ARTICLE IN PRESS

ISA Transactions xxx (xxxx) xxx

Contents lists available at ScienceDirect

ISA Transactions

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

Research article

FD-DE: Differential Evolution with fitness deviation based adaptation in parameter control

Zhenyu Meng [a,b,*], Zhenghao Song [a], Xueying Shao [a], Junyuan Zhang [a], Huarong Xu [c]

[a] Institute of Artificial Intelligence, Fujian University of Technology, Fuzhou, China
[b] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, China
[c] Department of Computer Science and Technology, Xiamen University of Technology, Xiamen, China

ARTICLE INFO

Article history:
Received 19 June 2022
Received in revised form 10 May 2023
Accepted 10 May 2023
Available online xxxx

Keywords:
Differential evolution
Fitness deviation
Parameter control
Population stagnation

ABSTRACT

Differential Evolution (DE) is arguably one of the most powerful stochastic optimization algorithms for different optimization applications, however, even the state-of-the-art DE variants still have many weaknesses. In this study, a new powerful DE variant for single-objective numerical optimization is proposed, and there are several contributions within it: First, an enhanced wavelet basis function is proposed to generate scale factor F of each individual in the first stage of the evolution; Second, a hybrid trial vector generation strategy with perturbation and t-distribution is advanced to generate different trial vectors regarding different stages of the evolution; Third, a fitness deviation based parameter control is proposed for the adaptation of control parameters; Fourth, a novel diversity indicator is proposed and a restart scheme can be launched if necessary when the quality of the individuals is detected bad. The novel algorithm is validated using a large test suite containing 130 benchmarks from the universal test suites on single-objective numerical optimization, and the results approve the big improvement in comparison with several well-known state-of-the-art DE variants. Moreover, our algorithm is also validated under real-world optimization applications, and the results also support its superiority.

© 2023 ISA. Published by Elsevier Ltd. All rights reserved.

1. Introduction

Differential Evolution (DE) [1], as one of the straightforward and efficient Evolutionary Algorithms (EAs), was initially put forth by Price and Storn in 1995 to solve Chebyshev polynomial fitting problem [2]. The good performance and easy implementation of DE have contributed to its progressive rise in popularity and practicality [3–6]. Same as other nature/bio-inspired algorithms [7–10], DE was also inspired by the biological principles of genetic inheritance and survival of the fittest, which enabled adaptation to its environment, therefore, good individuals were retained while the bad ones were discarded from the population [11–14]. The paper mainly focuses on improving the performance of DE for single-objective numerical optimization, which can be defined as follows:

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

where X denotes a solution candidate of the space Ω, $f(X)$ denotes the objective of a certain optimization application, X^* is the optimum of the minimum optimization and Ω^* is the set of all the proper X^*s. The DE algorithms will return a tolerable solution when the evolution process terminates by finding the optimum or arriving at the maximum number of function evaluations [15–18].

The performance of the DE algorithm, like that of other EAs, can be affected by the control parameters [19–22] including the scale factor F, the crossover rate CR and the population size PS. The ideal settings of these parameters are generally problem-dependent [23–26], and it is common to tweak the control parameter settings to get the desired outcomes when applying DE to real-world applications [27–30]. However, the tuning of the control parameters is time-consuming and it is also unfriendly to the engineers and scientists who are not DE professionals. Gradually, adaptive and self-adaptive DE algorithms drew much attention from the community. Brest et al. [2] proposed the first self-adaptive DE algorithm, namely jDE algorithm, in which control parameters evolved by encoding them into each individual, and the concept that better control parameters producing superior individuals who were more likely to survive and transmit their control parameters affected the development of algorithms in the DE branch significantly. Zhang and Sanderson [31] suggested a novel parameter adaption mechanism in which the control parameters CR and F obeyed semi-fixed Gaussian and Cauchy

* Corresponding author at: Institute of Artificial Intelligence, Fujian University of Technology, Fuzhou, China.
E-mail address: mzy1314@gmail.com (Z. Meng).

https://doi.org/10.1016/j.isatra.2023.05.005
0019-0578/© 2023 ISA. Published by Elsevier Ltd. All rights reserved.

Please cite this article as: Z. Meng, Z. Song, X. Shao et al., FD-DE: Differential Evolution with fitness deviation based adaptation in parameter control. ISA Transactions (2023), https://doi.org/10.1016/j.isatra.2023.05.005.

distribution respectively, and the values of μ_{CR} and μ_F were renewed at the end of each generation according to CR and F values that produced successful trial vectors in the generation. In other words, the success values of the control parameters F and CR were used for updating the distributions of themselves. However, the adaptations of F and CR in these algorithms were mixed together, which inevitably led to misleading interaction between control parameters. Meng et al. [32] developed a parameter adaptation technique in which the control parameters CR and F were separated and updated independently. This method significantly reduces the misleading interaction between parameters by using an efficient grouping scheme and a parameter-independent adaptation strategy. Tanabe and Fukunaga in SHADE algorithm [33] suggested that an entry pool recording the success μ_{CR} and μ_F pairs would improve the robustness of the parameter control, and the fitness-improvement-based adaptation schemes of CR and F were used for updating the $\mu_{CR} - \mu_F$ pair in the entry. The LSHADE algorithm [16] further improved the SHADE algorithm by incorporating a linear reduction technique by the same authors and won first place in the Congress on Evolutionary Computation (CEC2014) competition on single-objective numerical optimization. The later proposed state-of-the-art DE variants, for example, jSO [34], PaDE [35] and the algorithm in this paper, can be considered as further improvement of it.

Besides the parameter control, mutation strategy plays a vital role in the overall performance of DE [36–38]. Instead of employing the conventional mutation strategies in the literature, Zhang and Sanderson [31] developed a novel mutation strategy with an external archive in the JADE algorithm, where some individuals with poor performance were stored in the external archive and employing in enhancing the population diversity, thus achieving a better balance between exploration and exploitation. Deng et al. [39] suggested a new optimal mutation strategy which was chosen from five mutation strategies with complementary advantages, and the whole population was divided into various subpopulations for the consideration of diversity enhancement and useful information during the evolution was exchanged among them. The novel optimum mutation strategy boosts local search capacity while maintaining good global search capacity. Li et al. [40] proposed a method for evaluating the difficulty in problem-solving by quantitatively assessing the fitness distance information and obtaining relevance classification. The search strategy employed mixed mutation and fitness-distance-correlation-based parameter control to avoid falling into some local optima. Brest et al. [34] advocated that an inertia weight could be added in front of the scale factor F of the mutation strategy, and truncations of the scale factor F and crossover rate CR are conducted regarding different stages of the evolution. Though excellent results were obtained in its testing competition, it may be over-tuned according to the "no free lunch theorem" [41,42]. Meng and Yang [43] in Hip-DE presented a new mutation strategy based on depth information, in which the relationship of the individuals in the current generation and the relationship of the individuals between different generations are first factored into the mutation strategy. The main idea is that the historical information of the evolution can represent the global landscape of the objective, and is helpful in generating trial vector candidates during the evolution [44–46].

To summarize, there are three aspects of DE for its performance improvement when dealing with single-objective numerical optimization. The first is to apply various adaptive strategies in the generation of scale factor F at various phases in order to maintain a balance between exploration and exploitation. The second is a donor vector-based perturbation mechanism, which can improve the exploration capacity of DE and help it jump out of local optima by successfully perturbing individuals at different stages. The third method is the fitness-deviation-based adaptation of control parameters. All these three improvements are incorporated into our FD-DE algorithm and the main highlights of our algorithm are given as follows:

1. A wavelet basis function based generation of scale factor F is proposed instead of employing the commonly used Cauchy distribution.
2. A hybrid trial vector generation strategy with perturbation and t-distribution is advanced to generate different trial vectors regarding different stages of the evolution.
3. A fitness-deviation-based parameter control is proposed for the adaptation of control parameters during the evolution.
4. A novel diversity indicator is proposed and a restart scheme can be launched if necessary when the diversity is detected bad.

The rest of the paper is arranged as follows: Section 2 focuses on the well-known powerful DE algorithms, such as LSHADE, jSO, LPalmDE, PaDE, and Hip-DE. The proposed algorithm is introduced in Section 3. Section 4 discusses the experimental study and performance comparison of our algorithm with the state-of-the-art DE variants. Section 5 concludes the paper and presents limitations of existing work as well as some perspectives for future work.

2. Several powerful DE variants

In this part, several state-of-the-art DE variants are reviewed, and the LSHADE algorithm [16] is taken as the baseline because the recently proposed powerful DE variants, such as jSO [34], PaDE [35], LPalmDE [47], and Hip-DE [43], are all derived from it.

2.1. The LSHADE algorithm

The LSHADE algorithm is a famous DE variant as it won first place in the CEC2014 competition on single-objective numerical optimization. There are several characteristics mentioned in this algorithm: the first is that the historical success distributions of the control parameters are maintained in a candidate pool during the evolution; the second is the fitness-improvement-based adaptation scheme of control parameter; and the third is the linear population size reduction scheme. In parameter control, a new H-entry pool is allocated for the maintenance of the success distributions of F and CR, which improves the robustness of adaptation of the control parameters during the evolution. The control parameters F and CR follow the same distribution as JADE, and a randomly picked $\mu_F - \mu_{CR}$ pair from the H-entry pool is used for the generation of the corresponding F and CR for each individual. At the end of each generation, F and CR of the individuals that generate better trial vectors are employed in the adaptation of μ_F and μ_{CR}. Fitness-improvement-based adaptation scheme is also involved in the adaptation. Eqs. (2) and (3) present the detailed adaption schemes of μ_F and μ_{CR}, respectively:

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

$$\begin{cases} w_k = \dfrac{\Delta f_k}{\sum_{k=1}^{|S_{CR}|} \Delta f_k} \\ \Delta f_k = f(X_{k,G}) - f(U_{k,G}) \\ mean_{WL}(S_{CR}) = \dfrac{\sum_{k=1}^{|S_{CR}|} w_k \cdot S_{CR,k}^2}{\sum_{k=1}^{|S_{CR}|} w_k \cdot S_{CR,k}} \\ \mu_{CR,k,G+1} = \begin{cases} mean_{WL}(S_{CR}), & if\ S_{CR} \neq \emptyset \\ \mu_{CR,k,G}, & otherwise \end{cases} \end{cases} \tag{3}$$

Here Δf_k means the fitness difference of the kth individual in the population; $mean_{WL}(\cdot)$ represents the weighted Lehmer mean; $< \mu_F, \mu_{CR} >$ is a pair of record selected from a certain entry, e.g. the hth entry ($h \in [1, H]$) of the entry pool. Generally, only one entry will be updated in each generation. The generation of control parameter F and CR is also given in Eq. (4):

$$\begin{cases} F_i = randc_i(\mu_{F,r_i}, 0.1) \\ CR_i = \begin{cases} 0, & if\ \mu_{CR_i,r_i} = \emptyset \\ randn_i(\mu_{CR,r_i}, 0.1), & otherwise \end{cases} \end{cases} \tag{4}$$

F and CR should be truncated to (0,1] and [0,1] respectively before generating trial vectors. The linear population size reduction scheme proposed in the LSHADE algorithm is also given below in Eq. (5):

$$PS_{G+1} = round[\frac{PS_{\min} - PS_{\max}}{nfe_{\max}} \cdot nfe + PS_{\max}] \tag{5}$$

where $PS_{\max}$ is the initial population size, $PS_{\min}$ is the terminal population size, $nfe_{\max}$ is the maximum number of function evaluations, nfe is the number of current function evaluations, and round$(\cdot)$ means rounding to the nearest integer value. The size of the external archive $\mathbf{A}$ in LSHADE is dynamically adjusted according to $|A| = r^{arc} \cdot PS$ in accordance with the population size reduction.

2.2. The jSO algorithm

The jSO algorithm is a further improvement of LSHADE aiming at tackling the CEC2017 test suite for real-parameter single-objective optimization. An inertia weight is incorporated into the mutation strategy of LSHADE and then "DE/current-to-$p$best-ω/1" was proposed. The details of this mutation strategy are given below in Eq. (6):

$$V_{i,G} = X_{i,G} + F_\omega \cdot (X_{best,G}^p - X_{i,G}) + F \cdot (X_{r_1,G} - \widetilde{X}_{r_2,G}) \tag{6}$$

Here $X_{best,G}$ denotes individuals selected from the top $p \cdot 100\%$ of the population, and the parameter p obeys the adaption listed in Eq. (7):

$$p = (\frac{p_{\max} - p_{\min}}{\max_{nfes}}) \cdot nfes + p_{\min} \tag{7}$$

r_1 and r_2 denote the indices of randomly selected individuals from the current population $\mathbf{P}$ while $\widetilde{X}_{r_2,G}$ denotes individuals selected from the union of the current population and the external archive ($\widetilde{X}_{r_2} \in (\mathbf{P} \cup \mathbf{A})$). Moreover, the inertia-weight-based scale factor F_w is changed regarding different stages of the evolution, and the details are given in Eq. (7):

$$F_w = \begin{cases} 0.7 \cdot F & if\ nfes < 0.2 \cdot nfes_{\max} \\ 0.8 \cdot F & if\ nfes < 0.4 \cdot nfes_{\max} \\ 1.2 \cdot F & otherwise \end{cases} \tag{8}$$

Besides the slight change of mutation strategy, the adaptations of μ_F and μ_{CR} are also modified in the jSO algorithm, and the main idea is that the average of the old values and the new calculated mean values are of the better choice. The calculations of μ_F and μ_{CR} are given in Eqs. (9) and (10) respectively:

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

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

Then the control parameter F and CR can be generated according to the new μ_F and μ_{CR} in each generation and readjustments of the control parameters are also necessary for the jSO algorithm before generating the trial vectors. The readjustment of F and CR are given in Eqs. (11) and (12):

$$F_{i,G} = \begin{cases} \min(F_{i,G}, 0.7), & if\ nfes < 0.6 \cdot nfes_{\max} \\ 0.7 & otherwise \end{cases} \tag{11}$$

$$Cr_{i,G} = \begin{cases} \max(CR_{i,G}, 0.7), & if\ nfe < 0.25 \cdot nfe_{\max} \\ \max(CR_{i,G}, 0.6), & if\ nfe < 0.5 \cdot nfe_{\max} \\ CR_{i,G}, & otherwise \end{cases} \tag{12}$$

By the way, the jSO algorithm employs the same population size reduction scheme as the LSHADE algorithm which has already given in Eq. (5).

2.3. The LPalmDE algorithm

The LPalmDE algorithm is proposed to tackle the misleading interaction weakness in the parameter control of the LSHADE algorithm and the jSO algorithm. Generally, many DE variants proposed after the inception of the LSHADE algorithm mainly employed the same rule in the adaptations of control parameters F and CR. In other words, the control parameters F and CR that are used for generating success trial vectors are all considered to be good control parameters and then they are employed in updating their distributions. Consequently, there are cases that a better F and a worse CR (or a worse F and a better CR) may provide successful trial vector candidates, then the worse control parameter is used for updating its distributions. This weakness is named "misleading-interaction weakness". The main idea of the LPalmDE algorithm is to separate the control parameters into different groups and update them independently.

In the LPalmDE algorithm, all individuals in the population are divided into K groups according to the stochastic universal selection, and the selection probability of each group is initialized equal, $P(1) = P(2), \ldots, P(k) = \cdots = P(K) = \frac{1}{K}$ and then updated at the end of each generation according to Eq. (13):

$$\begin{cases} ns = \sum_{k=1}^{K} ns_k, \\ r_k = \begin{cases} \dfrac{ns_k^2}{ns \cdot (ns_k + nf_k)}, & if\ ns_k > 0 \\ \epsilon, & otherwise \end{cases} \\ P(k) = \dfrac{r_k}{\sum_{k=1}^{K}(r_k)} \end{cases} \tag{13}$$

where the number of winner individuals in the jth group is denoted by ns_j, the number of loser individuals in the jth group is represented by nf_j, and the number of all winner individuals in the population is represented by ns. By the way, the winner individuals are the ones that produced better offspring while the loser individuals are the ones that failed to produce better offspring. The adaptations of μ_F and μ_{CR} are given in Eqs. (14) and (15) respectively:

$$\begin{cases} \boldsymbol{S_F} = \bigcup_{k=1}^{K} \boldsymbol{S_{F_k}} \\ w_s = \dfrac{\Delta f_j}{\sum_{s=1}^{|\boldsymbol{S_F}|} \Delta f_j} \\ mean_{WL}(\boldsymbol{S_F}) = \dfrac{\sum_{s=1}^{|\boldsymbol{S_F}|} w_s \cdot \boldsymbol{S_F^2}(s)}{\sum_{s=1}^{|\boldsymbol{S_F}|} w_s \cdot \boldsymbol{S_F}(s)} \end{cases} \tag{14}$$

$$\mu_{CR} = \frac{\sum_{k=1}^{K} P(k) \cdot CR_k^2}{\sum_{k=1}^{K} P(k) \cdot CR_k} \tag{15}$$

Moreover, a times-tamp-based strategy is proposed in the LPalmDE algorithm which makes a balance between the mutation strategy "DE/target-to-pbest/1" with external archive and the mutation strategy without external archive.

2.4. The PaDE algorithm

The PaDE algorithm proposed by Meng et al. [35] can be considered as a further extension of the LPalmDE algorithm. An improved grouping strategy is proposed in the PaDE algorithm and the main idea is that the group containing more winner individuals should be classified more individuals into it the next generation. The initialization of the selection probability of each group is equal, $P(1) = P(2), \dots, P(k) = \cdots = P(K) = \frac{1}{K}$ and the probabilities is updated in the same way as the LPalmDE algorithm. However, the adaptations of the control parameters μ_F and μ_{CR} are improved from the LSHADE algorithm rather than the LPalmDE algorithm. The detailed adaptations of μ_F and μ_{CR} are given in Eqs. (16) and (17) respectively:

$$\begin{cases} w_s = \dfrac{\Delta f_s}{\sum_{s=1}^{|S_F|} \Delta f_s} \\ \Delta f_i = f(X_{i,G}) - f(U_{i,G}) \\ mean_{WL}(S_F) = \dfrac{\sum_{s=1}^{|S_F|} w_s \cdot S_F^2(s)}{\sum_{s=1}^{|S_F|} w_s \cdot S_F(s)} \\ \mu_{F,k,G+1} = \begin{cases} mean_{WL}(S_F), & if\ S_F \neq \emptyset \\ \mu_{F,k,G}, & otherwise \end{cases} \end{cases} \tag{16}$$

$$\begin{cases} w_s = \dfrac{\Delta f_s}{\sum_{s=1}^{|S_{CR}|} \Delta f_s} \\ \Delta f_i = f(X_{i,G}) - f(U_{i,G}) \\ mean_{WL}(S_{CR}) = \dfrac{\sum_{s=1}^{|S_{CR}|} w_s \cdot S_{CR}^2(s)}{\sum_{s=1}^{|S_{CR}|} w_s \cdot S_{CR}(s)} \\ \mu_{CR_{idx},G+1} = \begin{cases} mean_{WL}(S_{CR}), & if\ S_{CR} \neq \emptyset \& \max\{S_{CR}\} > 0 \\ 0, & if\ S_{CR} \neq \emptyset \& \mu_{CR_{idx},G} = 0 \\ \mu_{CR_{idx},G}, & otherwise \end{cases} \end{cases} \tag{17}$$

where S_F denotes the set of F values of the winner individuals and S_{CR} denotes the set of CR values of the winner individuals S_{CR}. Moreover, a novel parabolic population size reduction technique is also proposed in the PaDE algorithm with the exact formula given in Eq. (18):

$$PS_{G+1} = \begin{cases} \left\lceil \dfrac{y - PS_{ini}}{(x - PS_{ini})^2} \cdot (nfe - PS_{ini})^2 + PS_{ini} \right\rceil, & if\ nfe < x \\ \left\lfloor \dfrac{y - PS_{ini}}{x - PS_{ini}} \cdot (nfe - nfe_{\max}) + PS_{\min} \right\rfloor, & otherwise \end{cases} \tag{18}$$

Here the pivot point $Pivot = (x, y)$ is the connecting point of the parabola and the line segment. The main idea of this reduction scheme of population size is that the slower reduction of population size in the earlier stage of the evolution helps the algorithm to make a better exploration of the solution space.

2.5. The Hip-DE algorithm

The Hip-DE algorithm [43] was proposed by Meng and Yang in 2021, and the main characteristic of it is that historical information of the evolution was incorporated into the mutation strategy rather than just employing the current information of the population. The main idea of this innovation is that the historical information of the population can reflect the landscape of the objective. By incorporating the historical information into the mutation strategy, the Hip-DE algorithm can obtain better overall optimization performance. The details of the mutation strategy are given in Eq. (19):

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G}^{p} - X_{i,G}) + F \cdot (X_{r_1,G} - \widetilde{X}_{r_2,G}) \tag{19}$$

where $X_{r_2,G}$ is different from the above-mentioned DE variants and it denotes a randomly selected vector from the union, $\mathbf{P} \cup \mathbf{H}$. Here $\mathbf{P}$ also denotes the current population while $\mathbf{H}$ denotes the historical population.

Besides the mutation strategy, the adaptations of control parameters in Hip-DE algorithm are also different from the other DE variants. The adaptation of μ_F can be considered as a combination of the adaptations from JADE and LSHADE while the adaptation of μ_{CR} is modified from LSHADE. The details of the two adaptations are given in Eqs. (20) and (21) respectively:

$$\begin{cases} w_s = \dfrac{\Delta f_j}{\sum_{s=1}^{|S|} \Delta f_j} \\ mean_{WL}(S_F) = \dfrac{\sum_{s=1}^{|S_F|} w_s \cdot S_F^2(s)}{\sum_{s=1}^{|S_F|} w_s \cdot S_F(s)} \\ \mu_F = \begin{cases} (1-c) \cdot \mu_F + c \cdot mean_{WL}(S_F), & if\ S \neq \emptyset \\ \mu_F, & otherwise \end{cases} \end{cases} \tag{20}$$

$$\begin{cases} w_s = \dfrac{\Delta f_j}{\sum_{s=1}^{|S|} \Delta f_j} \\ mean_{WL}(S_{CR}) = \dfrac{\sum_{s=1}^{|S|} w_s \cdot S_{CR}^2(s)}{\sum_{s=1}^{|S|} w_s \cdot S_{CR}(s)} \\ \mu_{CR_{idx}} = \begin{cases} mean_{WL}(S_{CR}), & if\ S \neq \emptyset \\ \mu_{CR_{idx}}, & otherwise \end{cases} \end{cases} \tag{21}$$

Then, the control parameters F and CR can be regenerated according to Eqs. (22) and (23) after the updates of μ_F and μ_{CR}.

$$\begin{cases} F_{tmp} = randc(\mu_F, 0.1) \\ F_{tmp} = \begin{cases} randc(\mu_F, 0.1), & while\ F \leq 0 \\ 1. & if\ F > 1 \end{cases} \\ F_{i,G+1} = \begin{cases} F_{tmp}, & if\ rand_1 < \tau_1 \\ F_{i,G}, & otherwise \end{cases} \end{cases} \tag{22}$$

$$CR_{i,G+1} = \begin{cases} C(\mu_{CR_k}, 0.1), & if\ rand_2 < \tau_2 \& \mu_{Cr} \neq 0 \\ 0, & if\ rand_2 < \tau_2 \& \mu_{Cr} = 0 \\ CR_{i,G}, & otherwise \end{cases} \tag{23}$$

where the variables τ_1 and τ_2 denote the changing probabilities of the control parameters. Moreover, the calculation of the index *idx* is the same as the one in the PaDE algorithm. For the changing of population size *PS*, the Hip-DE algorithm employs fixed population size in the earlier stage of the evolution and then employs a linear reduction of the population at the later stage. Eq. (24) presents the reduction of the population size in the later part of the evolution:

$$PS = \lceil \frac{PS_{\min} - PS_{ini}}{nfe_{\max} - nfe_{st}} \cdot (nfe - nfe_{st}) + PS_{ini} \rceil \tag{24}$$

where nfe_{st} denotes the maximum number of function evaluations in the first stage of the evolution. By the way, other symbols have the same meanings as are explained above.

3. The novel FD-DE algorithm

The description of the novel FD-DE algorithm is divided into three parts: the first part presents the parameter adaptations, the second part gives the hybrid trial vector generation strategy, and the third part addresses the diversity adjustment by replacing the dimensions of the individuals.

3.1. Parameter adaptation

The scale factor F has a significant impact on the generation of donor vectors and consequently affects the overall performance of DE significantly, therefore, the adaptation of the scale factor F during the evolution is an important component in DE research. In our FD-DE algorithm, a wavelet basis function based parameter control for scale factor F is proposed for the earlier stage of the evolution instead of employing the commonly used Cauchy distribution during the whole evolution, and every F larger than 0.6 is truncated to 0.6 in this stage. The inspiration of this setting is because the wavelet basis function is more favorable for localization in the time and frequency domain [39], and can be served as an alternative for the generation of scale factor F. Eq. (25) presents the details in the generation of scale factors:

$$F_i = \begin{cases} \sqrt{2} \cdot \pi^{-\frac{1}{3}} \cdot (1 - \mu_F^2) \cdot e^{-\frac{\mu_F^2}{2}} + \varepsilon, & \text{if } nfe < nfe_{st} \\ randc_i(\mu_F, 0.1), & otherwise \end{cases} \tag{25}$$

where ε equals to $0.1 \cdot sin(\pi \cdot rand_i - 0.8)$, and Fig. 1 presents the comparison between Cauchy function and Wavelet basis function in the generation of F values.

The crossover rate CR also has a significant impact on the generation of trial vectors because it determines the probability of parameters in the donor vector inherited into the trial vector. The Gaussian distribution used in the literature [16,31] is also employed in our FD-DE algorithm, moreover, these CR values are restricted in [0,0.6] at the first stage of the evolution and also extends to [0,1] in the rest evolution. The detailed generation of CR is given in Eq. (26):

$$CR_i = \begin{cases} 0, & \text{if } \mu_{CR} = \emptyset \\ randn_i(\mu_{CR}, 0.1), & otherwise \end{cases} \tag{26}$$

Obviously, both the scale factor F and the crossover rate CR are all truncated to smaller values, no larger than 0.6, in the first stage of evolution while the ranges of F and CR are extended to (0,1] and [0,1] respectively in the rest evolution, which helps to prevent premature convergence and drawing the population out of some local optima.

The H-entry pool recording μ_F and μ_{CR} pairs proposed in LSHADE is also retained in our FD-DE algorithm for the robustness consideration of the parameter control. Only one entry was updated in each generation and the entry pool was updated in a circle from the 1st to the last and then back to the first. The idea that good control parameters are used for updating the distributions of themselves in JADE is further improved in our FD-DE algorithm with the details given in Eqs. (27) and (28) respectively:

$$\begin{cases} w_k = \dfrac{d_{f_i}}{\sum_{k \in S} d_{f_i}} \\ mean_{WL}(S_F) = \dfrac{\sum_{k=1}^{|S_F|} w_k \cdot S_F^2(k)}{\sum_{k=1}^{|S_F|} w_k \cdot S_F(k)} \\ \mu_{F,k,G+1} = \begin{cases} \frac{(mean_{WL}(S_F) + \mu_{F,k,G})}{2}, & \text{if } S_F \neq \emptyset \\ \mu_{F,k,G}, & otherwise \end{cases} \end{cases} \tag{27}$$

$$\begin{cases} w_k = \dfrac{d_{f_i}}{\sum_{k \in S} d_{f_i}} \\ mean_{WL}(S_{CR}) = \dfrac{\sum_{k=1}^{|S_{CR}|} w_k \cdot S_{CR}^2(k)}{\sum_{k=1}^{|S_{CR}|} w_k \cdot S_{CR}(k)} \\ \mu_{CR,k,G+1} = \begin{cases} mean_{WL}(S_{CR}), & \text{if } S_{CR} \neq \emptyset \\ \mu_{CR,k,G}, & otherwise \end{cases} \end{cases} \tag{28}$$

where d_{f_i} denotes the fitness deviation of the ith individual in the population and it can be calculated according to Eq. (29):

$$\begin{cases} d_{f_i} = \dfrac{\left| \Delta f_i - \overline{\Delta f} \right|}{\Delta f_i} \\ \Delta f_i = f(X_{i,G}) - f(U_{i,G}) \\ \overline{\Delta f} = \dfrac{\sum_{i=1}^{PS} (f(X_{i,G}) - f(U_{i,G}))}{PS} \end{cases} \tag{29}$$

Here $\overline{\Delta f}$ denotes the mean value of the fitness differences of all individuals in the population, and Δf_i denotes the fitness improvement of the ith individuals. By the way, the update of control parameters μ_F and μ_{CR} is conducted only if the success set S of individuals is not empty, in other words, Δf_i satisfies $\Delta f_i > 0$.

3.2. Trial vector generation strategy

As is known to all, the "DE/target-to-$p$best/1/bin" mutation strategy with external archive [31] obtained excellent performance in recent competitions, however, it still has premature convergence weakness and converges into some local optima when tackling some complex optimization problems. In order to tackle this weakness, t-distribution based perturbation strategy is proposed as a supplementary of the mutation strategy "DE/target-to-$p$best/1/bin", and the inspiration is that the standard deviation of global best can be considered as the perturbation range of the target vector.

The main mutation strategy of our FD-DE algorithm is given below in Eq. (30):

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G}^{p} - X_{i,G}) + F \cdot (X_{r_1,G} - \widetilde{X}_{r_2,G}). \tag{30}$$

Then the trial vector $U_{i,G}$ can be calculated by crossover operation according to a certain crossover rate CR. After that, the t-distribution based perturbation is employed in the trial vector under a certain probability τ, $\tau = 0.05$, and the details of the perturbation is shown in Eq. (31):

$$U_{j,i,G} = X_{j,i,G} + rand_{j,i,G} \cdot std(X_{gbest,G}) \cdot (tpdf(G, 1) + 1) \tag{31}$$

where $U_{j,i,G}$ and $X_{j,i,G}$ denote the jth parameter of the trial vector $U_{i,G}$ and the target vector $X_{i,G}$ respectively, $std(X_{gbest,G})$ denotes the standard deviation of $X_{gbest,G}$, and $tpdf(G, 1)$ denotes the probability density function of t-distribution. The pseudo code of the trial vector generation strategy is given in Algorithm 1.

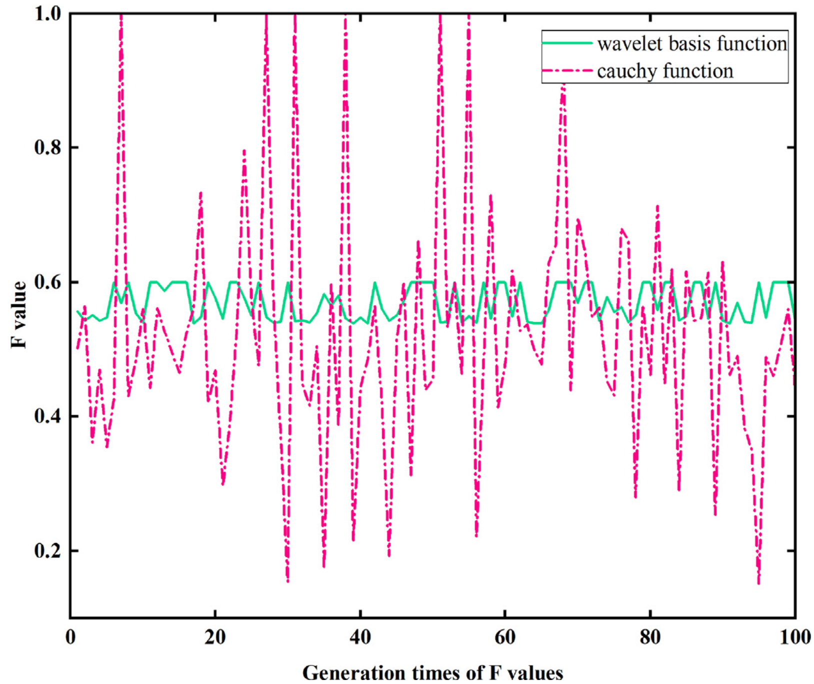


Fig. 1. The generation comparison of F by employing Cauchy function and Wavelet basis function.

Algorithm 1 Trial vector generation strategy

1: **for** $i = 1$ **to** PS **do**
2: Calculate $V_{i,G}$ according to Eq. (30);
3: Generate j_{rand} = randint (1, D);
4: **for** $j = 1$ **to** D **do**
5: **if** $j = j_{rand}$ or $rand(0, 1) < CR_i$ **then**
6: $U_{j,i,G} = V_{j,i,G}$;
7: **else**
8: **if** $rand > \tau$ **then**
9: $U_{j,i,G} = X_{j,i,G}$;
10: **else**
11: Perturb $U_{i,G}$ according to Eq. (31);
12: **end if**
13: **end if**
14: **end for**
15: **end for**

3.3. Dimensional replacement of individuals

Besides the perturbation in the trial vector, a diversity indicator of the population is proposed and dimensional replacement of individuals can be launched if necessary to draw the whole population out of some local optima. In order to detect the diversity of individuals, we offer a surrogate hyper-volume model to compute the geographic distribution of individuals in the population, and the model is an enhanced one in comparison with a former model proposed in [48]. The hyper-volume of the individuals can be calculated according to Eq. (32):

$$V_{hp} = \sqrt{\prod_{i=1}^{D} |u_i - l_i|} \tag{32}$$

where u_i and l_i denote the lower and upper bound of the ith dimension of the search domain. Then, the diversity indicator d_{in} can be calculated via Eq. (33):

$$\begin{cases} d_{in} = \sqrt{V_{pop}/V_{hp}} \\ V_{pop} = \sqrt{\prod_{i=1}^{D} |(u_{x_i} + l_{x_i})/2|} \end{cases} \tag{33}$$

where V_{pop} denotes the center cube of the population, u_{x_i} and l_{x_i} denote the lower and upper bound of the ith dimension of the population.

Besides the diversity indicator, a label recording the improvement status of each individual is also employed in the dimensional replacement mechanism. Different from the mechanism proposed in Tian's papers [39,49], the sum of the label that recording individuals without performance improvement is calculated according to Algorithm 2. When the algorithm enters a status satisfying $d_{in} < \xi$ ($\xi = 0.01$) and $C_n > k \cdot PS \cdot D$ ($k = 0.6$), dimensional replacement of individuals is launched according to Eq. (34):

$$X_{i,G}(j) = \begin{cases} X_{rand_i,G}(j), & if\ j \in R \\ X_{i,G}, & otherwise \end{cases} \tag{34}$$

which means that the jth dimension of the ith individual is replaced by a randomly selected individual's jth dimensional parameter. The whole pseudo code of our FD-DE algorithm is given in Algorithm 3, and it can be seen that the main body of the pseudo code is more or less the same as the algorithms in Section 2 and the main differences among these algorithms lie in the generation of trial vectors, the parameter control and the diversity enhancement technique. All the three aspects are also the main contributions of our work.

4. Experiment analysis

In this section, we mainly examine our algorithm in depth under a large test suite containing 130 benchmarks from the universal test suites for single-objective numerical optimization. The 88 benchmarks from CEC2013, CEC2014 and CEC2017 test suites

Algorithm 2 Calculate *counter* and C_n

$C_n = 0$
for $i = 1$ **to** PS **do**
  if $f(X_{i,G}) > f(U_{i,G})$ **then**
    $counter(i) = counter(i) + 1$
    $C_n = C_n + counter(i)$
  else
    $counter(i) = 0$;
  end if
end for

Algorithm 3 Pseudo code of the FD-DE algorithms

Require: Dimension D, bound constraints $[L, U]^D$, maximum number of function evaluations $nfes_{\max}$, objective $f(X)$.
Ensure: Global best value $f(X_{gbest})$, global best individual X_{gbest}, number of function evaluations $nfes$;
Initialize the population $P = \{X_1, X_2, \ldots, X_{PS}\}$, calculate the fitness values and label the global best individual;
Initialize the parameters and the external archive A;
while $nfes < nfes_{\max}$ **do**
  Calculate the control parameters and the trial vectors;
  for $i = 1$ **to** PS **do**
    Update the population;
    if $f(U_{i,G} < f(X_{i,G})$ **then**
      $X_{i,G} \rightarrow A$;
      $CR_{i,G} \rightarrow S_{CR}$, $F_{i,G} \rightarrow S_F$;
    end if
  end for
  Update A, μ_{CR} and μ_F;
  for $i = 1$ **to** PS **do**
    if $C_n > n$ && $d_{in} < 0.01$ **then**
      Adjust the i^{th} individual according to Eq. (34)
      Calculate the fitness value $f(X_{i,G})$;
      $nfes = nfes + 1$;
    end if
  end for
  Calculate PS_{G+1} according to Eq. (5);
  if $PS_G < PS_{G+1}$ **then**
    Individuals are sorted depending on their fitness;
    The worse PS_G - PS_{G+1} individuals are removed from the population;
  end if
  $G = G + 1$;
end while

for single-objective numerical optimization are the basic validation, and the extra total 42 benchmarks from CEC2019–CEC2022 test suites which includes 10 benchmarks from CEC2019 100-Digit challenge, 10 benchmarks from CEC2020 single-objective bound constrained numerical optimization, 10 benchmarks from CEC2021 single-objective bound constrained numerical optimization and 12 benchmarks from CEC2022 single-objective bound constrained optimization are suggested by a reviewer. Furthermore, the reviewer also suggested to validate our algorithm under the 57 real-world benchmarks from the CEC2020 competition on real-world single objective constrained optimization, so we also conducted further algorithm validation on real-world optimization applications. For the basic 88 benchmark-validation, there are 28 benchmarks from the CEC2013 test suite, 30 benchmarks from the CEC2014 test suite, and 30 from the CEC2017 test suite among the 88 benchmarks. Benchmarks from the CEC2013 test suite are marked as $f_{a_1} - f_{a_{28}}$ in our test suite, benchmarks from the CEC2014 test suite are denoted as $f_{b_1} - f_{b_{30}}$ and benchmarks from the CEC2017 test set are denoted as $f_{c_1} - f_{c_{30}}$ in our test suite. In algorithm validation, the maximum number of function evaluations is $10000 \cdot D$, and the fitness error statistics, e.g. mean and standard deviation, are obtained from 51 runs. By the way, all experiments are performed on MATLAB R2021a on a PC (Intel i74570 CUP3.20 GHz RAM4.00 GB) with Windows 10 x64 operating system.

4.1. Optimization accuracy

Five state-of-the-art DE variants are taken into consideration for algorithm comparison, and they are LSHADE [16], jSO [34], LPalmDE [32], PaDE [35], and Hip-DE [43]. All the settings listed in Table 1 of the aforementioned algorithms are their default values recommended by the authors. Statistical analysis is used for the comparison of the total 51-run results obtained by each algorithm on each benchmark. The mean value (Mean) and standard deviation (Std) of the fitness error $f - f^*$ of the total 51 runs are calculated and presented in Tables 2–11. Symbols "$>$", "$\approx$" and "$<$" in the parentheses behind "Mean/Std" denote "Better Performance", "Similar Performance" and "Worse Performance" respectively in comparison with our algorithm under Wilcoxon's signed rank test with the significant level $\alpha = 0.05$ on each benchmark, then, the comparison results on each benchmark can be considered as a Bernoulli distribution and the comparison under all the benchmarks can be considered as binomial distribution, which satisfies the conditions of sign test. "$w/d/l$" presented at the bottom of each table is the sum of the comparison results ($>$, $\approx$ and $<$) on all the benchmarks. The tier best results in the comparison are highlighted in *ITALIC* fonts while the best result in the comparison is emphasized in **BOLDFACE** on each benchmark.

Besides the comparison results on each test suite with different dimensional optimization, Table 12 also presents a summary of the comparison results from Tables 2–11, and we can see from the results that (1) our algorithm has overall better performance under the large test suite in comparison with the compared state-of-the-art DE variants; (2) our algorithm can outperform the winner DE variants, LSHADE and jSO, on their competitions; (3) our algorithm can also beat LPalmDE, PaDE and Hip-DE from the overall perspective of view though it fails to beat LPalmDE, PaDE and Hip-DE on some lower dimensional optimization; (4) our algorithm performs extremely better on CEC2014 and CEC2017 test suits rather than CEC2013; (5) Huge advantages of our algorithm can be observed for higher dimensional optimization. As the second reviewer suggested, we also conducted further algorithm validation in comparison with LSHADE-cnEpSin and LSHADE-SPACMA which ranked third and fourth in the CEC2017 competition, and the comparison results on 10D, 30D, 50D and 100D optimization are summarized in Table 13. Moreover, we also conducted algorithm validation on CEC2019–CEC2022 in the supplementary file as the third reviewer suggested. Again, it can be seen from the results that our algorithm is competitive as well.

4.2. Comparison on convergence speed

This part presents the algorithm comparison in terms of convergence speed, which was usually conducted by plotting the median value of 51-run results. The comparison figures of the algorithms including LSHADE, jSO, LPalmDE, PaDE, Hip-DE and our algorithm under the CEC2017 test suite for 30D optimization are given in Figs. 2–3. It can be seen from the figures that our approach outperforms LSHADE on benchmarks f_{c_1}, $f_{c_3} - f_{c_5}$, $f_{c_7} - f_{c_{10}}$, $f_{c_{14}} - f_{c_{22}}$, $f_{c_{24}} - f_{c_{30}}$; outperforms jSO on f_{c_1}, $f_{c_3} - f_{c_5}$, $f_{c_7} - f_{c_{10}}$, $f_{c_{12}} - f_{c_{15}}$, $f_{c_{17}} - f_{c_{19}}$, $f_{c_{21}} - f_{c_{30}}$; outperforms LPalmDE on $f_{c_1} - f_{c_5}$, $f_{c_8} - f_{c_{13}}$, $f_{c_{15}} - f_{c_{28}}$, $f_{c_{30}}$; outperforms PaDE on f_{c_1}, $f_{c_3} - f_{c_5}$, $f_{c_7} - f_{c_{12}}$, $f_{c_{15}} - f_{c_{24}}$, $f_{c_{26}} - f_{c_{30}}$; and outperforms Hip-DE on

Table 1
Recommended Parameter settings of all these contrasted algorithms.

Algorithm	Parameters initial settings
LSHADE	$\mu_F = 0.5$, $\mu_{CR} = 0.5$, $F\&CR$ same as JADE, $NP = 18 \cdot D \sim 4$, $r^{rac} = 2.6$, $p = 0.11$, $H = 6$
jSO	F, CR, r^{rac} same as iLSHADE, $\mu_F = 0.3$, $\mu_{CR} = 0.8$, $NP = 25 \cdot \ln D \cdot \sqrt{D} \sim 4$, $p = 0.25 \sim 0.125$, $H = 5$
LPalmDE	$\mu_{F_k} = \mu_{CR_k} = 0.5$, $F \sim CR(\mu_{F_k}, 0.2)$, $CR \sim N(\mu_{CR_k}, 0.1)$, $p = 0.11$, $k = 6$, $ps = 23 \cdot D \sim K$, $r^{rac} = 1.6$, $T_0 = 70$
PaDE	$\mu_F = 0.8$, $\mu_C R = 0.6$, $F\&CR$ same as LSHADE, $p = 0.11$, $NP = 25 \cdot \ln D \cdot \sqrt{D} \sim 4$, $r^{rac} = 1.6$, $r^d = 0.04$, $T_0 = 70$, $K = 4$
Hip-DE	$\mu_{F_k} = 0.6$, $\mu_{CR_k} = 0.8$, $F \sim CR(\mu_{F_k}, 0.1)$, CR same as LPalmDE, $p = 0.2 \sim 0.05$, $k = 6$, $ps = 15 \cdot D \sim K$, $r^{rac} = 5$, $\tau_F = \tau_{CR} = 0.9$, $N = \lceil r^p \cdot nfes_{\max}/ps \rceil$
Our algorithm	$F = 0.5$, $CR = 0.8$, $p = 0.11$, $k = 0.6$, $r^{rac} = 1.4$, $PS = 25 \cdot lnD\sqrt{D}$, $H = 4$, $nfes = D \cdot 10000$, $n = 0.6 \cdot PS \cdot D$, $C_n = 0$, $G = 1$, $G_{\max} = 3000$

Table 2
Algorithm comparisons between LSHADE, jSO, LPalmDE, PaDE, Hip-DE and our algorithms on 10-D optimization. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations max_{nfes} equalling to $10000 \cdot D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant max level $\alpha = 0.05$ in comparison with the new proposed algorithms.

DE variants	LSHADE	jSO	LPalmDE	PaDE	Hip-DE	Our algorithm
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{a_1}	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
f_{a_2}	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
f_{a_3}	6.9958E−03/2.1430E−02(≈)	1.3992E−03/9.9920E−03(≈)	6.9958E−03/2.1430E−02(≈)	6.9958E−03/2.1430E−02(≈)	1.3992E−03/9.9920E−03(>)	1.1193E−02/2.6209E−02
f_{a_4}	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
f_{a_5}	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
f_{a_6}	4.2328E+00/4.9081E+00(<)	1.9240E−01/1.3740E+00(≈)	3.6556E+00/4.7914E+00(<)	1.9240E+00/3.9346E+00(≈)	3.8480E+00/4.8384E+00(<)	1.1016E+00/3.0560E+00
f_{a_7}	5.1012E−06/1.1493E−05(<)	1.2811E−05/3.5711E−05(<)	1.1806E−05/4.1382E−05(<)	6.8743E−06/2.2339E−05(<)	5.8196E−06/1.0527E−05(<)	**9.0580E−07/3.7698E−06**
f_{a_8}	2.0234E+01/1.6500E−01(<)	2.0338E+01/6.9219E−02(<)	2.0170E+01/1.5794E−01(≈)	2.0185E+01/1.6226E−01(≈)	2.0190E+01/1.6379E−01(≈)	2.0152E+01/1.2764E−01
f_{a_9}	2.6146E+00/1.4096E+00(<)	5.4580E−01/8.0453E−01(<)	6.2155E−01/9.6573E−01(<)	1.1113E+00/1.3708E+00(<)	5.2625E−01/7.5240E−01(<)	**1.5513E−01/4.1253E−01**
$f_{a_{10}}$	3.1887E−03/5.4471E−03(>)	9.6684E−04/2.9617E−03(>)	1.1590E−03/3.7296E−03(>)	4.8761E−03/1.2935E−02(>)	1.9800E−03/5.9716E−03(>)	1.8788E−02/1.8428E−02
$f_{a_{11}}$	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
$f_{a_{12}}$	2.3714E+00/8.8618E−01(≈)	2.2631E+00/8.6924E−01(≈)	2.3801E+00/1.0162E+00(≈)	2.5456E+00/1.3641E+00(≈)	1.6059E+00/7.1994E−01(≈)	2.1925E+00/1.6381E+00
$f_{a_{13}}$	1.6942E+00/1.1617E+00(≈)	2.1161E+00/1.0569E+00(<)	2.3059E+00/1.6713E+00(<)	1.7455E+00/1.1427E+00(<)	1.4554E+00/6.0909E−01(≈)	1.3284E+00/8.2561E−01
$f_{a_{14}}$	2.2043E−02/3.7103E−02(≈)	4.0944E−02/5.5231E−02(<)	3.6738E−03/1.4841E−02(>)	1.3471E−02/3.3780E−02(≈)	7.3476E−03/2.0322E−02(>)	2.0818E−02/3.4586E−02
$f_{a_{15}}$	3.0936E+02/1.0250E+02(<)	2.7873E+02/9.6577E+01(≈)	3.8940E+02/1.6165E+02(<)	3.2704E+02/1.3075E+02(<)	3.1918E+02/1.5903E+02(<)	**2.5373E+02/1.1740E+02**
$f_{a_{16}}$	2.4399E−01/1.3480E−01(<)	1.0699E+00/1.9135E−01(<)	1.6171E−01/1.5357E−01(≈)	2.7084E−01/1.8882E−01(<)	1.1344E−01/1.2949E−01(≈)	1.2582E−01/1.4550E−01
$f_{a_{17}}$	1.0122E+01/1.5461E−14(≈)	1.0124E+01/2.9996E−03(<)	1.0122E+01/1.5461E−14(≈)	1.0122E+01/1.8501E−14(≈)	1.0122E+01/2.3613E−14(≈)	1.0122E+01/1.5461E−14
$f_{a_{18}}$	1.3780E+01/9.3860E−01(≈)	1.3904E+01/1.3513E+00(≈)	1.4739E+01/1.8785E+00(<)	1.5038E+01/2.0778E+00(<)	**1.3952E+01/1.4635E+00**(>)	1.4033E+01/1.5843E+00
$f_{a_{19}}$	2.1888E−01/3.5891E−02(>)	2.5004E−01/5.4417E−02(≈)	2.7790E−01/8.1843E−02(<)	2.2391E−01/3.2526E−02(>)	2.2487E−01/3.7121E−02(>)	2.4184E−01/3.7637E−02
$f_{a_{20}}$	1.9590E+00/3.3882E−01(<)	1.6501E+00/2.9398E−01(<)	1.7308E+00/3.2635E−01(<)	1.6365E+00/2.5202E−01(<)	1.4847E+00/3.3121E−01(<)	**1.3358E+00/3.5912E−01**
$f_{a_{21}}$	4.0019E+02/0.0000E+00(≈)	3.9627E+02/2.8033E+01(≈)	4.0019E+02/0.0000E+00(≈)	4.0019E+02/0.0000E+00(≈)	4.0019E+02/0.0000E+00(≈)	4.0019E+02/0.0000E+00
$f_{a_{22}}$	7.3504E+00/1.3629E+01(<)	8.0761E+00/1.3948E+01(<)	3.2694E+00/3.7856E+00(≈)	3.7927E+00/3.9378E+00(≈)	3.8246E+00/1.4344E+01(>)	5.2450E+00/1.4232E+01
$f_{a_{23}}$	2.6995E+02/1.4681E+02(<)	1.9197E+02/1.1895E+02(<)	3.0156E+02/1.6852E+02(<)	2.5629E+02/1.3776E+02(<)	2.2225E+02/1.3975E+02(<)	**1.4104E+02/1.0949E+02**
$f_{a_{24}}$	2.0312E+02/3.3689E+00(<)	2.0065E+02/1.9811E+00(≈)	2.0000E+02/0.0000E+00(≈)	1.9895E+02/7.4720E+00(≈)	2.0000E+02/6.9040E−06(≈)	2.0000E+02/3.2155E−14
$f_{a_{25}}$	2.0103E+02/2.0273E+00(<)	2.0009E+02/6.3574E−01(≈)	2.0000E+02/0.0000E+00(≈)	2.0000E+02/3.2155E−14(≈)	2.0000E+02/1.1191E−04(≈)	2.0000E+02/7.1902E−14
$f_{a_{26}}$	1.3171E+02/4.4867E+01(<)	1.0189E+02/8.9557E−01(>)	1.0824E+02/2.3217E+01(≈)	1.1020E+02/2.6558E+01(≈)	1.0535E+02/1.9328E+01(>)	1.1324E+02/2.9720E+01
$f_{a_{27}}$	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00
$f_{a_{28}}$	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00
>/≈/<	2/14/12	2/16/10	2/17/9	2/18/8	7/15/6	-/-/-

Table 3
Algorithm comparisons between LSHADE, jSO, LPalmDE, PaDE, Hip-DE and our algorithms on 30-D optimization. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations max_{nfes} equalling to $10000 \cdot D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant max level $\alpha = 0.05$ in comparison with the new proposed algorithms.

DE variants	LSHADE	jSO	LPalmDE	PaDE	Hip-DE	Our algorithm
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
fa_1	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
fa_2	3.1654E−13/2.5756E−13(<)	1.3390E−10/2.5586E−10(<)	8.2033E−13/1.4032E−12(<)	4.1016E−13/3.2482E−13(<)	6.6429E−13/2.4539E−12(<)	**0.0000E+00/0.0000E+00**
fa_3	1.2898E−01/6.9024E−01(<)	8.3236E−09/5.4693E−08(<)	1.0880E+00/7.4563E+00(<)	4.4021E−02/2.2015E−01(<)	5.9494E−01/2.5762E+00(<)	**1.1343E−10/8.0162E−10**
fa_4	5.7958E−14/1.0008E−13(<)	1.5961E−12/9.2470E−13(<)	1.0700E−13/1.1462E−13(<)	6.2416E−14/1.0248E−13(<)	2.2292E−14/6.8286E−14(<)	**0.0000E+00/0.0000E+00**
fa_5	1.1369E−13/0.0000E+00(<)	1.1146E−13/1.5919E−14(<)	1.1146E−13/1.5919E−14(<)	1.1369E−13/0.0000E+00(<)	1.1592E−13/1.5919E−14(<)	**7.5791E−14/5.4126E−14**
fa_6	8.3077E−09/4.4229E−08(<)	7.2368E−10/2.4841E−09(<)	6.3286E−12/4.0944E−11(<)	1.5597E−11/9.1139E−11(<)	6.4917E−11/4.4844E−10(<)	**3.5666E−14/5.7925E−14**
fa_7	6.0879E−01/4.4764E−01(<)	1.8790E−02/4.4427E−02(≈)	1.8771E−01/2.8314E−01(<)	1.3753E−01/1.8816E−01(<)	1.2862E−01/1.6020E−01(<)	1.3956E−02/3.4215E−02
fa_8	2.0842E+01/1.4562E−01(≈)	2.0937E+01/5.9054E−02(<)	2.0800E+01/1.3129E−01(≈)	2.0770E+01/1.8511E−01(≈)	2.0713E+01/1.7341E−01(>)	2.0846E+01/1.0254E−01
fa_9	2.6654E+01/1.4454E+00(<)	2.3500E+01/3.5501E+00(<)	1.8694E+01/4.0730E+00(≈)	2.5660E+01/2.6997E+00(<)	2.5445E+01/2.1807E+00(<)	1.5905E+01/8.6150E+00
fa_{10}	5.8008E−04/2.0082E−03(≈)	0.0000E+00/0.0000E+00(≈)	7.2472E−04/2.6666E−03(≈)	5.7989E−04/2.3938E−03(≈)	1.4502E−04/1.0357E−03(≈)	0.0000E+00/0.0000E+00
fa_{11}	7.4677E−14/4.1756E−14(>)	1.6161E−13/5.9480E−14(<)	3.4552E−14/2.8029E−14(>)	7.3562E−14/4.7278E−14(>)	**1.8948E−14/2.7063E−14**(>)	1.0923E−13/4.0742E−14
fa_{12}	5.5397E+00/1.7675E+00(≈)	9.0362E+00/2.5876E+00(<)	9.5214E+00/2.8088E+00(<)	6.9708E+00/1.6861E+00(<)	5.9854E+00/1.2339E+00(≈)	5.9607E+00/1.3459E+00
fa_{13}	6.0977E+00/3.2031E+00(>)	9.9228E+00/4.7778E+00(<)	1.3452E+01/6.0810E+00(<)	6.8556E+00/3.0585E+00(≈)	5.2605E+00/2.5633E+00(>)	7.3056E+00/3.5930E+00
fa_{14}	2.4901E−02/2.3191E−02(≈)	1.1759E+01/5.5189E+00(<)	**6.1233E−03/1.1992E−02**(>)	2.1636E−02/2.0382E−02(≈)	2.3677E−02/2.6668E−02(≈)	2.5310E−02/2.2925E−02
fa_{15}	2.6611E+03/2.7629E+02(<)	2.7096E+03/2.9775E+02(<)	2.8517E+03/4.0277E+02(<)	2.7470E+03/2.5305E+02(<)	2.6543E+03/3.0470E+02(<)	**2.5029E+03/2.4410E+02**
fa_{16}	7.5627E−01/1.4254E−01(≈)	2.2904E+00/3.1035E−01(<)	5.6364E−01/3.0153E−01(≈)	6.5732E−01/3.7700E−01(≈)	**3.1942E−01/2.5024E−01**(>)	6.8491E−01/3.6367E−01
fa_{17}	3.0434E+01/9.4299E−07(≈)	3.0673E+01/1.1707E−01(<)	3.0434E+01/1.8285E−06(≈)	3.0434E+01/9.4299E−07(≈)	3.0434E+01/5.4965E−14(≈)	3.0434E+01/5.4880E−14
fa_{18}	5.1891E+01/3.5319E+00(≈)	5.6899E+01/4.7627E+00(<)	**4.5282E+01/3.9113E+00**(>)	5.4183E+01/4.3282E+00(≈)	4.9676E+01/3.2152E+00(>)	5.3361E+01/3.1059E+00
fa_{19}	1.1673E+00/8.6400E−02(≈)	1.2583E+00/9.2718E−02(<)	1.1089E+00/1.6100E−01(>)	1.1302E+00/8.0877E−02(>)	1.1468E+00/9.1206E−02(>)	1.1945E+00/9.9496E−02
fa_{20}	1.0739E+01/1.7551E+00(<)	9.5726E+00/3.9013E−01(<)	9.1786E+00/4.0029E−01(<)	1.0480E+01/2.0089E+00(<)	9.0488E+00/3.5762E−01(≈)	**9.0103E+00/4.5617E−01**
fa_{21}	2.9608E+02/1.9604E+01(≈)	3.0342E+02/4.9361E+01(≈)	2.9804E+02/1.4003E+01(≈)	2.9608E+02/1.9604E+01(≈)	3.0085E+02/2.4726E+01(≈)	2.9608E+02/1.9604E+01
fa_{22}	1.0831E+02/2.5461E+00(<)	1.1945E+02/3.8193E+00(<)	1.0595E+02/3.2666E−01(>)	1.0620E+02/1.2563E+00(≈)	1.0590E+02/3.7285E−01(>)	1.0653E+02/1.9424E+00
fa_{23}	2.5258E+03/2.8207E+02(<)	2.5135E+03/3.2438E+02(<)	2.7840E+03/4.7290E+02(<)	2.7196E+03/3.5112E+02(<)	2.5666E+03/3.0098E+02(<)	**2.3539E+03/2.9944E+02**
fa_{24}	2.0047E+02/1.1076E+00(<)	2.0004E+02/4.9193E−02(≈)	2.0003E+02/8.6633E−02(≈)	2.0001E+02/8.8623E−03(>)	2.0001E+02/2.4049E−02(>)	2.0004E+02/6.4680E−02
fa_{25}	2.4096E+02/6.9199E+00(<)	2.3860E+02/4.7024E+00(<)	2.2094E+02/2.2883E+01(<)	2.2788E+02/2.0328E+01(<)	2.2045E+02/2.3039E+01(<)	**2.0813E+02/1.6720E+01**
fa_{26}	2.0000E+02/0.0000E+00(≈)	2.0000E+02/1.4171E−13(≈)	2.0000E+02/1.5891E−13(≈)	2.0000E+02/3.2155E−14(≈)	2.0000E+02/3.2155E−14(≈)	2.0000E+02/0.0000E+00
fa_{27}	3.0136E+02/3.0731E+00(≈)	3.0092E+02/1.1319E+00(≈)	3.0079E+02/1.4309E+00(≈)	3.0111E+02/3.5641E+00(≈)	3.0034E+02/6.8807E−01(≈)	3.0056E+02/8.6455E−01
fa_{28}	3.0000E+02/0.0000E+00(≈)	3.0000E+02/6.4311E−14(≈)	3.0000E+02/1.0168E−13(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00
>/≈/<	2/13/13	0/8/20	5/11/12	3/13/12	8/10/10	-/-/-

Table 4
Algorithm comparisons between LSHADE, jSO, LPalmDE, PaDE, Hip-DE and our algorithms on 50-D optimization. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations max_{nfes} equalling to $10000 \cdot D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant max level $\alpha = 0.05$ in comparison with the new proposed algorithms.

DE variants	LSHADE	jSO	LPalmDE	PaDE	Hip-DE	Our algorithm
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{a_1}	4.9041E−14/9.4449E−14(<)	5.7958E−14/1.0008E−13(<)	6.6875E−14/1.0463E−13(<)	2.6750E−14/7.3986E−14(<)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
f_{a_2}	6.7423E+02/8.6125E+02(<)	4.0540E+01/7.2598E+01(<)	3.9424E+03/5.1849E+03(<)	2.0307E+03/2.3340E+03(<)	2.1364E+03/2.2367E+03(<)	**1.9455E−06/1.2749E−05**
f_{a_3}	1.2671E+04/6.5125E+04(<)	**2.9443E+00/1.4549E+01**(>)	2.8044E+03/8.3476E+03(<)	1.7408E+03/5.0009E+03(<)	2.2823E+03/1.5217E+04(≈)	1.1544E+02/3.9016E+02
f_{a_4}	7.9086E−11/2.0827E−10(<)	1.5703E−08/2.5171E−08(<)	3.1994E−10/6.4786E−10(<)	1.1436E−11/9.6351E−12(<)	8.1498E−12/5.0062E−12(<)	**8.8274E−13/5.4313E−13**
f_{a_5}	1.4044E−13/4.8704E−14(<)	1.4489E−13/5.1240E−14(<)	1.6050E−13/5.6508E−14(<)	1.8056E−13/5.6508E−14(<)	2.2069E−13/3.5311E−14(<)	**1.1369E−13/0.0000E+00**
f_{a_6}	4.3447E+01/0.0000E+00(<)	4.3447E+01/1.6078E−14(<)	4.3447E+01/2.7019E−14(<)	4.3447E+01/0.0000E+00(<)	4.3447E+01/2.7019E−14(<)	**4.3221E+01/3.6247E−02**
f_{a_7}	2.1970E+00/1.5443E+00(<)	**1.1096E−01/9.8286E−02**(>)	1.3414E+00/1.1895E+00(<)	1.1727E+00/9.9422E−01(<)	5.9187E−01/5.3267E−01(<)	2.5576E−01/1.4433E−01
f_{a_8}	2.1081E+01/1.0125E−01(<)	2.1137E+01/3.6389E−02(<)	2.1034E+01/1.0688E−01(≈)	2.0986E+01/1.5825E−01(≈)	**2.0889E+01/1.8848E−01**(>)	2.1026E+01/1.2408E−01
f_{a_9}	5.3181E+01/1.7825E+00(<)	4.7390E+01/6.0204E+00(<)	4.1004E+01/5.4078E+00(<)	4.8642E+01/7.9741E+00(<)	4.0702E+01/1.2230E+01(<)	**1.3910E+01/3.5949E+00**
$f_{a_{10}}$	9.9034E−03/8.2053E−03(<)	1.3536E−03/2.9715E−03(<)	1.1206E−02/9.8886E−03(<)	1.1783E−02/9.7117E−03(<)	8.3560E−03/8.5167E−03(<)	**1.1146E−15/7.9597E−15**
$f_{a_{11}}$	6.9846E−11/8.9419E−11(<)	6.6023E−09/1.0054E−08(<)	**8.5822E−14/4.6001E−14**(>)	3.5444E−13/7.9246E−14(>)	2.0731E−13/9.2934E−14(>)	4.5698E−13/9.5768E−14
$f_{a_{12}}$	1.4404E+01/2.5423E+00(<)	1.8127E+01/4.0285E+00(<)	2.2598E+01/5.1639E+00(<)	1.7637E+01/2.3633E+00(<)	1.4549E+01/2.9729E+00(<)	**1.2253E+01/2.5801E+00**
$f_{a_{13}}$	2.1767E+01/8.3792E+00(<)	2.6687E+01/1.2368E+01(<)	4.8402E+01/1.5148E+01(<)	3.2929E+01/1.1175E+01(<)	2.5718E+01/6.9772E+00(<)	**1.8199E+01/7.3153E+00**
$f_{a_{14}}$	2.2927E−01/1.1510E−01(<)	6.2689E+01/1.6273E+01(<)	1.2305E−01/6.1027E−02(<)	**6.2610E−02/2.1402E−02**(>)	1.1777E−01/3.2191E−02(<)	8.5573E−02/2.5796E−02
$f_{a_{15}}$	6.3644E+03/4.5097E+02(≈)	6.3182E+03/3.9725E+02(≈)	6.0941E+03/5.7480E+02(≈)	6.3949E+03/4.0486E+02(≈)	6.3772E+03/4.0331E+02(≈)	6.2570E+03/3.7729E+02
$f_{a_{16}}$	1.1599E+00/2.0453E−01(≈)	3.1041E+00/5.1843E−01(<)	8.5795E−01/3.5467E−01(>)	1.0205E+00/5.2379E−01(≈)	**6.1986E−01/4.3286E−01**(>)	1.2336E+00/5.5031E−01
$f_{a_{17}}$	5.0788E+01/4.4648E−03(<)	5.2455E+01/3.4576E−01(<)	5.0786E+01/2.7668E−10(≈)	5.0786E+01/4.1392E−09(≈)	5.0786E+01/2.6464E−03(≈)	5.0786E+01/3.1537E−08
$f_{a_{18}}$	1.0282E+02/6.5621E+00(≈)	1.1094E+02/7.7117E+00(<)	**7.5292E+01/8.8673E+00**(>)	1.0465E+02/6.7557E+00(≈)	9.9416E+01/4.6798E+00(>)	1.0235E+02/6.3829E+00
$f_{a_{19}}$	2.5539E+00/1.3281E−01(<)	2.6398E+00/1.8963E−01(<)	2.3650E+00/3.0153E−01(≈)	2.3119E+00/1.4224E−01(>)	2.4978E+00/1.4014E−01(<)	2.3963E+00/1.3273E−01
$f_{a_{20}}$	1.8164E+01/5.1610E−01(<)	1.8836E+01/4.8032E−01(<)	1.7691E+01/5.7769E−01(<)	1.7824E+01/7.7552E−01(<)	1.7773E+01/4.4979E−01(<)	**1.7453E+01/5.1884E−01**
$f_{a_{21}}$	7.6054E+02/4.5472E+02(≈)	7.6004E+02/3.9949E+02(≈)	9.8441E+02/3.2088E+02(≈)	9.6632E+02/3.3846E+02(≈)	8.1606E+02/4.2597E+02(≈)	8.4102E+02/4.0700E+02
$f_{a_{22}}$	1.3713E+01/1.3369E+00(<)	5.5648E+01/1.0764E+01(<)	1.1485E+01/7.7771E−01(>)	1.1799E+01/2.9569E−01(>)	1.1718E+01/5.6189E−01(>)	1.2014E+01/5.8433E−01
$f_{a_{23}}$	5.8224E+03/3.3537E+02(≈)	5.5551E+03/5.6912E+02(≈)	6.0379E+03/5.7537E+02(<)	6.2467E+03/4.9915E+02(<)	6.1175E+03/4.6767E+02(<)	5.7254E+03/4.6894E+02
$f_{a_{24}}$	2.1102E+02/4.9491E+00(<)	**2.0055E+02/6.3083E−01**(>)	2.0505E+02/3.6720E+00(<)	2.0357E+02/3.0209E+00(<)	2.0116E+02/1.0748E+00(>)	2.0191E+02/6.5132E−01
$f_{a_{25}}$	2.7714E+02/6.1710E+00(>)	2.7652E+02/7.5487E+00(>)	2.8794E+02/6.6974E+00(≈)	2.8668E+02/7.1483E+00(≈)	2.8497E+02/5.9786E+00(≈)	2.8541E+02/7.3247E+00
$f_{a_{26}}$	2.5410E+02/5.3639E+01(≈)	**2.1214E+02/3.3582E+01**(>)	2.6044E+02/5.3225E+01(≈)	2.5842E+02/4.7252E+01(≈)	2.5451E+02/5.1935E+01(≈)	2.5742E+02/5.2569E+01
$f_{a_{27}}$	4.1463E+02/8.2186E+01(<)	3.3410E+02/6.3233E+01(>)	3.6513E+02/3.2188E+01(<)	3.4896E+02/3.7368E+01(<)	3.2085E+02/1.3853E+01(>)	3.3181E+02/1.3131E+01
$f_{a_{28}}$	4.0000E+02/0.0000E+00(≈)	4.0000E+02/2.8433E−13(≈)	4.0000E+02/2.8705E−13(≈)	4.0000E+02/0.0000E+00(≈)	4.0000E+02/7.1902E−14(≈)	4.0000E+02/1.4448E−13
>/≈/<	1/7/20	6/4/18	4/8/16	4/9/15	7/8/13	-/-/-

Table 5
Algorithm comparisons between LSHADE, jSO, LPalmDE, PaDE, Hip-DE and our algorithms on 10-D optimization. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations max_{nfes} equalling to $10000 \cdot D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant max level $\alpha = 0.05$ in comparison with the new proposed algorithms.

DE variants	LSHADE	jSO	LPalmDE	PaDE	Hip-DE	Our algorithm
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{b_1}	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
f_{b_2}	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
f_{b_3}	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
f_{b_4}	3.1541E+01/9.9473E+00(<)	1.9862E+01/1.7309E+01(<)	2.8046E+01/1.3785E+01(<)	2.5318E+01/1.5547E+01(<)	3.0689E+01/1.1317E+01(<)	**1.3716E+01/5.2858E+00**
f_{b_5}	1.4893E+01/8.3835E+00(≈)	1.5542E+01/8.0558E+00(≈)	1.5641E+01/8.2875E+00(≈)	1.2161E+01/8.9643E+00(≈)	1.4699E+01/8.6864E+00(≈)	1.7062E+01/6.9147E+00
f_{b_6}	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
f_{b_7}	2.1280E−03/5.9728E−03(>)	1.4011E−03/3.7606E−03(>)	**4.3468E−04/2.2947E−03**(>)	3.5748E−03/6.3340E−03(>)	4.3507E−04/1.7576E−03(>)	1.8073E−02/2.2128E−02
f_{b_8}	0.0000E+00/0.0000E+00(≈)	3.9018E−02/1.9505E−01(<)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
f_{b_9}	2.4399E+00/8.2825E−01(<)	2.3411E+00/9.0806E−01(≈)	2.7898E+00/1.4352E+00(<)	2.3258E+00/7.3560E−01(<)	1.8555E+00/8.4424E−01(≈)	1.8548E+00/1.1766E+00
$f_{b_{10}}$	3.6738E−03/1.4841E−02(≈)	1.4369E+00/3.0675E+00(<)	0.0000E+00/0.0000E+00(>)	4.8984E−03/1.6958E−02(≈)	1.2246E−03/8.7454E−03(>)	6.1230E−03/1.8757E−02
$f_{b_{11}}$	2.6262E+01/3.2833E+01(<)	7.9617E+01/1.0361E+02(<)	6.2484E+01/6.8591E+01(<)	2.6300E+01/2.8542E+01(<)	3.5211E+01/5.9586E+01(≈)	**1.8229E+01/2.3613E+01**
$f_{b_{12}}$	6.8688E−02/1.8600E−02(≈)	2.7570E−01/1.3066E−01(<)	**3.4043E−02/3.7980E−02**(>)	9.0799E−02/3.2253E−02(<)	5.0180E−02/3.5362E−02(≈)	6.8776E−02/5.6237E−02
$f_{b_{13}}$	5.0926E−02/1.2680E−02(<)	8.2189E−02/1.6372E−02(<)	3.5755E−02/1.7542E−02(>)	4.8641E−02/1.4137E−02(≈)	3.8281E−02/1.9252E−02(>)	4.6142E−02/1.2939E−02
$f_{b_{14}}$	7.6585E−02/3.0235E−02(<)	6.0012E−02/1.6503E−02(≈)	8.9253E−02/3.0849E−02(<)	8.5169E−02/2.5778E−02(<)	9.3147E−02/3.3580E−02(<)	6.1237E−02/2.5511E−02
$f_{b_{15}}$	3.7523E−01/7.1027E−02(≈)	4.8472E−01/1.2255E−01(<)	4.4600E−01/1.3391E−01(<)	3.9808E−01/6.7612E−02(<)	3.3934E−01/8.2343E−02(≈)	3.6876E−01/7.5188E−02
$f_{b_{16}}$	1.1545E+00/2.8465E−01(<)	1.1089E+00/2.9998E−01(<)	1.0983E+00/5.2778E−01(<)	9.9508E−01/3.2954E−01(<)	1.0054E+00/3.2978E−01(<)	**7.5842E−01/2.4470E−01**
$f_{b_{17}}$	9.8877E−01/9.0234E−01(≈)	3.0717E+00/5.8267E+00(<)	7.9561E−01/6.5280E−01(≈)	6.7040E−01/7.4298E−01(≈)	1.2062E+00/1.0777E+00(<)	1.1454E+00/2.2027E+00
$f_{b_{18}}$	2.1746E−01/1.7456E−01(<)	1.6407E−01/1.2149E−01(<)	5.3948E−02/6.4774E−02(≈)	1.1140E−01/1.1490E−01(<)	1.2709E−01/2.3341E−01(≈)	7.7439E−02/1.1637E−01
$f_{b_{19}}$	7.8554E−02/3.8712E−02(<)	3.8152E−02/2.4090E−02(≈)	5.6308E−02/3.4264E−02(≈)	5.3516E−02/2.6529E−02(<)	7.5280E−02/5.5976E−02(<)	4.5246E−02/2.1651E−02
$f_{b_{20}}$	1.3303E−01/1.0672E−01(≈)	2.9416E−01/2.0139E−01(<)	1.4108E−01/1.5277E−01(≈)	1.1694E−01/1.2425E−01(≈)	2.4634E−01/1.9335E−01(≈)	1.8400E−01/1.8034E−01
$f_{b_{21}}$	3.7363E−01/3.0179E−01(<)	4.8790E−01/3.2496E−01(<)	1.8875E−01/2.5821E−01(≈)	2.5773E−01/2.2981E−01(≈)	2.9136E−01/2.8283E−01(≈)	5.5165E−01/2.3311E+00
$f_{b_{22}}$	7.1247E−02/3.0817E−02(>)	1.1229E+00/3.9206E+00(<)	**3.4310E−02/4.1181E−02**(>)	8.2190E−02/3.2127E−02(>)	8.9447E−02/3.3048E−02(>)	1.1241E−01/4.0407E−02
$f_{b_{23}}$	3.2946E+02/0.0000E+00(≈)	3.2300E+02/4.6133E+01(≈)	3.2946E+02/2.8705E−13(≈)	3.2946E+02/0.0000E+00(≈)	3.2946E+02/0.0000E+00(≈)	3.2946E+02/0.0000E+00
$f_{b_{24}}$	1.0745E+02/2.0556E+00(<)	1.0812E+02/2.2690E+00(<)	1.0741E+02/3.1532E+00(<)	1.0649E+02/2.2496E+00(≈)	1.0522E+02/3.2771E+00(≈)	1.0609E+02/2.9599E+00
$f_{b_{25}}$	1.3992E+02/4.2483E+01(≈)	1.2348E+02/2.6870E+01(>)	1.3970E+02/3.5902E+01(≈)	1.2274E+02/2.6140E+01(>)	1.2733E+02/3.3534E+01(>)	1.3300E+02/2.9473E+01
$f_{b_{26}}$	1.0005E+02/1.4936E−02(<)	1.0008E+02/1.5389E−02(<)	**1.0003E+02/1.3563E−02**(>)	1.0006E+02/1.7953E−02(≈)	1.0005E+02/2.2598E−02(≈)	1.0004E+02/1.7499E−02
$f_{b_{27}}$	6.7747E+01/1.2854E+02(≈)	1.6986E+01/7.8178E+01(≈)	7.2412E+00/4.1816E+01(≈)	4.0325E+01/1.0930E+02(≈)	5.9910E+01/1.1975E+02(≈)	7.7825E+01/1.4126E+02
$f_{b_{28}}$	3.8668E+02/3.9914E+01(<)	3.8570E+02/4.8666E+01(≈)	3.8991E+02/5.3111E+01(≈)	3.9002E+02/5.1787E+01(≈)	3.9583E+02/5.3116E+01(≈)	**3.8325E+02/4.7179E+01**
$f_{b_{29}}$	2.2197E+02/4.1379E−01(<)	2.2181E+02/2.3045E−01(<)	2.2181E+02/2.2898E−01(≈)	2.2285E+02/2.8845E−01(<)	2.2283E+02/2.6392E−01(<)	**2.2165E+02/5.2063E−01**
$f_{b_{30}}$	**4.6321E+02/5.2134E+00**(>)	4.6675E+02/9.0090E+00(≈)	4.6974E+02/1.1657E+01(≈)	4.6994E+02/9.2280E+00(≈)	4.6372E+02/2.6359E+00(≈)	4.7046E+02/1.5872E+01
>/≈/<	3/14/13	2/12/16	6/17/7	3/17/10	5/19/6	-/-/-

Table 6
Algorithm comparisons between LSHADE, jSO, LPalmDE, PaDE, Hip-DE and our algorithms on 30-D optimization. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations max_{nfes} equalling to $10000 \cdot D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant max level $\alpha = 0.05$ in comparison with the new proposed algorithms.

DE variants	LSHADE	jSO	LPalmDE	PaDE	Hip-DE	Our algorithm
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{b_1}	9.4739E−15/6.7657E−15(<)	4.2633E−14/2.9944E−14(<)	1.4211E−14/5.6843E−15(<)	1.4211E−14/8.5265E−15(<)	1.0310E−14/7.0072E−15(<)	**2.7864E−16/1.9899E−15**
f_{b_2}	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00*
f_{b_3}	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*2.2292E−15/1.1144E−14(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00*
f_{b_4}	5.6843E−14/3.5951E−14(<)	5.9073E−14/1.9562E−14(<)	3.5666E−14/2.7756E−14(<)	4.5698E−14/2.5471E−14(<)	5.2385E−14/2.7481E−14(<)	**0.0000E+00/0.0000E+00**
f_{b_5}	2.0119E+01/2.3934E−02(>)	2.0759E+01/1.6120E−01(<)	*2.0066E+01/7.4238E−02(>)*	2.0166E+01/3.7188E−02(≈)	*2.0068E+01/5.1551E−02(>)*	2.0145E+01/7.8672E−02
f_{b_6}	*6.2972E−07/4.4971E−06(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*1.9680E−04/1.1748E−03*
f_{b_7}	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00(≈)*	*0.0000E+00/0.0000E+00*
f_{b_8}	1.2483E−13/3.4143E−14(>)	2.9658E+00/4.2518E+00(<)	5.5729E−14/5.7398E−14(>)	1.5381E−13/1.0123E−13(>)	**3.3437E−14/5.2316E−14**(>)	2.0508E−13/8.8119E−14
f_{b_9}	**6.4144E+00/1.5112E+00**(>)	1.1394E+01/3.0435E+00(<)	1.1782E+01/3.1407E+00(<)	8.2457E+00/1.8120E+00(≈)	7.1198E+00/1.1622E+00(≈)	7.7300E+00/1.7025E+00
$f_{b_{10}}$	3.6740E−03/8.0157E−03(<)	4.0943E+01/1.4683E+02(<)	**1.3197E−12/1.2643E−12**(>)	8.1644E−04/4.0814E−03(≈)	1.6329E−03/7.0209E−03(≈)	8.1644E−04/4.0814E−03
$f_{b_{11}}$	*1.2144E+03/1.8854E+02(≈)*	2.0396E+03/5.1897E+02(<)	1.4012E+03/2.8354E+02(<)	*1.2152E+03/2.1827E+02(≈)*	*1.2380E+03/2.0173E+02(≈)*	*1.1980E+03/2.1311E+02*
$f_{b_{12}}$	1.5811E−01/2.8434E−02(>)	7.4891E−01/4.4729E−01(<)	1.4456E−01/6.7727E−02(>)	1.8211E−01/4.0776E−02(≈)	**1.3268E−01/3.7064E−02**(>)	1.8472E−01/2.8697E−02
$f_{b_{13}}$	1.1687E−01/1.7832E−02(>)	1.5161E−01/2.2846E−02(<)	*1.0907E−01/2.6222E−02(>)*	*1.1825E−01/1.8391E−02(≈)*	*1.1358E−01/1.7614E−02(>)*	1.2358E−01/1.7047E−02
$f_{b_{14}}$	2.3833E−01/2.4433E−02(<)	2.0234E−01/2.7517E−02(<)	2.1999E−01/2.8498E−02(<)	2.1345E−01/2.3818E−02(<)	2.2812E−01/2.5689E−02(<)	**1.6163E−01/2.5255E−02**
$f_{b_{15}}$	*2.1264E+00/2.4035E−01(≈)*	3.2651E+00/6.1435E−01(<)	*2.0946E+00/4.8430E−01(≈)*	*2.2001E+00/2.2639E−01(≈)*	*2.1643E+00/2.2567E−01(≈)*	*2.1011E+00/2.7723E−01*
$f_{b_{16}}$	8.5244E+00/4.3208E−01(<)	9.3582E+00/7.1362E−01(<)	8.7497E+00/6.5272E−01(<)	8.6542E+00/5.3550E−01(<)	8.6613E+00/4.4900E−01(<)	**8.1404E+00/4.2265E−01**
$f_{b_{17}}$	2.1952E+02/1.0531E+02(<)	7.8189E+01/3.6285E+01(<)	2.2414E+02/1.0103E+02(<)	2.0448E+02/9.8168E+01(<)	2.1157E+02/9.7936E+01(<)	**6.6735E+01/3.5883E+01**
$f_{b_{18}}$	5.8581E+00/2.5188E+00(<)	*2.1400E+00/1.3711E+00(≈)*	6.8491E+00/3.3113E+00(<)	5.8067E+00/2.3911E+00(<)	8.7687E+00/4.1251E+00(<)	*2.4746E+00/1.6339E+00*
$f_{b_{19}}$	3.6776E+00/5.1518E−01(<)	2.3563E+00/5.9773E−01(<)	3.1043E+00/5.2631E−01(<)	3.4699E+00/5.7841E−01(<)	3.5779E+00/5.3213E−01(<)	**1.8612E+00/6.7950E−01**
$f_{b_{20}}$	*2.3661E+00/1.0688E+00(≈)*	*2.6261E+00/1.0587E+00(<)*	3.5070E+00/1.2706E+00(<)	3.1565E+00/1.4108E+00(<)	*2.8417E+00/1.1984E+00(<)*	*2.1129E+00/8.2522E−01*
$f_{b_{21}}$	6.2906E+01/6.6881E+01(<)	3.2913E+01/5.0125E+01(<)	9.2864E+01/7.7546E+01(<)	7.0374E+01/6.8848E+01(<)	6.4661E+01/6.3208E+01(<)	**1.2781E+01/2.8950E+01**
$f_{b_{22}}$	**2.7112E+01/1.7458E+01**(>)	6.7141E+01/5.5945E+01(≈)	8.3125E+01/6.1693E+01(≈)	6.1663E+01/4.9281E+01(≈)	7.4355E+01/5.7613E+01(≈)	6.1243E+01/5.2822E+01
$f_{b_{23}}$	3.1524E+02/3.5915E−13(<)	3.1524E+02/6.4311E−14(<)	3.1524E+02/4.0186E−13(<)	3.1524E+02/0.0000E+00(<)	3.1524E+02/0.0000E+00(<)	**3.1521E+02/1.4040E−02**
$f_{b_{24}}$	2.2384E+02/1.1857E+00(<)	*2.0043E+02/3.0427E+00(≈)*	2.2420E+02/1.9249E+00(<)	2.2348E+02/8.8410E−01(<)	2.2309E+02/9.8241E−01(<)	*2.0170E+02/5.8689E+00*
$f_{b_{25}}$	2.0261E+02/9.0783E−02(<)	2.0258E+02/3.6873E−02(<)	2.0274E+02/1.8059E−01(<)	2.0270E+02/1.6398E−01(<)	2.0269E+02/1.3055E−01(<)	**2.0253E+02/5.7468E−02**
$f_{b_{26}}$	1.0011E+02/1.5534E−02(≈)	1.0015E+02/2.4329E−02(<)	**1.0010E+02/2.7974E−02**(>)	1.0012E+02/1.6644E−02(≈)	1.0011E+02/1.4342E−02(>)	1.0012E+02/1.6842E−02
$f_{b_{27}}$	*3.0000E+02/1.2862E−13(≈)*	*3.0000E+02/1.9493E−13(≈)*	*3.0000E+02/9.0949E−14(≈)*	*3.0196E+02/1.4003E+01(≈)*	*3.0000E+02/6.4311E−14(≈)*	*3.0000E+02/0.0000E+00*
$f_{b_{28}}$	8.2834E+02/2.0412E+01(<)	8.2858E+02/2.3826E+01(<)	8.5985E+02/1.5568E+01(<)	8.5756E+02/1.7105E+01(<)	8.6139E+02/2.0977E+01(<)	**8.1501E+02/2.3528E+01**
$f_{b_{29}}$	7.1608E+02/2.7147E+00(<)	5.9936E+02/2.2692E+02(<)	6.5187E+02/1.8294E+02(<)	7.1834E+02/4.9874E+00(<)	7.1769E+02/3.2011E+00(<)	**4.9194E+02/2.8265E+02**
$f_{b_{30}}$	1.3276E+03/5.5426E+02(<)	4.3295E+02/2.5195E+01(<)	6.5393E+02/2.5669E+02(<)	5.5537E+02/1.7305E+02(<)	5.7850E+02/2.1047E+02(<)	**3.8332E+02/2.5117E+01**
>/≈/<	6/9/15	0/8/22	6/7/17	1/14/15	5/10/15	-/-/-

Table 7
Algorithm comparisons between LSHADE, jSO, LPalmDE, PaDE, Hip-DE and our algorithms on 50-D optimization. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations max_{nfes} equalling to $10000 \cdot D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant max level $\alpha = 0.05$ in comparison with the new proposed algorithms.

DE variants	LSHADE	jSO	LPalmDE	PaDE	Hip-DE	Our algorithm
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f_{b_1}	4.6323E+02/6.5743E+02(<)	1.8680E+01/5.2984E+01(<)	3.1719E+03/3.3603E+03(<)	6.5768E+02/1.0684E+03(<)	1.3129E+03/1.6470E+03(<)	**2.6024E−05/5.8085E−05**
f_{b_2}	*3.5666E−14/1.2510E−14(≈)*	5.8515E−14/1.8342E−14(<)	*3.9010E−14/1.3878E−14(≈)*	4.2354E−14/1.5434E−14(<)	4.4026E−14/1.5373E−14(<)	*3.3995E−14/1.1397E−14*
f_{b_3}	5.5729E−14/7.9597E−15(<)	6.3531E−14/1.8497E−14(<)	5.6843E−14/1.1369E−14(<)	5.5729E−14/7.9597E−15(<)	5.5729E−14/7.9597E−15(<)	**2.5635E−14/2.8566E−14**
f_{b_4}	5.0432E+01/4.7448E+01(<)	4.8279E+01/4.9343E+01(<)	*7.8758E+00/2.6585E+01(≈)*	*1.1724E+01/3.1855E+01(≈)*	1.4624E+01/3.2171E+01(<)	*5.1415E+00/1.9337E+01*
f_{b_5}	2.0266E+01/3.0727E−02(>)	2.1079E+01/1.3235E−01(<)	*2.0161E+01/1.4449E−01(>)*	2.0280E+01/7.9639E−02(>)	*2.0154E+01/1.0827E−01(>)*	2.0333E+01/7.3122E−02
f_{b_6}	3.7164E−01/6.2436E−01(>)	3.1160E−02/2.0308E−01(>)	7.7033E−01/2.8885E+00(>)	2.8863E−01/6.1836E−01(>)	**5.1826E−03/2.2898E−02**(>)	7.4575E−01/3.6682E−01
f_{b_7}	8.6937E−14/4.8704E−14(<)	1.0477E−13/3.0869E−14(<)	1.0254E−13/3.4143E−14(<)	9.5854E−14/4.1756E−14(<)	6.6875E−14/5.6508E−14(<)	**1.3375E−14/3.6993E−14**
f_{b_8}	1.1716E−10/1.6197E−10(<)	2.4691E−08/9.8723E−08(<)	**1.5827E−13/7.2183E−14**(>)	7.4454E−13/1.3889E−13(>)	3.9679E−13/1.8663E−13(>)	1.0143E−12/3.0913E−13
f_{b_9}	1.1381E+01/2.3529E+00(<)	1.6330E+01/2.8643E+00(<)	2.2467E+01/4.3700E+00(<)	1.5527E+01/2.4750E+00(<)	1.2618E+01/2.0666E+00(<)	**1.0081E+01/2.5147E+00**
$f_{b_{10}}$	4.4257E−02/2.1457E−02(<)	1.0186E+01/3.1509E+00(<)	*6.1685E−03/9.1693E−03(>)*	*8.6409E−03/9.4747E−03(>)*	1.0672E−02/1.1276E−02(>)	2.3701E−02/1.5921E−02
$f_{b_{11}}$	3.2807E+03/2.7862E+02(>)	3.3736E+03/3.1922E+02(>)	3.5727E+03/4.5657E+02(≈)	*3.1578E+03/3.0637E+02(>)*	*3.2697E+03/2.7049E+02(>)*	3.5229E+03/3.0303E+02
$f_{b_{12}}$	2.0778E−01/2.8704E−02(>)	3.7038E−01/4.8703E−01(<)	1.9923E−01/7.0786E−02(>)	2.2825E−01/4.6021E−02(>)	**1.9121E−01/3.5817E−02**(>)	2.4989E−01/4.0413E−02
$f_{b_{13}}$	1.6400E−01/1.6833E−02(<)	1.9599E−01/3.5732E−02(<)	1.6605E−01/2.7013E−02(<)	1.8438E−01/2.2301E−02(<)	1.7280E−01/1.5395E−02(<)	**7.2255E−02/1.3619E−02**
$f_{b_{14}}$	3.0734E−01/2.0586E−02(≈)	*2.8395E−01/4.7225E−02(>)*	3.0930E−01/2.5282E−02(≈)	*2.8821E−01/2.3309E−02(>)*	3.0511E−01/2.0451E−02(>)	3.1371E−01/3.7704E−02
$f_{b_{15}}$	5.1425E+00/4.9110E−01(<)	5.5266E+00/4.5823E−01(<)	*4.3985E+00/9.4224E−01(≈)*	5.1300E+00/4.0739E−01(<)	5.1936E+00/4.5943E−01(<)	*4.4314E+00/4.5177E−01*
$f_{b_{16}}$	*1.6934E+01/4.2713E−01(>)*	*1.6955E+01/4.5562E−01(>)*	*1.7034E+01/7.8666E−01(≈)*	*1.6750E+01/5.0161E−01(>)*	*1.7005E+01/4.8695E−01(≈)*	1.7171E+01/5.2245E−01
$f_{b_{17}}$	1.4631E+03/4.3951E+02(<)	3.5096E+02/1.8031E+02(<)	1.6354E+03/4.4952E+02(<)	1.6538E+03/3.8183E+02(<)	1.7422E+03/4.4036E+02(<)	**1.6296E+02/1.2283E+02**
$f_{b_{18}}$	1.0261E+02/1.5044E+01(<)	1.2017E+01/3.8417E+00(<)	1.0063E+02/1.3176E+01(<)	1.0100E+02/1.5585E+01(<)	1.0408E+02/1.3951E+01(<)	**8.1830E+00/3.4189E+00**
$f_{b_{19}}$	8.5149E+00/1.9699E+00(≈)	9.3987E+00/5.9690E−01(<)	8.1634E+00/1.6954E+00(>)	7.8896E+00/1.8827E+00(>)	**7.5140E+00/1.7928E+00**(>)	8.7430E+00/9.6323E−01
$f_{b_{20}}$	1.2638E+01/4.0514E+00(<)	5.6288E+00/1.7718E+00(<)	1.7466E+01/4.2271E+00(<)	1.4524E+01/5.9873E+00(<)	1.4155E+01/4.8204E+00(<)	**4.1451E+00/1.5549E+00**
$f_{b_{21}}$	4.8600E+02/1.4933E+02(<)	2.8415E+02/8.7876E+01(<)	6.0434E+02/1.6932E+02(<)	5.2876E+02/1.4034E+02(<)	5.1903E+02/1.5543E+02(<)	**2.0914E+02/7.1318E+01**
$f_{b_{22}}$	1.1532E+02/8.5949E+01(<)	1.4103E+02/8.3745E+01(<)	2.0456E+02/8.9644E+01(<)	1.3086E+02/5.7998E+01(<)	1.4872E+02/5.9586E+01(<)	**5.9820E+01/5.5243E+01**
$f_{b_{23}}$	3.4400E+02/1.1139E−13(<)	3.4400E+02/3.8986E−13(<)	3.4400E+02/5.1724E−13(<)	3.4400E+02/1.7667E−13(<)	3.4400E+02/9.0949E−14(<)	**3.4345E+02/5.9094E−01**
$f_{b_{24}}$	2.7512E+02/7.2698E−01(<)	2.7179E+02/2.0112E+00(<)	2.7513E+02/9.8034E−01(<)	2.7508E+02/1.0715E+00(<)	2.7491E+02/7.0369E−01(<)	**2.6836E+02/1.9560E+00**
$f_{b_{25}}$	2.0530E+02/3.2508E−01(<)	2.0500E+02/1.4826E−01(<)	2.0584E+02/4.4908E−01(<)	2.0586E+02/5.5987E−01(<)	2.0568E+02/3.4811E−01(<)	**2.0384E+02/3.5140E−01**
$f_{b_{26}}$	1.0016E+02/1.5470E−02(<)	1.0019E+02/3.1938E−02(<)	1.1582E+02/3.6669E+01(<)	1.0996E+02/2.9982E+01(<)	1.1583E+02/3.6666E+01(<)	**1.0009E+02/4.3558E−02**
$f_{b_{27}}$	3.3281E+02/3.3079E+01(<)	*3.0997E+02/1.9755E+01(≈)*	3.1599E+02/2.5979E+01(≈)	3.2435E+02/3.1123E+01(≈)	*3.1243E+02/2.4083E+01(≈)*	*3.0649E+02/2.6726E+00*
$f_{b_{28}}$	1.1142E+03/2.7419E+01(<)	1.0850E+03/2.9862E+01(<)	1.2747E+03/6.1095E+01(<)	1.2621E+03/4.5445E+01(<)	1.2504E+03/5.4025E+01(<)	**6.4287E+02/2.4120E+02**
$f_{b_{29}}$	8.1276E+02/4.5464E+01(<)	8.1535E+02/5.2901E+01(<)	6.1499E+02/1.2041E+02(<)	5.7430E+02/1.2837E+02(<)	6.1881E+02/1.0444E+02(<)	**4.6887E+02/5.2506E+01**
$f_{b_{30}}$	8.7786E+03/4.4313E+02(<)	8.3041E+03/3.2456E+02(<)	9.6906E+03/6.4480E+02(<)	9.4812E+03/6.4061E+02(<)	9.2567E+03/5.7401E+02(<)	**3.7069E+03/1.2982E+03**
>/≈/<	5/3/22	4/1/25	6/7/17	9/2/19	8/2/20	-/-/-

ARTICLE IN PRESS

Table 8
Algorithm comparisons between LSHADE, jSO, LPalmDE, PaDE, Hip-DE and our algorithms on 10-D optimization. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations max_{nfes} equalling to $10000 \cdot D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant max level $\alpha = 0.05$ in comparison with the new proposed algorithms.

DE variants	LSHADE	jSO	LPalmDE	PaDE	Hip-DE	Our algorithm
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
fc_1	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
fc_2	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
fc_3	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
fc_4	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
fc_5	2.6361E+00/8.1587E−01(<)	1.8729E+00/8.3593E−01(≈)	2.6337E+00/1.1226E+00(≈)	2.3242E+00/9.0518E−01(≈)	1.9525E+00/8.1945E−01(≈)	2.2249E+00/1.6950E+00
fc_6	4.4583E−15/2.2287E−14(≈)	0.0000E+00/0.0000E+00(≈)	1.7833E−14/4.1756E−14(<)	2.2292E−15/1.5919E−14(≈)	0.0000E+00/0.0000E+00(≈)	2.2292E−15/1.5919E−14
fc_7	1.2097E+01/6.3890E−01(≈)	1.2078E+01/7.2070E−01(≈)	1.2762E+01/1.3630E+00(≈)	1.2272E+01/1.1740E+00(≈)	**1.1763E+01/6.2667E−01**(>)	1.2407E+01/1.2329E+00
fc_8	2.4414E+00/8.7339E−01(≈)	2.0289E+00/8.4333E−01(≈)	2.5362E+00/1.1656E+00(≈)	2.4509E+00/8.8651E−01(≈)	2.1087E+00/7.8737E−01(≈)	2.4596E+00/1.7613E+00
fc_9	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
fc_{10}	2.8095E+01/4.2078E+01(<)	2.8622E+01/4.5822E+01(≈)	8.6440E+01/8.1759E+01(<)	3.1724E+01/4.6507E+01(<)	5.4977E+01/1.0564E+02(<)	**1.9349E+01/3.9317E+01**
fc_{11}	3.9590E−01/6.8874E−01(<)	4.4583E−15/3.1839E−14(≈)	0.0000E+00/0.0000E+00(≈)	6.9781E−01/8.1319E−01(<)	2.8376E−01/6.5957E−01(<)	0.0000E+00/0.0000E+00
fc_{12}	3.4019E+01/5.4563E+01(≈)	5.0792E+00/2.3459E+01(≈)	3.5466E+01/5.5162E+01(≈)	2.3887E+01/4.8343E+01(≈)	3.3247E+01/5.8590E+01(≈)	3.7404E+01/5.5570E+01
fc_{13}	3.5270E+00/2.2884E+00(<)	2.9004E+00/2.3060E+00(≈)	1.5215E+00/2.1529E+00(>)	1.8480E+00/2.3046E+00(≈)	2.1097E+00/2.5678E+00(≈)	2.5199E+00/2.5022E+00
fc_{14}	2.0907E−01/4.3055E−01(≈)	1.3656E−01/3.4579E−01(≈)	**6.0134E−02/2.3631E−01**(>)	1.8461E−01/4.1138E−01(≈)	8.5840E−01/7.9645E−01(<)	1.4829E−01/3.9563E−01
fc_{15}	1.8762E−01/2.1487E−01(<)	2.9709E−01/2.1862E−01(<)	1.1735E−01/1.8595E−01(≈)	1.5620E−01/2.0004E−01(<)	1.3775E−01/2.0125E−01(≈)	**7.3693E−02/1.3339E−01**
fc_{16}	3.4327E−01/1.7852E−01(≈)	5.1955E−01/2.9781E−01(<)	**1.9013E−01/1.6622E−01**(>)	3.8704E−01/1.8271E−01(≈)	4.4220E−01/2.4157E−01(<)	3.3010E−01/1.8807E−01
fc_{17}	1.3891E−01/1.4731E−01(≈)	4.9298E−01/3.9129E−01(<)	2.1111E−01/3.4588E−01(≈)	1.5382E−01/1.4829E−01(≈)	7.9989E−01/2.7181E+00(≈)	1.4921E−01/1.6454E−01
fc_{18}	2.3745E−01/2.0231E−01(≈)	2.5619E−01/1.9455E−01(≈)	2.3845E−01/2.2158E−01(≈)	1.9680E−01/1.9414E−01(≈)	1.5345E−01/1.6948E−01(≈)	1.8524E−01/2.1302E−01
fc_{19}	1.1055E−02/1.0560E−02(<)	9.4262E−03/1.0798E−02(≈)	1.3548E−02/1.2753E−02(≈)	1.0679E−02/1.0843E−02(≈)	1.4125E−02/2.5601E−02(≈)	**6.7524E−03/9.5801E−03**
fc_{20}	0.0000E+00/0.0000E+00(≈)	3.6114E−01/1.3055E−01(<)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	6.1210E−03/4.3713E−02(≈)	6.1210E−03/4.3713E−02
fc_{21}	1.5359E+02/5.1750E+01(≈)	1.4656E+02/5.1886E+01(≈)	1.5289E+02/5.2388E+01(≈)	1.4947E+02/5.1905E+01(≈)	1.6097E+02/5.0211E+01(≈)	1.6085E+02/5.1423E+01
fc_{22}	1.0001E+02/5.6308E−02(≈)	1.0000E+02/1.0925E−13(≈)	1.0000E+02/1.2368E−13(≈)	1.0001E+02/4.0631E−02(≈)	1.0000E+02/0.0000E+00(≈)	1.0001E+02/6.2390E−02
fc_{23}	3.0328E+02/1.4013E+00(<)	3.0146E+02/1.6873E+00(≈)	3.0135E+02/1.6340E+00(≈)	3.0157E+02/1.5799E+00(<)	3.0122E+02/1.4469E+00(≈)	3.0103E+02/1.6424E+00
fc_{24}	3.0366E+02/7.5121E+01(<)	2.7055E+02/1.0084E+02(≈)	2.7346E+02/9.2398E+01(>)	2.8226E+02/9.1087E+01(≈)	2.9062E+02/8.2896E+01(>)	3.0037E+02/7.4121E+01
fc_{25}	4.0881E+02/1.9694E+01(≈)	4.0953E+02/1.9977E+01(≈)	4.1752E+02/2.2713E+01(≈)	4.1931E+02/2.2889E+01(≈)	4.1663E+02/2.2574E+01(≈)	4.1665E+02/2.2663E+01
fc_{26}	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00(≈)	3.0000E+02/0.0000E+00
fc_{27}	3.8941E+02/2.1287E−01(>)	3.8936E+02/2.4015E−01(>)	3.9305E+02/1.7129E+00(≈)	3.9306E+02/1.7282E+00(≈)	3.9358E+02/1.2076E+00(<)	3.9298E+02/1.6554E+00
fc_{28}	3.5894E+02/1.2069E+02(<)	3.1340E+02/5.7489E+01(≈)	3.1113E+02/5.5625E+01(≈)	3.0556E+02/3.9732E+01(≈)	3.3950E+02/1.0007E+02(<)	3.1489E+02/6.0169E+01
fc_{29}	2.3410E+02/2.7555E+00(<)	2.3394E+02/3.0671E+00(<)	2.3188E+02/3.3384E+00(≈)	2.3331E+02/3.4771E+00(≈)	2.3325E+02/3.7920E+00(≈)	2.3206E+02/2.4702E+00
fc_{30}	3.2446E+04/1.6020E+05(≈)	3.9452E+02/4.3407E−02(≈)	3.9733E+02/1.1444E+01(>)	3.9639E+02/9.4408E+00(>)	3.9828E+02/1.3075E+01(≈)	7.5192E+03/5.0805E+04
>/≈/<	1/19/10	1/24/5	5/23/2	1/25/4	2/22/6	-/-/-

Table 9
Algorithm comparisons between LSHADE, jSO, LPalmDE, PaDE, Hip-DE and our algorithms on 30-D optimization. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations max_{nfes} equalling to $10000 \cdot D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant max level $\alpha = 0.05$ in comparison with the new proposed algorithms.

DE variants	LSHADE	jSO	LPalmDE	PaDE	Hip-DE	Our algorithm
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
fc_1	2.7864E−16/1.9899E−15(≈)	1.3932E−15/4.2679E−15(<)	8.3593E−16/3.3770E−15(≈)	0.0000E+00/0.0000E+00(≈)	5.5729E−16/2.7859E−15(≈)	0.0000E+00/0.0000E+00
fc_2	3.3437E−15/9.2483E−15(<)	2.7864E−15/8.5358E−15(<)	6.6875E−15/1.3437E−14(<)	5.5729E−15/1.1397E−14(<)	8.9166E−15/1.3319E−14(<)	**0.0000E+00/0.0000E+00**
fc_3	3.3437E−15/1.3508E−14(≈)	4.2354E−14/2.5019E−14(<)	1.2260E−14/2.3612E−14(<)	2.2292E−15/1.1144E−14(≈)	3.3437E−15/1.3508E−14(≈)	0.0000E+00/0.0000E+00
fc_4	5.8562E+01/2.1269E−14(<)	5.8562E+01/2.7847E−14(<)	5.3225E+01/1.7514E+01(<)	5.1443E+01/2.0364E+01(<)	5.8670E+01/7.7797E−01(<)	**2.9790E+01/7.2931E+00**
fc_5	6.3199E+00/1.5399E+00(<)	9.0011E+00/1.9905E+00(<)	1.0554E+01/2.7722E+00(<)	8.7278E+00/1.5651E+00(<)	7.2833E+00/1.4568E+00(<)	**4.0781E+00/1.7467E+00**
fc_6	5.3677E−09/2.6833E−08(>)	8.0803E−09/3.2519E−08(>)	1.2037E−13/2.7016E−14(>)	1.1369E−13/0.0000E+00(>)	6.0409E−09/2.7124E−08(>)	1.5391E−05/2.3758E−05
fc_7	3.7385E+01/1.3641E+00(<)	3.8746E+01/1.9227E+00(<)	4.2350E+01/3.7840E+00(<)	3.8705E+01/1.6217E+00(<)	3.5880E+01/9.6925E−01(≈)	3.5949E+01/1.4543E+00
fc_8	7.3618E+00/1.4181E+00(<)	8.8054E+00/1.6649E+00(<)	1.2915E+01/3.5624E+00(<)	9.4846E+00/1.7721E+00(<)	7.3763E+00/1.4044E+00(<)	**4.1943E+00/1.4406E+00**
fc_9	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00(≈)	0.0000E+00/0.0000E+00
fc_{10}	1.4191E+03/2.0794E+02(>)	1.5401E+03/2.3970E+02(>)	1.6161E+03/2.8206E+02(≈)	1.5063E+03/2.7206E+02(>)	1.4894E+03/2.2988E+02(>)	1.6581E+03/2.8190E+02
fc_{11}	1.8634E+01/2.4363E+01(<)	4.1677E+00/8.5999E+00(≈)	9.1092E+00/1.4165E+01(<)	1.7092E+01/2.3230E+01(<)	1.4561E+01/2.1375E+01(<)	6.3863E+00/1.3925E+01
fc_{12}	1.0373E+03/3.3002E+02(<)	1.6009E+02/9.2030E+01(<)	1.1286E+03/4.1334E+02(<)	1.1150E+03/3.7339E+02(<)	1.0934E+03/3.7777E+02(<)	**8.9338E+01/7.4437E+01**
fc_{13}	1.5708E+01/5.2229E+00(<)	1.6778E+01/3.3253E+00(<)	1.6681E+01/5.8629E+00(<)	1.4909E+01/6.2006E+00(≈)	1.5258E+01/6.3961E+00(≈)	**1.3025E+01/7.2480E+00**
fc_{14}	2.1348E+01/4.3814E+00(<)	2.1457E+01/3.2159E+00(<)	1.9551E+01/6.7897E+00(≈)	2.2486E+01/3.2953E+00(<)	2.2518E+01/1.1845E+00(<)	**2.0830E+01/2.8908E+00**
fc_{15}	4.0008E+00/2.3587E+00(<)	1.0103E+00/7.0740E−01(≈)	3.8024E+00/2.0026E+00(<)	3.0850E+00/1.5985E+00(<)	2.8528E+00/1.4990E+00(<)	9.5940E−01/8.0585E−01
fc_{16}	5.1347E+01/4.3559E+01(<)	5.4949E+01/6.7056E+01(<)	1.8316E+02/1.1987E+02(<)	8.8102E+01/8.3072E+01(<)	1.1182E+02/9.1892E+01(<)	**1.0184E+01/4.8945E+00**
fc_{17}	3.1879E+01/6.2712E+00(<)	3.3496E+01/6.4682E+00(<)	3.4552E+01/1.1333E+01(<)	3.1150E+01/7.6777E+00(<)	3.1126E+01/6.7203E+00(<)	**2.2077E+01/7.1517E+00**
fc_{18}	2.1763E+01/8.8785E−01(<)	2.0765E+01/3.3103E−01(<)	2.2592E+01/1.6834E+00(<)	2.1732E+01/1.1010E+00(<)	2.2606E+01/1.5667E+00(<)	**1.9892E+01/3.9753E+00**
fc_{19}	5.3009E+00/1.2616E+00(<)	4.6243E+00/1.7513E+00(<)	5.0615E+00/1.4097E+00(<)	4.9323E+00/1.5112E+00(<)	5.1407E+00/1.7925E+00(<)	**4.0577E+00/1.8777E+00**
fc_{20}	3.1817E+01/6.3336E+00(<)	2.9930E+01/7.9389E+00(<)	3.3118E+01/2.7299E+01(<)	3.7630E+01/1.7708E+01(<)	3.9194E+01/2.4154E+01(<)	**2.4325E+01/6.3803E+00**
fc_{21}	2.0710E+02/1.4812E+00(<)	2.0957E+02/1.8002E+00(<)	2.1128E+02/2.5408E+00(<)	2.0807E+02/1.7554E+00(<)	2.0747E+02/1.6222E+00(<)	**2.0493E+02/1.7643E+00**
fc_{22}	1.0000E+02/1.4352E−14(≈)	1.0000E+02/1.0047E−13(≈)	1.0000E+02/1.0047E−13(≈)	1.0000E+02/1.4352E−14(≈)	1.0000E+02/1.4352E−14(≈)	1.0000E+02/1.4352E−14
fc_{23}	3.4950E+02/2.7650E+00(≈)	3.5041E+02/3.8216E+00(<)	3.5013E+02/5.4847E+00(≈)	3.4559E+02/3.0598E+00(>)	3.4463E+02/4.2466E+00(>)	3.4863E+02/3.9621E+00
fc_{24}	4.2552E+02/1.7469E+00(<)	4.2729E+02/2.5303E+00(<)	4.2357E+02/2.9915E+00(≈)	4.2130E+02/2.0215E+00(>)	4.2094E+02/2.2052E+00(>)	4.2237E+02/3.1337E+00
fc_{25}	3.8674E+02/2.3732E−02(<)	3.8670E+02/1.0235E−02(<)	3.8678E+02/3.2896E−02(<)	3.8677E+02/2.7835E−02(<)	3.8678E+02/2.7130E−02(<)	**3.8054E+02/6.1941E−01**
fc_{26}	9.2546E+02/3.4801E+01(<)	9.3878E+02/4.2029E+01(<)	9.2447E+02/4.7704E+01(<)	8.6890E+02/5.7770E+01(≈)	8.7924E+02/3.1269E+01(≈)	8.5195E+02/4.7912E+01
fc_{27}	5.0334E+02/5.8744E+00(<)	4.9773E+02/6.7188E+00(<)	5.0712E+02/4.4763E+00(<)	5.0792E+02/4.8855E+00(<)	5.0545E+02/5.8970E+00(<)	**4.8624E+02/7.4449E+00**
fc_{28}	3.3305E+02/5.2139E+01(<)	3.1320E+02/3.6533E+01(≈)	3.2898E+02/4.7641E+01(≈)	3.3279E+02/5.1725E+01(<)	3.2842E+02/4.9133E+01(<)	**3.1320E+02/3.6529E+01**
fc_{29}	4.3156E+02/7.1649E+00(<)	4.3590E+02/1.1192E+01(<)	4.2772E+02/9.7203E+00(<)	4.3185E+02/7.8449E+00(<)	4.3469E+02/7.4613E+00(<)	**4.2236E+02/7.7182E+00**
fc_{30}	1.9910E+03/5.1984E+01(<)	1.9708E+03/2.6363E+01(<)	2.0690E+03/6.8697E+01(<)	2.0457E+03/5.1264E+01(<)	2.0514E+03/5.5328E+01(<)	**1.5406E+03/9.1584E+01**
>/≈/<	2/5/23	2/5/23	1/8/21	4/6/20	4/6/20	-/-/-

Table 10
Algorithm comparisons between LSHADE, jSO, LPalmDE, PaDE, Hip-DE and our algorithms on 50-D optimization. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations max_{nfes} equalling to $10000 \cdot D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant max level $\alpha = 0.05$ in comparison with the new proposed algorithms.

DE variants	LSHADE	jSO	LPalmDE	PaDE	Hip-DE	Our algorithm
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
fc_{1}	*1.7555E−14/6.0880E−15(≈)*	2.9258E−14/8.7194E−15(<)	2.2570E−14/7.6139E−15(<)	2.3127E−14/8.9701E−15(<)	2.5914E−14/6.7891E−15(<)	*1.8948E−14/6.7657E−15*
fc_{2}	*1.0092E−12/5.7168E−12(≈)*	*1.8391E−13/7.1747E−13*(<)	*6.8546E−14/1.7159E−13*(<)	*5.2942E−14/3.9390E−14*(<)	*4.5140E−14/2.9633E−14*(<)	*4.2911E−14/1.0158E−13*
fc_{3}	1.6496E−13/5.8253E−14(<)	2.6750E−13/7.2166E−14(<)	1.5381E−13/4.7278E−14(<)	1.5938E−13/4.6901E−14(<)	1.4824E−13/4.9657E−14(<)	**6.5760E−14/2.6351E−14**
fc_{4}	7.4401E+01/5.0815E+01(<)	*5.2699E+01/4.3116E+01(≈)*	*6.9341E+01/5.1779E+01*(<)	8.3537E+01/4.4029E+01(<)	8.0188E+01/4.7197E+01(<)	*4.9855E+01/4.1050E+01*
fc_{5}	1.2781E+01/2.0421E+00(<)	1.6472E+01/3.3435E+00(<)	2.3334E+01/4.6458E+00(<)	1.6896E+01/2.8434E+00(<)	1.4647E+01/1.9140E+00(<)	**1.1776E+01/2.3310E+00**
fc_{6}	*1.3996E−07/2.6902E−07*(>)	4.5306E−07/1.0754E−06(>)	1.0171E−03/2.1880E−03(>)	*2.0087E−04/8.5648E−04*(>)	*2.5672E−05/1.8260E−04*(>)	5.8717E−03/3.8666E−03
fc_{7}	6.2661E+01/2.0497E+00(<)	6.6291E+01/2.8331E+00(<)	7.1044E+01/5.4122E+00(<)	6.5188E+01/2.7437E+00(<)	6.1420E+01/1.3780E+00(<)	**5.9553E+01/1.3399E+00**
fc_{8}	1.2592E+01/2.1023E+00(<)	1.7579E+01/2.9018E+00(<)	2.4214E+01/5.7974E+00(<)	1.6944E+01/2.5199E+00(<)	1.5243E+01/2.0900E+00(<)	**1.0899E+01/2.3772E+00**
fc_{9}	6.0187E−14/5.7310E−14(<)	7.5791E−14/5.4126E−14(<)	1.7555E−03/1.2536E−02(<)	4.0125E−14/5.4870E−14(<)	*1.1146E−14/3.4143E−14(≈)*	*1.7555E−03/1.2536E−02*
fc_{10}	*3.1037E+03/3.2509E+02*(>)	*3.1900E+03/2.9785E+02*(>)	3.2257E+03/3.8735E+02(>)	*3.1435E+03/2.9592E+02*(>)	*3.1438E+03/3.1760E+02*(>)	3.3594E+03/3.2790E+02
fc_{11}	4.9702E+01/8.0592E+00(<)	2.7732E+01/3.2713E+00(<)	7.2471E+01/1.3860E+01(<)	6.5681E+01/1.2045E+01(<)	4.7692E+01/7.6651E+00(<)	**2.2621E+01/3.6345E+00**
fc_{12}	2.1818E+03/4.5692E+02(<)	1.6365E+03/4.2894E+02(<)	2.2169E+03/4.9276E+02(<)	2.0999E+03/5.1909E+02(<)	2.3693E+03/6.1342E+02(<)	**9.9477E+02/3.3906E+02**
fc_{13}	4.9925E+01/2.3914E+01(<)	3.1840E+01/1.7620E+01(<)	7.1488E+01/2.9691E+01(<)	5.7240E+01/2.9138E+01(<)	6.5626E+01/2.2012E+01(<)	**1.3007E+01/1.3494E+01**
fc_{14}	2.9563E+01/3.3497E+00(<)	2.4844E+01/1.9528E+00(<)	3.0943E+01/3.0322E+00(<)	3.0181E+01/3.9907E+00(<)	3.2397E+01/4.2989E+00(<)	**2.3771E+01/1.8471E+00**
fc_{15}	3.8002E+01/8.6118E+00(<)	2.3728E+01/3.0491E+00(<)	4.6376E+01/1.6567E+01(<)	4.1413E+01/1.1554E+01(<)	5.1579E+01/1.5661E+01(<)	**1.7965E+01/2.2339E+00**
fc_{16}	3.4340E+02/1.2380E+02(<)	4.4427E+02/1.3553E+02(<)	4.0800E+02/1.1478E+02(<)	3.7479E+02/1.0736E+02(<)	3.7666E+02/1.0247E+02(<)	**1.5983E+02/4.8806E+01**
fc_{17}	2.4250E+02/6.8884E+01(<)	2.7824E+02/9.4804E+01(<)	3.1111E+02/1.1558E+02(<)	2.9307E+02/7.9127E+01(<)	2.9637E+02/5.9646E+01(<)	**1.4582E+02/6.6330E+01**
fc_{18}	4.0906E+01/1.4363E+01(<)	2.4922E+01/2.1399E+00(<)	5.2376E+01/2.9651E+01(<)	4.5133E+01/1.4021E+01(<)	5.3481E+01/1.5632E+01(<)	**2.1390E+01/8.3270E−01**
fc_{19}	2.4516E+01/6.3755E+00(<)	1.4514E+01/2.5875E+00(<)	3.4783E+01/1.6103E+01(<)	2.9531E+01/9.4244E+00(<)	4.0751E+01/1.4438E+01(<)	**1.0173E+01/2.2333E+00**
fc_{20}	1.5170E+02/5.2989E+01(<)	*1.2969E+02/7.4729E+01(≈)*	*1.6196E+02/1.1808E+02*(<)	1.7527E+02/6.7373E+01(<)	1.6931E+02/6.9511E+01(<)	*1.0765E+02/4.6360E+01*
fc_{21}	*2.1289E+02/2.5945E+00(≈)*	2.1880E+02/3.1300E+00(<)	2.2331E+02/3.7662E+00(<)	2.1695E+02/2.5911E+00(<)	2.1625E+02/2.2651E+00(<)	*2.1220E+02/2.5567E+00*
fc_{22}	2.0179E+03/1.7758E+03(<)	1.5263E+03/1.8031E+03(<)	7.7586E+02/1.4763E+03(<)	3.3344E+02/9.2080E+02(<)	*1.0261E+02/1.5993E+01(≈)*	*1.0023E+02/8.1129E−01*
fc_{23}	4.3021E+02/3.9508E+00(<)	4.3274E+02/6.0158E+00(<)	4.3736E+02/7.8816E+00(<)	*4.2713E+02/5.2016E+00(≈)*	*4.2759E+02/6.9648E+00*(<)	*4.2534E+02/8.9364E+00*
fc_{24}	5.0668E+02/2.9836E+00(<)	5.0930E+02/3.7851E+00(<)	5.0459E+02/4.9137E+00(<)	5.0525E+02/6.8755E+00(<)	5.0766E+02/5.8008E+00(<)	**5.0208E+02/5.4284E+00**
fc_{25}	4.8336E+02/1.2014E+01(<)	4.8116E+02/3.1464E+00(<)	4.9531E+02/2.3657E+01(<)	5.0349E+02/2.8545E+01(<)	4.8183E+02/4.0653E+00(<)	**4.7999E+02/1.5698E+00**
fc_{26}	1.1394E+03/4.4333E+01(<)	1.1578E+03/5.1982E+01(<)	1.1605E+03/7.1254E+01(<)	1.1462E+03/7.4496E+01(<)	1.1307E+03/7.1604E+01(<)	**1.0646E+03/6.5387E+01**
fc_{27}	5.3343E+02/1.8105E+01(<)	5.1152E+02/1.0414E+01(<)	5.4156E+02/1.3098E+01(<)	5.3914E+02/1.1183E+01(<)	5.3516E+02/7.5460E+00(<)	**4.7883E+02/9.6464E+00**
fc_{28}	4.6938E+02/2.0290E+01(<)	4.6172E+02/1.1607E+01(<)	4.9429E+02/2.2015E+01(<)	4.9716E+02/2.0290E+01(<)	4.8758E+02/2.4279E+01(<)	**4.5744E+02/7.7066E−01**
fc_{29}	*3.5285E+02/1.0007E+01*(>)	3.6471E+02/1.4654E+01(<)	*3.5149E+02/1.7455E+01*(>)	*3.5320E+02/1.2396E+01(≈)*	3.7017E+02/1.2459E+01(<)	3.5628E+02/1.4653E+01
fc_{30}	6.5923E+05/7.0107E+04(<)	6.0465E+05/3.5617E+04(<)	6.2226E+05/4.2168E+04(<)	6.2350E+05/3.4581E+04(<)	6.1904E+05/3.5079E+04(<)	**1.5784E+05/4.7688E+04**
>/≈/<	3/3/24	2/2/26	3/0/27	2/2/26	2/2/26	-/-/-

Table 11
Algorithm comparisons between LSHADE, jSO, LPalmDE, PaDE, Hip-DE and our algorithms on 100-D optimization. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations max_{nfes} equalling to $10000 \cdot D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant max level $\alpha = 0.05$ in comparison with the new proposed algorithms.

DE variants	LSHADE	jSO	LPalmDE	PaDE	Hip-DE	Our algorithm
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
fc_{1}	7.3896E−13/8.9686E−13(>)	1.7047E−08/7.6147E−12(>)	*2.8255E−13/8.0178E−13*(>)	*2.0536E−13/2.6495E−13*(>)	2.3256E−12/5.4658E−12(>)	3.2667E−09/6.2892E−09
fc_{2}	3.0065E+02/1.6475E+03(>)	9.0966E+01/3.7725E+01(>)	*5.1674E+00/3.1154E+01*(>)	*1.8276E−06/1.0739E−05*(>)	6.2828E+02/2.4471E+03(>)	1.1679E+149/6.3996E+149
fc_{3}	8.5931E−07/1.0218E−06(<)	3.1320E−07/3.1046E−07(<)	1.8297E−07/2.2898E−07(<)	*7.6206E−08/1.4680E−07(≈)*	1.9368E−07/4.1047E−07(<)	*6.1426E−08/6.5685E−08*
fc_{4}	1.9668E+02/1.1574E+01(<)	1.9378E+02/2.1805E+01(<)	*1.1581E+02/6.8223E+01*(>)	*1.2463E+02/6.9534E+01*(>)	1.9590E+02/3.3743E+00(<)	1.8042E+02/1.7292E+01
fc_{5}	3.7352E+01/5.6033E+00(<)	**2.9772E+01/6.0453E+00**(>)	5.5712E+01/8.9897E+00(<)	4.5139E+01/5.1004E+00(<)	3.9906E+01/4.3826E+00(<)	3.4861E+01/4.3439E+00
fc_{6}	6.5851E−03/3.8162E−03(>)	**2.0632E−04/4.8850E−04**(>)	1.5295E−02/1.1984E−02(>)	1.5603E−02/1.6361E−02(>)	1.0858E−03/1.1299E−03(>)	3.8179E−01/8.2776E−02
fc_{7}	1.4008E+02/3.7571E+00(<)	1.3125E+02/8.0417E+00(<)	1.5164E+02/7.7896E+00(<)	1.4014E+02/3.9842E+00(<)	1.3220E+02/3.7489E+00(<)	**1.2459E+02/2.9152E+00**
fc_{8}	3.9456E+01/4.5198E+00(<)	**3.0340E+01/4.4281E+00**(>)	5.7677E+01/7.9064E+00(<)	4.5875E+01/4.8086E+00(<)	4.3640E+01/4.3073E+00(<)	3.6864E+01/3.8873E+00
fc_{9}	5.3220E−01/5.6448E−01(>)	**1.2037E−13/2.7016E−14**(>)	1.3239E+00/8.5952E−01(>)	9.6747E−01/6.2137E−01(>)	3.8620E−02/7.6503E−02(>)	2.7110E+00/1.0453E+00
fc_{10}	1.0346E+04/4.9641E+02(<)	1.1346E+04/7.7446E+02(<)	*9.6959E+03/6.4184E+02(≈)*	*9.7105E+03/5.0214E+02*(<)	1.0383E+04/6.5123E+02(<)	*9.4618E+03/5.3458E+02*
fc_{11}	4.6550E+02/1.1355E+02(<)	1.1919E+02/3.2361E+01(<)	6.3934E+02/1.1335E+02(<)	5.7635E+02/8.6944E+01(<)	4.5567E+02/7.8064E+01(<)	**3.8916E+01/1.7212E+01**
fc_{12}	2.1519E+04/6.7812E+03(<)	1.8260E+04/8.8971E+03(<)	1.8717E+04/6.9982E+03(<)	2.2884E+04/8.8568E+03(<)	2.3315E+04/8.6249E+03(<)	**9.3716E+03/4.3983E+03**
fc_{13}	4.9293E+02/3.9935E+02(<)	2.1862E+02/6.2157E+01(<)	1.2089E+03/8.1210E+02(<)	7.2575E+02/3.9125E+02(<)	1.8580E+03/8.0657E+02(<)	**1.6816E+02/6.3682E+01**
fc_{14}	2.5249E+02/3.2490E+01(<)	7.0393E+01/1.2821E+01(<)	2.6310E+02/3.8153E+01(<)	2.7586E+02/4.0309E+01(<)	2.5379E+02/2.9448E+01(<)	**3.0973E+01/2.2990E+00**
fc_{15}	2.5019E+02/5.5026E+01(<)	1.9962E+02/4.2506E+01(<)	2.4976E+02/4.9758E+01(<)	2.5913E+02/4.3604E+01(<)	2.3688E+02/4.6663E+01(<)	**5.7259E+01/2.6434E+01**
fc_{16}	1.7080E+03/2.5313E+02(<)	1.7876E+03/3.7649E+02(<)	1.6725E+03/2.9246E+02(<)	*1.4224E+03/2.6291E+02(≈)*	1.8912E+03/3.0987E+02(<)	*1.4926E+03/3.0336E+02*
fc_{17}	1.1030E+03/1.7351E+02(<)	1.2518E+03/2.8538E+02(<)	1.2486E+03/2.6139E+02(<)	1.1762E+03/1.7418E+02(<)	1.2923E+03/1.7638E+02(<)	**8.4950E+02/2.1230E+02**
fc_{18}	2.2707E+02/5.1504E+01(<)	1.9851E+02/4.0267E+01(<)	2.2742E+02/5.2416E+01(<)	2.1779E+02/4.7641E+01(<)	2.1770E+02/4.6402E+01(<)	**3.4279E+01/4.9706E+00**
fc_{19}	1.7210E+02/2.5433E+01(<)	1.4435E+02/1.7947E+01(<)	1.7882E+02/2.6508E+01(<)	1.8357E+02/3.0051E+01(<)	1.7315E+02/2.2927E+01(<)	**4.1226E+01/4.0991E+00**
fc_{20}	1.5994E+03/1.8845E+02(<)	1.3490E+03/2.5210E+02(<)	1.4440E+03/2.3029E+02(<)	1.5452E+03/2.1643E+02(<)	1.6243E+03/1.7776E+02(<)	**1.1451E+03/1.8884E+02**
fc_{21}	*2.5871E+02/6.5915E+00(≈)*	*2.5955E+02/4.6235E+00(≈)*	2.8439E+02/9.7938E+00(<)	2.6978E+02/5.9218E+00(<)	2.6713E+02/5.9443E+00(<)	*2.5778E+02/4.7030E+00*
fc_{22}	1.1167E+04/1.5373E+03(<)	1.1442E+04/8.0037E+02(<)	1.0523E+04/1.6274E+03(<)	1.1011E+04/5.3478E+02(<)	1.1635E+04/5.5910E+02(<)	**9.7706E+03/6.5640E+02**
fc_{23}	**5.6892E+02/8.6729E+00**(>)	5.9110E+02/1.0543E+01(<)	5.8529E+02/1.2811E+01(≈)	5.9339E+02/1.9521E+01(<)	6.0766E+02/1.2630E+01(<)	5.8624E+02/1.1831E+01
fc_{24}	9.1118E+02/7.4354E+00(<)	9.0300E+02/1.1805E+01(<)	9.3472E+02/1.6291E+01(<)	9.2374E+02/1.2208E+01(<)	9.2928E+02/1.3172E+01(<)	**8.8605E+02/1.9122E+01**
fc_{25}	7.4622E+02/2.8359E+01(<)	7.1784E+02/4.2077E+01(<)	7.2855E+02/3.7814E+01(<)	7.3038E+02/4.5545E+01(<)	7.3344E+02/4.1661E+01(<)	**6.6859E+02/4.5924E+01**
fc_{26}	3.3042E+03/8.9139E+01(<)	3.1735E+03/1.1047E+02(<)	3.4736E+03/1.4629E+02(<)	3.3906E+03/1.1060E+02(<)	3.3343E+03/9.3665E+01(<)	**2.9579E+03/8.9566E+01**
fc_{27}	6.2777E+02/1.9876E+01(<)	6.0566E+02/1.7111E+01(<)	6.6124E+02/2.2129E+01(<)	6.5690E+02/2.1078E+01(<)	6.3851E+02/1.3582E+01(<)	**5.7450E+02/1.3210E+01**
fc_{28}	5.2971E+02/3.2387E+01(<)	5.3478E+02/3.0677E+01(<)	5.3301E+02/3.3932E+01(<)	5.2179E+02/2.9053E+01(≈)	5.2982E+02/2.8022E+01(≈)	**5.1843E+02/3.0554E+01**
fc_{29}	1.2654E+03/1.7396E+02(<)	1.4931E+03/2.3702E+02(<)	1.3722E+03/2.0603E+02(<)	1.1774E+03/1.9076E+02(<)	1.2337E+03/1.8583E+02(<)	**1.0829E+03/1.4762E+02**
fc_{30}	2.3863E+03/1.5950E+02(<)	2.4451E+03/1.4975E+02(<)	2.5425E+03/1.6137E+02(<)	2.5845E+03/1.5411E+02(<)	2.5758E+03/1.6872E+02(<)	**1.6010E+03/1.2247E+02**
>/≈/<	5/1/24	6/1/23	5/2/23	5/3/22	4/1/25	-/-/-

Table 12
Summary of the comparison results between our algorithms algorithm and the other contrasted algorithms under the CEC2013, CEC2014 and CEC2017 benchmarks on 10D, 30D and 50D optimization respectively.

A given algorithm versus our algorithm											
Test suit:	CEC2013			CEC2014			CEC2017				All
$>/\approx/<$	D = 10	D = 30	D = 50	D = 10	D = 30	D = 50	D = 10	D = 30	D = 50	D = 100	$\sum$
LSHADE	2/14/12	2/13/13	1/7/20	3/14/13	6/9/15	5/3/22	1/19/10	2/5/23	3/3/24	5/1/24	30/88/176
jSO	2/16/10	0/8/20	6/4/18	2/12/16	0/8/22	4/1/25	1/24/5	2/5/23	2/2/26	6/1/23	25/81/188
LPalmDE	2/17/9	5/11/12	4/8/16	6/17/7	6/7/17	6/7/17	5/23/2	1/8/21	3/0/27	5/2/23	43/100/151
PaDE	2/18/8	3/13/12	4/9/15	3/17/10	1/14/15	9/2/15	1/25/4	4/6/20	3/1/26	5/3/22	35/108/151
Hip-DE	7/15/6	8/10/10	7/8/13	5/19/6	5/10/15	8/2/20	2/22/6	4/6/20	2/2/26	4/1/25	52/95/147

$f_{c_1}, f_{c_3} - f_{c_5}, f_{c_8} - f_{c_{10}}, f_{c_{11}} - f_{c_{22}}$ and $f_{c_{25}} - f_{c_{30}}$. Moreover, we also made a summary of these results in Table 14 and listed the cases on which the (LSHADE, jSO, LPalmDE, PaDE, Hip-DE) algorithms achieve the best or tier-best performance under CEC2017 test suite with the objectives indexed $f_{c_1} - f_{c_{30}}$ in our test suite in Table 15, and we can see that the proposed algorithm also secures very good optimization performance from the convergence speed view. In summary, the proposed algorithm performs best on simple multimodal and hybrid functions which indicates that the local search capability is improved. In addition, our algorithm outperforms other algorithms in terms of convergence speed for composition functions though performs worse in terms of the average optimization accuracy.

4.3. Complexity analysis

In this section, we compare the time complexity of our algorithm to that of previous DE variants. The time complexity comparison is conducted according to the instruction of the CEC2013 competition, and the results are provided in Table 16. T_0 represents the time spent on basic arithmetic expressions, T_1 represents the computation time spent on function call of 30D f_{14} of the CEC2013 test suite under the 200,000 function evaluations, and $\widehat{T}_2$ represents the overall cost of an algorithm optimizing 30D f_{14} of the CEC2013 test suite under the 200,000 function evaluations. Then, the value $\frac{\widehat{T}_2 - T_1}{T_0}$ is used for denoting the algorithm complexity of each algorithm.

The results show that our algorithm is of less time complexity than LPalmDE and Hip-DE, and it consumes more time than LSHADE, jSO and PaDE because performance improvements usually cost extra time consumption. As a reviewer suggested, we also give a further explanation of the complexity in terms of **O**. The computational complexity of our algorithm in each run is $\mathbf{O}(PS \cdot D \cdot (U - L))$ where PS denotes the population size, D denotes the dimension number, U denotes the upper bound and L denotes the lower bound of the solution space. The total computational complexity of the algorithm under the fixed cost criterion is $\mathbf{O}(nfe_{\max} \cdot D \cdot (U - L))$ where $nfe_{\max}$ denotes the maximum number of function evaluations. Obviously, the algorithms in the comparison are of no difference in terms of **O** notation. Moreover, the time of function call usually consumes much more time in comparison with the time consumed by the algorithm itself. That is also why the Congress on Evolutionary Computation (CEC) 2013, 2014 and 2017 competitions give their recommendations rather than using **O** for algorithm complexity analysis.

4.4. Analysis of the diversity indicator

Here in this part, we mainly examine the parameter settings of the diversity indicator in our algorithm. As is mentioned in the algorithm description section, there are two parameters involved in the diversity indicator, one is ξ, the threshold of d_{in} and the other is k, the coefficient factor of $PS \cdot D$. Generally, the parameter ξ is of less sensitivity for the optimization performance, and a default value $\xi = 0.01$ is recommended in our algorithm. Different from ξ, we make a deeper investigation of the parameter k and the results are presented in Table 17, in which six different values are examined. From the comparison results, we can see that the default setting $k = 0.6$ performs the best.

Moreover, we also make a further comparison between our algorithm with and without the diversity indicator. The detailed results are given in Table 18, and it is clear that our default algorithm leads to 19 performance improvements in comparison with the one without the indicator. It also can be seen from the detailed results that the proposed strategy significantly improves the performance of the algorithm on simple multimodal functions and composition functions. It may reveal that the strategy may be of big probability for further performance improvement of DE algorithms, and it will be one of the next work in the near future.

4.5. Application

A constrained optimization from the real world, the tension/compression spring design is taken into consideration for our algorithm validation, and the main objective of it is to get the optimal weight of a tension or compression spring [50]. The formula of the objective is given in Eq. (35):

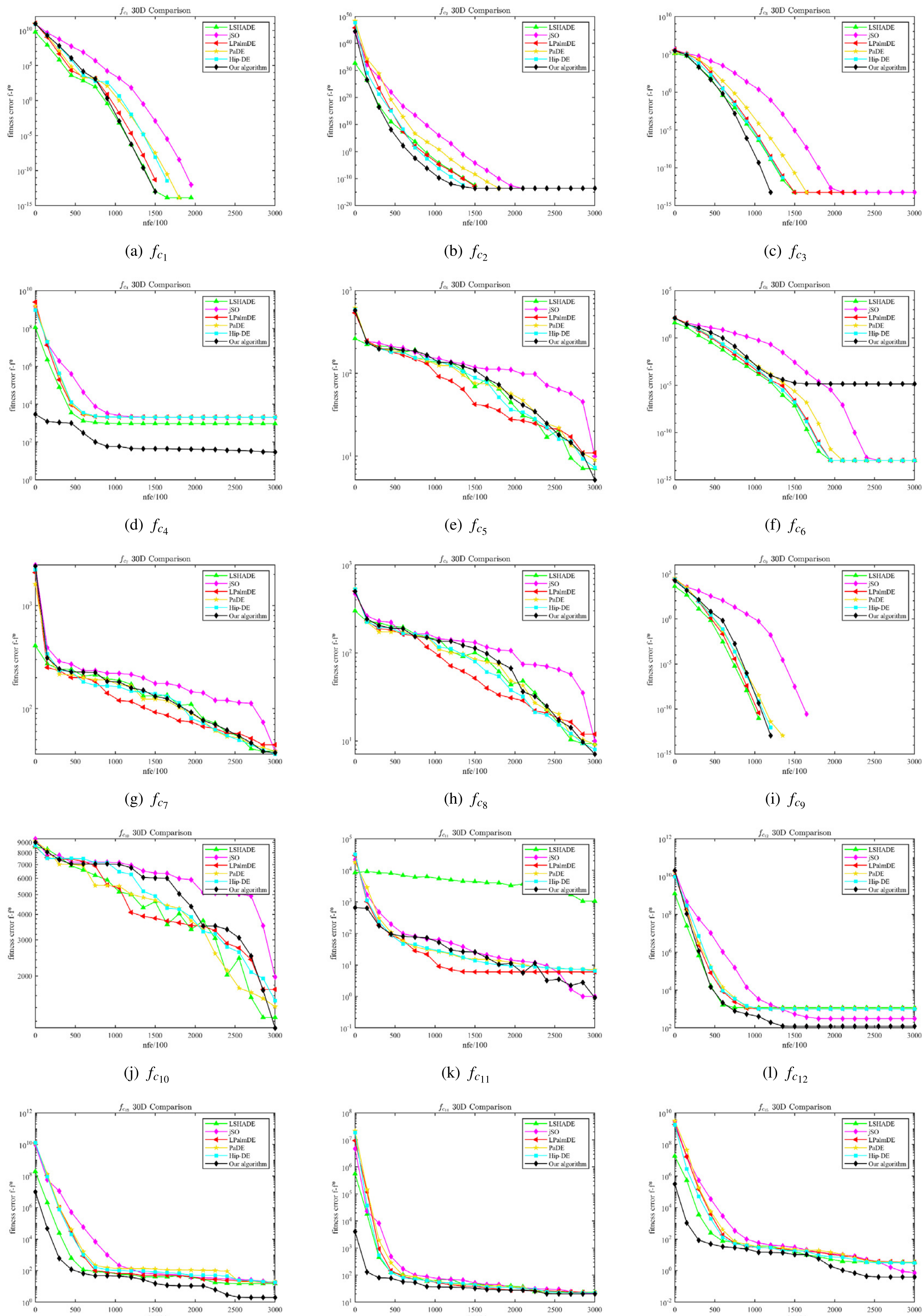
$$\min f(x) = x_1^2 \cdot x_2 \cdot (x_3 + 2) \tag{35}$$

where three variables (the diameter, $x_1 \in [0.05, 2.00]$, of the wire, the mean, $x_2 \in [0.25, 1.30]$, of the diameter of coil, and the number, $x_3 \in [2.00, 15.0]$, of the active coils) is used for the calculation of the weight while four constraints in Eq. (36) should be satisfied.

$$\begin{cases} g_1(x) = 1 - \dfrac{x_2^3 \cdot x_3}{71785 \cdot x_1^4} \le 0 \\ g_2(x) = \dfrac{4 \cdot x_2^2 - x_1 \cdot x_2}{12566(x_2 \cdot x_1^3 - x_1^4)} + \dfrac{1}{5108 \cdot x_1^2} - 1 \le 0 \\ g_3(x) = 1 - \dfrac{140.45 \cdot x_1}{x_2^2 \cdot x_3} \le 0 \\ g_4(x) = \dfrac{x_1 + x_2}{1.5} - 1 \le 0 \end{cases} \tag{36}$$

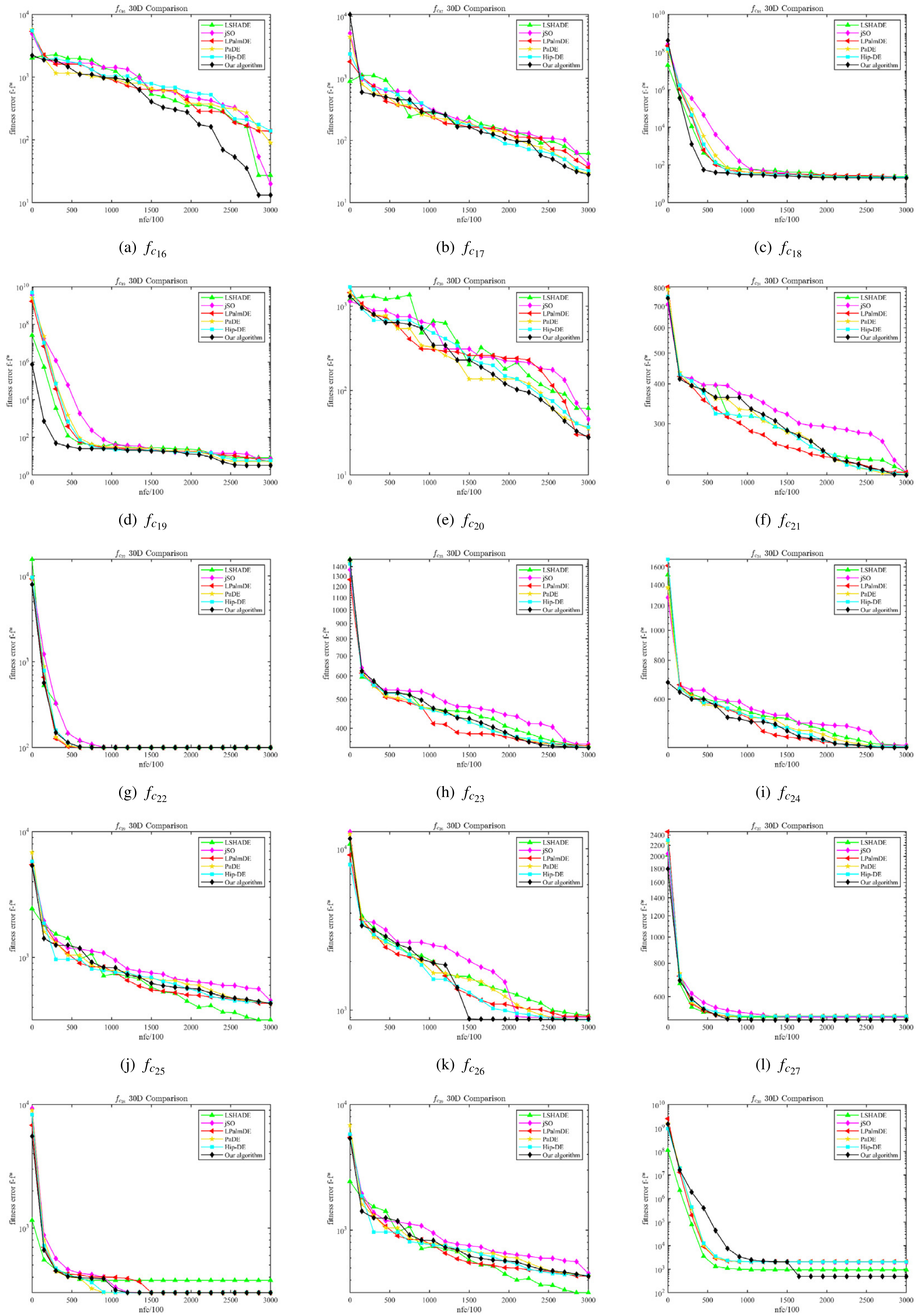
The problem addresses how to minimize the volume V of a helical spring under a constant tension/compression load, and the problem is composed of three design variables, including the number of spring's active coils $P = x_1 \in [2, 15]$, the diameter of the winding $D = x_2 \in [0.25, 1.3]$, the diameter of the wire $d = x_3 \in [0.05, 2]$.

The recent state-of-the-art competitors for solving this real-world constrained optimization problem were proposed in the Genetic and Evolutionary Computation Conference (GECCO) 2020, and the best-performing algorithms such as sCMAgES [51], EnMODE and COLSHADE [52] are taken into comparison for algorithm validation. The best result ("Best"), worst result ("Worst"),



(a) f_{c_1} (b) f_{c_2} (c) f_{c_3} (d) f_{c_4} (e) f_{c_5} (f) f_{c_6} (g) f_{c_7} (h) f_{c_8} (i) f_{c_9} (j) $f_{c_{10}}$ (k) $f_{c_{11}}$ (l) $f_{c_{12}}$ (m) $f_{c_{13}}$ (n) $f_{c_{14}}$ (o) $f_{c_{15}}$

Fig. 2. Here we present the convergence comparison by employing the median value of 51 runs on 30D optimization under f_{c_1}–f_{c_8} of our test suite.



(a) $f_{c_{16}}$ (b) $f_{c_{17}}$ (c) $f_{c_{18}}$

(d) $f_{c_{19}}$ (e) $f_{c_{20}}$ (f) $f_{c_{21}}$

(g) $f_{c_{22}}$ (h) $f_{c_{23}}$ (i) $f_{c_{24}}$

(j) $f_{c_{25}}$ (k) $f_{c_{26}}$ (l) $f_{c_{27}}$

(m) $f_{c_{28}}$ (n) $f_{c_{29}}$ (o) $f_{c_{30}}$

Fig. 3. Here we present the convergence comparison by employing the median value of 51 runs on 30D optimization under f_{c_9}–$f_{c_{16}}$ of our test suite.

ARTICLE IN PRESS

Table 13

Algorithm comparisons between LSHADE-SPACMA and LSHADE-cnEpSin and our algorithms on (10,30,50,100)-D optimization under the CEC2017 test suite. The results are calculated under 51 independent runs with the fixed maximum number of function evaluations max_{nfes} equalling to $10000 \cdot D$. The overall performance of each algorithm is measured under Wilcoxon's signed rank test with the significant max level $\alpha = 0.05$ in comparison with the new proposed algorithms.

Test suit:	10D			30D			50D			100D		
>/≈/<	LSHADE-SPACMA	LSHADE-cnEpSin	Our algorithm	LSHADE-SPACMA	LSHADE-cnEpSin	Our algorithm	LSHADE-SPACMA	LSHADE-cnEpSin	Our algorithm	LSHADE-SPACMA	LSHADE-cnEpSin	Our algorithm
f_{c_1}	0.0000E+00/0.0000e+00(≈)	0.0000E+00/0.0000e+00(≈)	0.0000E+00/0.0000e+00	0.0000E+00/0.0000e+00(≈)	3.1614E−02/6.0497e−02(<)	0.0000E+00/0.0000e+00	**0.0000E+00/0.0000e+00**(>)	1.5369E+01/7.2292e+01(<)	1.8948E−14/6.7657e−15	**4.4583E−15/6.6595e−15**(>)	4.6343E+03/1.6056e+04(<)	3.2667E−09/6.2892e−09
f_{c_2}	0.0000E+00/0.0000e+00(≈)	3.3437E−15/1.3508e−14(≈)	0.0000E+00/0.0000e+00	7.2447E−15/1.2510e−14(<)	5.9737E+01/4.2627e+02(<)	**0.0000E+00/0.0000e+00**	3.5666E−14/2.4675e−14(≈)	1.3245E+01/4.1494e+01(<)	4.2911E−14/1.0158e−13	**2.6354E+00/1.8397e+01**(>)	2.5750E+13/1.1801e+14(>)	1.1679E+149/6.3996e+149
f_{c_3}	0.0000E+00/0.0000e+00(≈)	0.0000E+00/0.0000e+00(≈)	0.0000E+00/0.0000e+00	0.0000E+00/0.0000e+00(≈)	4.8730E−02/2.8094e−02(<)	0.0000E+00/0.0000e+00	**0.0000E+00/0.0000e+00**(>)	3.0547E+01/1.7148e+01(<)	6.5760E−14/2.6351e−14	**0.0000E+00/0.0000e+00**(>)	5.1697E+03/2.5065e+03(<)	6.1426E−08/6.5685e−08
f_{c_4}	0.0000E+00/0.0000e+00(≈)	0.0000E+00/0.0000e+00(≈)	0.0000E+00/0.0000e+00	5.8562E+01/1.3924e−14(<)	4.0316E+01/7.7282e+00(<)	**2.9790E+01/7.2931e+00**	3.6199E+01/3.8460e+01(≈)	6.9748E+01/3.9963e+01(≈)	4.9855E+01/4.1050e+01	2.0182E+02/1.0536e+01(<)	2.1569E+02/3.0488e+01(<)	**1.8042E+02/1.7292e+01**
f_{c_5}	1.6228E+00/8.2059e−01(≈)	1.7386E+00/6.8417e−01(≈)	2.2249E+00/1.6950e+00	**3.4652E+00/2.4757e+00**(>)	3.9963E+01/4.9690e+00(<)	4.0781E+00/1.7467e+00	**5.9698E+00/2.2425e+00**(>)	9.7220E+01/8.7241e+00(<)	1.1776E+01/2.3310e+00	**1.2271E+01/3.1253e+00**(>)	3.7362E+02/1.8281e+01(<)	3.4861E+01/4.3439e+00
f_{c_6}	0.0000E+00/0.0000e+00(≈)	0.0000E+00/0.0000e+00(≈)	2.2292E−15/1.5919e−14	**2.8979E−14/5.0038e−14**(>)	4.2507E−04/1.6041e−04(<)	1.5391E−05/2.3758e−05	**1.1146E−13/1.5919e−14**(>)	2.9711E−02/7.4551e−03(<)	5.8717E−03/3.8666e−03	**1.1369E−13/0.0000e+00**(>)	8.1032E−01/1.5210e−01(<)	3.8179E−01/8.2776e−02
f_{c_7}	**1.0967E+01/3.4877e−01**(>)	1.1803E+01/5.1855e−01(>)	1.2407E+01/1.2329e+00	**3.4106E+01/1.1883e+00**(>)	8.1595E+01/7.9000e+00(<)	3.5949E+01/1.4543e+00	**5.7207E+01/1.1483e+00**(>)	1.8034E+02/1.6916e+01(<)	5.9553E+01/1.3399e+00	**1.1189E+02/1.3131e+00**(>)	2.3168E+02/1.3323e+02(<)	1.2459E+02/2.9152e+00
f_{a_8}	**9.9564E−01/8.2071e−01**(>)	1.7192E+00/6.9096e−01(≈)	2.4596E+00/1.7613e+00	3.6100E+00/1.6882e+00(≈)	4.4738E+01/5.7873e+00(<)	4.1943E+00/1.4406e+00	**5.8917E+00/2.2145e+00**(>)	1.0069E+02/9.4456e+00(<)	1.0899E+01/2.3772e+00	**1.0749E+01/2.7789e+00**(>)	3.5149E+02/9.0855e+01(<)	3.6864E+01/3.8873e+00
f_{c_9}	0.0000E+00/0.0000e+00(≈)	0.0000E+00/0.0000e+00(≈)	0.0000E+00/0.0000e+00	0.0000E+00/0.0000e+00(≈)	1.3800E+02/5.2678e+01(<)	0.0000E+00/0.0000e+00	0.0000E+00/0.0000e+00(≈)	1.0001E+03/2.1615e+02(<)	1.7555E−03/1.2536e−02	**0.0000E+00/0.0000e+00**(>)	9.4317E+03/1.1283e+03(<)	2.7110E+00/1.0453e+00
$f_{c_{10}}$	1.9763E+01/3.9691e+01(≈)	1.5676E+01/3.3826e+01(≈)	1.9349E+01/3.9317e+01	**1.3950E+03/2.2739e+02**(>)	1.6673E+03/2.3019e+02(≈)	1.6581E+03/2.8190e+02	3.7320E+03/7.6557e+02(<)	3.5462E+03/3.1271e+02(<)	**3.3594E+03/3.2790e+02**	9.7451E+03/8.6532e+02(≈)	1.1091E+04/4.6750e+02(<)	9.4618E+03/5.3458e+02
$f_{c_{11}}$	0.0000E+00/0.0000e+00(≈)	0.0000E+00/0.0000e+00(≈)	0.0000E+00/0.0000e+00	2.2037E+01/2.7421e+01(<)	2.0417E+01/2.3340e+01(<)	**6.3863E+00/1.3925e+01**	3.1461E+01/4.4259e+00(<)	4.1453E+01/4.8695e+00(<)	**2.2621E+01/3.6345e+00**	4.3449E+01/1.8287e+01(<)	4.3222E+02/8.3199e+01(<)	**3.8916E+01/1.7212e+01**
$f_{c_{12}}$	1.2193E+02/8.2872e+01(<)	1.2070E+02/7.3096e+01(<)	**3.7404E+01/5.5570e+01**	5.5231E+02/2.9108e+02(<)	1.0688E+03/3.1570e+02(<)	**8.9338E+01/7.4437e+01**	1.6022E+03/3.5650e+02(<)	7.5877E+03/5.8879e+03(<)	**9.9477E+02/3.3906e+02**	**4.8631E+03/7.3545e+02**(>)	1.0363E+05/3.5685e+04(<)	9.3716E+03/4.3983e+03
$f_{c_{13}}$	3.5908E+00/2.3581e+00(<)	3.4273E+00/2.5106e+00(≈)	**2.5199E+00/2.5022e+00**	1.5169E+01/4.0367e+00(≈)	2.6008E+01/1.5394e+01(<)	1.3025E+01/7.2480e+00	4.0567E+01/2.4313e+01(<)	1.3797E+02/4.8540e+01(<)	**1.3007E+01/1.3494e+01**	1.5241E+02/3.6138e+01(≈)	7.2063E+02/2.2426e+02(<)	1.6816E+02/6.3682e+01
$f_{c_{14}}$	2.5389E−01/5.5717e−01(≈)	1.9509E−01/3.9896e−01(≈)	1.4829E−01/3.9563e−01	2.2824E+01/1.3273e+00(<)	2.1623E+01/3.7940e+00(<)	**2.0830E+01/2.8908e+00**	2.9237E+01/3.0793e+00(<)	3.6927E+01/5.8909e+00(<)	**2.3771E+01/1.8471e+00**	7.4529E+01/1.1766e+01(<)	2.2316E+02/3.0461e+01(<)	**3.0973E+01/2.2990e+00**
$f_{c_{15}}$	3.6262E−01/1.9400e−01(<)	2.3424E−01/1.9610e−01(<)	**7.3693E−02/1.3339e−01**	5.0107E+00/2.5891e+00(<)	8.8846E+00/4.0992e+00(<)	**9.5940E−01/8.0585e−01**	3.1227E+01/6.0121e+00(<)	6.9896E+01/2.8872e+01(<)	**1.7965E+01/2.2339e+00**	1.0560E+02/2.7808e+01(<)	2.8804E+02/6.0345e+01(<)	**5.7259E+01/2.6434e+01**
$f_{c_{16}}$	7.7578E−01/3.3932e−01(<)	6.8290E−01/3.5444e−01(<)	**3.3010E−01/1.8807e−01**	4.2464E+01/6.4755e+01(<)	2.5851E+02/9.7268e+01(<)	**1.0184E+01/4.8945e+00**	4.4905E+02/2.1505e+02(<)	6.7373E+02/1.1490e+02(<)	**1.5983E+02/4.8806e+01**	**1.2443E+03/3.7716e+02**(>)	2.2939E+03/2.3819e+02(<)	1.4926E+03/3.0336e+02
$f_{c_{17}}$	1.1982E−01/1.3148e−01(≈)	1.1634E+00/4.0362e+00(<)	1.4921E−01/1.6454e−01	3.0508E+01/1.0539e+01(<)	4.9539E+01/8.5739e+00(<)	**2.2077E+01/7.1517e+00**	2.7123E+02/1.1904e+02(<)	4.9641E+02/1.0718e+02(<)	**1.4582E+02/6.6330e+01**	9.4972E+02/3.9147e+02(≈)	1.2035E+03/1.8832e+02(<)	8.4950E+02/2.1230e+02
$f_{c_{18}}$	3.9665E+00/7.6526e+00(<)	1.9629E+00/5.3826e+00(<)	**1.8524E−01/2.1302e−01**	2.3081E+01/3.0666e+00(<)	2.4938E+01/3.3403e+00(<)	**1.9892E+01/3.9753e+00**	3.2476E+01/6.6407e+00(<)	6.7634E+01/2.3264e+01(<)	**2.1390E+01/8.3270e−01**	1.2905E+02/3.2881e+01(<)	3.0420E+02/2.5778e+02(<)	**3.4279E+01/4.9706e+00**
$f_{c_{19}}$	2.1669E−01/3.6752e−01(<)	1.2422E−01/3.3261e−01(<)	**6.7524E−03/9.5801e−03**	9.7972E+00/2.1131e+00(<)	7.7947E+00/1.6251e+00(<)	**4.0577E+00/1.8777e+00**	2.0384E+01/4.1476e+00(<)	3.1828E+01/8.4263e+00(<)	**1.0173E+01/2.2333e+00**	7.4232E+01/1.1526e+01(<)	1.8927E+02/2.9060e+01(<)	**4.1226E+01/4.0991e+00**
$f_{c_{20}}$	3.1944E−01/1.7731e−01(<)	3.9404E−01/2.7855e−01(<)	**6.1210E−03/4.3713e−02**	6.6019E+01/3.9960e+01(<)	4.8118E+01/1.2834e+01(<)	**2.4325E+01/6.3803e+00**	1.8644E+02/1.1838e+02(<)	2.2643E+02/8.0916e+01(<)	**1.0765E+02/4.6360e+01**	1.3964E+03/2.8261e+02(<)	1.4440E+03/2.2498e+02(<)	**1.1451E+03/1.8884e+02**
$f_{c_{21}}$	**1.0160E+02/3.9725e+00**(>)	1.7232E+02/4.7152e+01(≈)	1.6085E+02/5.1423e+01	2.0842E+02/4.4356e+00(<)	2.4619E+02/5.8943e+00(<)	**2.0493E+02/1.7643e+00**	2.1539E+02/1.0176e+01(≈)	3.0266E+02/1.0687e+01(<)	2.1220E+02/2.5567e+00	**2.4578E+02/1.8091e+01**(>)	4.5231E+02/1.6342e+02(<)	2.5778E+02/4.7030e+00
$f_{c_{22}}$	1.0003E+02/9.7205e−02(≈)	1.0002E+02/7.8531e−02(≈)	1.0001E+02/6.2390e−02	1.0000E+02/1.7533e−13(≈)	1.0000E+02/9.5779e−08(≈)	1.0000E+02/1.4352e−14	1.4755E+03/2.0880e+03(<)	8.5275E+02/1.6024e+03(<)	**1.0023E+02/8.1129e−01**	9.9648E+03/9.1041e+02(≈)	1.2293E+04/1.7588e+03(<)	9.7706E+03/6.5640e+02
$f_{c_{23}}$	3.0227E+02/1.6377e+00(<)	3.0080E+02/1.2387e+00(≈)	3.0103E+02/1.6424e+00	3.5592E+02/3.7823e+00(<)	3.9361E+02/5.7075e+00(<)	**3.4863E+02/3.9621e+00**	4.4028E+02/6.0131e+00(<)	4.9365E+02/5.1331e+01(<)	**4.2534E+02/8.9364e+00**	5.8096E+02/8.1264e+00(>)	5.8215E+02/1.0925e+01(≈)	5.8624E+02/1.1831e+01
$f_{c_{24}}$	2.9896E+02/7.8662e+01(<)	3.1390E+02/5.4049e+01(≈)	3.0037E+02/7.4121e+01	4.2806E+02/2.8642e+00(<)	4.8268E+02/1.0666e+01(<)	**4.2237E+02/3.1337e+00**	5.1540E+02/7.3594e+00(<)	5.5620E+02/7.5698e+01(<)	**5.0208E+02/5.4284e+00**	9.1824E+02/1.8189e+01(<)	9.0884E+02/8.5990e+00(<)	**8.8605E+02/1.9122e+01**
$f_{c_{25}}$	4.1671E+02/2.2523e+01(≈)	4.2385E+02/2.2823e+01(<)	4.1665E+02/2.2663e+01	3.8670E+02/8.8616e−03(<)	3.8212E+02/3.9044e+00(≈)	3.8054E+02/6.1941e−01	4.8048E+02/1.6568e+00(<)	4.8658E+02/1.5054e+01(<)	**4.7999E+02/1.5698e+00**	7.0172E+02/4.5339e+01(<)	7.3562E+02/3.5395e+01(<)	**6.6859E+02/4.5924e+01**
$f_{c_{26}}$	3.0000E+02/0.0000e+00(≈)	3.0000E+02/0.0000e+00(≈)	3.0000E+02/0.0000e+00	9.4286E+02/4.5633e+01(<)	**4.7066E+02/4.1805e+02**(>)	8.5195E+02/4.7912e+01	1.1536E+03/5.7706e+01(<)	1.8696E+03/6.0143e+02(<)	**1.0646E+03/6.5387e+01**	3.1408E+03/8.6962e+01(<)	3.2135E+03/9.5935e+01(<)	**2.9579E+03/8.9566e+01**
$f_{c_{27}}$	**3.8945E+02/1.8149e−01**(>)	3.9191E+02/2.2269e+00(>)	3.9298E+02/1.6554e+00	5.0698E+02/5.0219e+00(<)	**4.3592E+02/3.3567e+00**(>)	4.8624E+02/7.4449e+00	5.3414E+02/1.5654e+01(<)	**4.3149E+02/3.2142e+00**(>)	4.7883E+02/9.6464e+00	6.0650E+02/1.8352e+01(<)	6.2961E+02/3.5201e+01(<)	**5.7450E+02/1.3210e+01**
$f_{c_{28}}$	3.0783E+02/4.2607e+01(≈)	3.9916E+02/1.2855e+02(<)	3.1489E+02/6.0169e+01	3.1746E+02/4.0922e+01(≈)	3.1581E+02/3.7244e+01(≈)	3.1320E+02/3.6529e+01	4.5981E+02/6.8398e+00(<)	**4.5819E+02/8.5165e+00**(>)	4.5744E+02/7.7066e−01	5.1386E+02/1.9588e+01(≈)	5.6696E+02/2.8771e+01(<)	5.1843E+02/3.0554e+01
$f_{c_{29}}$	2.3166E+02/2.9645e+00(≈)	**2.2827E+02/2.4533e+00**(>)	2.3206E+02/2.4702e+00	4.4626E+02/1.3604e+01(<)	4.5517E+02/3.2720e+01(<)	**4.2236E+02/7.7182e+00**	3.7896E+02/4.4149e+01(<)	5.3508E+02/6.4281e+01(<)	**3.5628E+02/1.4653e+01**	1.5675E+03/3.0635e+02(<)	1.1032E+03/1.4332e+02(≈)	1.0829E+03/1.4762e+02
$f_{c_{30}}$	1.6451E+04/1.1442e+05(<)	4.0996E+04/1.6392e+05(<)	**7.5192E+03/5.0805e+04**	2.0058E+03/6.1891e+01(<)	2.0130E+03/5.9142e+01(<)	**1.5406E+03/9.1584e+01**	6.5421E+05/7.4416e+04(<)	6.1820E+05/4.3140e+04(<)	**1.5784E+05/4.7688e+04**	2.4184E+03/1.6823e+02(<)	2.5852E+03/1.5940e+02(<)	**1.6010E+03/1.2247e+02**
>/≈/<	4/16/10	3/17/10	-/-/-	4/7/19	2/4/24	-/-/-	6/4/20	2/1/27	-/-/-	12/5/13	1/2/27	-/-/-

Table 14
Our algorithm obtains the best or tie best in comparison with a given algorithm on 30D optimization under $f_{c_1} - f_{c_{30}}$ of our test suite.

Algorithm	Benchmarks on which our algorithm wins or obtains similar performance
LSHADE	$f_{c_1} - f_{c_5}, f_{c_7} - f_{c_9}, f_{c_{11}} - f_{c_{30}}$
jSO	$f_{c_1} - f_{c_5}, f_{c_7} - f_{c_9}, f_{c_{11}} - f_{c_{30}}$
LPalmDE	$f_{c_1} - f_{c_5}, f_{c_7} - f_{c_{30}}$
PaDE	$f_{c_1} - f_{c_5}, f_{c_7} - f_{c_9}, f_{c_{11}} - f_{c_{22}}, f_{c_{23}} - f_{c_{30}}$
Hip-DE	$f_{c_1} - f_{c_5}, f_{c_7} - f_{c_9}, f_{c_{11}} - f_{c_{22}}, f_{c_{25}} - f_{c_{30}}$

Table 15
A certain algorithm obtains the best or tier best performance on 30D optimization under $f_{c_1} - f_{c_{30}}$ of our test suite.

Algorithm	Benchmark on which a certain algorithm wins or obtains similar performance	Total
LSHADE	$f_{c_6}, f_{c_9}, f_{c_{10}}$	3
jSO	$f_{c_6}, f_{c_9}, f_{c_{10}}$	3
LPalmDE	f_{c_6}, f_{c_9}	1
PaDE	$f_{c_6}, f_{c_9}, f_{c_{10}}, f_{c_{23}}$	4
Hip-DE	$f_{c_6}, f_{c_9}, f_{c_{10}}, f_{c_{22}}, f_{c_{23}}$	5
Our algorithm	$f_{c_1} - f_{c_4}, f_{c_4}, f_{c_9} - f_{c_{25}}, f_{c_{28}} - f_{c_{30}}$	25

Table 16
The time complexity comparison on benchmark $f_{a_{14}}$ of our test suite for real-parameter single-objective optimization. The comparison is conducted according to the suggestion of CEC2013 competition.

Algorithms	T_0	T_1	$\widehat{T}_2$	$\frac{\widehat{T}_2 - T_1}{T_0}$
LSHADE	0.0583	0.6117	1.4834	14.9520
jSO			1.5077	15.3688
PaDE			1.7969	20.3293
LPalmDE			3.4237	48.2332
Hip-DE			2.0617	24.8713
Our algorithm			1.9325	22.6552

average and standard deviation ("Ave/Std") are the commonly used metrics for the comparison and they are summarized in Table 19, from which we can see that our FD-DE algorithm is competitive in the real-world constrained optimization.

5. Conclusion

In this paper, a novel DE algorithm with fitness-deviation-based adaptation in parameter control is proposed. The new algorithm mainly has four characteristics. First, an enhanced wavelet basis function is proposed to generate the scale factor F of each individual in the first stage of the evolution. The reason why Cauchy distribution gradually replaced Gaussian distribution in the generation of scale factor F is because of its high probability of making longer jumps. Though the high jumps of F value in the later part of the evolution stage usually help the algorithm jump out some local optima, we found that high jumps in the earlier stage of the evolution may hamper the convergence. That is why an enhanced wavelet basis function is proposed to generate scale factor F of each individual in the first stage of the evolution. Second, a hybrid trial vector generation strategy with perturbation and t-distribution is advanced to generate different trial vectors regarding different stages of the evolution. As is

Table 17
The analysis of k in reduce population stagnation on 30D optimization under benchmarks f_{c_1}–$f_{c_{30}}$ of our test suite.

k	$k = 0.3$	$k = 0.4$	$k = 0.5$	$k = 0.7$	$k = 0.8$	$k = 0.6$
NO.	Mean/Std.	Mean/Std.	Mean/Std.	Mean/Std.	Mean/Std.	Mean/Std.
fc_1	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
fc_2	1.1703E−14/1.5228E−14(>)	**1.0588E−14/1.3878E−14**(>)	1.1703E−14/1.4127E−14(>)	1.2818E−14/1.5373E−14(>)	1.4489E−14/1.8307E−14(>)	1.5047E−14/1.5414E−14
fc_3	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
fc_4	4.5877E+01/8.5547E+00(<)	4.5423E+01/5.5455E+00(>)	4.5540E+01/8.5248E+00(>)	**4.3562E+01/1.0324E+01**(>)	4.6980E+01/6.0228E+00(<)	4.5604E+01/7.5595E+00
fc_5	6.6482E+00/1.3977E+00(>)	6.9681E+00/1.5462E+00(<)	**6.4729E+00/1.1045E+00**(>)	6.6713E+00/1.4116E+00(>)	6.5886E+00/1.3036E+00(>)	6.6765E+00/1.5024E+00
fc_6	2.4465E−05/6.2162E−05(<)	1.8130E−05/3.4389E−05(<)	**1.3024E−05/1.9782E−05**(>)	1.8823E−05/2.6862E−05(<)	2.5599E−05/5.2628E−05(<)	1.3867E−05/2.3357E−05
fc_7	3.7090E+01/1.3147E+00(<)	3.7142E+01/1.2235E+00(<)	3.6911E+01/1.3227E+00(<)	3.7089E+01/1.0443E+00(<)	3.6846E+01/1.2731E+00(<)	**3.6822E+01/1.1578E+00**
fc_8	**6.8377E+00/1.3214E+00**(>)	7.1105E+00/1.4316E+00(>)	6.9740E+00/1.3957E+00(>)	7.4133E+00/1.5900E+00(<)	6.9185E+00/1.6195E+00(>)	7.2335E+00/1.3592E+00
fc_9	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0(≈)*	*0/0*
fc_{10}	1.4274E+03/2.5402E+02(<)	1.4820E+03/2.0053E+02(<)	1.4671E+03/2.3469E+02(<)	1.4079E+03/1.8958E+02(<)	1.4602E+03/1.8267E+02(<)	**1.3946E+03/2.2057E+02**
fc_{11}	5.3465E+00/8.3452E+00(<)	4.4425E+00/8.5643E+00(>)	8.5955E+00/1.5443E+01(<)	**3.7289E+00/6.7348E+00**(>)	5.4596E+00/1.1418E+01(<)	4.5023E+00/8.2185E+00
fc_{12}	1.2680E+02/9.2133E+01(<)	8.6977E+01/8.1550E+01(>)	**8.5155E+01/8.0996E+01**(>)	1.0243E+02/8.2567E+01(<)	1.0090E+02/7.6893E+01(<)	9.9866E+01/7.9844E+01
fc_{13}	1.3161E+01/6.7446E+00(<)	1.1073E+01/6.7981E+00(<)	1.2103E+01/6.7298E+00(<)	1.2688E+01/6.0639E+00(<)	1.1775E+01/6.6325E+00(<)	**1.1035E+01/7.2168E+00**
fc_{14}	2.1148E+01/3.0860E+00(<)	2.0615E+01/4.8100E+00(>)	**2.0488E+01/4.6345E+00**(>)	2.0749E+01/4.2188E+00(>)	2.1592E+01/1.0230E+00(<)	2.1050E+01/2.9524E+00
fc_{15}	1.0535E+00/7.5018E−01(<)	1.0298E+00/6.7783E−01(<)	**8.3783E−01/6.7706E−01**(>)	1.0154E+00/7.8698E−01(<)	1.1916E+00/9.2214E−01(<)	9.6653E−01/8.2778E−01
fc_{16}	7.8764E+01/7.8342E+01(<)	**5.8752E+01/6.8006E+01**(>)	7.6873E+01/8.4157E+01(<)	7.2303E+01/6.9816E+01(<)	8.8243E+01/7.4914E+01(<)	6.6835E+01/7.0080E+01
fc_{17}	**2.4050E+01/6.9667E+00**(>)	2.7850E+01/6.1067E+00(<)	2.5221E+01/7.6662E+00(>)	2.6495E+01/6.7503E+00(>)	2.7161E+01/7.3413E+00(<)	2.7119E+01/6.8624E+00
fc_{18}	2.0564E+01/1.5376E−01(>)	**2.0179E+01/2.8184E+00**(>)	2.0235E+01/2.7022E+00(>)	2.0593E+01/2.4965E−01(<)	2.0566E+01/2.0230E−01(<)	2.0565E+01/1.5357E−01
fc_{19}	4.1610E+00/1.1645E+00(>)	4.4853E+00/1.6122E+00(<)	4.7866E+00/1.8590E+00(<)	4.3181E+00/1.8710E+00(<)	**4.1005E+00/1.5257E+00**(>)	4.3084E+00/1.4229E+00
fc_{20}	2.9412E+01/8.6283E+00(>)	3.1280E+01/1.7611E+01(<)	**2.7243E+01/6.7929E+00**(>)	3.0875E+01/1.8348E+01(<)	2.8765E+01/7.4602E+00(>)	3.0627E+01/1.7944E+01
fc_{21}	2.0705E+02/2.0017E+00(<)	2.0713E+02/1.5496E+00(<)	2.0677E+02/1.6608E+00(>)	**2.0670E+02/1.3885E+00**(>)	2.0679E+02/1.4908E+00(>)	2.0680E+02/1.4528E+00
fc_{22}	*1.0000E+02/1.0808E−13(≈)*	*1.0000E+02/6.3901E−14(≈)*	*1.0000E+02/8.9787E−14(≈)*	*1.0000E+02/1.4352E−14(≈)*	*1.0000E+02/1.4352E−14(≈)*	*1.0000E+02/6.3901E−14*
fc_{23}	3.4478E+02/2.7793E+00(<)	3.4440E+02/3.6992E+00(<)	3.4483E+02/3.0194E+00(<)	3.4389E+02/3.0522E+00(<)	3.4388E+02/2.8514E+00(<)	**3.4376E+02/3.0726E+00**
fc_{24}	4.2122E+02/3.1292E+00(<)	4.2134E+02/2.5840E+00(<)	4.2193E+02/2.4895E+00(<)	4.2145E+02/2.0710E+00(<)	4.2204E+02/2.3652E+00(<)	**4.2106E+02/2.3367E+00**
fc_{25}	3.8606E+02/4.6486E−01(<)	3.8607E+02/4.2912E−01(<)	3.8608E+02/3.8915E−01(<)	**3.8594E+02/4.3678E−01**(>)	3.8597E+02/4.3584E−01(<)	3.8596E+02/4.8819E−01
fc_{26}	8.8720E+02/4.0802E+01(<)	8.8643E+02/3.8244E+01(<)	**8.8021E+02/4.3704E+01**(>)	8.8394E+02/4.6362E+01(<)	8.8204E+02/5.1293E+01(<)	8.8190E+02/3.4251E+01
fc_{27}	4.9357E+02/1.0363E+01(<)	4.9362E+02/7.8831E+00(<)	4.9400E+02/9.3004E+00(<)	4.9237E+02/9.1376E+00(<)	4.9226E+02/9.4838E+00(<)	**4.9168E+02/1.0514E+01**
fc_{28}	3.0873E+02/3.0247E+01(<)	3.1948E+02/4.2560E+01(<)	3.0405E+02/2.0248E+01(<)	3.0873E+02/3.0246E+01(<)	3.1096E+02/3.3612E+01(<)	**3.0000E+02/1.9493E−13**
fc_{29}	4.3058E+02/6.3837E+00(<)	4.3074E+02/6.3438E+00(<)	4.3104E+02/5.5691E+00(<)	4.2995E+02/5.8832E+00(<)	**4.2788E+02/6.5869E+00**(>)	4.2957E+02/8.4912E+00
fc_{30}	1.9530E+03/3.6589E+01(>)	1.9673E+03/3.7095E+01(<)	1.9548E+03/3.2544E+01(>)	**1.9528E+03/2.8054E+01**(>)	1.9632E+03/3.1270E+01(<)	1.9614E+03/2.6331E+01
>/≈/<	8/4/18	8/4/18	13/4/13	10/4/16	7/4/19	-/-/-

Table 18
The summary of the experiment results between our algorithm with and without diversity indicator on 30D optimization under f_{c_1} - $f_{c_{30}}$ of our test suite.

30D	Without indicator	With indicator
NO.	Mean/Std.	Mean/Std.
f_{c_1}	*0/0(≈)*	*0/0*
f_{c_2}	1.4489E−14/1.6448E−14(<)	**1.1703E−14/1.4127E−14**
f_{c_3}	8.9166E−15/2.0878E−14(<)	**0/0**
f_{c_4}	4.5886E+01/8.4582E+00(<)	**4.5540E+01/8.5248E+00**
f_{c_5}	6.5419E+00/1.5850E+00(<)	**6.4729E+00/1.1045E+00**
f_{c_6}	3.9138E−05/1.9544E−04(<)	**1.3024E−05/1.9782E−05**
f_{c_7}	3.7099E+01/1.0819E+00(<)	**3.6911E+01/1.3227E+00**
f_{c_8}	**6.7623E+00/1.5399E+00**(>)	6.9740E+00/1.3957E+00
f_{c_9}	*0/0(≈)*	*0/0*
$f_{c_{10}}$	1.4953E+03/2.2416E+02(<)	**1.4671E+03/2.3469E+02**
$f_{c_{11}}$	**6.5865E+00/1.3889E+01**(>)	8.5955E+00/1.5443E+01
$f_{c_{12}}$	9.9315E+01/7.6863E+01(<)	**8.5155E+01/8.0996E+01**
$f_{c_{13}}$	1.4835E+01/5.3403E+00(<)	**1.2103E+01/6.7298E+00**
$f_{c_{14}}$	**2.0185E+01/5.4760E+00**(>)	2.0488E+01/4.6345E+00
$f_{c_{15}}$	9.3783E−01/6.9517E−01(<)	**8.3783E−01/6.7706E−01**
$f_{c_{16}}$	**6.3117E+01/7.3142E+01**(>)	7.6873E+01/8.4157E+01
$f_{c_{17}}$	2.7622E+01/6.1478E+00(<)	**2.5221E+01/7.6662E+00**
$f_{c_{18}}$	2.0566E+01/1.6143E−01(<)	**2.0235E+01/2.7022E+00**
$f_{c_{19}}$	**4.2666E+00/1.3339E+00**(>)	4.7866E+00/1.8590E+00
$f_{c_{20}}$	3.0904E+01/1.3956E+01(<)	**2.7243E+01/6.7929E+00**
$f_{c_{21}}$	**2.0649E+02/1.5558E+00**(>)	2.0677E+02/1.6608E+00
$f_{c_{22}}$	*1.0000E+02/1.4352E−14(≈)*	*1.0000E+02/8.9787E−14*
$f_{c_{23}}$	3.4512E+02/3.1718E+00(<)	**3.4483E+02/3.0194E+00**
$f_{c_{24}}$	4.2201E+02/2.4237E+00(<)	**4.2193E+02/2.4895E+00**
$f_{c_{25}}$	**3.8602E+02/3.6537E−01**(>)	3.8608E+02/3.8915E−01
$f_{c_{26}}$	8.8813E+02/4.4653E+01(<)	**8.8021E+02/4.3704E+01**
$f_{c_{27}}$	**4.9267E+02/9.8627E+00**(>)	4.9400E+02/9.3004E+00
$f_{c_{28}}$	3.1522E+02/3.8588E+01(<)	**3.0405E+02/2.0248E+01**
$f_{c_{29}}$	4.3208E+02/6.5504E+00(<)	**4.3104E+02/5.5691E+00**
$f_{c_{30}}$	1.9583E+03/2.8843E+01(<)	**1.9548E+03/3.2544E+01**
>/≈/<	8/3/19	-/-/-

Table 19
The optimization results comparison among sCMAgES, EnMODE, COLSHADE and our FD-DE for the real-world tension/compression spring design optimization.

Algorithms	Best	Worst	Ave/Std
sCMAgES	1.2665233061E−02	1.2683764357E−02	1.2667735915E−02/4.6998E−05
EnMODE	1.2665233897E−02	1.2718978442E−02	1.2719354786E−02/2.0138E−05
COLSHADE	1.2665232989E−02	1.2666738416E−02	1.2665358735E−02/1.0625E−11
Our algorithm	**1.2665232798E−02**	**1.2665264873E−02**	**1.2665235751E−02/1.4510E−05**

known to all, the trial vector generation strategy dominates the overall performance of DE variants, and DE variants equipped with JADE's mutation strategy DE/target-to-*p*best/1/bin secure front rank in the recent competitions, however, the quality of individuals is bad in the later part of the evolution when tackling complex optimization problems. That is why we proposed a hybrid trial vector generation strategy with perturbation and t-distribution. Third, a fitness-deviation-based parameter control is proposed for the adaptation of control parameters during the evolution. This adaptation scheme of parameter control can be considered as a further extension of the parameter control in recent winner DE variants. The fitness deviation rather than the basic fitness improvement is taken into consideration. Fourth, a novel diversity indicator is proposed and a restart scheme can be launched if necessary when the diversity is detected bad. The indicator is actually a quality indicator, and restart is launched when the quality of the individuals is detected to be bad. Our algorithm is validated under a large test suite containing 130 benchmarks from the universal test suites on single-objective optimization and the experiment results support the superiority of our algorithm. Moreover, our algorithm is also validated under real-world optimization applications, and the results also support its superiority.

Declaration of competing interest

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

Acknowledgments

This work is supported by the Natural Science Foundation of Fujian Province, China (Grant No. 2021J05227, 2022Y0077), Fuxiaquan National Independent Innovation Demonstration Zone collaborative innovation platform project (3502ZCQXT2021009) and Scientific Research Startup Foundation of Fujian University of Technology, China (GY-Z19013).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.isatra.2023.05.005.

References

[1] Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 1997;11(4):341–59.
[2] Brest J, Greiner S, Bošković B, Mernik M, Zumer V. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. Evol Comput IEEE Trans 2006;10(6):646–57.
[3] Das S, Abraham A, Chakraborty UK, Konar A. Differential evolution using a neighborhood-based mutation operator. IEEE Trans Evol Comput 2009;13(3):526–53.
[4] Yu Y, Gao S, Zhou M, Wang Y, Lei Z, Zhang T, et al. Scale-free network-based differential evolution to solve function optimization and parameter estimation of photovoltaic models. Swarm Evol Comput 2022;74:101142.
[5] Gao S, Yu Y, Wang Y, Wang J, Cheng J, Zhou M. Chaotic local search-based differential evolution algorithms for optimization. IEEE Trans Syst Man Cybern 2019;51(6):3954–67.
[6] Yu Y, Lei Z, Wang Y, Zhang T, Peng C, Gao S. Improving dendritic neuron model with dynamic scale-free network-based differential evolution. IEEE/CAA J Autom Sin 2021;9(1):99–110.
[7] Nanda SJ, Panda G. A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm Evol Comput 2014;16:1–18.
[8] Meng Z, Pan J-S. Monkey king evolution: A new memetic evolutionary algorithm and its application in vehicle fuel consumption optimization. Knowl-Based Syst 2016;97c:144–57.
[9] Meng Z, Pan J-S. QUasi-affine TRansformation evolution with external ARchive (QUATRE-EAR): an enhanced structure for differential evolution. Knowl-Based Syst 2018;155:35–53.
[10] Chen X, Li K. Collective information-based particle swarm optimization for multi-fuel CHP economic dispatch problem. Knowl-Based Syst 2022;248:108902.
[11] Brest J, Maučec MS. Population size reduction for the differential evolution algorithm. Appl Intell 2008;29(3):228–47.
[12] Gao S, Wang K, Tao S, Jin T, Dai H, Cheng J. A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models. Energy Convers Manage 2021;230:113784.
[13] Meng Z, Yang C. Two-stage differential evolution with novel parameter control. Inform Sci 2022;596:321–42.
[14] Chen X, Shen A. Self-adaptive differential evolution with Gaussian–Cauchy mutation for large-scale CHP economic dispatch problem. Neural Comput Appl 2022;34(14):11769–87.
[15] Brest J, Zamuda A, Fister I, Boskovic B. Some improvements of the self-adaptive jde algorithm. In: 2014 IEEE symposium on differential evolution. IEEE; 2014, p. 1–8.
[16] Tanabe R, Fukunaga AS. Improving the search performance of SHADE using linear population size reduction. In: 2014 IEEE congress on evolutionary computation. IEEE; 2014, p. 1658–65.
[17] Chu S-C, Xu X-W, Yang S-Y, Pan J-S. Parallel fish migration optimization with compact technology based on memory principle for wireless sensor networks. Knowl-Based Syst 2022;241:108124.
[18] Pan J-S, Zhang L-G, Wang R-B, Snášel V, Chu S-C. Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems. Math Comput Simulation 2022;202:343–73.
[19] Wu G, Mallipeddi R, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. Technical report, 2017, p. 1–16.
[20] Pan J-S, Meng Z, Chu S-C, Xu H-R. Monkey king evolution: an enhanced ebb-tide-fish algorithm for global optimization and its application in vehicle navigation under wireless sensor network environment. Telecommun Syst 2017;65(3):351–64.
[21] Meng Z, Pan J-S, Zheng W-m. Differential evolution utilizing a handful top superior individuals with bionic bi-population structure for the enhancement of optimization performance. Enterp Inf Syst 2018;1–22.
[22] Meng Z. Dimension improvements based adaptation of control parameters in differential evolution: A fitness-value-independent approach. Expert Syst Appl 2023;223:119848.
[23] Islam SM, Das S, Ghosh S, Roy S, Suganthan PN. An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. IEEE Trans Syst Man Cybern B 2011;42(2):482–500.
[24] Yu W-J, Shen M, Chen W-N, Zhan Z-H, Gong Y-J, Lin Y, et al. Differential evolution with two-level parameter adaptation. IEEE Trans Cybern 2013;44(7):1080–99.
[25] Cai Y, Liao J, Wang T, Chen Y, Tian H. Social learning differential evolution. Inform Sci 2018;433:464–509.
[26] Wu T-Y, Wang T, Lee Y-Q, Zheng W, Kumari S, Kumar S. Improved authenticated key agreement scheme for fog-driven IoT healthcare system. Secur Commun Netw 2021;2021.
[27] Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans Evol Comput 2011;15(1):55–66.
[28] Gong W, Fialho Á, Cai Z, Li H. Adaptive strategy selection in differential evolution for numerical optimization: an empirical study. Inform Sci 2011;181(24):5364–86.
[29] Gong W, Cai Z. Differential evolution with ranking-based mutation operators. IEEE Trans Cybern 2013;43(6):2066–81.
[30] Wu T, Guo X, Chen Y, Kumari S, Chen C. Amassing the security: An enhanced authentication protocol for drone communications over 5G networks. Drones 2022;6(1):10.
[31] Zhang J, Sanderson AC. JADE: adaptive differential evolution with optional external archive. Evol Comput IEEE Trans 2009;13(5):945–58.
[32] Meng Z, Pan J-S, Kong L. Parameters with adaptive learning mechanism (PALM) for the enhancement of differential evolution. Knowl-Based Syst 2018;141:92–112.
[33] Tanabe R, Fukunaga A. Success-history based parameter adaptation for differential evolution. In: 2013 IEEE congress on evolutionary computation. IEEE; 2013, p. 71–8.
[34] Brest J, Maučec MS, Bošković B. Single objective real-parameter optimization: Algorithm jSO. In: 2017 IEEE congress on evolutionary computation. IEEE; 2017, p. 1311–8.
[35] Meng Z, Pan J-S, Tseng K-K. PaDE: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization. Knowl-Based Syst 2019;168:80–99.
[36] Meng Z, Pan J-S, Xu H. QUasi-affine TRansformation evolutionary (QUATRE) algorithm: a cooperative swarm based algorithm for global optimization. Knowl-Based Syst 2016;109:104–21.
[37] Cui L, Li G, Zhu Z, Lin Q, Wong K-C, Chen J, et al. Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism. Inform Sci 2018;422:122–43.
[38] Meng Z, Pan J-S. HARD-DE: Hierarchical ARchive based mutation strategy with depth information of evolution for the enhancement of differential evolution on numerical optimization. IEEE Access 2019;7:12832–54.
[39] Deng W, Xu J, Song Y, Zhao H. Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem. Appl Soft Comput 2021;100:106724.
[40] Li W, Meng X, Huang Y. Fitness distance correlation and mixed search strategy for differential evolution. Neurocomputing 2021;458:514–25.
[41] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1(1):67–82.
[42] Duénez-Guzmán EA, Vose MD. No free lunch and benchmarks. Evol Comput 2013;21(2):293–312.
[43] Meng Z, Yang C. Hip-DE: Historical population based mutation strategy in differential evolution with parameter adaptive mechanism. Inform Sci 2021;562:44–77.
[44] Meng Z, Pan J-S. Quasi-affine transformation evolutionary (QUATRE) algorithm: the framework analysis for global optimization and application in hand gesture segmentation. In: 2016 IEEE 13th international conference on signal processing. IEEE; 2016, p. 1832–7.
[45] Meng Z, Pan J-S. A competitive quasi-affine transformation evolutionary (C-QUATRE) algorithm for global optimization. In: 2016 IEEE international conference on systems, man, and cybernetics. IEEE; 2016, p. 001644–9.
[46] Meng Z, Pan J-S, Li X. The quasi-affine transformation evolution (QUATRE) algorithm: an overview. In: The Euro-China conference on intelligent data analysis and applications. Springer; 2017, p. 324–33.
[47] Meng Z, Zhong Y, Yang C. CS-DE: Cooperative strategy based differential evolution with population diversity enhancement. Inform Sci 2021;577:663–96.
[48] Osuna-Enciso V, Cuevas E, Castañeda BM. A diversity metric for population-based metaheuristic algorithms. Inform Sci 2022;586:192–208.
[49] Tian M, Gao X. Differential evolution with neighborhood-based adaptive evolution mechanism for numerical optimization. Inform Sci 2019;478:422–48.
[50] Kumar A, Wu G, Ali MZ, Mallipeddi R, Suganthan PN, Das S. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. Swarm Evol Comput 2020;56:100693.
[51] Kumar A, Das S, Zelinka I. A modified covariance matrix adaptation evolution strategy for real-world constrained optimization problems. In: Proceedings of the 2020 genetic and evolutionary computation conference companion. 2020, p. 11–2.
[52] Gurrola-Ramos J, Hernàndez-Aguirre A, Dalmau-Cedeño O. COLSHADE for real-world single-objective constrained optimization problems. In: 2020 IEEE congress on evolutionary computation. IEEE; 2020, p. 1–8.