003/3.3190E+002(>)	3.0503E+003/4.6513E+002(<)	2.7682E+003/2.9282E+002
fa_{24}	2.0857E+002/1.1226E+001(<)	2.0399E+002/3.8307E+000(<)	2.0062E+002/1.3859E+000(<)	2.0069E+002/4.1575E+000(<)	2.0003E+002/3.2709E−002(=)	2.0003E+002/3.3724E−002(=)	2.0003E+002/5.6801E−002
fa_{25}	2.8170E+002/6.3689E+000(<)	2.7390E+002/1.5541E+001(<)	2.4130E+002/4.0186E+000(<)	2.4000E+002/4.6140E+000(<)	2.3924E+002/6.8915E+000(<)	2.0806E+002/1.7642E+001(>)	2.1450E+002/2.0867E+001
fa_{26}	2.1581E+002/4.4406E+001(<)	2.0205E+002/1.4639E+001(<)	2.0000E+002/0(=)	2.0784E+002/2.7153(<)	2.0000E+002/0(=)	2.0000E+002/0(=)	2.0000E+002/0
fa_{27}	6.6268E+002/2.4023E+002(<)	4.0486E+002/1.4821E+002(<)	3.0182E+002/5.6276E+000(<)	3.0104E+002/2.6188E+000(>)	3.0083E+002/9.8159E−001(>)	3.0180E+002/3.7048E+000(<)	3.0148E+002/2.9330E+000
fa_{28}	3.0000E+002/0(=)	3.0000E+002/0(=)	3.0000E+002/2.8705E−013(=)	3.0000E+002/0(=)	3.0000E+002/0(=)	3.0000E+002/0(=)	3.0000E+002/0
w/d/l	2/4/22	5/3/20	8/4/16	8/3/17	8/4/16	8/4/16	-/-/-

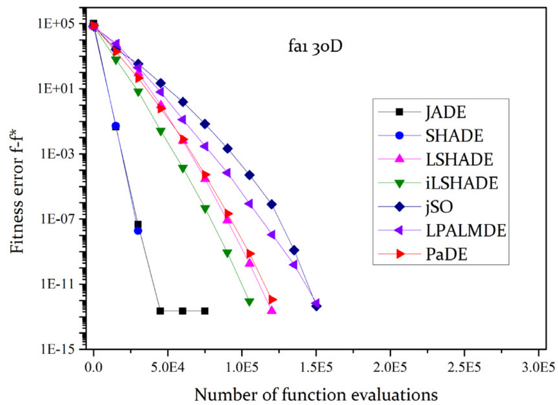
Table 4
Mean/Std fitness error $\Delta f = f - f*$ comparison on 100D optimization among JADE, SHADE, LSHADE, iLSHADE, jSO, LPALMDE and PaDE is presented here. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations $nfe_{\max}$ equalling to $10000D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant level $\alpha = 0.05$ in comparison with the new proposed PaDE.

D = 100	JADE	SHADE	LSHADE	iLSHADE	jSO	LPALMDE	PaDE
fa_1	4.0571E−013/9.4449E−014(<)	3.7450E−013/1.0974E−013(<)	2.2737E−013/0.0000E+000(=)	2.2737E−013/0.0000E+000(=)	2.2737E−013/0.0000E+000(=)	2.2737E−013/0.0000E+000(=)	2.2737E−013/0.0000E+000
fa_2	1.2279E+005/3.7060E+004(>)	1.2202E+005/3.7554E+004(>)	1.7878E+005/5.3312E+004(<)	1.9953E+005/6.3163E+004(<)	1.3015E+005/3.5894E+004(>)	1.6705E+005/4.0402E+004(>)	1.7534E+005/5.8392E+004
fa_3	2.1092E+008/1.6278E+008(<)	3.1471E+007/2.6142E+007(<)	7.5640E+006/7.4181E+006(<)	3.0230E+006/3.5940E+006(<)	1.4529E+006/1.9593E+006(>)	2.6527E+006/2.7162E+006(<)	2.6223E+006/2.9413E+006
fa_4	1.4714E+004/3.4810E+004(<)	1.3375E−003/1.2431E−003(<)	6.1703E−004/4.2628E−004(>)	1.0048E−003/8.0535E−004(>)	6.9722E−004/6.3211E−004(>)	4.7353E−004/3.8816E−004(>)	1.1760E−003/8.1546E−004
fa_5	5.0602E−013/9.4610E−014(<)	4.7258E−013/7.9978E−014(<)	4.6143E−013/8.9262E−014(<)	3.7673E−013/5.7925E−014(<)	4.3246E−013/7.8829E−014(<)	3.7227E−013/6.8509E−014(<)	3.6112E−013/6.3118E−014
fa_6	5.6062E+001/5.9790E+001(<)	6.7331E+001/5.5336E+001(<)	2.3071E+002/1.8034E+001(<)	2.2097E+002/2.3747E+001(<)	2.2378E+002/2.5710E+001(<)	1.4838E+002/5.1100E+001(<)	5.5889E+001/5.1655E+001
fa_7	7.0939E+001/1.3472E+001(<)	5.4879E+001/1.2067E+001(<)	9.7795E+000/2.7101E+000(<)	5.2205E+000/1.4767E+000(>)	3.6708E+000/1.1489E+000(>)	7.5835E+000/2.4607E+000(>)	8.2602E+000/2.2874E+000
fa_8	2.1281E+001/5.3025E−002(<)	2.1180E+001/9.0840E−002(>)	2.1270E+001/3.5575E−002(<)	2.1267E+001/4.8478E−002(<)	2.1302E+001/1.9751E−002(<)	2.1250E+001/6.4663E−002(<)	2.1237E+001/8.7589E−002
fa_9	1.3330E+002/3.6820E+000(<)	1.3582E+002/3.6686E+000(<)	1.3271E+002/2.7772E+000(<)	1.2736E+002/5.3133E+000(>)	1.2796E+002/5.2037E+000(>)	1.0576E+002/1.0430E+001(>)	1.3005E+002/3.0948E+000
fa_{10}	2.9602E−002/1.9738E−002(<)	3.3421E−002/2.0966E−002(<)	1.6857E−002/1.2939E−002(<)	1.1493E−002/9.6781E−003(>)	1.4650E−002/1.0970E−002(<)	1.9506E−002/1.2301E−002(<)	1.3787E−002/1.1045E−002
fa_{11}	5.6843E−014/0.0000E+000(>)	1.9509E−002/1.3932E−001(<)	1.4364E−003/8.2487E−004(<)	7.8586E−004/8.0194E−004(<)	9.7672E−003/7.0492E−003(<)	6.3581E+001/9.3262E+000(<)	3.9221E−011/3.9669E−011
fa_{12}	2.0026E+002/2.6591E+001(<)	1.7033E+002/2.1186E+001(<)	5.7103E+001/9.1399E+000(>)	4.4432E+001/8.4773E+000(>)	4.9987E+001/8.4602E+000(>)	6.2370E+001/8.8914E+000(>)	7.9220E+001/6.8795E+000
fa_{13}	4.4650E+002/4.9379E+001(<)	4.1199E+002/4.3590E+001(<)	1.3246E+002/1.8750E+001(>)	1.2041E+002/3.0058E+001(>)	1.0624E+002/2.7836E+001(>)	1.9179E+002/2.8501E+001(>)	2.0277E+002/2.2412E+001
fa_{14}	8.2138E−002/1.6663E−002(>)	6.2563E−002/1.5488E−002(>)	8.0272E+001/1.6855E+001(<)	4.5094E+001/1.3753E+001(<)	4.7276E+002/1.0907E+002(<)	4.9957E+003/9.0124E+002(<)	8.2958E−001/2.9351E−001
fa_{15}	1.5153E+004/6.6257E+002(<)	1.4653E+004/6.1380E+002(<)	1.5775E+004/6.6234E+002(<)	1.4769E+004/1.0302E+003(<)	1.4887E+004/8.4703E+002(<)	1.3386E+004/1.0695E+003(>)	1.4599E+004/5.3127E+002
fa_{16}	2.0600E+000/6.0289E−001(<)	1.8262E+000/1.9833E−001(<)	1.8449E+000/1.7424E−001(<)	2.4574E+000/8.3543E−001(<)	3.6002E+000/6.1409E−001(<)	1.2786E+000/4.2421E−001(>)	1.6144E+000/4.4532E−001
fa_{17}	1.0156E+002/7.9986E−014(=)	1.0156E+002/1.0573E−013(=)	1.0319E+002/2.7838E−001(<)	1.0233E+002/2.6284E−001(<)	1.1321E+002/1.4435E+000(<)	2.1934E+002/2.2770E+001(<)	1.0156E+002/2.3268E−003
fa_{18}	3.7462E+002/2.7656E+001(<)	3.2847E+002/2.0488E+001(<)	2.8250E+002/1.3429E+001(<)	2.5689E+002/2.5990E+001(<)	2.7877E+002/1.9041E+001(<)	1.5481E+002/1.1489E+001(>)	2.5618E+002/1.2650E+001
fa_{19}	9.3176E+000/9.6727E−001(<)	9.2418E+000/1.1772E+000(<)	7.3842E+000/4.5265E−001(<)	6.9047E+000/3.6801E−001(<)	7.1857E+000/3.3944E−001(<)	1.0972E+001/1.4814E+000(<)	5.8257E+000/2.7538E−001
fa_{20}	5.0000E+001/3.6490E−012(<)	5.0000E+001/9.2395E−009(<)	5.0000E+001/1.4971E−011(<)	4.9990E+001/6.8654E−002(<)	5.0000E+001/2.4305E−009(<)	4.9927E+001/1.8375E−001(=)	4.9927E+001/1.8375E−001
fa_{21}	3.6863E+002/4.6862E+001(>)	3.7647E+002/4.2840E+001(>)	3.3725E+002/4.8829E+001(>)	3.9020E+002/3.0033E+001(>)	3.9608E+002/1.9604E+001(=)	3.9608E+002/1.9604E+001(=)	3.9608E+002/1.9604E+001
fa_{22}	5.8663E+001/4.9993E+001(<)	5.2358E+001/4.8887E+001(<)	1.1049E+002/1.8891E+001(<)	6.7045E+001/2.2028E+001(<)	3.0604E+002/8.3948E+001(<)	5.0150E+003/9.0948E+002(<)	1.9168E+001/7.4270E−001
fa_{23}	1.6921E+004/1.0419E+003(<)	1.7760E+004/1.0586E+003(<)	1.4471E+004/7.8273E+002(<)	1.2943E+004/9.0314E+002(>)	1.3489E+004/9.2693E+002(>)	1.3029E+004/1.2066E+003(>)	1.4222E+004/6.3852E+002
fa_{24}	3.2670E+002/1.5103E+001(<)	3.0192E+002/1.4430E+001(<)	2.3981E+002/6.8972E+000(<)	2.3846E+002/1.0385E+001(<)	2.3379E+002/1.1879E+001(<)	2.3958E+002/8.3904E+000(<)	2.3151E+002/7.5471E+000
fa_{25}	6.1160E+002/2.6703E+001(<)	5.4202E+002/7.1608E+001(<)	3.9183E+002/9.8452E+000(<)	3.7650E+002/1.1055E+001(<)	3.7059E+002/1.2074E+001(<)	4.1381E+002/1.1798E+001(<)	3.6060E+002/1.1799E+001
fa_{26}	5.8459E+002/5.7564E+001(<)	4.7272E+002/7.3806E+001(<)	3.3138E+002/5.3041E+000(>)	3.2327E+002/5.6862E+000(>)	3.1630E+002/5.3031E+000(>)	3.5004E+002/7.6157E+000(<)	3.4313E+002/5.1142E+000
fa_{27}	2.5000E+003/8.2523E+002(<)	1.5901E+003/1.9602E+002(<)	7.3118E+002/1.0971E+002(>)	6.5416E+002/1.2749E+002(>)	5.7853E+002/1.2868E+002(>)	7.5354E+002/1.0104E+002(<)	7.5310E+002/7.7290E+001
fa_{28}	3.3770E+003/1.0121E+003(<)	2.9441E+003/7.1665E+002(<)	3.2547E+003/1.0107E+003(<)	2.7713E+003/7.2579E+002(<)	2.9533E+003/8.8931E+002(<)	2.5162E+003/2.5042E+001(>)	2.5770E+003/2.9758E+002
w/d/l	4/1/23	4/1/23	6/1/21	10/1/17	10/2/16	11/3/14	-/-/-

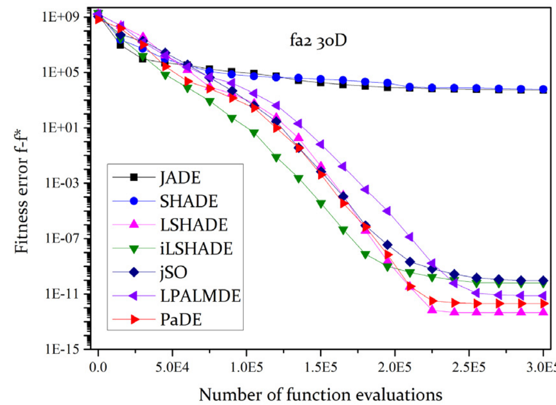
Table 5

Best, mean and standard deviation comparison under benchmarks fb_1–fb_{30} between the state-of-the-art DE variant jSO and the new proposed PaDE algorithm. All these values are selected/calculated from 51 runs with the maximum function evaluation equalling to $nfe_{\max} = 10000 \times D$. Symbols "$>$", "$=$" and "$<$" in the parentheses of the "Best" column are measured by the arithmetic values while symbols in the "Mean/Std" column are measured under Wilcoxon's signed rank test with a level of significance $\alpha = 0.05$, and the overall performances are summarized in last row of the table.

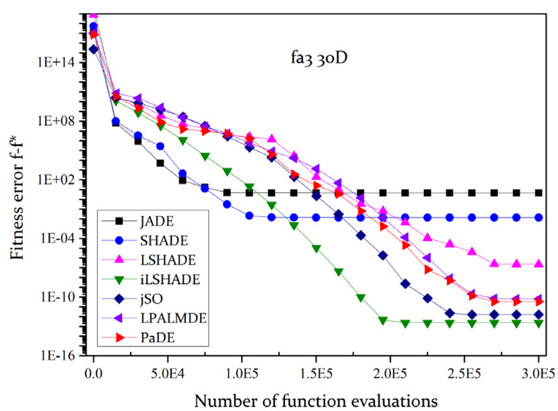
30D	Best		Mean/Std	
No.	jSO	PaDE	jSO	PaDE
fb_1	0(=)	0	4.1797E−015/6.5395E−015(<)	1.9505E−015/4.9388E−015
fb_2	0(=)	0	1.3932E−014/2.6887E−014(<)	1.2265E−014/1.9899E−014
fb_3	0(=)	0	5.2385E−014/1.5434E−014(<)	2.6750E−014/2.8656E−014
fb_4	5.8562E+001(<)	0	5.8562E+001/3.2700E−014(<)	3.1413E+001/3.0249E+001
fb_5	4.9748E+000(>)	6.2370E+000	8.6497E+000/2.0934E+000(>)	1.0591E+001/2.5426E+000
fb_6	0(=)	0	3.2367E−008/9.3835E−008(<)	2.7544E−009/1.9161E−008
fb_7	3.4600E+001(<)	3.4343E+001	3.9339E+001/1.8841E+000(<)	3.9318E+001/2.1670E+000
fb_8	4.9752E+000(>)	6.7005E+000	9.4583E+000/2.1279E+000(>)	1.1245E+001/2.3278E+000
fb_9	0(=)	0	0/0(=)	0/0
fb_{10}	9.6500E+002(>)	1.1165E+003	1.4808E+003/2.5423E+002(<)	1.4605E+003/3.4110E+002
fb_{11}	1.1369E−012(>)	1.9899E+000	7.1131E+000/1.6029E+001(>)	1.3315E+001/1.6171E+001
fb_{12}	1.1244E+001(>)	2.4819E+002	2.2700E+002/1.6507E+002(>)	9.0897E+002/3.3191E+002
fb_{13}	3.8406E+000(<)	2.9849E+000	1.7375E+001/2.6842E+000(<)	1.5433E+001/5.4733E+000
fb_{14}	2.0020E+001(<)	1.9899E+000	2.1897E+001/1.0315E+000(<)	2.1709E+001/6.8972E+000
fb_{15}	4.3465E−001(>)	4.4851E−001	1.2503E+000/9.0304E−001(>)	3.5652E+000/1.7714E+000
fb_{16}	1.3400E+001(>)	1.7522E+001	5.0255E+001/6.3397E+001(>)	1.7415E+002/9.8178E+001
fb_{17}	1.3509E+001(<)	1.2819E+000	3.1718E+001/7.5669E+000(>)	3.5069E+001/6.8571E+000
fb_{18}	4.6254E−001(>)	2.0302E+001	2.0419E+001/2.8814E+000(>)	2.2243E+001/1.1125E+000
fb_{19}	2.9251E+000(<)	2.4359E+000	5.0249E+000/1.7148E+000(<)	4.4831E+000/1.1755E+000
fb_{20}	1.3195E+001(>)	2.3239E+001	2.8954E+001/5.8323E+000(>)	4.2850E+001/1.8831E+001
fb_{21}	2.0580E+002(<)	2.0340E+002	2.1030E+002/1.8239E+000(<)	2.0966E+002/2.2975E+000
fb_{22}	1.0000E+002(=)	1.0000E+002	1.0000E+002/1.2158E−013(=)	1.0000E+002/1.0047E−013
fb_{23}	3.4976E+002(<)	3.3885E+002	3.5735E+002/3.3985E+000(<)	3.4679E+002/3.3955E+000
fb_{24}	4.2673E+002(<)	4.1361E+002	4.3131E+002/2.8783E+000(<)	4.2171E+002/2.9981E+000
fb_{25}	3.8669E+002(>)	3.8671E+002	3.8669E+002/5.1663E−003(>)	3.8681E+002/4.8595E−002
fb_{26}	8.7006E+002(<)	8.1172E+002	9.9050E+002/4.5187E+001(<)	9.1365E+002/4.6710E+001
fb_{27}	4.9150E+002(>)	4.9410E+002	5.0229E+002/5.7289E+000(>)	5.0579E+002/4.3393E+000
fb_{28}	3.0000E+002(=)	3.0000E+002	3.1341E+002/3.7087E+001(<)	3.0852E+002/2.9535E+001
fb_{29}	3.4949E+002(<)	3.3952E+002	4.3420E+002/1.5194E+001(<)	4.3130E+002/1.2676E+001
fb_{30}	1.9417E+003(>)	1.9435E+003	1.9694E+003/1.7286E+001(>)	2.0461E+003/5.8612E+001
w/d/l	12/7/11	-/-/-	12/2/16	-/-/-



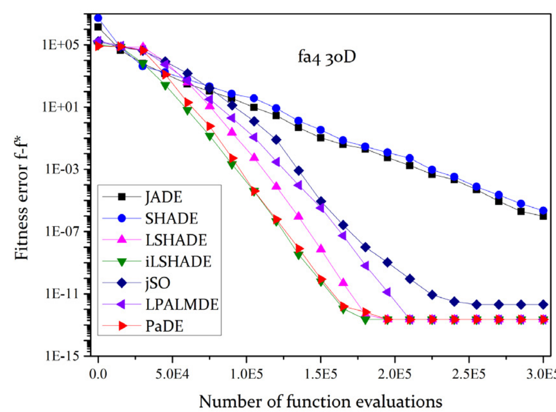
(a) Convergence comparison on benchmark f_1



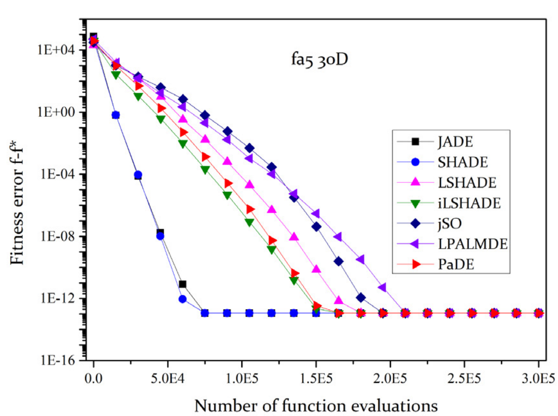
(b) Convergence comparison on benchmark f_2



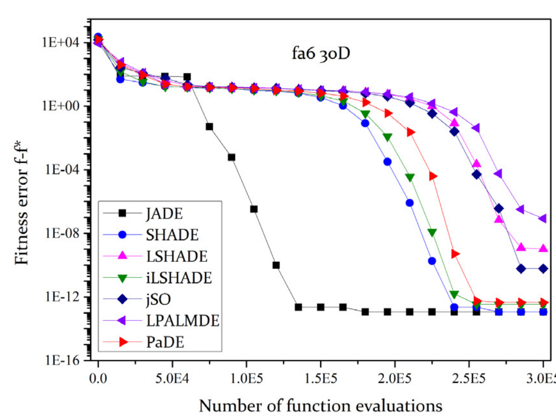
(c) Convergence comparison on benchmark f_3



(d) Convergence comparison on benchmark f_4

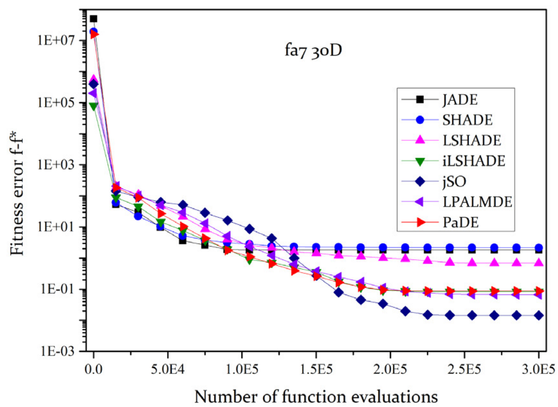


(e) Convergence comparison on benchmark f_5

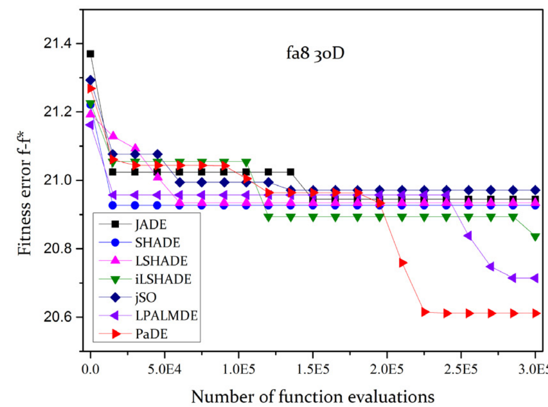


(f) Convergence comparison on benchmark f_6

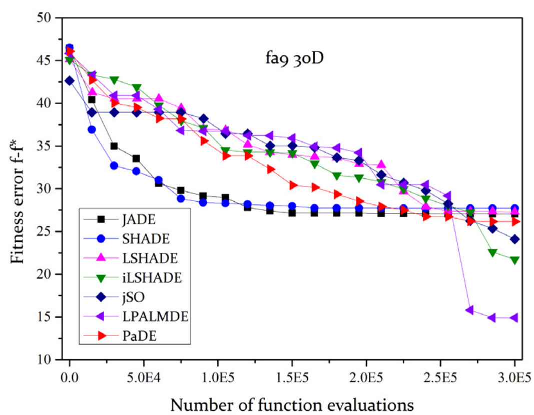
Fig. 6. Here presents the convergence speed comparison by employing the median value of 51 runs obtained by each algorithm on 30-D optimization. There are total 28 comparison figures and the first 6 figures are presented here.



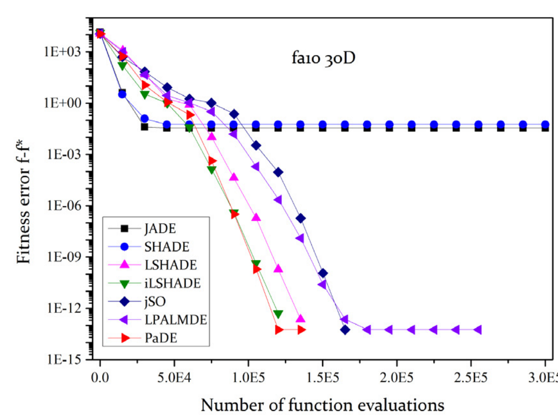
(a) Convergence comparison on benchmark f_7



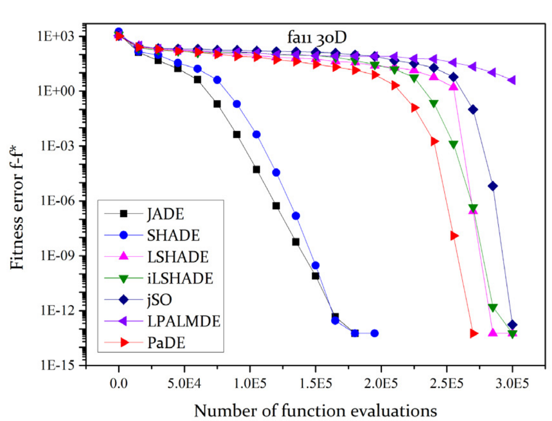
(b) Convergence comparison on benchmark f_8



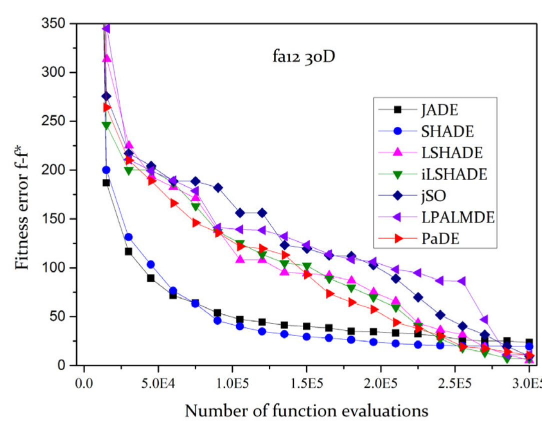
(c) Convergence comparison on benchmark f_9



(d) Convergence comparison on benchmark f_{10}

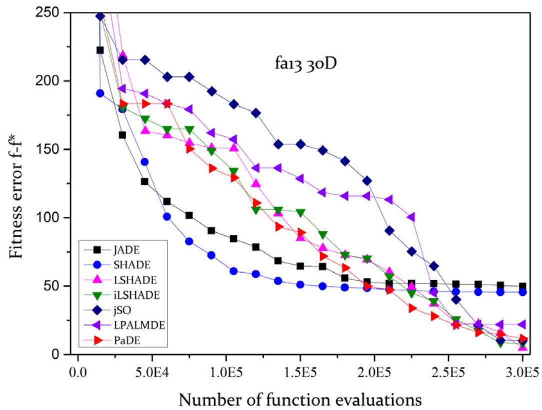


(e) Convergence comparison on benchmark f_{11}

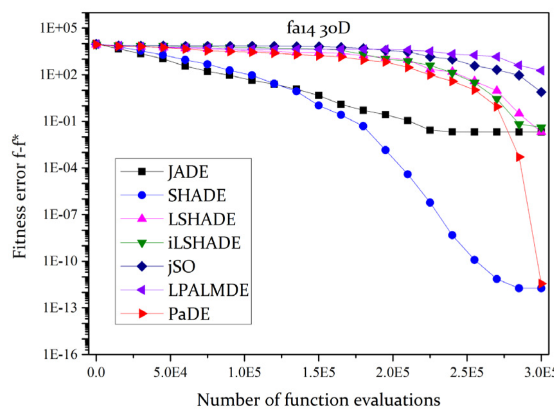


(f) Convergence comparison on benchmark f_{12}

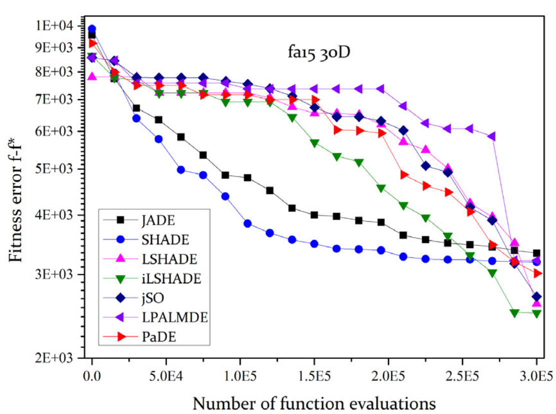
Fig. 7. Continued from Fig. 6, comparisons on f_7–f_{12} are presented here.



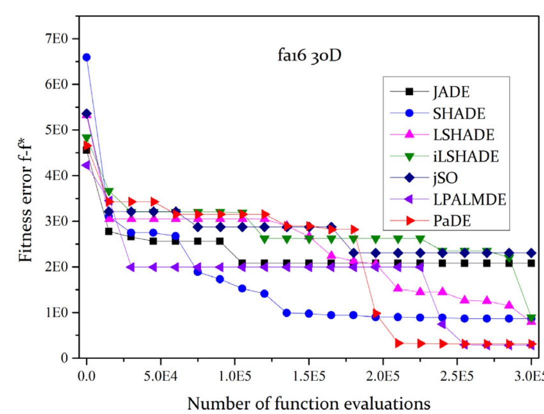
(a) Convergence comparison on benchmark f_{13}



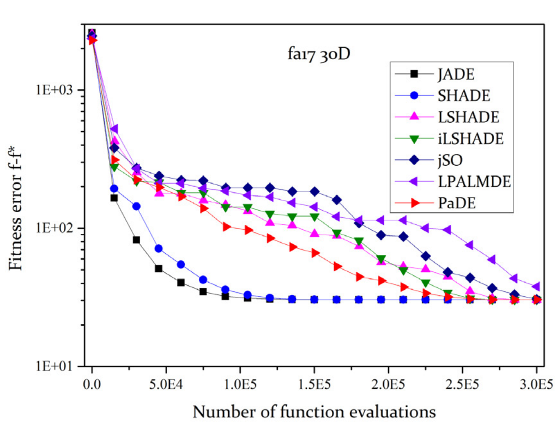
(b) Convergence comparison on benchmark f_{14}



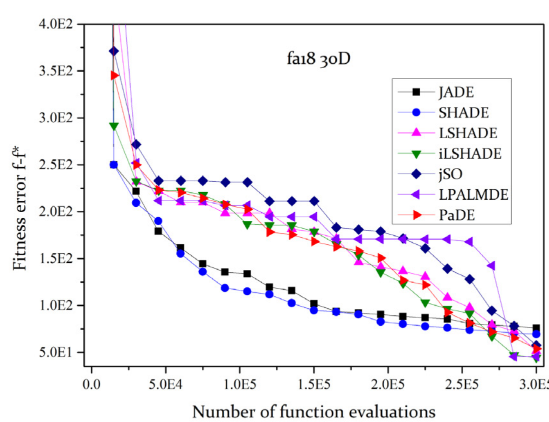
(c) Convergence comparison on benchmark f_{15}



(d) Convergence comparison on benchmark f_{16}

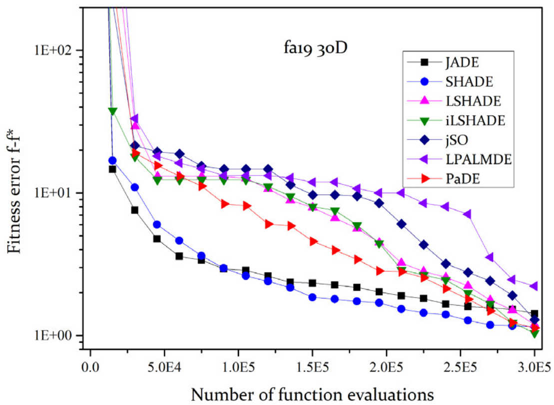


(e) Convergence comparison on benchmark f_{17}

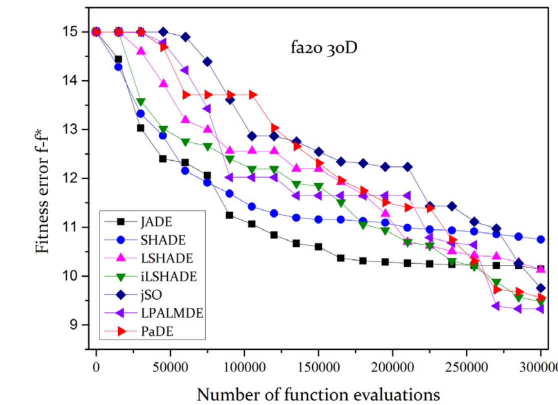


(f) Convergence comparison on benchmark f_{18}

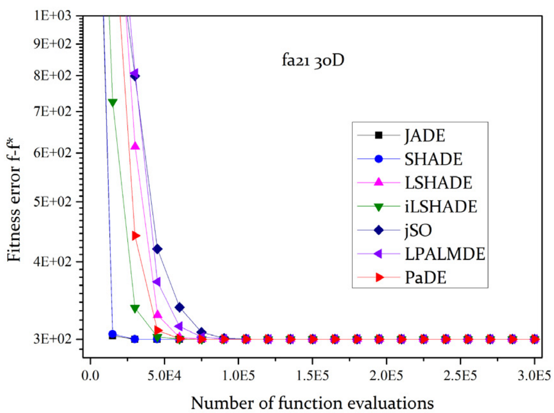
Fig. 8. Continued from Fig. 7, comparisons on f_{13}–f_{18} are presented here.



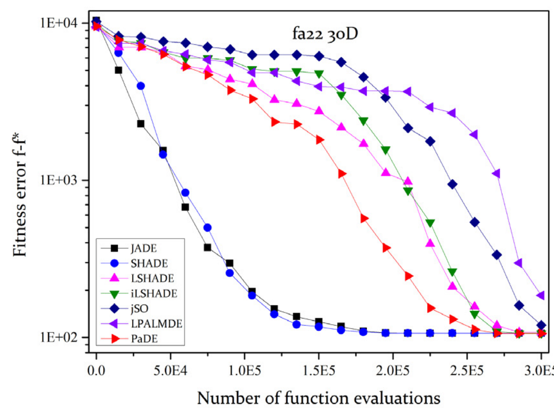
(a) Convergence comparison on benchmark f_{19}



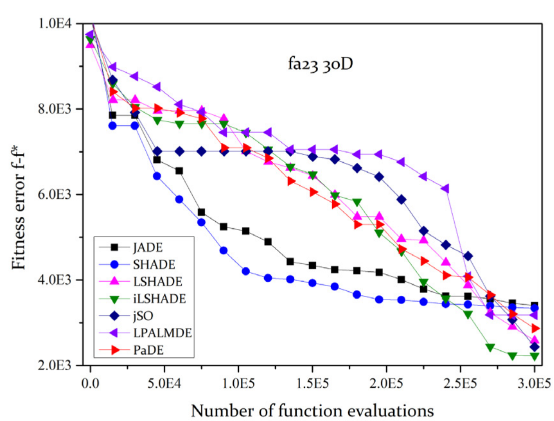
(b) Convergence comparison on benchmark f_{20}



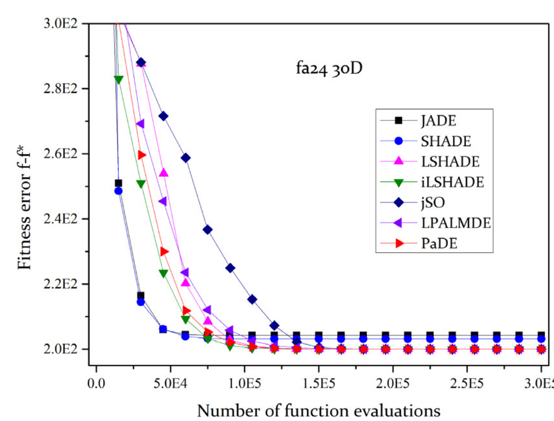
(c) Convergence comparison on benchmark f_{21}



(d) Convergence comparison on benchmark f_{22}



(e) Convergence comparison on benchmark f_{23}



(f) Convergence comparison on benchmark f_{24}

Fig. 9. Continued from Fig. 8, comparisons on f_{19}–f_{24} are presented here.

fitness errors obtained by all these algorithms on each benchmark is selected out for such comparison. Fig. 6 presents the first part containing 6 figures, fa_1–fa_6, of the total 28-figure comparison, Fig. 7 presents the second part also containing 6 figures, fa_7–fa_{12}, of the total comparison, Fig. 8 presents the third part 6 figures from f_{13} to f_{18}, Fig. 9 presents the fourth part 6 figures from f_{19} to f_{24} and Fig. 10 presents the last part 4 figures from f_{25} to f_{28}. From all these convergence figures, we can see that the proposed PaDE algorithm outperforms JADE on fa_2–fa_4, fa_7–fa_{10}, fa_{12}–fa_{16}, fa_{20}, fa_{21}, and fa_{24}–fa_{28}; it also outperforms SHADE on fa_2–fa_4, fa_7–fa_{10}, fa_{12}, fa_{13}, fa_{15}, fa_{16}, fa_{18}, fa_{20}, fa_{21}, and fa_{23}–fa_{28}; it also outperforms LSHADE on fa_3–fa_{11}, fa_{14}, fa_{16}, fa_{17}, fa_{19}–fa_{22}, and fa_{24}–fa_{28}; it also outperforms iLSHADE on fa_2, fa_7, fa_8, fa_{10}, fa_{11}, fa_{14}, fa_{16}, fa_{17}, fa_{21}, fa_{22}, fa_{24}–fa_{26}, and fa_{28}; it also outperforms jSO on fa_1, fa_2, fa_4– fa_6, fa_8, fa_{10}–fa_{12}, fa_{14}, fa_{16}–fa_{19}, fa_{21}, fa_{22}, and fa_{24}–fa_{28}; it also outperforms LPALMDE on fa_1–fa_6, fa_8, fa_{10}, fa_{11}, fa_{13}–fa_{17}, fa_{19}, fa_{21}–fa_{24} and fa_{25}–fa_{28}. Furthermore, the proposed PaDE algorithm outperforms all the other algorithms on fa_4, fa_8, fa_{10}, fa_{16}, fa_{17}, fa_{21}, fa_{24}, fa_{26}–fa_{28}. As a result, the proposed PaDE is also competitive with other state-of-the-art DE variants on convergence speed perspective.

The proposed PaDE algorithm is also contrasted with jSO under CEC2017 test suite on 30D optimization. Benchmark functions in CEC2017 test suite are mapped into our test suite and labelled as fb_1–fb_{30}. The "Best", and "Mean/std" values are selected/calculated for algorithm evaluation, all of which are presented in Table 5.

All these values are obtained from 51 runs under each benchmark with the maximum function evaluations equalling to $nfe_{\max} = 10000 \times D$. Symbols "$>$", "$=$" and "$<$" in the parentheses of the "Best" column are measured by the arithmetic values while symbols in the "Mean/std" column are measured under Wilcoxon's signed rank test with a level of significance $\alpha = 0.05$. The overall performances are summarized in last row of the table, "w", "d" and "l" denote "win", "draw" and "lose" respectively. We can see from the table that the new proposed PaDE and jSO can find the global optima during the 51-run on benchmarks fb_1–fb_3, fb_6 and fb_9. The proposed PaDE algorithm can also find the global optimum during the 51-run on benchmark fb_4. Both the new proposed PaDE and jSO can find tier best on benchmarks fb_{22} and fb_{28} besides the benchmarks that the two algorithms can find the global optima. From the "Mean/Std" perspective, both the new proposed PaDE and jSO perform equally well on fb_9 and fb_{22}. To summarize, the new proposed PaDE algorithm obtained better or similar improvements in 19 out of 30 benchmark functions from the "Best" perspective, and it also obtained better or similar improvements in 18 out of 30 benchmark functions from the "Mean/std" perspective.

Furthermore, by comparing the performance of jSO algorithm in Tables 3 and 5, we can see that jSO algorithm performs better under the fb_1–fb_{30} (CEC2017) benchmarks in comparison with our new proposed PaDE algorithm than that under the fa_1–fa_{28} (CEC2013) benchmarks. As is known to all that the jSO algorithm involved some brutal changes of control parameters, these parameter adaptation schemes may be over-tuned under the CEC2017 test suite, therefore, an initiative to use more benchmarks or more than one Congress on Evolutionary Computation (CEC) Competition test suites for algorithm evaluation is advanced for evolutionary algorithms' evaluation. That is also the reason why we employ two test suites, CEC2013 and CEC2017, containing 58 benchmark functions for the evaluations of the new proposed PaDE algorithm, and from the obtained optimization results we can see that the proposed PaDE algorithm is competitive with the other state-of-the-art DE variants.

4.4. Discussion on parabolic population size reduction scheme

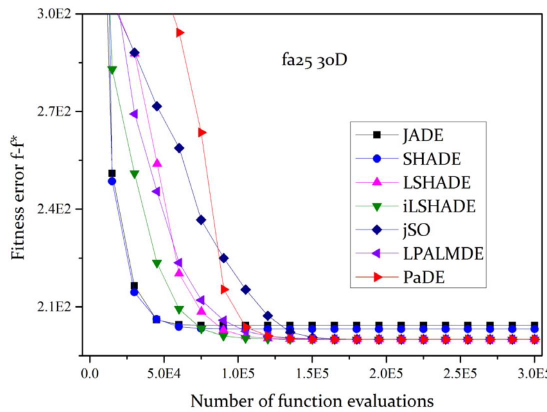
In this part we mainly discuss the new proposed parabolic population size reduction scheme, the pivot based population size reduction scheme and the former proposed linear population size reduction scheme. As is known to all that the CEC Competitions employed the fixed maximum number of function evaluations, $nfe_{\max}$ equalling to 10000·D, in algorithm evaluation. Therefore, an algorithm employing a larger population size means that fewer generations are available during the whole evolution. Fig. 11 presents an example to illustrate the generation number comparison under larger fixed ps ($ps = 25\log(D)\sqrt{(D)}$), smaller fixed ps ($ps = 100$), linear reduction scheme of ps (in Eq. (24)) and the novel parabolic reduction scheme of ps(in Eq. (23)) on 30-D optimization. Generally speaking, a certain DE variant with a large population size usually can obtain a better perception of the whole landscape of a target objective while more generations during the evolution usually help the DE variant search more deeply of some potential local areas. Furthermore, employing larger population size in each generation during the evolution will definitely limit the exploitation capacity of an optimization algorithm while employing a smaller population size usually associates with falling into some local optima. This is a dilemma, therefore, when fixed maximum number of function evaluation are employed, we need to balance the population size during each generation and the total generations available during the evolution.

Recent research in the literature recommended a relative larger initial population size with a reduction scheme during evolution, and the linear population size reduction scheme proposed in LSHADE [10] was proven to be an excellent reduction scheme for many optimization problems. Nevertheless, we see that the quick reduction of population size at the beginning of the evolution usually leads to a bad perception of the landscape of some objective functions, therefore, we proposed the parabolic population size reduction scheme to maintain a relative larger population size at the front part of the evolution, and we also proposed a pivot scheme to make a balance between the parabolic population size reduction scheme and the linear population size reduction scheme. The optimization performance comparisons under CEC2013 test suite on 30-D optimization of all the three PaDE variants are given in Table 6.

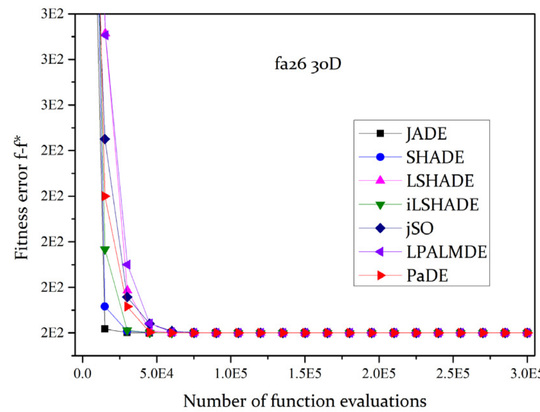
We can see from the table that the default PaDE (PaDE-Pivot) outperforms PaDE-Linear on 9 benchmarks out of the total 28 benchmarks from "Best" perspective; it outperforms PaDE-Linear on 11 out of 28 benchmarks from "Median" perspective; and it also outperforms PaDE-Linear on 16 out of 28 benchmarks from the "Mean/Std" perspective. The comparison results in the table also reveal that the default PaDE outperforms PaDE-Para on 10 out of 28 benchmarks from "Best" perspective, on 14 out of 28 benchmarks from "Median" perspective, and on 17 out of 28 benchmarks from "Best/Std" perspective. Furthermore, the PaDE-Para performs extremely well on some of the unimodal benchmarks, e.g. f_1–f_5 and some of the multimodal benchmarks, e.g. f_{10}. To summarize, the novel parabolic population size reduction scheme in the front part of the evolution was meaningful and effective, and the proposed PaDE algorithm, the default one, obtains an overall better performance in comparison with the other two PaDE variants including PaDE-Linear and PaDE-Para.

4.5. Discussion on the mutation strategy in PaDE algorithm

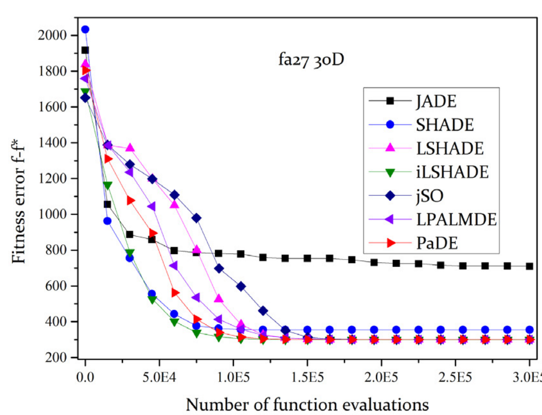
The mutation strategy with time stamp scheme in PaDE algorithm can be considered as a balance between mutation strategy with and without external archive. When the time stamp threshold T_0 in Eq. (27) equals to $+\infty$, $T_0 = +\infty$, or the decay rate r^d in Eq. (27) equals to 0, $r^d = 0$, the mutation strategy is degraded into the mutation strategy with external archive, which is the same as the mutation strategy with archive in JADE algorithm. When the threshold T_0 equals to 0, $T_0 = 0$, or the decay rate r^d equals to T_0, then mutation strategy in PaDE algorithm is degenerated into mutation strategy without external archive which have already mentioned in JADE [8] and LPALMDE [5]. Furthermore, the decay rate can also be dynamically decreased in PaDE algorithm according to the ratio of better solutions found by the trial vector during



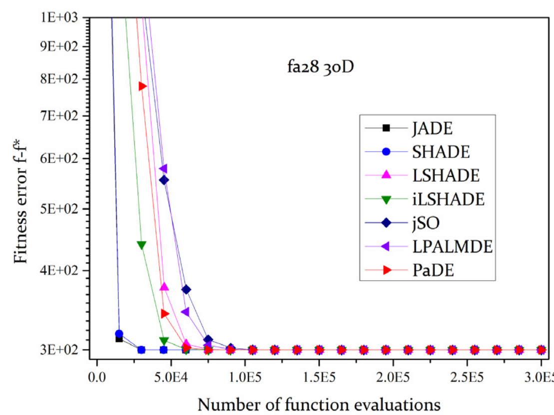
(a) Convergence comparison on benchmark f_{25}



(b) Convergence comparison on benchmark f_{26}



(c) Convergence comparison on benchmark f_{27}



(d) Convergence comparison on benchmark f_{28}

Fig. 10. Continued from Fig. 9, comparisons on the last 4 benchmarks are presented here.

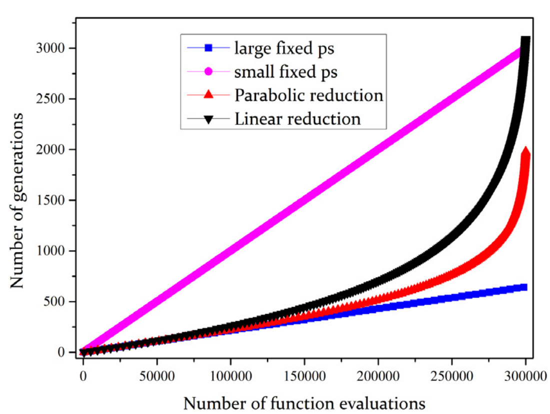


Fig. 11. Illustration of the number of generations between the novel parabolic approach and linear approach.

the evolution. However, for simplicity, we employ the constant value $r^d = 0.04$ as default in the novel PaDE algorithm.

The optimization performance comparisons among “PaDE $T_0 = 0$”, “PaDE $T_0 = +\infty$”, and the default PaDE are given in Table 7.

From the table we can see that the proposed PaDE algorithm outperforms “PaDE $T_0 = 0$” on 14 out of 28 benchmarks from “Best” perspective, on 14 out of 28 benchmarks from “Median” perspective, and on 16 out of 28 benchmarks from “Mean/Std” perspective under CEC2013 test suit. The comparison also reveals that the proposed PaDE algorithm outperforms “PaDE $T_0 = +\infty$” on 8 out of 28 benchmarks from “Best” perspective, on 15 out of 28 benchmarks from “Median” perspective, and on 15 out of 28 benchmarks from “Mean/Std” perspective under CEC2013 test suite. As a result, the mutation strategy with time stamp mechanism proposed in our PaDE algorithm is competitive with other mutation strategies in the literature.

5. Conclusion

DE variants are powerful population-based stochastic algorithms for many complex optimization problems, however, there are still some weaknesses existing in these state-of-the-art DE variants, and these weaknesses can be summarized into two categories: one is the weakness in parameters control, the other is weakness in mutation strategy. Here in this paper, a new Parameter adaptation DE variant called PaDE is proposed to tackle the weaknesses both in parameters control and in mutation strategy. In the PaDE algorithm, a novel grouping strategy with adaptation schemes for control parameters F and Cr is proposed for the tackling improper adaptation schemes in some DE variants, and a novel parabolic population size reduction scheme is proposed to tackle the weakness in linear population size reduction scheme. Furthermore, an enhanced time stamp based mutation strategy is also advanced to avoid too old and harmful inferior solution being archive-residents in comparison with JADE variants as well

Table 6
Minimum (Best), Median, Mean/Std of fitness error Δf^* over 51 runs under each benchmark on 30-D optimization are presented here for comparisons. ">", "=" and "<" denote better performance, similar performance and worse performance in comparison with the default PaDE algorithm (PaDE-Pivot).

30D	PaDE-Linear			PaDE-Para			Default PaDE (PaDE-Pivot)		
No.	Best	Median	Mean/Std	Best	Median	Mean/Std	Best	Median	Mean/Std
1	0(=)	0(=)	0/0(=)	0(=)	0(=)	0/0(=)	0	0	0/0
2	4.5475E−013(<)	7.2760E−012(<)	2.0852E−010/5.9378E−010(<)	0 (>)	2.2737E−013(>)	3.2991E−013/2.6638E−013(>)	2.2737E−013	2.0464E−012	4.2688E−011/2.1931E−010
3	4.5475E−013(<)	7.3585E−009(<)	4.6737E−002/3.2926E−001(<)	0 (>)	6.8212E−013(<)	3.4162E−004/2.1891E−003(>)	2.2737E−013	3.3651E−011	2.5361E−004/1.2253E−003
4	0 (=)	2.2737E−013(=)	2.5412E−013/1.5520E−013(<)	0 (=)	0 (>)	1.1146E−013/1.1480E−013(>)	0	2.2737E−013	2.1846E−013/1.0129E−013
5	1.1369E−013(<)	1.1369E−013(=)	1.1369E−013/0 (<)	1.1369E−013(<)	1.1369E−013(=)	1.1369E−013/0 (<)	0	1.1369E−013	1.1146E−013/1.5919E−014
6	1.1369E−013(=)	4.5475E−013(>)	6.1529E−009/4.3629E−008(>)	2.2737E−013(<)	1.5348E−011(<)	1.1729E−005/8.2960E−005(<)	1.1369E−013	4.3201E−012	1.3097E−006/9.2577E−006
7	7.4832E−005(<)	9.7086E−002(<)	2.1178E−001/4.6163E−001(<)	2.6257E−004(<)	2.4787E−002(>)	8.1695E−002/1.2973E−001(>)	5.5242E−005	8.7066E−002	1.2669E−001/1.4861E−001
8	2.0345E+001(>)	2.0662E+001(>)	2.0701E+001/2.2232E−001(>)	2.0494E+001(>)	2.0871E+001(<)	2.0842E+001/1.2433E−001(<)	2.0553E+001	2.0826E+001	2.0818E+001/1.4319E−001
9	1.5820E+001(>)	2.6254E+001(<)	2.5860E+001/2.1793E+000(<)	2.0564E+001(>)	2.5821E+001(>)	2.5645E+001/1.3253E+000(<)	2.2064E+001	2.5968E+001	2.5498E+001/1.4712E+000
10	0 (=)	7.3960E−003(=)	5.4119E−003/5.4435E−003(<)	0 (=)	0 (>)	4.2532E−003/4.8315E−003(>)	0	7.3960E−003	5.0749E−003/5.0128E−003
11	0 (=)	0 (>)	2.3406E−014/2.8254E−014(>)	0 (=)	5.6843E−014(=)	6.6875E−014/2.4662E−014(<)	0	5.6843E−014	3.5666E−014/2.7756E−014
12	5.4112E+000(<)	8.3798E+000(<)	8.2911E+000/1.5252E+000(<)	6.7497E+000(<)	1.0225E+001(<)	1.0237E+001/1.5067E+000(<)	5.3671E+000	8.3481E+000	8.1114E+000/1.4309E+000
13	3.8770E+000(<)	1.1682E+001(<)	1.1782E+001/4.3688E+000(<)	5.9237E+000(<)	1.3540E+001(<)	1.3873E+001/4.3016E+000(<)	3.1393E+000	9.5692E+000	9.8167E+000/4.3167E+000
14	1.8190E−012(=)	3.6380E−012(>)	1.2247E−002/1.5115E−002(>)	1.1360E−006(<)	5.9252E−002(<)	5.4526E−002/3.2490E−002(<)	1.8190E−012	2.0819E−002	2.5718E−002/2.1063E−002
15	2.1435E+003(>)	2.8579E+003(<)	2.9057E+003/3.3397E+002(<)	1.9352E+003(=)	3.1481E+003(<)	3.0615E+003/3.0184E+002(<)	2.1979E+003	2.8231E+003	2.8145E+003/3.1207E+002
16	4.4689E−002(>)	3.1321E−001(>)	4.6304E−001/3.6419E−001(>)	2.1607E−001(<)	9.1804E−001(<)	9.0116E−001/3.7612E−001(<)	1.1345E−001	6.4598E−001	6.3143E−001/3.0285E−001
17	3.0434E+001(=)	3.0434E+001(=)	3.0434E+001/4.1577E−014(=)	3.0434E+001(=)	3.0434E+001(=)	3.0434E+001/1.6718E−005(=)	3.0434E+001	3.0434E+001	3.0434E+001/9.4299E−007
18	3.8957E+001(>)	5.3960E+001(>)	5.3582E+001/5.7683E+000(<)	4.1511E+001(>)	6.5407E+001(<)	6.4391E+001/6.5837E+000(<)	4.6473E+001	5.3968E+001	5.5348E+001/5.0973E+000
19	8.9388E−001(>)	1.1306E+000(>)	1.1105E+000/8.4557E−002(>)	1.0669E+000(<)	1.3959E+000(<)	1.3751E+000/1.0803E−001(<)	9.2167E−001	1.1334E+000	1.1203E+000/8.7538E−002
20	8.2796E+000(<)	9.5611E+000(<)	1.0693E+001/2.0732E+000(<)	8.0989E+000(>)	9.5904E+000(<)	1.0560E+001/1.9444E+000(<)	8.2481E+000	9.4231E+000	1.0359E+001/1.8654E+000
21	2.0000E+002(=)	3.0000E+002(=)	2.9412E+002/2.3764E+001(>)	2.0000E+002(=)	3.0000E+002(=)	3.0171E+002/3.4946E+001(<)	2.0000E+002	3.0000E+002	2.9608E+002/1.9604E+001
22	1.0500E+002(=)	1.0598E+002(=)	1.0592E+002/3.2534E−001(>)	1.0541E+002(<)	1.0611E+002(<)	1.0651E+002/1.1517E+000(<)	1.0500E+002	1.0598E+002	1.0632E+002/1.0848E+000
23	2.3082E+003(<)	2.8698E+003(<)	2.8816E+003/3.0672E+002(<)	1.9507E+003(<)	2.9743E+003(<)	2.9985E+003/2.9277E+002(<)	1.9176E+003	2.7602E+003	2.7682E+003/2.9282E+002
24	2.0000E+002(=)	2.0004E+002(=)	2.0007E+002/9.4364E−002(<)	2.0000E+002(=)	2.0001E+002(>)	2.0004E+002/8.4725E−002(=)	2.0000E+002	2.0002E+002	2.0004E+002/5.6801E−002
25	2.0000E+002(=)	2.3682E+002(<)	2.2249E+002/2.2456E+001(<)	2.0000E+002(=)	2.0001E+002(=)	2.1803E+002/2.1900E+001(<)	2.0000E+002	2.0001E+002	2.1450E+002/2.0867E+001
26	2.0000E+002(=)	2.0000E+002(=)	2.0000E+002/0 (=)	2.0000E+002(=)	2.0000E+002(=)	2.0000E+002/0 (=)	2.0000E+002	2.0000E+002	2.0000E+002/0
28	3.0000E+002(=)	3.0000E+002(=)	3.0000E+002/0 (=)	3.0000E+002(=)	3.0000E+002(=)	3.0000E+002/0 (=)	3.0000E+002	3.0000E+002	3.0000E+002/0
> /= /<	6/13/9	7/10/11	8/4/16	7/11/10	6/8/14	6/5/17	-/-/-	-/-/-	-/-/-

Table 7
Minimum, Median, Mean/Std of fitness error Δf^* over 51 runs under each benchmark on 30-D optimization are presented here for comparisons. ">", "=" and "<" denote better performance, similar performance and worse performance in comparison with the default PaDE algorithm (PaDE-Pivot).

30D	PaDE $T0 = 0$			PaDE $T0 = +\infty$			Default PaDE (PaDE-Pivot)		
No.	Best	Median	Mean/Std	Best	Median	Mean/Std	Best	Median	Mean/Std
1	0 (=)	0 (=)	4.4583E−015/3.1839E−014(<)	0 (=)	0 (=)	0/0 (=)	0	0	0/0
2	4.5727E−008(<)	2.3980E−004(<)	3.4186E−002/2.0554E−001(<)	2.2737E−013(=)	4.0927E−012(<)	9.6746E−009/4.3206E−008(<)	2.2737E−013	2.0464E−012	4.2688E−011/2.1931E−010
3	4.5475E−013(<)	2.9287E−007(<)	8.9815E−004/3.3482E−003(<)	2.2737E−013(=)	8.8016E−010(<)	5.0651E−003/2.0540E−002(<)	2.2737E−013	3.3651E−011	2.5361E−004/1.2253E−003
4	3.1832E−012(<)	8.2991E−011(<)	2.7533E−009/9.4675E−009(<)	0 (=)	2.2737E−013(=)	2.7196E−013/1.8201E−013(<)	0	2.2737E−013	2.1846E−013/1.0129E−013
5	1.1369E−013(<)	1.1369E−013(=)	1.1815E−013/2.2287E−014(<)	1.1369E−013(<)	1.1369E−013(=)	1.1369E−013/0 (<)	0	1.1369E−013	1.1146E−013/1.5919E−014
6	6.8212E−013(<)	4.7324E−001(<)	5.5945E−001/4.3154E−001(<)	1.1369E−013(=)	5.6843E−013(<)	5.8517E−009/4.1635E−008(>)	1.1369E−013	4.3201E−012	1.3097E−006/9.2577E−006
7	3.0504E−003(<)	1.9703E−001(<)	2.7136E−001/2.4918E−001(<)	4.5239E−005(>)	1.0532E−001(<)	1.8128E−001/2.2492E−001(<)	5.5242E−005	8.7066E−002	1.2669E−001/1.4861E−001
8	2.0384E+001(>)	2.0738E+001(>)	2.0711E+001/2.1779E−001(>)	2.0331E+001(>)	2.0843E+001(<)	2.0801E+001/1.5737E−001(>)	2.0553E+001	2.0826E+001	2.0818E+001/1.4319E−001
9	1.1978E+001(>)	2.5517E+001(>)	2.4921E+001/3.2340E+000(>)	2.1746E+001(>)	2.5986E+001(<)	2.5877E+001/1.1722E+000(<)	2.2064E+001	2.5968E+001	2.5498E+001/1.4712E+000
10	5.6843E−014(<)	1.7236E−002(<)	1.9029E−002/1.1209E−002(<)	0 (=)	7.3960E−003(=)	5.8459E−003/5.9110E−003(<)	0	7.3960E−003	5.0749E−003/5.0128E−003
11	0 (=)	5.6843E−014(=)	4.0125E−014/2.6158E−014(<)	0 (=)	5.6843E−014(=)	3.0094E−014/2.8655E−014(>)	0	5.6843E−014	3.5666E−014/2.7756E−014
12	6.2972E+000(<)	1.0513E+001(<)	1.0549E+001/1.4271E+000(<)	4.0523E+000(>)	8.3950E+000(<)	8.2790E+000/1.6984E+000(<)	5.3671E+000	8.3481E+000	8.1114E+000/1.4309E+000
13	8.9228E+000(<)	1.8638E+001(<)	1.8857E+001/4.9515E+000(<)	4.3889E+000(<)	1.1705E+001(<)	1.2407E+001/5.3045E+000(<)	3.1393E+000	9.5692E+000	9.8167E+000/4.3167E+000
14	1.8190E−012(=)	2.0819E−002(=)	2.2044E−002/2.1800E−002(>)	1.8190E−012(=)	2.0819E−002(=)	1.9595E−002/1.8812E−002(>)	1.8190E−012	2.0819E−002	2.5718E−002/2.1063E−002
15	2.2074E+003(<)	2.8168E+003(>)	2.7909E+003/2.5360E+002(>)	1.7252E+003(>)	2.9348E+003(<)	2.9150E+003/3.3827E+002(>)	2.1979E+003	2.8231E+003	2.8145E+003/3.1207E+002
16	4.9361E−002(>)	3.1886E−001(>)	4.5840E−001/3.6093E−001(>)	1.1883E−001(<)	8.3053E−001(<)	7.3456E−001/3.5730E−001(<)	1.1345E−001	6.4598E−001	6.3143E−001/3.0285E−001
17	3.0434E+001(=)	3.0434E+001(=)	3.0434E+001/9.4299E−007(=)	3.0434E+001(=)	3.0434E+001(=)	3.0434E+001/4.7545E−014(=)	3.0434E+001	3.0434E+001	3.0434E+001/9.4299E−007
18	3.8120E+001(>)	5.4951E+001(<)	5.4496E+001/4.4483E+000(>)	4.1842E+001(>)	5.4849E+001(<)	5.4711E+001/4.1672E+000(>)	4.6473E+001	5.3968E+001	5.5348E+001/5.0973E+000
19	9.7905E−001(<)	1.1103E+000(>)	1.1107E+000/7.0340E−002(>)	9.2945E−001(<)	1.1289E+000(>)	1.1232E+000/8.4581E−002(<)	9.2167E−001	1.1334E+000	1.1203E+000/8.7538E−002
20	8.2263E+000(>)	9.6101E+000(<)	1.0719E+001/1.9884E+000(<)	8.3989E+000(<)	9.3893E+000(>)	1.0277E+001/1.9734E+000(>)	8.2481E+000	9.4231E+000	1.0359E+001/1.8654E+000
21	2.0000E+002(=)	3.0000E+002(=)	2.9975E+002/3.7738E+001(<)	2.0000E+002(=)	3.0000E+002(=)	2.9975E+002/3.7738E+001(<)	2.0000E+002	3.0000E+002	2.9608E+002/1.9604E+001
22	1.0494E+002(>)	1.0598E+002(=)	1.0601E+002/2.9419E−001(>)	1.0451E+002(<)	1.0598E+002(=)	1.0578E+002/4.2161E−001(>)	1.0500E+002	1.0598E+002	1.0632E+002/1.0848E+000
23	1.7426E+003(>)	2.8426E+003(<)	2.8214E+003/3.5949E+002(<)	2.0936E+003(<)	2.8116E+003(<)	2.7945E+003/3.5447E+002(<)	1.9176E+003	2.7602E+003	2.7682E+003/2.9282E+002
24	2.0005E+002(<)	2.0047E+002(<)	2.0067E+002/6.2222E−001(<)	2.0000E+002(=)	2.0003E+002(<)	2.0006E+002/7.7840E−002(<)	2.0000E+002	2.0002E+002	2.0004E+002/5.6801E−002
25	2.0001E+002(<)	2.0015E+002(<)	2.0822E+002/1.7556E+001(>)	2.0000E+002(=)	2.0011E+002(<)	2.1915E+002/2.2402E+001(>)	2.0000E+002	2.0001E+002	2.1450E+002/2.0867E+001
26	2.0000E+002(=)	2.0000E+002(=)	2.0000E+002/1.2107E−008(=)	2.0000E+002(=)	2.0000E+002(=)	2.0000E+002/1.7953E−013(=)	2.0000E+002	2.0000E+002	2.0000E+002/0
27	3.0066E+002(<)	3.0660E+002(<)	3.0866E+002/7.8257E+000(<)	3.0002E+002(<)	3.0150E+002(<)	3.0339E+002/5.3615E+000(<)	3.0001E+002	3.0021E+002	3.0148E+002/2.9330E+000
28	3.0000E+002(=)	3.0000E+002(=)	3.0000E+002/0 (=)	3.0000E+002(=)	3.0000E+002(=)	3.0000E+002/6.4311E−014(=)	3.0000E+002	3.0000E+002	3.0000E+002/0
> /= /<	7/7/14	5/9/14	9/3/16	6/14/8	2/11/15	9/4/15	-/-/-	-/-/-	-/-/-

as to enhance the diversity of external individuals in comparison with PALMDE variants. An initiative is also proposed in the paper that the evaluation of evolutionary algorithms should be verified under more benchmarks or more than one test suites to avoid the over-fitting problem, and consequently, our proposed PaDE algorithm is measured under two test suites, CEC2013 test suite and CEC2017 test suite containing 58 benchmark functions. The experiment results show that the new proposed PaDE algorithm is competitive with these state-of-the-art DE variants both on optimization accuracy and on convergence speed.

References

[1] K.V. Price, Differential evolution vs. the functions of the 2nd ICEO, in: Evolutionary Computation, 1997., IEEE International Conference on, Indianapolis, IN, 1997, pp. 153–157.

[2] S. Wright, J. Nocedal, Numerical optimization, Springer Sci. 35 (67–68) (1999) 7.

[3] Z. Meng, J.-S. Pan, Monkey King Evolution: A new memetic evolutionary algorithm and its application in vehicle fuel consumption optimization, Knowl.-Based Syst. 97c (2016) 144–157.

[4] J. Liang, B. Qu, P. Suganthan, A.G. Hernández-Díaz, Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization, in: Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report, Vol. 201212, 2013.

[5] Z. Meng, J.-S. Pan, L. Kong, Parameters with Adaptive Learning Mechanism (PALM) for the enhancement of differential evolution, Knowl.-Based Syst. 141 (2018) 92–112.

[6] R. Storn, K. Price, Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces, Berkeley, CA: Int. Comput. Sci. Inst. (1995).

[7] K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: a Practical Approach to Global Optimization, Springer Science & Business Media, 2006.

[8] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.

[9] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for Differential Evolution, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 71–78.

[10] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 1658–1665.

[11] J. Brest, M.S. Maučec, B. Bošković, iL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization, in: Evolutionary Computation (CEC), 2016 IEEE Congress on, IEEE, 2016, pp. 1188–1195.

[12] J. Brest, M.S. Maučec, B. Bošković, Single objective real-parameter optimization: Algorithm jSO, in: Evolutionary Computation (CEC), 2017 IEEE Congress on, IEEE, 2017, pp. 1311–1318.

[13] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, U Michigan Press, 1975.

[14] C.D.G. Kirkpatrick, M.P. Vecchi., Optimization by simmulated annealing., Science 220 (4598) (1983) 671–680.

[15] J.-S. Pan, Z. Meng, H. Xu, X. Li, QUasi-Affine TRansformation Evolution (QUATRE) Algorithm: A new simple and accurate structure for global optimization, in: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2016, pp. 657–667.

[16] Z. Meng, J.S. Pan, QUasi-Affine TRansformation Evolutionary (QUATRE) algorithm: A parameter-reduced differential evolution algorithm for optimization problems, in: 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 4082–4089.

[17] S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.

[18] E. Mezura-Montes, J. Velázquez-Reyes, C.A. Coello Coello, A comparative study of differential evolution variants for global optimization, in: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM, 2006, pp. 485–492.

[19] R. Gämperle, S.D. Müller, P. Koumoutsakos, A parameter study for differential evolution, Adv. Intell. Syst. Fuzzy Syst. Evol. Comput. 10 (10) (2002) 293–298.

[20] J. Brest, S. Greiner, B. Bošković, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657.

[21] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.

[22] Y. Wang, Z. Cai, Q. Zhang, Differential Evolution With Composite Trial Vector Generation Strategies and Control Parameters, IEEE Trans. Evol. Comput. 15 (1) (2011) 55–66.

[23] H. Holland John, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, USA: Univ. Michigan (1975).

[24] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.

[25] A. Zamuda, J. Brest, B. Boskovic, V. Zumer, Large Scale Global Optimization using Differential Evolution with self-adaptation and cooperative co-evolution, in: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, pp. 3718–3725.

[26] J. Brest, M.S. Maučec, Population size reduction for the differential evolution algorithm, Appl. Intell. 29 (3) (2008) 228–247.

[27] J. Brest, A. Zamuda, B. Boskovic, M.S. Maucec, V. Zumer, High-dimensional real-parameter optimization using Self-Adaptive Differential Evolution algorithm with population size reduction, in: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, pp. 2032–2039.

[28] J. Teo, Exploring dynamic self-adaptive populations in differential evolution, Soft Comput. 10 (8) (2006) 673–686.

[29] V. Feoktistov, S. Janaqi, Generalization of the strategies in differential evolution, in: Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International, IEEE, 2004, p. 165.

[30] K.V. Price, Differential evolution: a fast and simple numerical optimizer, in: Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American, IEEE, 1996, pp. 524–527.

[31] J.-S. Pan, Z. Meng, H. Xu, X. Li, A matrix-based implementation of DE algorithm: the compensation and deficiency, in: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2017, pp. 72–81.

[32] S. Rahnamayan, H.R. Tizhoosh, M.M. Salama, Opposition-based differential evolution, IEEE Trans. Evol. Comput. 12 (1) (2008) 64–79.

[33] S. Das, A. Abraham, U.K. Chakraborty, A. Konar, Differential evolution using a neighborhood-based mutation operator, IEEE Trans. Evol. Comput. 13 (3) (2009) 526–553.

[34] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.

[35] W. Gong, Z. Cai, Differential evolution with ranking-based mutation operators, IEEE Trans. Cybern. 43 (6) (2013) 2066–2081.

[36] Z. Meng, J.-S. Pan, H. Xu, QUasi-Affine TRansformation Evolutionary (QUATRE) algorithm: A cooperative swarm based algorithm for global optimization, Knowl.-Based Syst. 109 (2016) 104–121.

[37] Z. Meng, J.-S. Pan, QUasi-Affine TRansformation Evolution with External ARchive (QUATRE-EAR): an enhanced structure for differential evolution, Knowl.-Based Syst. 155 (2018) 35–53.

[38] F. Peng, K. Tang, G. Chen, X. Yao, Multi-start JADE with knowledge transfer for numerical optimization, in: 2009 IEEE Congress on Evolutionary Computation, 2009, pp. 1889–1895.

[39] J.E. Baker, Reducing bias and inefficiency in the selection algorithm, in: Proceedings of the second international conference on genetic algorithms, 1987, pp. 14–21.

[40] J. Liang, B. Qu, P. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, in: Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2013.

[41] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.-P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, KanGAL Rep. 2005005 (2005) 2005.

[42] N. Awad, M. Ali, J. Liang, B. Qu, P. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization, in: Technical Report, Nanyang Technological University Singapore, 2016, pp. 1–34.