Information Sciences 579 (2021) 33–54

Contents lists available at ScienceDirect

Information Sciences

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

NFDDE: A novelty-hybrid-fitness driving differential evolution algorithm

Xuewen Xia [a,c,*], Lei Tong [b], Yinglong Zhang [a,c,*], Xing Xu [a,c], Honghe Yang [a,c], Ling Gui [a,c], Yuanxiang Li [c,d], Kangshun Li [c,e]

[a] College of Physics and Information Engineering, Minnan Normal University, Zhangzhou, China
[b] School of Tourism Management, Wuhan Business University, Wuhan, China
[c] Key Lab of Intelligent Optimization and Information Processing, Minnan Normal University, Zhangzhou, China
[d] School of Computer, Wuhan University, Wuhan, China
[e] College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China

ARTICLE INFO

Article history:
Received 7 May 2021
Received in revised form 22 July 2021
Accepted 23 July 2021
Available online 31 July 2021

Keywords:
Differential evolution
Novelty-based driving force
Adaptive scaling factors
Adjustment of population size

ABSTRACT

In differential evolution algorithm (DE), it is a widely accepted method that selecting individuals with higher fitness to generate a mutant vector. In this case, the population evolution is under a fitness-based driving force. Although the driving force is beneficial for the exploitation, it sacrifices performance on the exploration. In this paper, a **n**ovelty-hybrid-**f**itness **d**riving force is introduced to trade off contradictions between the exploration and the exploitation of DE. In the new proposed DE, named as NFDDE, both fitness and novelty values of individuals are considered when choosing individuals to create mutant vectors. In addition, two adaptive scaling factors are proposed to adjust the weights of the fitness-based driving force and the novelty-based driving force, respectively, and then distinct properties of the two driving forces can be effectively utilized. At last, to save computational resources, some individuals with lower novelty are deleted when the population has converged to a certain extent. The comprehensive performance of NFDDE is extensively evaluated by comparisons between it and other 9 state-of-art DE variants based on CEC2017 test suite. In addition, distinct properties of the newly introduced strategies and involved parameters are further confirmed by a set of experiments.

1. Introduction

In recent years, metaheuristics have attracted many scholars' attention, and have been widely applied in many real applications. Generally, most of the metaheuristics are inspired by various biological phenomena, physical phenomena, or social phenomena. For instance, gray wolf optimizer [21] and harris hawk optimizer [12] are inspired by biological phenomena, while gravitational search [30] and heat transfer search [25] are enlightened by some common physical phenomena. Moreover, teaching learning based optimization [29] can be regarded as a social phenomena inspired metaheuristic. Compared with traditional gradient-based algorithms, the metaheuristics have more competitive advantages, such as favorable global search ability, easy implementation, and not requiring gradient information.

* Corresponding authors at: College of Physics and Information Engineering, Minnan Normal University, Zhangzhou, China.
E-mail addresses: xwxia@whu.edu.cn (X. Xia), zhang_yinglong@126.com (Y. Zhang).

https://doi.org/10.1016/j.ins.2021.07.082
0020-0255/

The differential evolution (DE) algorithm proposed in 1997 [34] is a popular metaheuristic algorithm, especially for numerical optimization. In the last decades, DE has been successfully applied in various fields for its simple implementation and promising performance [31,47,50]. Similar to other evolutionary algorithms (EAs), an individual in DE commonly denoted as a vector is regarded as a candidate solution to a problem. In each generation, individuals in DE breed their offspring based on three operators, i.e., mutation, crossover, and selection operators. However, distinct from other EAs, the salient characteristic of DE lies in its mutation operator. Concretely, the mutation operator in DE is used to generate differential information among different individuals, while direct selecting gene values from different individuals is widely accepted in mutation operators of other EAs.

In DE, an individual adjusts its search direction and step length mainly depending on a "guidance" of the differential information. Thus, the differential information generated by the mutation operator can be regarded as a "driving force" for the individual. Consequently, how to generate an efficient "driving force" (i.e., the differential information) by a proper mutation operator as well as corresponding parameters is crucial for the performance of DE.

Generally, it is a popular method that defining a fitness function based on a given task to evaluate candidate solutions' quality. Without loss of generality, a fitness value means an objective function value for a give problem in this paper. Hence, various driving forces based on the fitness value are generated in different DEs. For example, an elite individual is widely accepted as a driving force for other individuals' evolution, such as the "DE/best/1" and "DE/current-to-best/1" mutation operators. Although the fitness-based driving force is intuitively reasonable, it is associated with a few issues especially in complicated multimodal problems of which deception is one of the most prominent [10,14,15]. Moreover, as the complexity of a problem increasing, crafting an appropriate fitness function becomes more difficult because the fitness-based driving force is more vulnerable to deception, which is harmful for the exploration. To improve the exploration ability, integrating randomness, which can be regarded as a random-based driving force, into the fitness-based driving force is an easy and popular strategy for optimizing multimodal problems because it can enhance the population diversity. Commonly, "DE/rand/1" and the "DE/rand/2", which are two popular mutation operators in DE, can be treated as the random-based driving forces.

Although the fitness-based driving force and the randomness-based driving force offer favorable performance on unimodal problems and multimodal problems, respectively, it is very difficult to predefine an efficient and accurate driving force for a population in advance due to that many real applications are "black-box" problems. In other words, it is infeasible in DE to design the most favorable mutation operator for different problems. In fact, not only different problems have different characteristics, but also a problem have distinct properties in different fitness landscapes. Thus, it is also unrealistic that applying a trial-and-error scheme to determine the most efficient driving force, i.e., the most favorable mutation operator in DE, for a specific problem.

In contrast to focusing on the objective optimization in many EAs, a few researchers in artificial life pay close attention to systems without explicit objectives [5,33]. A widely accepted approach in the field is creating a complex artificial system with a more novelty rather than a higher fitness. Some experiments verify that the novelty-based search is immune to the problems of deception and local optima, which is inherent in the fitness-based search, because the novelty-based search entirely ignores an explicit objective. The study results suggest a counter-intuitive conclusion that ignoring (or partial ignoring) an objective in this way may be conducive to searching for the objective [10,14,15]. However, it is dangerous to suggest that the novelty-based search is better than the traditional fitness-based search in general. In fact, the novelty-based search sacrifices performance of the exploitation.

Inspired by the above discussions, a novelty-hybrid-fitness driving DE (NFDDE) is proposed in this study. In NFDDE, fitness and novelty are two basic driving forces for the population evolution. In each generation, both the fitness and the novelty of individuals are computed. Based on the measured results, those individuals with higher fitness and novelty values are selected to perform the mutation operator, and then the novelty-hybrid-fitness driving force generated by the operator is used to guide the population to promising areas. Furthermore, to efficiently utilize merits of the two driving forces, two adaptive scaling factors are introduced in NFDDE based on their historical performance. Lastly, considering that rational allocating computational resources is another important issue in EAs, hence, in this study, some individuals with low novelty are deleted from the current population when the population has converged to a certain extent. Thus, the computational resources can be saved during the evolutionary process without compromising the solution accuracy.

The remainder of this paper is organized as follows. We first discuss the related work of DE in Section 2 and then give a detail description of NFDDE in Section 3. Extensive experiments between NFDDE and other state-of-art DE variants are presented in Section 4. Moreover, to verify the performance of the newly introduced strategies in NFDDE, sensitivity analysis of the strategies is also detailed in this section. Finally, conclusions are provided in Section 5.

2. Related work

As one of the simplest EAs, DE has been proven to be a very promising metaheuristic for numerical optimization. In this section, four basic operators and two related parameters of DE are briefly introduced. Furthermore, various improvements of DE are also discussed in this section.

2.1. DE

2.1.1. Initialization

An initial population $\left\{\mathbf{X}_i^t = \left(x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,D}^t\right) \mid i = 1, 2, \ldots, N\right\}$ of DE is randomly generated in a feasible space, where D is the number of variables, and N is the population size. After that, the population consecutively applies mutation, crossover, and selection operators to generate offspring in each generation.

2.1.2. Mutation

In each generation t, a mutation operator is used to create a mutant vector $\mathbf{V}_i^t = \left(v_{i,1}^t, v_{i,2}^t, \ldots, v_{i,D}^t\right)$ for a target vector $\mathbf{X}_i^t$. Generally, there are 6 widely applicable mutation strategies, the details of which are described as follows.

$$DE/rand/1: \quad \mathbf{V}_i^t = \mathbf{X}_{r_1}^t + F \cdot \left(\mathbf{X}_{r_2}^t - \mathbf{X}_{r_3}^t\right) \tag{1}$$

$$DE/rand/2: \quad \mathbf{V}_i^t = \mathbf{X}_{r_1}^t + F \cdot \left(\mathbf{X}_{r_2}^t - \mathbf{X}_{r_3}^t + \mathbf{X}_{r_4}^t - \mathbf{X}_{r_5}^t\right) \tag{2}$$

$$DE/best/1: \quad \mathbf{V}_i^t = \mathbf{X}_{best}^t + F \cdot \left(\mathbf{X}_{r_1}^t - \mathbf{X}_{r_2}^t\right) \tag{3}$$

$$DE/best/2: \quad \mathbf{V}_i^t = \mathbf{X}_{best}^t + F \cdot \left(\mathbf{X}_{r_1}^t - \mathbf{X}_{r_2}^t + \mathbf{X}_{r_3}^t - \mathbf{X}_{r_4}^t\right) \tag{4}$$

$$DE/current-to-best/1: \quad \mathbf{V}_i^t = \mathbf{X}_i^t + F \cdot \left(\mathbf{X}_{best}^t - \mathbf{X}_i^t + \mathbf{X}_{r_1}^t - \mathbf{X}_{r_2}^t\right) \tag{5}$$

$$DE/rand-to-best/1: \quad \mathbf{V}_i^t = \mathbf{X}_{r_1}^t + F \cdot \left(\mathbf{X}_{best}^t - \mathbf{X}_{r_1}^t + \mathbf{X}_{r_2}^t - \mathbf{X}_{r_3}^t\right) \tag{6}$$

In the above equations, r_1, r_2, r_3, r_4, and r_5 are mutually exclusive integers randomly selected from the set $\{1, 2, \ldots, N\} \setminus \{i\}$; the vector $\mathbf{X}_{best}^t$ denotes the best individual in the population at the generation t; the parameter F regarded as a scaling factor is a positive value, which is used to scale the difference between two different individuals. For instance, the generation of a mutation vector $\mathbf{V}_i$ based on the mutation strategy "*DE/rand-to-best/2*" can be demonstrated by Fig. 1.

Note that, if $v_{i,j}^t$ exceeds out of a feasible region $[L_j, U_j]$, it needs to be reset within the feasible region by Eq. (7).

$$v_{i,j}^t = \begin{cases} min\left\{U_j, 2L_j - v_{i,j}^t\right\}, & \text{if } v_{i,j}^t < L_j \\ max\left\{L_j, 2U_j - v_{i,j}^t\right\}, & \text{if } v_{i,j}^t > U_j \end{cases} \tag{7}$$

where U_j and L_j are the upper and lower boundaries of the j^{th} variable, respectively.

2.1.3. Crossover

Based on the target vector $\mathbf{X}_i^t$ and the corresponding mutant vector $\mathbf{V}_i^t$, a crossover operator is executed to create a trial vector $\mathbf{U}_i^t = \left(u_{i,1}^t, u_{i,2}^t, \ldots, u_{i,D}^t\right)$. The binomial crossover, defined as Eq. (8), is widely accepted in many DE variants [28,40,48].

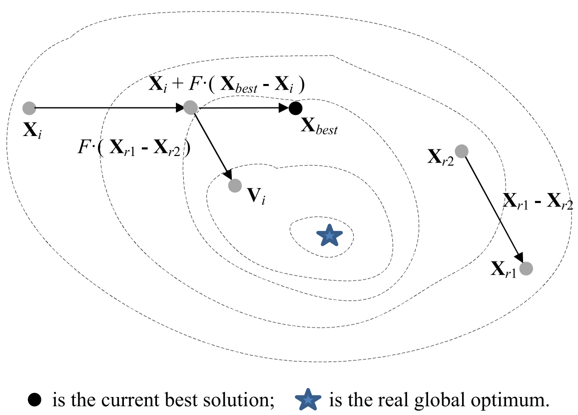


Fig. 1. Illustration of the "DE/rand-to-best/1" mutation strategy. The dashed curves denote the contours of the optimization problem. $\mathbf{V}_i$ is the mutation vector for the individual $\mathbf{X}_i$.

$$u_{i,j}^t = \begin{cases} v_{i,j}^t, & \text{if } rand_j(0,1) \leqslant CR \text{ or } j = j_{rand} \\ x_{i,j}^t, & \text{otherwise} \end{cases} \tag{8}$$

where $rand_j$ is a random number uniformly distributed in the range [0, 1]; CR is a crossover rate applied to determine the probability that $u_{i,j}^t$ is copied from $v_{i,j}^t$ or $x_{i,j}^t$; and j_{rand} is an integer randomly generated in the range [1, D].

2.1.4. Selection

Once the trial vector $\mathbf{U}_i^t$ is produced, the better one between $\mathbf{X}_i^t$ and $\mathbf{U}_i^t$ is selected to the next generation. For a minimization problem, the selection operator can be described as Eq. (9).

$$\mathbf{X}_i^{t+1} = \begin{cases} \mathbf{X}_i^t, & \text{if } f(\mathbf{X}_i^t) < f(\mathbf{U}_i^t) \\ \mathbf{U}_i^t, & \text{otherwise} \end{cases} \tag{9}$$

where $f(x)$ is the function value of the solution x.

2.2. Literature review

The performance of DE highly depends on its trial vector generation strategy and corresponding parameters. Thus, extensive research in the last decades focused on studying distinct characteristics of them. For example, it has been observed that "*DE/rand/1*" and "*DE/rand/2*" are favorable for maintaining population diversity but less efficient in speeding up of convergence [48], whereas "*DE/best/1*" and "*DE/best/2*" are conducive to the exploitation though they cannot offer promising performance on the exploration. Thus, to alleviate contradictions between the exploitation and the exploration, "*DE/current-to-best/1*" and "*DE/rand-to-best/1*" are proposed [23]. Moreover, many empirical study also investigated the performance of the parameters in DE [27,41]. For example, Storn suggested that a favorable range of F is in the range [0.4, 1.0] [34], and N should be between 5D and 10D [27]. In addition, many other studies also put forward different suggestions for the parameters.

Although distinct characteristic of different strategies and parameter settings have been verified by extensive experiments [23,24,26], it is still very difficult to choose the most promising parameters or strategies in advance due to that correlations among the different parameters or strategies have not been completely understood. Furthermore, not only different problems have their own characteristics, but also different fitness landscapes in a problem have distinct properties. Consequently, a set of static parameters and strategies is not suitable for complicated problems, and even for different fitness landscapes in a same problem. Hence, various hybrid strategies and (self-) adaptive mechanisms are applied in DE to satisfy distinct requirements of different problems or fitness landscapes [1,17,18,28,35,48]. The main motivation of the study is applying different strategies to improve population diversity and speed up the convergence at initial evolution stage and later evolution stage, respectively [42,46].

Considering different mutation strategies have their own distinct properties, many scholars pour much attention on hybrid various mutation strategies. In SaDE [28], for instance, a population adaptive selects "*DE/rand/1*" or "*DE/current-to-best/1*" as its mutation operator in each generation according to historical success ratios of the two mutation operators. Motivated by the research, Wang [40] uses three popular mutation operators as basic operators saved in a candidate pool. In each generation, an individual randomly chooses a mutation operator from the pool to generate offspring. In TSDE [17], three mutation operators conducive to the exploration are used to improve the exploration in the initial evolution stage. On the contrary, two mutation operators beneficial for the exploitation are adopted to speed up the convergence process in the later evolution stage. Moreover, introducing dynamic population size strategy is also an feasible method to trade-off the exploration and the exploitation based on an intuition that more population size is beneficial for global search while less population size is favorable for local search [2,4,35,38].

Furthermore, some adaptive strategies for F and CR are also investigated in many studies [16,36,39,48]. For example, Liu et al. [16] introduces a fuzzy adaptive DE (FADE), in which two fuzzy logic controllers are adopted to tune F and CR during the evolution process. Moreover, it is also a popular and widely accepted strategy that applying a set of parameters who display favorable historical performance to generate offspring in the last generations. In JADE [48], for instance, information of recent successful F and CR is applied to adjust the subsequent F and CR, respectively. Based on it, a state-of-art DE algorithm named as SHADE is proposed by Tanabe [36] in 2013. Furthermore, the neighbor topology popularly applied in particle swarm optimization algorithm is also introduced in DE [7,19,20,38,45,49], and then corresponding adaptive methods for mutation strategy and parameters are developed. For instance, multi-population mechanisms and the dynamic neighborhood models offer promising performance on the exploration ability [19,45,49]. Some study also verify that organically integrating DE with other EAs or search strategies can bring helpful and positive improvements for DE [8,31,42].

The core motivation of this type of adaptive mechanisms is based on an assumption that the most suitable mutation strategy and parameters in the last few generations may display more promising performance in future generations [2,9,11,17,40]. Due to that these adaptive mechanisms only take advantages of historical experiences of the population rather than a user's prior knowledge, it can display very pleasurable and robust performance on different problems. However, the fitness value is the only metric when performing the adaptive adjustments. Thus, the diversity loss problem caused by over-reliance on the fitness-based driving force is still cannot solved elegantly. In other words, it remains to be studied that how to balance the contradiction between the exploration and the exploitation during the evolutionary process.

3. Novelty-hybrid-fitness driving DE

3.1. Motivations

In EA research field, the most common strategy is evaluating individuals with respect to a specific objective function as their fitness values, and then creating drive forces based on the values to guide the population evolution. Although the evolution process under the fitness-based driving force may be intuitively reasonable, it may cause rapid loss of diversity, and then leads the population to be easily trapped into local optima.

In contrast to the focus on the fitness value in EA community, many scholars in artificial life often study problems without explicit objectives. A typical approach is to create an open-ended system by searching directly for behavioral novelty instead of pursuing a higher fitness [10,14]. Some researches on robotics systems [10,15] and neural networks [13] verify that driving force based on the behavioral novelty can display very favorable performance on many tasks.

Inspired by the above discussions, a novelty-hybrid-fitness driving DE (NFDDE) is proposed, in which fitness and novelty are two metrics when generating efficient and helpful driving forces. In each generation, not only the fitness of individuals are measured, but also the novelty of the individuals are also considered. As a result, when an individual selecting other individuals to perform the mutation operator, both the fitness and the novelty values are considered. The main steps in NFDDE are detailed as follows.

3.2. Framework of NFDDE

From the motivations of NFDDE we known that a main difference between it and the original DE algorithm is the selection of individuals applied in the mutation operator. With the new proposed mutation operator in NFDDE, the population evolution is driven by two types of individuals, i.e., elite individuals and novel individuals. Furthermore, an adaptive strategy is used to adjust the scaling factor and crossover rate in NFDDE. These new proposed strategies are detailed as follows.

In NFDDE, "novelty" is a new introduced "driving force". Thus, how to measure an individual's novelty should to be firstly dealt with. Generally, a novelty value measures how unique an individual's behavior is. In DE, the novelty of an individual can be measured relative to other individuals in the population. Hence, a simple method to characterize an individual's novelty is measure the average distance between the individual and its the K-nearest neighbors. The novelty of the individual $\mathbf{X}_i$ can be defined as Eq. (10).

$$\mathcal{N}_i^t = \frac{1}{K}\sum_{j=1}^{K} dist\left(\mathbf{X}_i, \mu_j\right) \tag{10}$$

where μ_j is the jth-nearest neighbor of $\mathbf{X}_i$ with respect to the distance metric *dist*, which is a domain-dependent measure of behavioral difference between two individuals in the search space. In this study, for simplicity, $dist\left(\mathbf{X}_i, \mu_j\right)$ denotes Euclidean distance between $\mathbf{X}_i$ and μ_j.

It can be observed from Eq. (10) that the value of K determines an individual's novelty in a population. In fact, different values of K can bring different novelties for individuals in a same population distribution. From the distribution of a population in a 2-D space demonstrated by Fig. 2, for instance, the individual A (the solid point in the figure) has the lowest novelty when $K = 1$. On the contrary, when $K = 2$ or 3, the individual A has the highest novelty. In this study, $K = 2$ is adopted based on a set of experiments.

Based on the novelty value, the new mutation operator introduced in NFDDE is defined as Eq. (11).

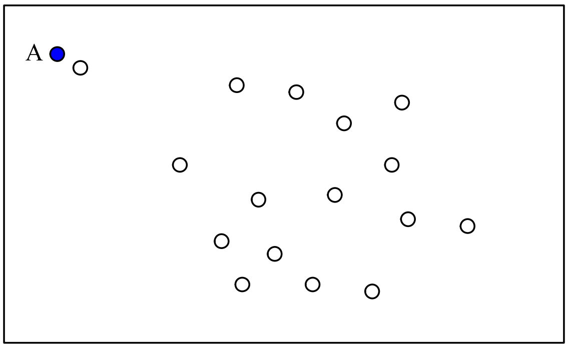


Fig. 2. Based on the distribution of a population, the individual A (the solid circle) has the lowest novelty when it has one neighbor. On the contrary, it has the highest novelty when it has 2 or 3 neighbors.

$$\mathbf{V}_i^t = \mathbf{X}_i^t + F_{1,i}^t \cdot \left(\mathbf{X}_{pBest}^t - \mathbf{X}_i^t\right) + F_{2,i}^t \cdot \left(\mathbf{X}_{pNovel}^t - \mathbf{X}_{r_1}^t\right) \tag{11}$$

where $\mathbf{X}_{pBest}^t$ and $\mathbf{X}_{pNovel}^t$ are two individuals which are randomly selected from the top p best elite individuals and the top p novelty individuals at the generation t, respectively.

Note that, the parts "$\mathbf{X}_{pBest}^t - \mathbf{X}_i^t$" and "$\mathbf{X}_{pNovel}^t - \mathbf{X}_{r1}^t$" in Eq. (11) can be regarded as the "fitness driving" part and the "novelty driving" part, respectively. Unlike only one scale factor is applied in some popular DE algorithms, there are two scale factors in NFDDE. Concretely, $F_{1,i}^t$ and $F_{2,i}^t$ are two scaling factors for the "fitness driving" part and the "novelty driving" part, respectively. Since the primary objectives of the initial stage and the later stage are enhancing the exploration and the exploitation, respectively, the value of p should be changed during the search process. In NFDDE, a linearly decreasing p is introduce to tune strengths of the two driving force, and then to satisfy the different requirements of the different stages. In this study, the value of p at the generation t is defined as Eq. (12).

$$p^t = 0.5 - \frac{t}{T} \times 0.4 \tag{12}$$

where T is the predefined the maximum generation.

Furthermore, a history-based scheme applied in SHADE [36] is used to adjust $F_{1,i}^t, F_{2,i}^t$ and CR_i^t in NFDDE.

$$F_{m,i}^t = randc(\mu f_m, 0.1), \;\; m = 1, 2 \tag{13}$$

$$CR_i^t = randc(\mu cr, 0.1) \tag{14}$$

$$\mu f_m = mean_{WL}(S_{f_m}), \;\; m = 1, 2 \tag{15}$$

$$\mu cr = mean_{WL}(S_{cr}) \tag{16}$$

As in [48], the weighted Lehmer mean $mean_{WL}(S)$ computed by Eq. (17) is used to influence the parameter adaptation (S refers to S_{f_m} or S_{CR}).

$$mean_{WL}(S) = \frac{\sum_{k=1}^{|S|} w_k \cdot S_k^2}{\sum_{k=1}^{|S|} w_k \cdot S_k} \tag{17}$$

$$w_k = \frac{\Delta f_k}{\sum_{j=1}^{|S|} \Delta f_j}, \;\; k = 1, 2 \tag{18}$$

Note that, when computing $mean_{WL}(S_{f_1})$ and $mean_{WL}(S_{cr}), \Delta f_k$ in Eq. (18) denotes the improvement of all the improved individuals, i.e., $\Delta f_{cr} = \Delta f_1 = |f\left(\mathbf{X}_k^{t+1}\right) - f(\mathbf{X}_k^t)|$. On the contrary, when measuring $mean_{WL}(S_{f_2}), \Delta f_2$ indicates the improvement of those improved individuals with a higher novelty. Concretely, before computing $mean_{WL}(S_{f_2})$, the novelty of an individual is computed by Eq. (10). Based on the results, the best half individuals measured by the novelty values are selected to calculate the improvement.

In each generation, unlike the update method proposed in SHADE [36], all individuals in NFDDE use a same μf_m and μcr to generate its scaling factors and crossover rates, respectively. During the evolution process, the successful values of $F_{m,i}^t$ and CR_i^t, which are able to generate a more favorable offspring, are used to update μf_m and μcr based on Eqs. (17) and (18). Note that μf_m (m = 1, 2) and μcr are initialized as 0.5 at the first generation.

During the evolution process, to save computational resources, those individuals with the lowest novelty are deleted from the current population if the population has converged to a certain extent. The deleting process of individuals is defined as Eq. (19).

$$NP = \begin{cases} N - N_{del}, & \text{if } max\left(\mathcal{N}_i^t\right) < 0.1 \times \|U_{pop} - L_{pop}\| \quad and \quad N > N_{min} \\ N, & \text{otherwise} \end{cases} \tag{19}$$

where N_{del} means the number of deleted individuals; U_{pop} and L_{pop} are the last upper and the lower boundaries of the population, respectively; $\|U_{pop} - L_{pop}\|$ denotes Euclidean distance between U_{pop} and L_{pop}; and N_{min} is a predefined minimum population size.

When deleting individuals from the current population, only those individuals with a lower novelty are chosen. Concretely, we first sort the individuals according to their novelty values, and then delete N_{del} individuals who have a lower novelty from the current population. In order to guarantee the convergence speed, the current best individual cannot be removed from the population even if it has the lowest novelty. Note that, the values of U_{pop} and L_{pop} are updated only if the population size is decreased.

Based on the above introductions, the pseudo-code of NFDDE is presented in Algorithm 1.

Algorithm 1. NFDDE ()

```
t = 1, N = N_max, K = 3, μf_1 = μf_2 = μCR = 0.5;
Initialize a population {X_1^t, X_2^t, ··· X_N^t}, and evaluate all individuals in it;
Compute 𝒩_i^t for each individual i based on Eq. (10);
While not meet stop conditions
    Generate F_{1,i}^t, F_{2,i}^t, and CR_i^t (1 ⩽ i ⩽ N) by Eqs. (13) and (14);
    Generate U_i^t for each individual by Eqs. (11), (8) and (7);
    Select offspring X_i^{t+1} by Eq. (9);
    Adaptation of μf_m (m=1, 2) and μcr by Eqs. (15)–(18);
    t = t + 1;
    If N > N_min
      Compute 𝒩_i^t for each individual i based on Eq. (10);
      N_del = ⌈N × 0.1⌉;
      Delete some individuals with a lower novelty by Eq. (19);
    End If
End While
```

4. Experimental results and discussions

4.1. Peer algorithms and benchmark functions

To testify the performance of NFDDE, other 9 state-of-art DE variants are selected in this research. The basic information of all the competitors are presented in Table 1. Note that all operators and parameter settings in the 9 peer DE variants are the same as that given in corresponding references.

In this work, CEC2017 test suite is utilized to testify performance and characteristics of NFDDE. In the test suite, 30 benchmark functions are categorized into 4 different types, i.e., unimodal functions (f_1–f_3), simple multimoal functions (f_4–f_{10}), hybrid functions (f_{11}–f_{20}), and composition functions functions (f_{21}–f_{30}). Detailed information of the functions can refer to the literature [3]. To obtain statistical results, each algorithm is carried out 51 independent runs on each benchmark function. The allowed number of the maximum function evaluations (*MaxFEs*) for each run is set as 10 000$\times D$.

4.2. Experiment 1: Solution accuracy

In this section, a set of experiments is conducted to compare the performance of all the algorithms on CEC2017 test suite with D = 30. The comparison results listed in Tables 2–4 include the mean values (*Mean*) and standard deviations (*S.D.*) of achieved error values, while (#)*Best* denotes the number of the best mean result obtained by an algorithm. Furthermore, results of *t*-test between NFDDE and other 9 peer algorithms as well as rank values denoted as *Avg.*(*rank*) (the lower the better) of all peer algorithms are also presented in the tables. Concretely, symbols "+", "−", and "=" denote that NFDDE is significantly better than, significantly worse than, and almost the same as the corresponding competitor algorithms, respectively, measured by *t*-test results. (#)+, (#)−, and (#)= denote the number of "+", "−", and "=" in each column, respectively. Note that, in the tables, the best result of *Mean* on each function is shown in bold [1–50].

Table 1
Basic information of all the peer algorithms when D = 30.

Algorithm	Year	Characteristics
NFDDE	–	$N_{max} = 20 \times D, N_{min} = 2 \times D$
CoDE [40]	2011	Composite 3 mutation strategies and 3 settings of F and CR, N = 30
TSDE [17]	2016	Composite 5 mutation strategies and 3 settings of F and CR, N = 30
MPEDE [43]	2016	Composite 3 mutation strategies; multiple populations; $\lambda_1 = \lambda_2 = \lambda_3 = 0.2$, N = 250, ng = 20
LSHADE-snEpSin [4]	2017	$N^{init} = 18{*}D$, H = 5, $freq$ = 0.5, ps = 0.5, pc = 0.4, r^{arc} = 2.6
EFADE [22]	2018	Composite 3 mutation strategies, adaptive scheme for F and CR, N = 50
EDEV [44]	2018	$\lambda_1 = \lambda_2 = \lambda_3 = 0.1$, $\lambda_4 = 0.7$, ng = 20, N = 50
DI-DE [37]	2019	Composite 3 mutation strategies, $G_0 = 0.15{*}G_{max}, ps_1 = ps_2 = 0.35$, p = 0.1, σ = 0.8∼0.3, N = 60
DPADE [6]	2019	"DE/rand/1", c = 0.1, P_{min} = 0.1, ε = 0.001, N = 100
SHA_SNS [32]	2020	Adapted by SHADE, N = 100, $\delta = 0.1 * N$

Table 2
Comparison results of solution accuracy on 3 unimodal functions (f_1–f_3) in CEC2017 test suite (D=30).

		NFDDE	CoDE	TSDE	MPEDE	LSHADE-cnEpSin	EFADE	EDEV	DI-DE	DPADE	SHA_SNS
f_1	*Mean*	1.11E−14	1.45E−14	3.62E−15	1.95E−15	**0.00E+00**	1.50E−14	8.36E−16	2.26E−14	1.90E+02	1.87E−14
		(5)	(6=)	(4-)	(3−)	(1−)	(7+)	(2−)	(9+)	(10+)	(8+)
	S.D.	2.20E−15	1.12E−14	9.36E−15	5.70E−15	0.00E+00	8.24E−15	3.38E−15	1.03E−14	6.24E+02	1.81E−14
f_2	*Mean*	**0.00E+00**	1.02E+00	**0.00E+00**	3.51E+00	**0.00E+00**	**0.00E+00**	2.73E+00	6.05E+08	4.33E+10	4.41E+00
		(1)	(5+)	(1=)	(7+)	(1=)	(1=)	(6+)	(9+)	(10+)	(8+)
	S.D.	0.00E+00	1.19E+00	0.00E+00	1.30E+01	0.00E+00	0.00E+00	1.95E+01	4.32E+09	2.14E+11	1.53E+01
f_3	*Mean*	4.79E−14	3.15E−13	**0.00E+00**	1.61E−11	4.46E−15	2.22E−13	4.24E−11	4.05E+02	5.21E−01	1.33E−13
		(3)	(6=)	(1−)	(7=)	(2=)	(5=)	(8+)	(10+)	(9+)	(4=)
	S.D.	2.09E−14	8.99E−13	0.00E+00	8.20E−11	1.54E−14	2.63E−13	3.03E−10	1.31E+03	1.11E+00	4.35E−14
(#)	*Best*	1	0	2	0	2	1	0	0	0	0
Avg.	(*rank*)	3.00	5.67	2.00	5.67	1.33	4.33	5.33	9.33	9.66	6.67
(#)	+		1	0	1	0	1	2	3	3	2
(#)	=		2	1	1	2	2	0	0	0	0
(#)	−		0	2	1	1	0	1	0	0	1

The best result of Mean on each function is shown in bold.

Table 3
Comparison results of solution accuracy on 7 simple multimodal functions (f_4–f_{10}) in CEC2017 test suite (D=30).

		NFDDE	CoDE	TSDE	MPEDE	LSHADE-cnEpSin	EFADE	EDEV	DI-DE	DPADE	SHA_SNS
f_4	*Mean*	3.59E+01	3.11E+01	5.04E+01	5.44E+01	5.86E+01	3.74E+01	**1.02E+00**	6.53E+01	9.06E+01	2.17E+01
		(4)	(3=)	(6+)	(7+)	(8+)	(5=)	(1−)	(9+)	(10+)	(2−)
	S.D.	2.95E+01	2.96E+01	2.25E+01	1.78E+01	3.59E−14	2.91E+01	1.75E+00	1.34E+01	1.01E+01	2.87E+01
f_5	*Mean*	2.17E+01	3.92E+01	3.91E+01	2.56E+01	**1.21E+01**	4.20E+01	3.05E+01	3.24E+01	3.74E+01	3.55E+01
		(2)	(9+)	(8+)	(3+)	(1−)	(10+)	(4+)	(5+)	(7+)	(6+)
	S.D.	5.92E+00	9.19E+00	1.14E+01	7.78E+00	2.16E+00	1.32E+01	7.40E+00	1.17E+01	1.58E+01	1.21E+01
f_6	*Mean*	1.14E−13	1.67E−13	1.11E−13	6.71E−10	7.07E−10	1.25E−13	**1.09E−13**	6.39E−05	1.14E−13	3.68E−06
		(3)	(6+)	(2=)	(7=)	(8=)	(5+)	(1=)	(10+)	(3=)	(9=)
	S.D.	0.00E+00	5.73E−14	1.59E−14	4.79E−09	4.86E−09	3.41E−14	2.23E−14	9.80E−05	0.00E+00	2.62E−05
f_7	*Mean*	4.88E+01	6.74E+01	6.88E+01	5.62E+01	**4.33E+01**	9.51E+01	6.42E+01	6.69E+01	8.05E+01	5.96E+01
		(2)	(7+)	(8+)	(3+)	(1=)	(10+)	(5+)	(6+)	(9+)	(4+)
	S.D.	5.71E+00	1.02E+01	1.03E+01	6.85E+00	2.14E+00	2.54E+01	5.25E+00	1.16E+01	2.35E+01	1.09E+01
f_8	*Mean*	1.93E+01	3.94E+01	3.95E+01	2.90E+01	**1.36E+01**	4.54E+01	2.76E+01	3.34E+01	4.51E+01	3.74E+01
		(2)	(7+)	(8+)	(4+)	(1−)	(10+)	(3+)	(5+)	(9+)	(6+)
	S.D.	4.52E+00	1.20E+01	1.27E+01	7.94E+00	1.89E+00	1.36E+01	7.96E+00	9.82E+00	2.10E+01	1.48E+01
f_9	*Mean*	2.23E−15	9.80E−02	4.45E−02	1.78E−02	**0.00E+00**	2.23E−14	1.69E−01	2.10E−01	8.92E−15	1.07E−02
		(2)	(8+)	(7+)	(6=)	(1=)	(4+)	(9+)	(10+)	(3=)	(5=)
	S.D.	1.59E−14	3.50E−01	1.36E−01	8.91E−02	0.00E+00	4.56E−14	3.49E−01	2.58E−01	3.09E−14	6.46E−02
f_{10}	*Mean*	2.66E+03	1.96E+03	1.94E+03	2.69E+03	**1.36E+03**	3.57E+03	2.16E+03	2.66E+03	3.88E+03	3.11E+03
		(6)	(3−)	(2+)	(7=)	(1−)	(9+)	(4−)	(5=)	(10+)	(8+)
	S.D.	1.04E+03	5.30E+02	4.35E+02	3.44E+02	2.00E+02	7.36E+02	4.88E+02	6.31E+02	3.81E+02	7.80E+02
(#)	*Best*	0	0	0	0	5	0	2	0	0	0
Avg.	(*rank*)	**3.00**	6.14	5.86	5.29	**3.00**	7.57	3.86	7.14	7.29	5.71
(#)	+		5	6	4	1	6	4	6	5	4
(#)	=		1	1	3	3	1	1	1	2	2
(#)	−		1	0	0	3	0	2	0	0	1

The best result of Mean on each function is shown in bold.

4.2.1. Unimodal functions (f_1–f_3)

From Table 2 we can see that LSHADE-cnEpSin and TSDE yield the best performance on 2 out of the 3 unimodal functions, in terms of *Mean*, while NFDDE and EFADE achieve the best result on one test function. Furthermore, LSHADE-cnEpSin offers the most promising performance, in terms of average rank values ($Avg.(rank)$), followed by TSDE and NFDDE. Although LSHADE-cnEpSin, TSDE, NFDDE, and EFADE attain very outstanding performance on the unimodal functions, LSHADE-cnEpSin displays the most favorable and comprehensive performance. Based on the results we can say that the covariance matrix learning strategy can efficiently catch characteristics of fitness landscapes in unimodal functions, and then guide a population to search for promising regions. Furthermore, the performance of NFDDE also verifies that the novelty-hybrid-fitness driving force can offer a favorable comprehensive performance on the unimodal functions.

Table 4
Comparison results of solution accuracy on 20 complicated multimodal functions (f_{11}–f_{30}) in CEC2017 test suite (D=30).

		NFDDE	CoDE	TSDE	MPEDE	LSHADE-cnEpSin	EFADE	EDEV	DI-DE	DPADE	SHA_SNS
f_{11}	*Mean*	1.22E+01(2)	2.23E+01(8+)	1.70E+01(6=)	2.38E+01(9+)	1.35E+01(4+)	1.30E+01(3=)	1.88E+01(7+)	**9.46E+00**(1=)	1.51E+01(5=)	5.01E+01(10+)
	S.D.	9.34E+00	2.06E+01	1.66E+01	1.26E+01	2.10E+01	4.99E+00	7.54E+00	1.12E+01	1.46E+01	2.63E+01
f_{12}	*Mean*	6.06E+02(2)	8.30E+03(8+)	8.27E+03(7+)	9.22E+02(3+)	**4.50E+02**(1−)	7.33E+03(6+)	5.72E+03(5+)	1.34E+04(10+)	1.03E+04(9+)	1.31E+03(4+)
	S.D.	3.08E+02	5.87E+03	9.72E+03	3.81E+02	2.58E+02	7.21E+03	5.78E+03	1.33E+04	7.08E+03	5.69E+02
f_{13}	*Mean*	**1.65E+01**(1)	5.66E+01(10+)	2.88E+01(5+)	2.31E+01(3+)	1.72E+01(2+)	3.29E+01(6+)	4.93E+01(9+)	4.73E+01(8+)	2.65E+01(4+)	3.72E+01(7+)
	S.D.	6.85E+00	9.96E+01	1.28E+01	7.60E+00	3.91E+00	2.70E+01	8.06E+01	2.65E+01	9.93E+00	1.87E+01
f_{14}	*Mean*	1.93E+01(7)	1.32E+01(3−)	1.23E+01(2−)	1.55E+01(6=)	2.30E+01(8+)	**1.11E+01**(1−)	1.49E+01(5−)	2.39E+01(9+)	1.32E+01(4−)	6.10E+01(10+)
	S.D.	9.38E+00	8.85E+00	7.91E+00	1.23E+01	1.42E+00	4.26E+00	1.08E+01	5.96E+00	8.41E+00	3.12E+01
f_{15}	*Mean*	5.33E+00(2)	1.13E+01(7+)	8.58E+00(4+)	8.60E+00(5+)	**2.76E+00**(1−)	7.91E+00(3+)	1.17E+01(8+)	1.23E+01(9+)	1.02E+01(6+)	2.65E+01(10+)
	S.D.	2.11E+00	5.09E+00	3.81E+00	3.07E+00	1.27E+00	3.22E+00	1.02E+01	3.03E+00	3.23E+00	1.61E+01
f_{16}	*Mean*	7.39E+01(2)	4.84E+02(7+)	5.48E+02(9+)	3.09E+02(4+)	**3.78E+01**(1−)	4.47E+02(6+)	4.99E+02(8+)	2.09E+02(3+)	3.79E+02(5+)	6.21E+02(10+)
	S.D.	1.13E+02	2.26E+02	1.96E+02	1.87E+02	3.86E+01	1.76E+02	1.70E+02	1.37E+02	1.98E+02	2.27E+02
f_{17}	*Mean*	4.85E+01(3)	5.19E+01(6=)	7.18E+01(8+)	5.18E+01(5=)	**2.69E+01**(1−)	6.17E+01(7+)	4.69E+01(2=)	4.99E+01(4=)	7.86E+01(9+)	1.39E+02(10+)
	S.D.	1.63E+01	5.14E+01	7.60E+01	1.96E+01	7.20E+00	3.89E+01	2.93E+01	1.10E+01	2.75E+01	9.95E+01
f_{18}	*Mean*	2.40E+01(2)	1.38E+02(9+)	8.45E+01(7+)	2.44E+01(3=)	**2.35E+01**(1=)	9.96E+01(8+)	5.25E+01(5=)	1.41E+02(10+)	2.92E+01(4+)	5.75E+01(6+)
	S.D.	1.74E+00	1.51E+02	6.06E+01	6.08E+00	1.06E+00	1.11E+02	1.34E+02	2.35E+02	1.02E+01	3.77E+01
f_{19}	*Mean*	**4.43E+00**(1)	5.54E+00(2+)	5.92E+00(4+)	7.11E+00(6+)	5.85E+00(3+)	6.35E+00(5+)	7.71E+00(7+)	1.12E+01(9+)	8.97E+00(8+)	3.98E+01(10+)
	S.D.	1.29E+00	1.73E+00	1.83E+00	2.19E+00	2.20E+00	2.27E+00	4.99E+00	2.08E+00	1.99E+00	2.75E+01
f_{20}	*Mean*	**1.88E+01**(1)	1.14E+02(8+)	1.10E+02(7+)	7.49E+01(5+)	3.02E+01(3+)	1.16E+02(9+)	3.31E+01(4+)	2.76E+01(2+)	7.85E+01(6+)	1.76E+02(10+)
	S.D.	9.43E+00	9.15E+01	8.99E+01	5.19E+01	8.23E+00	6.08E+01	4.91E+01	1.11E+01	7.12E+01	1.29E+02
f_{21}	*Mean*	2.20E+02(2)	2.38E+02(7+)	2.40E+02(8+)	2.28E+02(3+)	**2.13E+02**(1=)	2.42E+02(10+)	2.36E+02(5+)	2.33E+02(4+)	2.42E+02(9+)	2.36E+02(6+)
	S.D.	6.08E+00	9.27E+00	8.80E+00	7.22E+00	2.11E+00	1.10E+01	7.86E+00	1.04E+01	1.84E+01	1.39E+01
f_{22}	*Mean*	**1.00E+02**(1=)	**1.00E+02**(1=)	**1.00E+02**(1=)	**1.00E+02**(1=)	**1.00E+02**(1=)	**1.00E+02**(1=)	2.79E+02(10+)	**1.00E+02**(1=)	**1.00E+02**(1=)	**1.00E+02**(1=)
	S.D.	3.44E−01	5.01E−13	1.00E+02	0.00E+00	0.00E+00	4.49E−13	6.20E+02	1.50E−12	2.30E−13	3.24E−13
f_{23}	*Mean*	3.61E+02(2)	3.85E+02(7+)	3.86E+02(8+)	3.81E+02(4+)	**3.55E+02**(1=)	3.92E+02(10+)	3.83E+02(5+)	3.79E+02(3+)	3.83E+02(6+)	3.87E+02(9+)
	S.D.	8.47E+00	1.08E+01	9.14E+00	9.83E+00	3.80E+00	1.21E+01	8.17E+00	8.39E+00	1.47E+01	1.29E+01
f_{24}	*Mean*	4.36E+02(2)	4.56E+02(7+)	4.60E+02(8+)	4.44E+02(3+)	**4.28E+02**(1=)	4.70E+02(10+)	4.53E+02(5+)	4.47E+02(4+)	4.63E+02(9+)	4.56E+02(6+)
	S.D.	6.36E+00	9.81E+00	1.10E+01	9.91E+00	2.82E+00	1.35E+01	7.43E+00	7.27E+00	1.57E+01	1.30E+01
f_{25}	*Mean*	3.87E+02(5)	3.87E+02(7=)	3.87E+02(6=)	**3.78E+02**(1−)	3.87E+02 (3=)	3.87E+02(4=)	3.83E+02(2−)	3.87E+02(10=)	3.87E+02(9=)	3.87E+02(8=)
	S.D.	7.44E−02	1.03E−01	5.13E−01	4.56E−01	9.29E−03	3.79E−02	6.63E+00	1.91E−01	1.83E−01	7.63E−01
f_{26}	*Mean*	1.04E+03(2)	1.37E+03(9+)	1.39E+03(10+)	1.18E+03(4+)	**9.64E+02**(1−)	1.07E+03(3=)	1.31E+03(6+)	1.23E+03(5+)	1.31E+03(8+)	1.31E+03(7+)
	S.D.	1.04E+02	2.17E+02	1.26E+02	1.63E+02	3.50E+01	5.37E+02	9.85E+01	9.32E+01	1.06E+02	1.64E+02
f_{27}	*Mean*	**4.94E+02**(1)	5.01E+02(7+)	4.98E+02(5+)	5.02E+02(8+)	5.06E+02(9+)	4.98E+02(4+)	5.00E+02(6+)	4.98E+02(3+)	4.97E+02(2+)	5.08E+02(10+)
	S.D.	5.03E+00	7.45E+00	7.98E+00	7.16E+00	4.01E+00	7.27E+00	1.38E−04	6.58E+00	7.38E+00	7.11E+00
f_{28}	*Mean*	**3.09E+02**(1)	3.30E+02(4+)	3.31E+02(5+)	3.37E+02(7+)	3.13E+02(2+)	3.14E+02(3=)	4.44E+02(10+)	3.42E+02(8+)	3.95E+02(9+)	3.31E+02(6+)
	S.D.	2.95E+01	5.04E+01	5.14E+01	5.66E+01	3.48E+01	3.59E+01	7.95E+01	5.17E+01	3.50E+01	5.26E+01

(*continued on next page*)

Table 4 (*continued*)

		NFDDE	CoDE	TSDE	MPEDE	LSHADE-cnEpSin	EFADE	EDEV	DI-DE	DPADE	SHA_SNS
f_{29}	*Mean*	4.13E+02(2)	4.32E+02(3+)	4.61E+02(9+)	4.57E+02(8+)	4.32E+02(4+)	4.40E+02(6+)	**4.09E+02**(1=)	4.38E+02(5+)	4.52E+02(7+)	5.23E+02(10+)
	S.D.	1.81E+01	5.28E+01	8.11E+01	3.01E+01	6.31E+00	5.17E+01	6.59E+01	3.78E+01	5.16E+01	8.91E+01
f_{30}	*Mean*	1.99E+03(3)	2.09E+03(5+)	2.09E+03(6+)	2.00E+03(4+)	1.98E+03(2=)	2.27E+03(10+)	**2.23E+02**(1−)	2.18E+03(9+)	2.13E+03(8+)	2.10E+03(7+)
	S.D.	1.75E+01	1.01E+02	9.79E+01	4.64E+01	4.14E+01	1.94E+02	2.19E+01	1.66E+02	1.08E+02	9.49E+01
(#)	*Best*	6	1	1	2	10	2	2	2	1	1
Avg.	(*rank*)	**2.35**	6.25	6.25	4.60	2.50	5.75	5.55	5.85	6.40	7.65
(#)	+		16	16	15	8	14	14	16	16	18
(#)	=		3	3	4	7	5	3	4	3	2
(#)	−		1	1	1	5	1	3	0	1	0

The best result of Mean on each function is shown in bold.

Table 5
t-test results of between NFDDE and other 9 peer algorithms on CEC2017 test suite (D = 30).

NFDDE	vs.	CoDE	TSDE	MPEDE	LSHADE-cnEpSin	EFADE	EDEV	DI-DE	DPADE	SHA_SNS
(#)	+	24	16	19	9	17	19	21	24	20
(#)	=	3	3	6	12	5	3	4	1	6
(#)	–	3	11	5	9	8	8	5	5	4

Table 6
Friedman-test results on different types of functions in CEC2017 test suite (D = 30).

	f_1–f_3		f_4–f_{10}		f_{11}–f_{20}		f_{21}–f_{30}		Overall	
	Algorithm	*R.V.*	Algorithm	*R.V.*	Algorithm	*R.V.*	Algorithm	*R.V.*	Algorithm	*R.V.*
1	LSHADE-cnEpSin	1.83	NFDDE	3.00	NFDDE	2.30	NFDDE	2.65	NFDDE	2.70
2	TSDE	2.50	LSHADE-cnEpSin	3.00	LSHADE-cnEpSin	2.50	LSHADE-cnEpSin	3.20	LSHADE-cnEpSin	2.78
3	NFDDE	3.50	EDEV	3.86	MPEDE	4.90	MPEDE	4.70	MPEDE	5.00
4	EFADE	4.83	MPEDE	5.29	EFADE	5.40	EDEV	5.30	EDEV	5.20
5	EDEV	5.33	SHA_SNS	5.71	TSDE	5.90	DI-DE	5.35	TSDE	5.90
6	CoDE	5.67	TSDE	5.86	DPADE	5.95	CoDE	6.10	EFADE	6.28
7	MPEDE	5.67	CoDE	6.14	EDEV	6.00	EFADE	6.70	CoDE	6.32
8	SHA_SNS	6.67	DI-DE	7.21	DI-DE	6.50	DPADE	6.85	DI-DE	6.57
9	DI-DE	9.33	DPADE	7.36	CoDE	6.85	TSDE	6.95	DPADE	6.95
10	DPADE	9.67	EFADE	7.57	SHA_SNS	8.70	SHA_SNS	7.20	SHA_SNS	7.30

4.2.2. Simple multimodal functions (f_4–f_{10})

From the comparison results on the simple multimodal functions presented in Table 3 we observe that LSHADE-cnEpSin attains the best result on 5 out of the 7 functions, while EDEV yields the best result on other 2 simple multimodal function, in terms of *Mean*. Although NFDDE, as well as other 7 competitors, cannot yield the best result on any test function, it attains the second best result on 4 functions. Thus, NFDDE exhibits the same favorable performance as LSHADE-cnEpSin, in terms of the values of *Avg.* (*rank*). The stable and promising property of NFDDE testifies that the hybrid driving force can obtain a trade-off between keeping population diversity and speeding up convergence on some simple multimodal functions.

4.2.3. Hybrid functions (f_{11}–f_{20}) and composition functions (f_{21}–f_{30})

Hybrid functions and composition functions are two types of complicated functions which are very difficult to be optimized. From the results on f_{11}–f_{30} listed in Table 4 we can see that NFDDE and LSHADE-cnEpSin exhibit more competitive performance than other peer algorithms. Concretely, NFDDE attains the best result, measured by (#)*Best*, on 6 out of the 20 difficult multimodal functions, while LSHADE-cnEpSin attains the best result on 10 functions. Although LSHADE-cnEpSin dominates NFDDE, in terms (#)*Best*, NFDDE outperforms LSHADE-cnEpSin in terms of the value of *Avg.*(*rank*). The comparison results indicate that the hybrid driving force is helpful for trading off the exploration and the exploitation, and then enable the population have a reliable search behavior. Together with the results presented in Tables 2 and 3, we can see that the hybrid driving force contributes more significant positive effective on the complicated multimodal functions than on the unimodal functions and the simple multimodal functions.

4.3. Experiment 2: Statistical test

Applying statistic tests to analyze experimental results is a popular method in the field of computational intelligence. In this section, two popular statistic tests, i.e., the two-tailed t-test and the Friedman-test with a significance level of α = 0.05, are used to analyze the results.

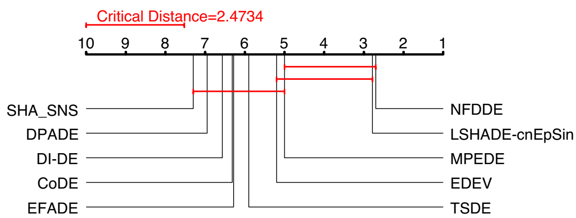


Fig. 3. Comparison results of the critical difference (D = 30). When the distance between two algorithms is less than the critical distance, there is no significant difference between the two algorithms.

Table 7
Comparison results of solution accuracy on CEC2017 test suite (D = 10).

		NFDDE	CoDE	TSDE	MPEDE	LSHADE-cnEpSin	EFADE	EDEV	DI-DE	DPADE	SHA_SNS
f_1	*Mean*	**0.00E+00**(1)	8.64E−15(8+)	**0.00E+00**(1=)	4.11E−09(9+)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)	1.63E−08(10+)	**0.00E+00**(1=)	2.79E−16(7+)
f_2	*Mean*	**0.00E+00**(1)	4.31E−01(10+)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)
f_3	*Mean*	**0.00E+00**(1)	2.67E−14(8+)	**0.00E+00**(1=)	1.49E−13(9+)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)	9.62E−08(10+)	**0.00E+00**(1=)	**0.00E+00**(1=)
f_4	*Mean*	**0.00E+00**(1)	4.35E−14(7+)	**0.00E+00**(1=)	6.03E−09(8+)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)	3.12E−03(9+)	1.84E−01(10+)	**0.00E+00**(1=)
f_5	*Mean*	4.27E+00(3)	6.03E+00(8+)	4.33E+00(4=)	5.86E+00(7+)	**1.87E+00**(1−)	6.49E+00(9+)	2.19E+00(2−)	4.99E+00(6+)	4.86E+00(5=)	8.23E+00(10+)
f_6	*Mean*	**0.00E+00**(1)	2.90E−14(7+)	**0.00E+00**(1=)	2.36E−05(10+)	**0.00E+00**(1=)	4.68E−14(9+)	**0.00E+00**(1=)	1.11E−14(6+)	**0.00E+00**(1=)	4.24E−14(8+)
f_7	*Mean*	1.45E+01(4)	1.51E+01(5=)	1.44E+01(3=)	1.74E+01(8+)	**1.18E+01**(1−)	1.91E+01(10+)	1.34E+01(2−)	1.61E+01(6+)	1.62E+01(7+)	1.85E+01(9+)
f_8	*Mean*	4.49E+00(5)	6.15E+00(7+)	4.47E+00(4=)	6.32E+00(8+)	**1.87E+00**(1−)	7.68E+00(9+)	2.11E+00(2−)	4.80E+00(6=)	4.44E+00(3=)	8.04E+00(10+)
f_9	*Mean*	**0.00E+00**(1)	3.34E−14(9+)	**0.00E+00**(1=)	6.24E−14(10+)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)	**0.00E+00**(1=)
f_{10}	*Mean*	1.45E+02(3)	2.29E+02(5+)	1.12E+02(2=)	2.86E+02(7+)	**1.21E+01**(1−)	2.76E+02(6+)	1.84E+02(4=)	2.96E+02(8+)	3.14E+02(9+)	4.22E+02(10+)
f_{11}	*Mean*	1.17E−01(2)	8.14E−01(7+)	2.15E−01(4=)	2.24E+00(10+)	**0.00E+00**(1−)	1.40E+00(8+)	2.06E−02(3=)	2.51E−01(5+)	2.54E−01(6=)	2.16E+00(9+)
f_{12}	*Mean*	**5.33E−01**(1)	2.59E+01(5+)	4.33E+00(3=)	1.10E+01(4+)	1.10E+02(9+)	1.62E+00(2=)	4.07E+01(7+)	6.98E+01(8+)	2.83E+01(6+)	3.63E+02(10+)
f_{13}	*Mean*	**1.90E+00**(1)	4.14E+00(5+)	2.15E+00(2=)	5.03E+00(8+)	4.76E+00(7+)	3.60E+00(4+)	4.21E+00(6+)	5.40E+00(10+)	2.24E+00(3=)	5.12E+00(9+)
f_{14}	*Mean*	5.46E−01(7)	9.17E−01(8+)	7.80E−02(3−)	4.87E+00(9+)	2.13E−02(2−)	3.85E−01(4=)	**6.55E−03**(1−)	4.79E−01(6=)	4.29E−01(5=)	9.67E+00(10+)
f_{15}	*Mean*	1.23E−01(3)	2.85E−01(7+)	**4.46E−02**(1−)	7.20E−01(9+)	2.46E−01(6+)	1.26E−01(4=)	5.03E−02(2−)	3.96E−01(8+)	1.33E−01(5=)	1.52E+00(10+)
f_{16}	*Mean*	4.93E−01(3)	5.19E+00(9=)	**2.27E−01**(1−)	2.34E+00(8+)	5.36E−01(4=)	8.59E−01(7+)	4.77E−01(2=)	6.96E−01(6+)	5.98E−01(5=)	2.42E+01(10+)
f_{17}	*Mean*	8.88E+00(9)	4.34E+00(6=)	4.21E−01(2−)	7.26E+00(8−)	**1.34E−01**(1−)	1.59E+00(5−)	8.25E−01(3−)	5.93E+00(7−)	9.51E−01(4−)	1.65E+01(10+)
f_{18}	*Mean*	1.81E−01(4)	1.58E−01(3=)	**4.10E−02**(1−)	2.79E+00(9+)	2.76E+00(8+)	2.45E−01(7=)	1.08E−01(2=)	2.24E−01(5=)	2.35E−01(6=)	1.29E+01(10+)
f_{19}	*Mean*	1.60E−02(5)	1.42E−02(4=)	1.22E−02(2−)	5.52E−01(9+)	1.85E−02(7=)	1.40E−02(3=)	2.73E−02(6+)	6.03E−02(8+)	**8.00E−03**(1−)	1.09E+00(10+)
f_{20}	*Mean*	1.32E−01(8)	8.19E−02(7=)	**0.00E+00**(1−)	9.46E−01(9+)	2.33E−01(8+)	**0.00E+00**(1−)	6.12E−03(4−)	6.12E−03(5−)	6.12E−03(3−)	5.59E+00(10+)
f_{21}	*Mean*	1.31E+02(3)	1.70E+02(9+)	1.59E+02(7+)	**1.23E+02**(1=)	1.54E+02(5+)	1.26E+02(2=)	1.61E+02(8+)	1.58E+02(6+)	1.36E+02(4=)	1.99E+02(10+)
f_{22}	*Mean*	9.81E+01(6)	8.90E+01(3=)	8.81E+01(2−)	9.22E+01(4=)	1.00E+02(9=)	**8.06E+01**(1−)	9.81E+01(5=)	9.81E+01(7=)	9.81E+01(8=)	1.00E+02(10=)
f_{23}	*Mean*	3.06E+02(6)	3.08E+02(8+)	3.00E+02(2=)	3.07E+02(7=)	3.02E+02(3−)	3.08E+02(9+)	3.05E+02(5=)	**2.97E+02**(1=)	3.05E+02(4−)	3.09E+02(10+)
f_{24}	*Mean*	2.68E+02(4)	3.10E+02(9+)	2.85E+02(6=)	**2.24E+02**(1−)	2.85E+02(5=)	2.39E+02(3=)	3.04E+02(8+)	3.03E+02(7=)	2.36E+02(2=)	3.30E+02(10+)
f_{25}	*Mean*	4.05E+02(3)	4.14E+02(7+)	4.06E+02(4=)	4.10E+02(6=)	4.21E+02(9+)	**4.00E+02**(1−)	4.19E+02(8+)	4.07E+02(5=)	4.03E+02(2=)	4.28E+02(10+)
f_{26}	*Mean*	3.00E+02(2)	3.00E+02(2=)	3.00E+02(2=)	3.02E+02(10=)	3.00E+02(2=)	3.00E+02(2=)	3.00E+02(2=)	3.00E+02(2=)	3.00E+02(2=)	**2.99E+02**(1=)
f_{27}	*Mean*	3.89E+02(6)	3.91E+02(9+)	3.89E+02(2−)	3.89E+02(5−)	3.90E+02(7+)	3.89E+02(3+)	**3.71E+02**(1−)	3.89E+02(4−)	3.90E+02(8+)	3.94E+02(10+)
f_{28}	*Mean*	**3.00E+02**(1)	3.28E+02(7+)	**3.00E+02**(1=)	3.17E+02(6=)	4.16E+02(8+)	**3.00E+02**(1=)	4.60E+02(9+)	3.05E+02(4=)	3.06E+02(5=)	4.77E+02(10+)
f_{29}	*Mean*	**2.30E+02**(1)	2.37E+02(5+)	2.32E+02(3+)	2.49E+02(8+)	**2.30E+02**(1=)	2.52E+02(9+)	2.40E+02(6+)	2.41E+02(7+)	2.35E+02(4+)	2.64E+02(10+)
f_{30}	*Mean*	3.97E+02(4)	1.65E+04(9=)	4.12E+02(5+)	3.96E+02(2=)	4.85E+04(9+)	3.97E+02(3=)	**2.05E+02**(1−)	4.13E+02(6+)	4.18E+02(7+)	1.86E+05(10+)
(#)	*Best*	10	0	11	3	13	9	9	3	6	5
Avg.	(*rank*)	3.27	6.90	**2.43**	7.00	4.00	4.30	3.50	6.00	4.30	8.23
(#)	+		21	3	18	11	11	8	14	7	23
(#)	=		9	18	9	11	15	13	13	19	7
(#)	−		0	9	3	8	4	9	3	4	0
	N	*D**20	30	30	250	*D**18	75	40	75	100	60

The best result of Mean on each function is shown in bold.

Table 8
Comparison results of solution accuracy on CEC2017 test suite (D = 50).

		NFDDE	CoDE	TSDE	MPEDE	LSHADE-cnEpSin	EFADE	EDEV	DI-DE	DPADE	SHA_SNS
f_1	*Mean*	5.96E−13(3)	3.11E+03(8+)	1.66E−04(6+)	2.70E−14(2=)	**1.73E−14**(1=)	8.60E−01(7+)	9.38E−12(5+)	5.40E+03(10+)	3.27E+03(9+)	1.87E−12(4+)
f_2	*Mean*	1.96E−01(2)	1.19E+33(9+)	1.69E+06(5+)	8.99E+08(6+)	8.63E−01(3=)	**0.00E+00**(1−)	3.40E+09(7+)	8.30E+29(8+)	1.29E+38(10+)	2.29E+02(4+)
f_3	*Mean*	1.32E−08(2)	2.19E+02(7+)	2.54E−08(3+)	2.49E−04(6=)	**1.56E−13**(1−)	7.69E−05(5+)	1.60E−05(4+)	6.80E+03(9+)	9.99E+03(10+)	1.08E+03(8+)
f_4	*Mean*	2.99E+01(2)	9.04E+01(8+)	6.89E+01(7+)	4.60E+01(5=)	3.91E+01(3+)	4.19E+01(4=)	**1.05E+00**(1−)	1.30E+02(10+)	1.06E+02(9+)	4.56E+01(6=)
f_5	*Mean*	3.70E+01(2)	4.64E+02(10+)	7.49E+01(6+)	5.90E+01(4+)	**2.71E+01**(1−)	9.54E+01(9+)	8.48E+01(7+)	7.83E+01(5+)	8.51E+01(8+)	5.65E+01(3+)
f_6	*Mean*	6.27E−04(6)	7.89E−03(10+)	1.99E−08(3−)	1.70E−03(8=)	2.63E−07(4−)	1.79E−13(2−)	7.19E−04(7=)	5.48E−03(9+)	**1.16E−13**(1−)	9.66E−07(5−)
f_7	*Mean*	**8.13E+01**(2)	5.60E+02(10+)	1.21E+02(6+)	1.07E+02(4+)	8.25E+01(2=)	1.80E+02(9+)	1.30E+02(7+)	1.20E+02(5+)	1.37E+02(8+)	9.59E+01(3+)
f_8	*Mean*	3.82E+01(2)	4.59E+02(10+)	7.31E+01(6+)	5.20E+01(3+)	**2.92E+01**(1−)	9.10E+01(9+)	7.39E+01(7+)	6.97E+01(5+)	8.68E+01(8+)	6.03E+01(4+)
f_9	*Mean*	1.12E+00(8)	8.33E−02(4−)	7.79E−02(3−)	8.69E−01(7=)	**9.36E−14**(1−)	5.26E−03(2−)	6.89E+00(10+)	4.39E+00(9+)	3.77E−01(6−)	1.12E−01(5−)
f_{10}	*Mean*	9.86E+03(9)	1.69E+04(10+)	4.39E+03(3−)	4.78E+03(4−)	**2.97E+03**(1−)	8.00E+03(8−)	4.19E+03(2−)	5.25E+03(5−)	6.55E+03(7−)	5.47E+03(6−)
f_{11}	*Mean*	5.59E+01(5)	6.19E+01(6+)	4.49E+01(3−)	9.69E+01(9+)	**2.70E+01**(1−)	4.65E+01(4−)	7.66E+01(8+)	3.59E+01(2−)	6.89E+01(7+)	1.10E+02(10+)
f_{12}	*Mean*	3.23E+03(2)	6.30E+04(10+)	4.55E+04(5+)	1.14E+04(4+)	**1.35E+03**(1−)	4.79E+04(6+)	6.39E+04(7+)	1.00E+05(8+)	1.30E+05(9+)	4.17E+03(3=)
f_{13}	*Mean*	**5.70E+01**(1)	6.10E+02(5+)	6.71E+02(6+)	9.14E+01(3+)	7.00E+01(2+)	2.69E+03(10+)	1.39E+03(8+)	8.50E+02(7+)	2.30E+03(9+)	1.05E+02(4+)
f_{14}	*Mean*	3.51E+01(2)	1.06E+02(8+)	4.42E+01(4+)	6.53E+01(5+)	**2.73E+01**(1−)	4.39E+01(3+)	1.77E+03(10+)	7.49E+01(6+)	8.19E+01(7+)	1.90E+02(9+)
f_{15}	*Mean*	3.59E+01(2)	1.10E+02(6+)	4.37E+01(3+)	6.86E+01(5+)	**2.45E+01**(1−)	1.19E+02(7+)	1.30E+02(8+)	5.60E+01(4+)	1.10E+03(10+)	2.09E+02(9+)
f_{16}	*Mean*	**2.77E+02**(1)	1.59E+03(10+)	1.00E+03(9+)	9.89E+02(8+)	3.53E+02(2+)	9.00E+02(5+)	9.39E+02(6+)	7.77E+02(3+)	8.42E+02(4+)	9.75E+02(7+)
f_{17}	*Mean*	**2.12E+02**(1)	1.99E+03(10+)	5.87E+02(6+)	5.98E+02(7+)	2.15E+02(2+)	6.00E+02(8+)	5.59E+02(5+)	3.60E+02(3+)	4.61E+02(4+)	7.59E+02(9+)
f_{18}	*Mean*	3.23E+01(2)	6.90E+02(7+)	3.53E+02(5+)	1.36E+02(4+)	**2.47E+01**(1−)	3.03E+03(8+)	6.81E+02(6+)	1.59E+04(10+)	6.33E+03(9+)	8.47E+01(3+)
f_{19}	*Mean*	1.90E+01(3)	8.69E+01(9+)	**1.38E+01**(1−)	4.10E+01(6+)	1.77E+01(2=)	2.18E+01(4=)	4.54E+01(7+)	2.73E+01(5+)	1.03E+03(10+)	7.23E+01(8+)
f_{20}	*Mean*	6.37E+02(10)	5.89E+02(9=)	4.17E+02(7−)	3.92E+02(4−)	**1.25E+02**(1=)	3.92E+02(5−)	3.99E+02(6−)	3.10E+02(2−)	3.63E+02(3−)	5.65E+02(8=)
f_{21}	*Mean*	2.36E+02(2)	6.60E+02(10+)	2.69E+02(6+)	2.49E+02(3+)	**2.29E+02**(1=)	2.89E+02(8+)	2.74E+02(7+)	2.69E+02(5+)	2.95E+02(9+)	2.56E+02(4+)
f_{22}	*Mean*	6.41E+03(8)	1.46E+03(10+)	4.30E+03(5−)	2.44E+03(2−)	**1.31E+03**(1−)	6.47E+03(9=)	4.25E+03(4−)	5.45E+03(7=)	3.80E+03(3−)	5.30E+03(6=)
f_{23}	*Mean*	**4.52E+02**(1)	8.90E+02(10+)	5.10E+02(7+)	4.80E+02(3+)	4.54E+02(2=)	5.20E+02(8+)	4.90E+02(6+)	4.89E+02(5+)	5.19E+02(9+)	4.85E+02(4+)
f_{24}	*Mean*	**5.24E+02**(1)	5.60E+02(6+)	5.61E+02(7+)	5.39E+02(3+)	5.26E+02(2=)	5.85E+02(9+)	5.69E+02(8+)	5.51E+02(5+)	5.92E+02(10+)	5.51E+02(4+)
f_{25}	*Mean*	5.20E+02(8)	5.09E+02(7=)	5.07E+02(6−)	**4.44E+02**(1−)	4.85E+02(4−)	4.90E+02(5−)	4.45E+02(2−)	4.81E+02(3−)	5.26E+02(9=)	5.32E+02(10+)
f_{26}	*Mean*	**1.46E+03**(1)	4.59E+03(10+)	1.85E+03(6+)	1.59E+03(3+)	1.50E+03(2=)	2.07E+03(9+)	1.99E+03(8+)	1.72E+03(4+)	1.97E+03(7+)	1.74E+03(5+)
f_{27}	*Mean*	5.29E+02(4)	5.06E+02(2−)	5.11E+02(3−)	5.48E+02(8+)	5.31E+02(6=)	5.30E+02(5=)	**4.99E+02**(1−)	5.34E+02(7=)	5.43E+02(9+)	5.46E+02(10+)
f_{28}	*Mean*	4.79E+02(6)	4.65E+02(4−)	4.62E+02(3−)	4.90E+02(7=)	4.76E+02(5=)	**4.60E+02**(1−)	4.93E+02(8+)	4.61E+02(2−)	4.97E+02(9+)	5.01E+02(10+)
f_{29}	*Mean*	**3.46E+02**(1)	2.69E+03(10+)	4.40E+02(5+)	4.42E+02(6+)	3.50E+02(2=)	4.87E+02(7+)	4.99E+02(8+)	4.01E+02(3+)	4.12E+02(4+)	6.95E+02(9+)
f_{30}	*Mean*	5.89E+05(2)	6.17E+05(6+)	5.92E+05(3=)	7.09E+05(10+)	6.46E+05(9+)	6.09E+05(5+)	**5.00E+02**(1−)	6.19E+05(7+)	6.50E+05(8+)	6.05E+05(4+)
(#)	*Best*	8	0	1	1	14	2	3	0	1	0
Avg.	(*rank*)	3.33	8.03	4.90	5.10	**2.20**	6.00	6.03	5.77	7.50	6.13
(#)	+		24	19	20	5	18	23	23	24	22
(#)	=		3	1	6	11	4	2	2	1	5
(#)	−		3	10	4	14	8	5	5	5	3
	N	D*20	60	60	250	D*18	50	100	60	100	150

The best result of Mean on each function is shown in bold.

Table 9
Comparison results of solution accuracy on CEC2017 test suite (D = 100).

		NFDDE	CoDE	TSDE	MPEDE	LSHADE-cnEpSin	EFADE	EDEV	DI-DE	DPADE	SHA_SNS
f_1	*Mean*	1.16E−07(5)	4.55E+03(8+)	7.14E+01(7+)	**1.40E−13**(1−)	1.90E−12(2−)	1.46E+01(6+)	1.94E−09(4−)	8.85E+03(10+)	5.33E+03(9+)	9.40E−11(3−)
f_2	*Mean*	4.87E+18(3)	3.99E+84(10+)	2.92E+26(4+)	4.88E+47(6+)	5.20E+11(2−)	**7.58E+01**(1−)	1.23E+59(7+)	3.14E+77(9+)	5.41E+68(8+)	1.90E+27(5+)
f_3	*Mean*	2.31E+01(5)	3.91E+04(8+)	3.32E−01(3−)	2.79E+01(6=)	7.84E−09(1−)	4.75E+02(7+)	2.59E+00(4−)	6.71E+04(9+)	7.04E+04(10+)	1.03E−03(2−)
f_4	*Mean*	1.30E+02(4)	2.51E+02(10+)	1.61E+02(5+)	08.58E+01(2−)	2.05E+02(6+)	2.06E+02(7+)	8.83E+01(3−)	2.29E+02(9+)	2.13E+02(8+)	**4.15E+01**(1−)
f_5	*Mean*	1.27E+02(2)	3.13E+02(9+)	1.67E+02(5+)	1.49E+02(4+)	**5.59E+01**(1−)	1.77E+02(6+)	2.61E+02(8+)	1.92E+02(7+)	3.64E+02(10+)	1.30E+02(3=)
f_6	*Mean*	2.48E−02(8)	1.13E−02(7−)	1.12E−07(3−)	2.07E−01(10+)	3.24E−05(4−)	**1.78E−13**(1−)	9.96E−03(6−)	9.32E−02(9+)	4.90E−12(2−)	1.81E−03(5−)
f_7	*Mean*	2.02E+02(3)	9.43E+02(10+)	2.77E+02(5+)	2.85E+02(6+)	**1.59E+02**(1−)	2.67E+02(4+)	4.09E+02(8+)	2.93E+02(7+)	5.33E+02(9+)	2.00E+02(2=)
f_8	*Mean*	1.27E+02(2)	3.68E+02(10+)	1.69E+02(5+)	1.50E+02(4+)	**5.26E+01**(1−)	1.81E+02(7+)	2.50E+02(8+)	1.73E+02(6+)	3.29E+02(9+)	1.37E+02(3+)
f_9	*Mean*	1.97E+01(4)	2.08E+02(8+)	3.54E+01(6+)	3.20E+01(5+)	**1.27E−13**(1−)	1.39E+01(3−)	1.10E+03(10+)	8.54E+01(7+)	2.25E+02(9+)	7.14E+00(2−)
f_{10}	*Mean*	2.83E+04(9)	3.46E+04(10+)	1.12E+04(4−)	1.09E+04(3−)	**9.94E+03**(1−)	1.78E+04(7−)	1.07E+04(2−)	1.55E+04(6−)	1.91E+04(8−)	1.33E+04(5−)
f_{11}	*Mean*	2.59E+02(4)	6.27E+02(8+)	1.46E+02(3−)	8.44E+02(9+)	**4.49E+01**(1−)	1.30E+02(2−)	5.20E+02(5+)	5.23E+02(6+)	5.55E+02(7+)	8.84E+02(10+)
f_{12}	*Mean*	7.79E+04(4)	1.66E+06(10+)	1.96E+05(5+)	3.10E+04(3−)	**5.25E+03**(1−)	2.25E+05(6+)	4.46E+05(7+)	6.37E+05(8+)	1.44E+06(9+)	1.82E+04(2−)
f_{13}	*Mean*	1.62E+03(4)	2.19E+03(7=)	2.18E+03(6=)	8.02E+02(3−)	**1.12E+02**(1−)	2.55E+03(9+)	3.19E+03(10+)	2.21E+03(8=)	1.94E+03(5=)	1.41E+02(2−)
f_{14}	*Mean*	2.07E+02(2)	1.81E+04(10+)	4.88E+03(7+)	5.06E+02(5+)	**5.15E+01**(1−)	8.70E+03(9+)	1.65E+03(6+)	6.83E+03(8+)	2.76E+02(3+)	4.39E+02(4+)
f_{15}	*Mean*	2.33E+02(2)	2.58E+03(9+)	1.84E+03(8+)	4.39E+02(4+)	**8.50E+01**(1−)	1.07E+03(7+)	1.02E+03(6+)	2.65E+03(10+)	4.68E+02(5+)	3.38E+02(3+)
f_{16}	*Mean*	**1.55E+03**(1)	3.14E+03(10+)	2.86E+03(7+)	2.62E+03(3+)	1.61E+03(2=)	2.86E+03(8+)	2.90E+03(9+)	2.77E+03(6+)	2.67E+03(4+)	2.73E+03(5+)
f_{17}	*Mean*	**7.80E+02**(1)	2.78E+03(10+)	1.79E+03(5+)	2.01E+03(7+)	1.12E+03(2+)	1.76E+03(4+)	1.87E+03(6+)	1.70E+03(3+)	2.03E+03(8+)	2.32E+03(9+)
f_{18}	*Mean*	2.40E+02(2)	1.57E+05(10+)	4.67E+04(8+)	7.68E+02(3+)	**8.28E+01**(1−)	4.06E+04(7+)	8.21E+03(5+)	7.38E+04(9+)	3.25E+04(6+)	1.86E+03(4+)
f_{19}	*Mean*	1.50E+02(2)	2.68E+03(10+)	1.09E+03(6+)	2.30E+02(4+)	**5.18E+01**(1−)	1.15E+03(7+)	1.16E+03(8+)	1.24E+03(9+)	6.31E+02(5+)	1.98E+02(3+)
f_{20}	*Mean*	4.62E+03(10)	2.08E+03(7−)	1.75E+03(4−)	1.84E+03(6−)	**1.33E+03**(1−)	1.69E+03(2−)	1.84E+03(5−)	1.73E+03(3−)	2.70E+03(9−)	2.45E+03(8−)
f_{21}	*Mean*	**3.31E+02**(1)	6.24E+02(10+)	3.99E+02(5+)	3.65E+02(4+)	3.36E+02(2=)	4.17E+02(7+)	4.58E+02(8+)	4.01E+02(6+)	5.65E+02(9+)	3.55E+02(3+)
f_{22}	*Mean*	2.97E+04(9)	3.55E+04(10+)	1.24E+04(4−)	1.21E+04(3−)	**1.01E+04**(1−)	1.74E+04(7−)	1.16E+04(2−)	1.62E+04(6−)	2.01E+04(8−)	1.43E+04(5−)
f_{23}	*Mean*	6.40E+02(2)	1.43E+03(10+)	6.72E+02(4+)	6.90E+02(6+)	**5.90E+02**(1−)	7.11E+02(7+)	6.81E+02(5+)	7.28E+02(8+)	8.06E+02(9+)	6.72E+02(3+)
f_{24}	*Mean*	**9.58E+02**(1)	1.11E+03(8+)	1.04E+03(5+)	1.03E+03(4+)	9.62E+02(2=)	1.05E+03(6+)	1.14E+03(10+)	1.06E+03(7+)	1.13E+03(9+)	9.99E+02(3+)
f_{25}	*Mean*	7.79E+02(8)	7.98E+02(9=)	7.45E+02(3−)	7.44E+02(2−)	7.78E+02(7=)	**7.24E+02**(1−)	7.53E+02(4−)	7.77E+02(6=)	8.00E+02(10+)	7.60E+02(5−)
f_{26}	*Mean*	4.00E+03(2)	4.93E+03(7+)	4.80E+03(6+)	4.57E+03(4+)	**3.16E+03**(1−)	4.95E+03(8+)	6.34E+03(10+)	4.59E+03(5+)	5.42E+03(9+)	4.32E+03(3+)
f_{27}	*Mean*	6.33E+02(5)	6.38E+02(6=)	6.33E+02(4=)	6.97E+02(10+)	5.94E+02(2−)	6.53E+02(7+)	**5.00E+02**(1−)	6.94E+02(9+)	6.31E+02(3=)	6.82E+02(8+)
f_{28}	*Mean*	5.58E+02(8)	6.04E+02(10+)	5.42E+02(6−)	5.28E+02(4−)	5.15E+02(2−)	5.39E+02(5−)	**5.06E+02**(1−)	5.57E+02(7=)	5.93E+02(9+)	5.26E+02(3−)
f_{29}	*Mean*	**1.21E+03**(1)	4.33E+03(10+)	2.10E+03(5+)	2.42E+03(9+)	1.22E+03(2=)	2.11E+03(6+)	1.86E+03(3+)	2.07E+03(4+)	2.19E+03(7+)	2.22E+03(8+)
f_{30}	*Mean*	**2.33E+03**(1)	4.30E+03(9+)	2.81E+03(5+)	2.62E+03(4+)	2.39E+03(2+)	3.55E+03(6+)	3.92E+03(7+)	4.21E+03(8+)	4.51E+03(10+)	2.40E+03(3+)
(#)	*Best*	6	0	0	1	17	3	2	0	0	1
Avg.	(*rank*)	**3.83**	9.00	5.10	4.80	1.80	5.67	5.93	7.17	7.53	4.17
(#)	+		25	20	20	3	21	20	24	24	16
(#)	=		3	2	1	6	0	0	3	2	2
(#)	−		2	8	9	21	9	10	3	4	12
	N	D*20	100	200	100	D*18	200	200	100	400	200

The best result of Mean on each function is shown in bold.

4.3.1. t-test results

The *t*-test results between NFDDE and the 9 competitors on each test function are also presented in Tables 2–4.

The results on the 3 unimodal functions show that NFDDE is not inferior to CoDE, EFADE, DI-DE, and DPADE on all the unimodal funcsions, in terms of the *t*-test results. On the contrary, NFDDE is dominated by LSHADE-cnEpSin and TSDE. Moreover, NFDDE yields the same performance as, or is slightly better than other 3 other competitors. On the 7 simple multimodal functions, NFDDE and LSHADE-cnEpSin achieve more outstanding performance than other algorithms. Concretely, NFDDE outperforms all the other competitors except LSHADE-cnEpSin on the majority of the simple multimodal functions. Although LSHADE-cnEpSin dominates NFDDE on 3 out of the 7 functions, both of them display the same performance on the test functions, in terms of *Avg.* (*rank*). The performance verifies that the novelty-hybrid-fitness driving force can offer a stable performance on the multimodal functions. However, for those 20 complicated problems, i.e., f_{11}–f_{30}, NFDDE exhibits more promising performance than LSHADE-cnEpSin because that NFDDE is significantly better than LSHADE-cnEpSin on 8 functions while NFDDE is significantly dominated by LSHADE-cnEpSin on 5 functions. The results manifest that the hybrid driving force introduced in NFDDE is beneficial for the complicated multimodal functions. In addition, NFDDE significantly outperforms other competitors on the majority of the functions. Concretely, NFDDE is significantly superior to the other 8 competitors on at least 15 functions except EFADE and DI-DE who are dominated by NFDDE on 14 functions.

The overall performance of *t*-test results on three different types of functions are presented in Table 5. The overall results manifest that NFDDE dominates all the peer algorithms except LSHADE-cnEpSin on the majority of the benchmark functions, while NFDDE and LSHADE-cnEpSin offer the same performance, in terms of the *t*-test results. From the above discussion we can obtain a preliminary conclusion that the novelty-hybrid-fitness driving force causes NFDDE have a very comprehensive performance, especially on complicated multimodal problems.

4.3.2. Friedman-test results

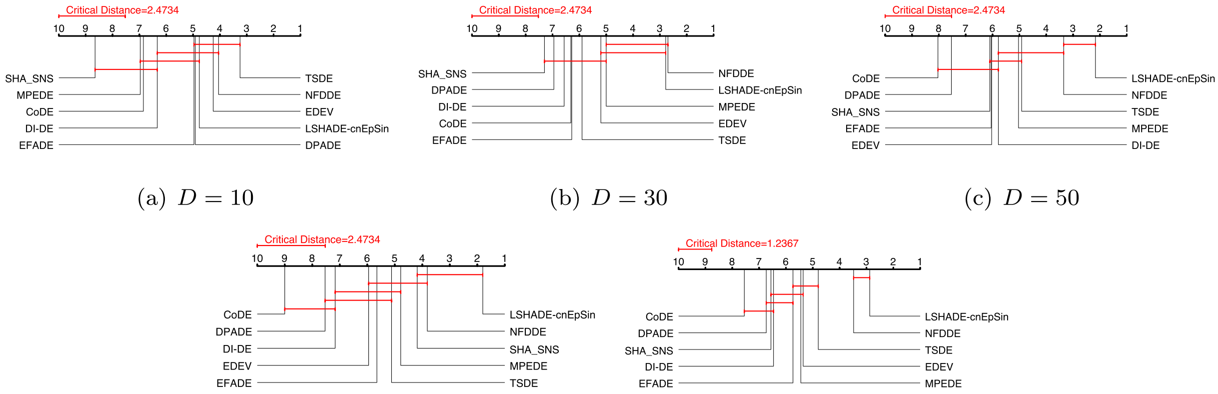
The results of the Friedman-test on CEC2017 test suite ($D = 30$) are listed in Table 6, in which each algorithm and its ranking values (*R.V.*) are listed in ascending order (the lower the better). Moreover, the Friedman-test results on different types of functions (i.e., unimodal functions (f_1–f_3), simple multimodal functions (f_4–f_{10}), hybrid functions (f_{11}–f_{20}), and composition functions (f_{21}–f_{30})) are also presented in Table 6.

From Table 6 we see that LSHADE-cnEpSin yields the most favorable performance on the unimodal functions, on which NFDDE attains the third best result. Since fitness landscapes of unimodal functions and simple multimodal functions are easy to search, thus, we can see that the positive feedback mechanism of *F* and *CR* implied in LSHADE-cnEpSin makes the population converge quickly. However, the novelty-based driving force applied in NFDDE causes individuals search for many novelty regions rather than the obtained optimal solutions. Thus, NFDDE offers relatively unfavorable performance on the simple functions. On the contrary, NFDDE exhibits the most favorable performance on the hybrid and composite multimodal functions (f_{11}–f_{30}), which are very complicated, followed by LSHADE-cnEpSin and MPEDE. Moreover, the overall performance measured by the Friedman-test result indicate that NFDDE and LSHADE-cnEpSin still offer more promising characteristics than other competitors. From the comparison results on the different types of functions, we can obtain a preliminary conclusion that the novelty-based driving force can bring more opportunities for a population to explore undetected regions, and then enhance the population's global search capability. Furthermore, we also find out that the novelty-hybrid-fitness driving force in NFDDE plays more positive performance on difficult multimodal functions than on the unimodal and simple multimodal functions.

In addition, a graphical approach is utilized to show critical difference with the Friedman rankings, the result of which is demonstrated by Fig. 3. The critical difference results testify that NFDDE is significantly better than EDEV, TSDE, EFADE, CoDE, DI-DE, DPADE, and SHA_SNS on CEC2017 test suite when $D = 30$, while there is no significant difference between NFDDE and other 2 DE variants on the test suite.

Table 10
Overall results of Friedman-test on CEC2017 test suite.

	$D = 10$		$D = 30$		$D = 50$		$D = 100$		Overall	
	Algorithm	*R.V.*	Algorithm	*R.V.*	Algorithm	*R.V.*	Algorithm	*R.V.*	Algorithm	*R.V.*
1	TSDE	3.25	NFDDE	2.70	LSHADE-cnEpSin	2.17	LSHADE-cnEpSin	1.79	LSHADE-cnEpSin	2.89
2	NFDDE	4.05	LSHADE-cnEpSin	2.78	NFDDE	3.35	NFDDE	3.84	NFDDE	3.48
3	EDEV	4.25	MPEDE	5.00	TSDE	4.92	SHA_SNS	4.16	TSDE	4.80
4	LSHADE-cnEpSin	4.77	EDEV	5.20	MPEDE	5.03	MPEDE	4.74	EDEV	5.34
5	DPADE	4.92	TSDE	5.90	DI-DE	5.78	TSDE	5.16	MPEDE	5.44
6	EFADE	4.97	EFADE	6.28	EDEV	6.03	EFADE	5.81	EFADE	5.78
7	DI-DE	6.33	CoDE	6.32	EFADE	6.05	EDEV	5.91	DI-DE	6.44
8	CoDE	6.85	DI-DE	6.57	SHA_SNS	6.10	DI-DE	7.10	SHA_SNS	6.57
9	MPEDE	6.97	DPADE	6.95	DPADE	7.53	DPADE	7.52	DPADE	6.72
10	SHA_SNS	8.65	SHA_SNS	7.30	CoDE	8.03	CoDE	8.97	CoDE	7.53



(a) $D = 10$ (b) $D = 30$ (c) $D = 50$

(d) $D = 100$ (e) All dimensions

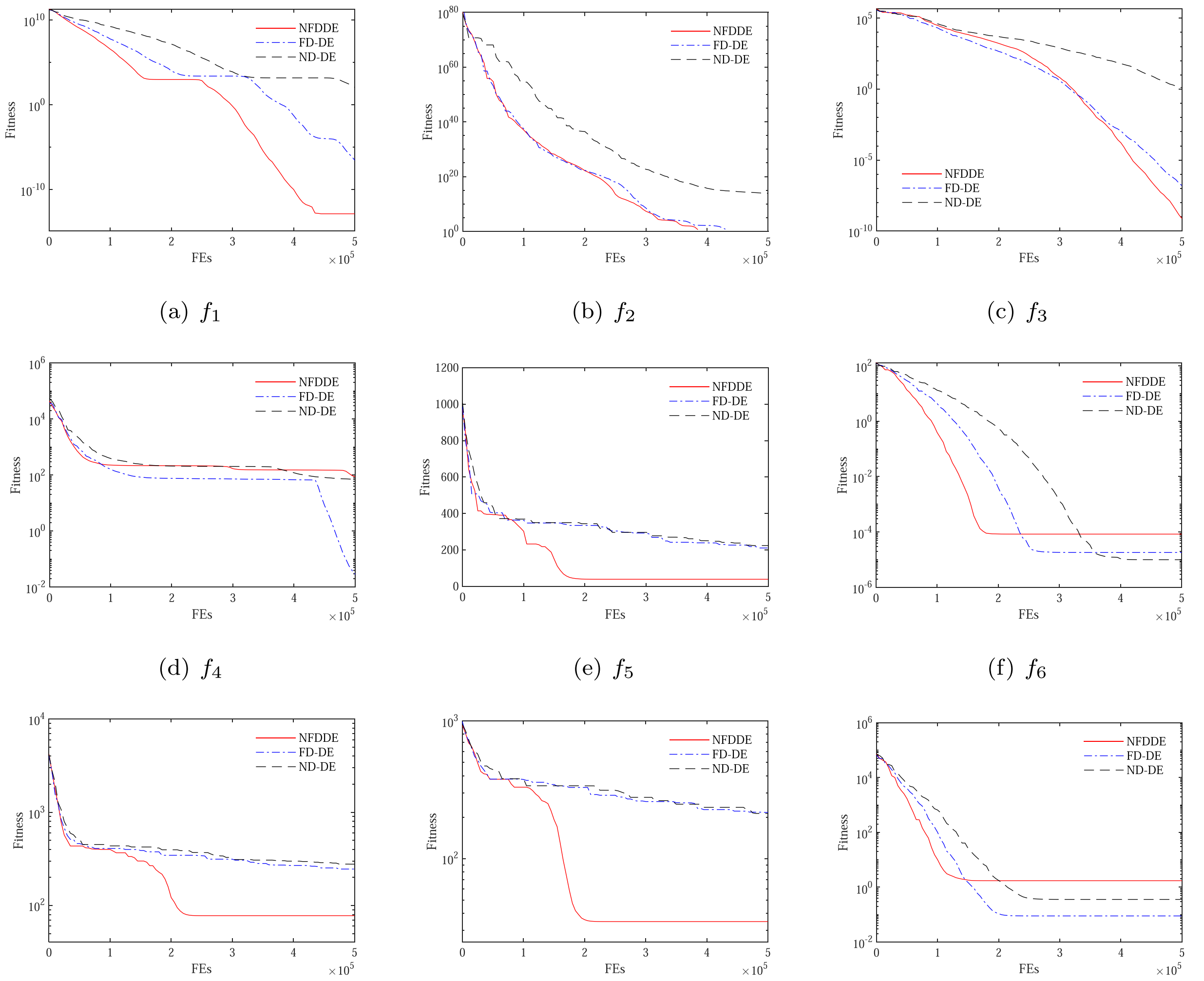
Fig. 4. Comparison results of the critical difference on 4 different dimensionality cases (D = 10, 30, 50, and 100). Moreover, the overall comparison result on the 4 cases is demonstrated in Fig. 4e.

Table 11
Comparison results of the hybrid, fitness-based, and novelty-based driving strategies.

	NFDDE	FD-DE	ND-DE
f_1	5.96E−13 ± 1.69E−13	1.47E−07 ± 3.82E−07(+)	1.79E+01 ± 7.31E+01(+)
f_2	1.96E−01 ± 3.15E−01	3.82E+02 ± 1.83E+03(=)	1.04E+17 ± 5.25E+17(+)
f_3	1.32E−08 ± 1.56E−08	3.17E−06 ± 1.74E−05(=)	2.08E+00 ± 4.72E+00(+)
f_4	2.99E+01 ± 4.12E+01	5.18E+01 ± 4.66E+01(+)	8.28E+01 ± 4.72E+01(+)
f_5	3.70E+01 ± 8.44E+00	2.00E+02 ± 1.12E+01(+)	2.24E+02 ± 1.22E+01(+)
f_6	6.27E−04 ± 2.06E−03	8.99E−04 ± 4.85E−03(=)	4.96E−04 ± 2.22E−03(=)
f_7	8.13E+01 ± 7.90E+00	2.38E+02 ± 9.74E+00(+)	2.74E+02 ± 1.35E+01(+)
f_8	3.82E+01 ± 7.39E+00	2.01E+02 ± 9.04E+00(+)	2.24E+02 ± 1.31E+01(+)
f_9	1.12E+00 ± 1.46E+00	3.62E−01 ± 5.44E−01(−)	3.40E−01 ± 3.22E−01(−)
f_{10}	9.86E+03 ± 9.46E+02	9.88E+03 ± 5.19E+02(=)	9.69E+03 ± 4.85E+02(=)
f_{11}	5.59E+01 ± 1.29E+01	6.61E+01 ± 1.68E+01(+)	8.87E+01 ± 3.08E+01(+)
f_{12}	3.23E+03 ± 1.97E+03	1.62E+03 ± 4.75E+02(−)	2.70E+03 ± 1.38E+03(=)
f_{13}	5.70E+01 ± 3.59E+01	1.91E+02 ± 2.82E+01(+)	2.04E+02 ± 3.08E+01(+)
f_{14}	3.51E+01 ± 4.70E+00	9.42E+01 ± 8.88E+00(+)	1.10E+02 ± 9.86E+00(+)
f_{15}	3.59E+01 ± 7.71E+00	8.22E+01 ± 2.52E+01(+)	1.18E+02 ± 2.16E+01(+)
f_{16}	2.77E+02 ± 1.54E+02	1.61E+03 ± 1.64E+02(+)	1.63E+03 ± 1.70E+02(+)
f_{17}	2.12E+02 ± 1.25E+02	1.18E+03 ± 1.55E+02(+)	1.25E+03 ± 1.36E+02(+)
f_{18}	3.23E+01 ± 5.97E+00	5.31E+01 ± 8.13E+00(+)	6.21E+01 ± 8.10E+00(+)
f_{19}	1.90E+01 ± 3.71E+00	5.16E+01 ± 3.87E+00(+)	5.69E+01 ± 3.99E+00(+)
f_{20}	6.37E+02 ± 4.81E+02	1.24E+03 ± 1.58E+02(+)	1.22E+03 ± 1.36E+02(+)
f_{21}	2.36E+02 ± 7.28E+00	3.99E+02 ± 9.30E+00(+)	4.18E+02 ± 1.35E+01(+)
f_{22}	6.41E+03 ± 4.78E+03	8.08E+03 ± 2.51E+03(+)	6.96E+03 ± 2.68E+03(=)
f_{23}	4.52E+02 ± 1.34E+01	6.05E+02 ± 1.67E+01(+)	6.28E+02 ± 1.49E+01(+)
f_{24}	5.24E+02 ± 9.42E+00	5.11E+02 ± 1.80E+01(−)	5.83E+02 ± 7.40E+01(+)
f_{25}	5.20E+02 ± 2.60E+01	5.16E+02 ± 2.91E+01(=)	5.11E+02 ± 3.10E+01(=)
f_{26}	1.46E+03 ± 1.34E+02	2.06E+03 ± 6.96E+02(+)	3.10E+03 ± 2.08E+02(+)
f_{27}	5.29E+02 ± 2.10E+01	5.75E+02 ± 4.15E+01(+)	5.70E+02 ± 5.23E+01(+)
f_{28}	4.80E+02 ± 2.36E+01	4.81E+02 ± 2.28E+01(=)	4.76E+02 ± 2.49E+01(=)
f_{29}	3.46E+02 ± 2.01E+01	1.07E+03 ± 1.13E+02(+)	1.18E+03 ± 1.50E+02(+)
f_{30}	5.89E+05 ± 1.44E+04	5.90E+05 ± 1.57E+04(=)	8.14E+05 ± 5.26E+04(+)
(#)+		20	23
(#)=		7	6
(#)−		3	1

4.4. Experiment 3: Performance of scalability

In this section, three additional dimension cases (i.e., D = 10, 50, and 100) of CEC2017 test suite are used to testify the scalability of the 10 peer algorithms. The comparison results of the 3 dimension cases are presented in Table 7, Table 8, and Table 9, respectively. Note that, in this set of experiments, parameters except the population size of each algorithm are the same as that in the above experiment. It is a popular strategy that applying different population sizes for different



(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

(g) f_7 (h) f_8 (i) f_9

Fig. 5. Convergence process under the different driving forces. NFDDE is the proposed algorithm in this study. The mutation operators in FD-DE and ND-DE are defined as Eqs. (20) and (21), respectively, while other strategies in them are the same as that in NFDDE.

scale problems. Thus, in the set of experiments, different population sizes of all the competitors tuned by us based on a few experiments are presented in the last row of the 3 tables.

The results demonstrated in Table 7 indicate that LSHADE-cnEpSin and TSDE offer the most competitive and comprehensive performance on the low dimension functions, in terms of (#)*Best*. Although LSHADE-cnEpSin dominates NFDDE on the index, NFDDE outperforms LSHADE-cnEpSin in terms of the *t*-test result. Moreover, from the perspective of *Avg.* (*rank*) we can observe that TSDE and NFDDE offer more reliable performance than other competitors. Although NFDDE attains the optimal solutions on some simple functions (i.e., f_1–f_4, f_6, and f_9), it shows very unpromising performance on the 20 complicated functions (f_{11}–f_{30}), on which TSDE offers the best performance, followed by EDEV.

When the dimensionality of the functions being expanded to 50, LSHADE-cnEpSin and NFDDE yield the best result on 14 and 8 functions, respectively. Although NFDDE does not display favorable performance on the unimodal functions (f_1–f_3) and simple multimodal functions (f_4–f_{10}), it performs very promising characteristics on the complicated multimodal functions (f_{11}–f_{30}). In contrast with NFDDE, TSDE and EFADE do not exhibit a promising performance on any complicated functions though they yield the best result on 5 and 4 complicated functions, respectively, when $D = 10$.

From the results presented in Table 9, it can be observed that LSHADE-cnEpSin and NFDDE still demonstrate more robust and reliable performance on the test functions than other 8 competitors when $D = 100$. Although NFDDE cannot attain the best result on these functions, on which LSHADE-cnEpSin offers the best result on 17 out of the 30 functions, NFDDE obtains the best solution on 6 complicated multimodal functions (f_{11}–f_{30}), on which LSHADE-cnEpSin yields the best result on 12 functions. In the contrast with the favorable performance of NFDDE and LSHADE-cnEpSin, other competitors do not display

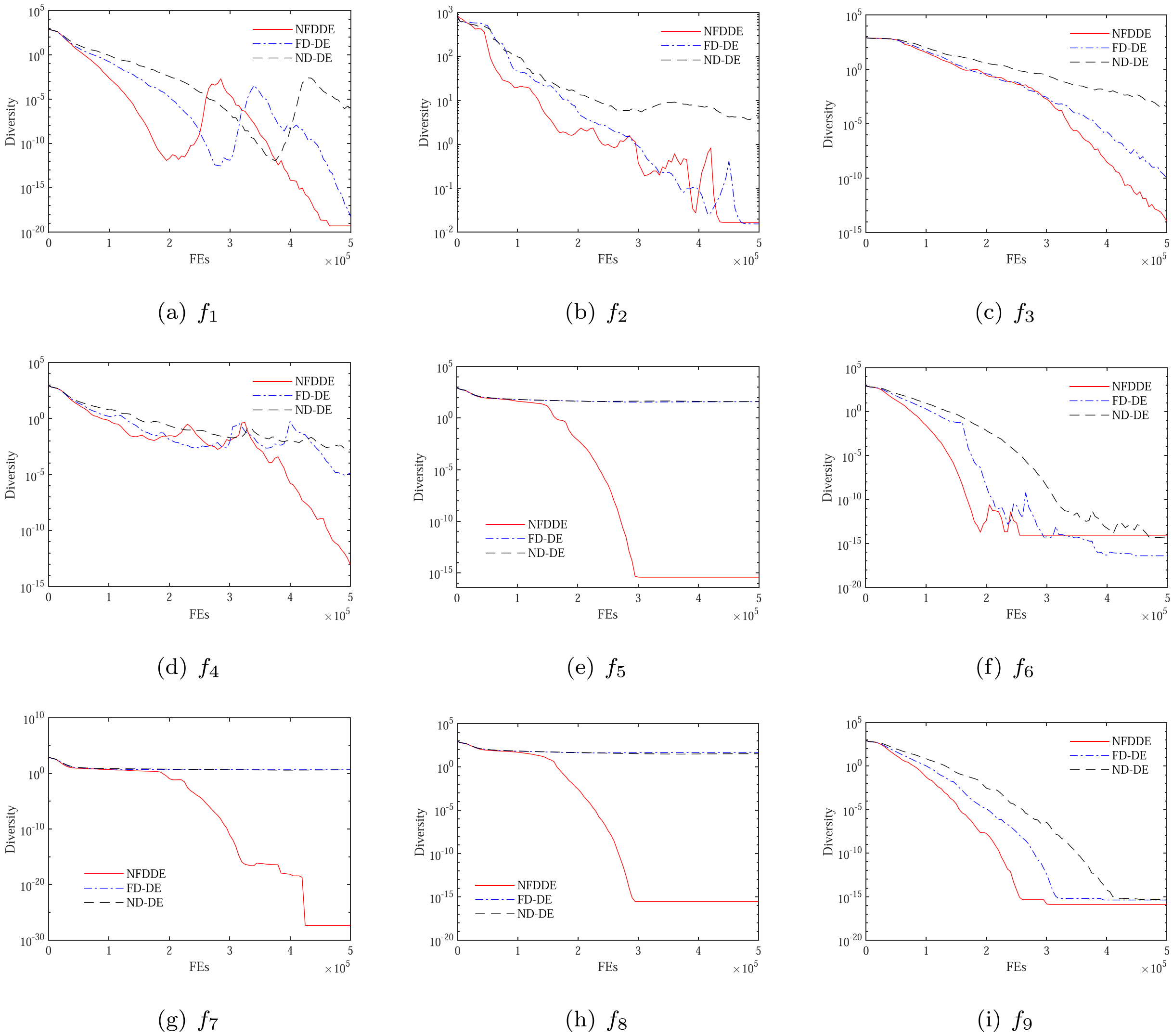


Fig. 6. Fluctuation of population diversity under the different driving forces. NFDDE is the proposed algorithm in this study. The mutation operators in FD-DE and ND-DE are defined as Eqs. (20) and (21), respectively, while other strategies in them are the same as that in NFDDE.

promising performance on the complicated functions. In addition, although NFDDE is dominated by LSHADE-cnEpSin, in terms of (#)*Best* and the *t*-test result, NFDDE exhibits the most promising property measured by *Avg*. (*rank*). The results validate that the hybrid driving force applied in NFDDE has a reliable performance on different functions.

To describe the scalability of all the peer algorithms more clearly, 5 Friedman-tests for the 4 different dimension cases (i.e., D = 10, 30, 50, and 100) and an overall performance are conducted in this section, the results of which are presented in Table 10. In addition, the critical difference of the 5 Friedman-tests are also executed, the details of which are demonstrated by Fig. 4.

The results presented in Table 10 show that TSDE displays the best performance on the lowest dimension cases, i.e., D = 10. However, it does not yield outstanding characteristics on the higher dimension cases. On the contrary, FDDE as well as LSHADE-cnEpSin exhibits very salient and promising performance when the dimensionality of the problems increases. The results verify that the novelty-hybrid-fitness driving force applied in NFDDE and the covariance metrix learning strategy introduced in LSHADE-cnEpSin can relieve the "dimensionality curse" problem to a certain extent.

The critical difference results demonstrated in Fig. 4 show that NFDDE significantly outperforms other 4, 7, 5, and 3 competitors when dimensions are 10, 30, 50, and 100, respectively. On the contrary, NFDDE does not be significantly dominated by any peer algorithm though LSHADE-cnEpSin shows more favorable performance. When considering all the dimensionality cases, we observe that NFDDE and LSHADE-cnEpSin significantly outperforms all the other 8 peer algorithms, and there is no significantly difference between NFDDE and LSHADE-cnEpSin.

These results introduced above verify that NFDDE attains favorable and competitive performance on the benchmark functions with a higher dimensionality. Thus, we can draw a preliminary conclusion that the new introduced strategies in it are more beneficial for complicated multimodal functions, and have a promising scalability on the test suite.

4.5. Experiment 4: Performance of proposed strategies

In NFDDE, there are a few novel strategies are introduced. To understand the characteristics of the new proposed strategies, the performance of some introduced strategies are verified by CEC2017 with 50 variables (D = 50) in this section.

4.5.1. Performance of the two driving forces

In this study, a hybrid driving forces based on the fitness-based and the novelty-based driving forces are proposed. Thus, the performance of the new proposed driving force is testified by a set of experiments. The results measured by solution accuracy are presented in Table 11, while the convergence processes of the different driving forces are demonstrated by Fig. 5. Due to the space limitation, only the convergence processes of f_1–f_9 are presented in this paper. In the set of experiments, note that, mutation operators in FD-DE and ND-DE are defined as Eqs. (20) and (21), respectively, while other strategies in them are the same as that in NFDDE. In Eq. (20), the main driving force is caused by $\mathbf{X}^t_{pBest}$ while the main driving force in Eq. (21) is generated by $\mathbf{X}^t_{pNovelty}$. Thus, we can say that the driving forces implied in Eqs. (20) and (21) can be regarded as a fitness-based driving force and a novelty-based driving force, respectively.

$$\mathbf{V}^t_i = \mathbf{X}^t_i + F^t_{1,i} \cdot \left(\mathbf{X}^t_{pBest} - \mathbf{X}^t_i + \mathbf{X}^t_{r_1} - \mathbf{X}^t_{r_2}\right) \tag{20}$$

$$\mathbf{V}^t_i = \mathbf{X}^t_i + F^t_{2,i} \cdot \left(\mathbf{X}^t_{pNovelty} - \mathbf{X}^t_i + \mathbf{X}^t_{r_1} - \mathbf{X}^t_{r_2}\right) \tag{21}$$

The comparison results presented in Table 11 show that the performance difference between NFDDE and FD-DE is not significant than that between NFDDE and ND-DE on the 3 unimodal functions (i.e., f_1–f_3), in terms of the solution accuracy. The results of convergence process on f_1–f_3 demonstrate that NFDDE and FD-DE achieve almost the same convergence speed

Table 12
Comparisons of the performance of F_1 and F_2.

	NFDDE	NFDDE/F_1	NFDDE/F_2
f_1	1.20E−08 ± 1.21E−08	6.35E−14 ± 2.02E−14(−)	5.88E−14 ± 2.29E−14(−)
f_2	1.30E+01 ± 2.31E+01	1.37E+00 ± 1.92E+00(+)	9.80E−01 ± 1.53E+00(+)
f_3	1.77E−05 ± 1.99E−05	1.63E−10 ± 1.87E−10(−)	2.65E−10 ± 3.05E−10(−)
f_4	2.99E+01 ± 4.12E+01	2.17E+01 ± 2.95E+01(=)	3.43E+01 ± 4.43E+01(=)
f_5	3.70E+01 ± 8.44E+00	1.99E+02 ± 1.03E+01(+)	1.98E+02 ± 1.34E+01(+)
f_6	6.27E−04 ± 2.06E−03	1.41E−03 ± 4.45E−03(=)	7.81E−04 ± 3.18E−03(=)
f_7	8.13E+01 ± 7.90E+00	2.35E+02 ± 1.15E+01(+)	2.33E+02 ± 9.75E+00(+)
f_8	3.82E+01 ± 7.39E+00	1.97E+02 ± 1.04E+01(+)	1.95E+02 ± 1.10E+01(+)
f_9	1.12E+00 ± 1.46E+00	1.72E−01 ± 2.61E−01(−)	2.23E−01 ± 2.03E−01(−)
f_{10}	9.86E+03 ± 9.46E+02	1.01E+04 ± 6.98E+02(=)	9.74E+03 ± 7.24E+02(=)
f_{11}	5.59E+01 ± 1.29E+01	5.67E+01 ± 1.26E+01(=)	5.66E+01 ± 1.40E+01(=)
f_{12}	3.23E+03 ± 1.97E+03	1.82E+03 ± 4.80E+02(−)	1.82E+03 ± 4.41E+02(−)
f_{13}	5.70E+01 ± 3.59E+01	1.96E+02 ± 3.48E+01(+)	1.89E+02 ± 2.24E+01(+)
f_{14}	3.51E+01 ± 4.70E+00	9.57E+01 ± 7.29E+00(+)	9.44E+01 ± 7.58E+00(+)
f_{15}	3.59E+01 ± 7.71E+00	8.42E+01 ± 2.27E+01(+)	7.65E+01 ± 1.97E+01(+)
f_{16}	2.77E+02 ± 1.54E+02	1.57E+03 ± 1.65E+02(+)	1.54E+03 ± 2.00E+02(+)
f_{17}	2.12E+02 ± 1.25E+02	1.16E+03 ± 1.38E+02(+)	1.19E+03 ± 1.48E+02(+)
f_{18}	3.23E+01 ± 5.97E+00	5.31E+01 ± 6.60E+00(+)	5.63E+01 ± 6.70E+00(+)
f_{19}	1.90E+01 ± 3.71E+00	5.07E+01 ± 4.35E+00(+)	5.09E+01 ± 4.69E+00(+)
f_{20}	6.37E+02 ± 4.81E+02	1.22E+03 ± 1.56E+02(+)	1.21E+03 ± 1.42E+02(+)
f_{21}	2.36E+02 ± 7.28E+00	3.94E+02 ± 8.96E+00(+)	3.99E+02 ± 1.29E+01(+)
f_{22}	6.41E+03 ± 4.78E+03	8.25E+03 ± 2.95E+03(+)	8.82E+03 ± 2.79E+03(+)
f_{23}	4.52E+02 ± 1.34E+01	6.06E+02 ± 1.07E+01(+)	5.93E+02 ± 3.94E+01(+)
f_{24}	5.24E+02 ± 9.42E+00	5.12E+02 ± 2.70E+01(−)	5.08E+02 ± 8.39E+00(−)
f_{25}	5.20E+02 ± 2.60E+01	5.11E+02 ± 3.19E+01(=)	5.09E+02 ± 3.06E+01(=)
f_{26}	1.46E+03 ± 1.34E+02	2.02E+03 ± 6.76E+02(+)	1.90E+03 ± 6.83E+02(+)
f_{27}	5.29E+02 ± 2.10E+01	5.61E+02 ± 4.85E+01(+)	5.71E+02 ± 4.50E+01(+)
f_{28}	4.80E+02 ± 2.36E+01	4.73E+02 ± 2.16E+01(=)	4.71E+02 ± 2.04E+01(=)
f_{29}	3.46E+02 ± 2.01E+01	1.08E+03 ± 1.08E+02(+)	1.07E+03 ± 1.12E+02(+)
f_{30}	5.89E+05 ± 1.44E+04	5.91E+05 ± 1.93E+04(=)	5.90E+05 ± 1.74E+04(=)
(#)+		18	18
(#)=		7	7
(#)−		5	5

on f_2 and f_3, while NFDDE offers a higher convergence speed than FD-DE. Furthermore, ND-DE not only offers the worst performance measured by the solution accuracy, but also displays the slowest convergence speed. From the results we can obtain a conclusion that the fitness-based driving force is favorable for unimodal functions. In the contrast, the novelty-based driving force without cooperation of the fitness-based driving force cannot display a satisfactory performance on unimodal functions.

The experimental results on f_4–f_{30} measured by solution accuracy indicate that NFDDE dominates FD-DE and ND-DE on majority of the multimodal functions. FD-DE and ND-DE only overwhelming NFDDE on 3 and 1 multimodal functions, respectively. Moreover, ND-DE offers more unfavorable performance than FD-DE. The convergence curves in Fig. 5(d)–(i) demonstrate that NFDDE yields the highest convergence speed on 3 out of the 6 multimodal functions while NFDDE and FD-DE display a higher convergence speed on all the 6 multimodal functions in the early evolutionary stage. On the contrary, ND-DE offers unfavorable performance, in terms of the convergence speed, on 4 multimoal functions.

Furthermore, we also use an experiment to investigate how the three driving forces affect the population diversity, the results of which are demonstrated in Fig. 6. Similar to the last experiment, only the results on f_1–f_9 are considered in the experiment due to the space limitation. In this study, the population diversity is defined as Eq. (22).

$$diversity(ps) = \frac{1}{N \cdot |L|} \cdot \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{D} (x_{ij} - \bar{x}_j)^2} \tag{22}$$

where N is the population size, $|L|$ is the length of the longest diagonal in the searching space, D is the dimensionality of the problem, x_{ij} is the j^{th} value of the i^{th} particle, and $\bar{x}_j$ is the j^{th} value of the average point $\bar{x}$.

From the results demonstrated in Fig. 6(a)–(c) we see that ND-DE yields the most outstanding performance on f_2 and f_3 measured by the population diversity, while the population diversities in NFDDE and FD-DE drop quickly on the two simple unimodal functions. There is an interesting phenomenon in Fig. 6a that a large fluctuation appears in both the 3 curves. The reason of it is that many individuals with a lower novelty, which means that the individuals are located in dense areas, are periodically deleted from current population. After that, thus, the population can display a higher population diversity. Furthermore, it can be observed that the time of the diversity fluctuation of ND-DE is the latest. In other words, the time that the deleting individual process conducted in ND-DE is the latest. From these results we attain that ND-DE has the most favorable performance, in terms of the population diversity, on the 3 simple unimodal functions. The results demonstrated in Fig. 6(d)–(f) manifest that ND-DE has the highest population diversity on the 6 multimodal functions though it cannot yield very promising fitness values on them, while NFDDE yields the highest convergence speed especially in the later evolution stage. The results indicate that the novelty-based driving force is beneficial for the population diversity while it sacrifices the performance on solution accuracy. On the contrary, hybrid with the fitness-based driving force and the novelty-based driving force can obtain a trade-off between the population diversity and the solution accuracy.

4.5.2. Performance of F_1 and F_2

In NFDDE, two driving forces are utilized to generate a mutant vector. Thus, to adjust the weights of the two driving forces, two scaling factors are introduced in NFDDE (see Eq. (11)). In this section, to verify the performance of the two factors, a set of experiments is conducted, the results of which are presented in Table 12. Note that, NFDDE/F_1 and NFDDE/F_1 denote two NFDDE variants in which only F_1 and F_2 are used to generate mutant vectors, respectively.

From Table 12 we observe that NFDDE/F_1 and NFDDE/F_2 display more outstanding performance on the 3 unimodal functions since both of them dominate NFDDE on 2 out of the 3 simple functions. For the 7 simple multimodal functions, NFDDE fares marginally better performance because that NFDDE outperforms NFDDE/F_1 and NFDDE/F_2 on 3 functions while NFDDE is surpassed by the two competitors on 1 function. However, for the 20 complicated multimodal functions, i.e. f_{11}–f_{30}, NFDDE exhibits distinct advantages because it significantly dominates NFDDE/F_1 and NFDDE/F_2 on the majority of the complicated multimodal functions. The t-test results presented at the last three rows of the table show that NFDDE achieves the significantly favorable performance. The results validate the hypothesis that using the two adaptive scaling factors allows more efficient utilization of both the fitness-based force and the novelty-based drive force than only one scaling factors.

5. Conclusions and future work

In the majority study of DEs for the single objective optimization, applying fitness as a selection pressure is a widely accepted choice. Although the fitness-driving force plays a positive role in speeding up convergence process, it sacrifices the exploration capability when optimizing complicated multimodal functions. On the contrary, the novelty-based search entirely ignoring the objective is immune to the complicated functions of deception and local optima.

In this paper, a novelty-hybrid-fitness driving strategy is introduce to make up the deficiency caused by the fitness-based force in DEs. In the new proposed DE algorithm, named as NFDDE, both fitness and novelty values are considered when selecting individuals to create mutant vectors. Moreover, two scaling factors adaptively tuned are used to adjust the weights of two driving forces, i.e., the fitness-based driving force and the novelty-based driving force, during the evolution process. At last, to rationally allocate computational resources, some individuals with a lower novelty are deleted from the current population when the population has converged to a certain extent.

To testify the comprehensive performance of NFDDE, a set of experiments were conducted. Furthermore, the effectiveness of the new introduced strategies was also analyzed by extensive experiments. From the experimental results we obtain some preliminary conclusions. Firstly, with an organic integration strategy, advantages of the two driving forces can be effectively utilized, and then NFDDE can trade off contradictions between the exploration and the exploitation. Secondly, the hybrid driving force contributes to complicated multimodal problems. Moreover, the proposed strategy has an outstanding scalability. In other words, it manifests more competitive performance on large scale functions. In addition, with two adaptive scaling factors, the promising characteristics of the hybrid driving force can be utilized more effectively. Hence, NFDDE yields more reliable and comprehensive performance, especially on complicated multimodal functions.

Considering "No Free Lunch theorem", it is dangerous to suggest that the novelty-based driving force is better than the traditional fitness-based driving force in general. Although introducing the novelty-based driving force casts the performance of DE in a new perspective, some issues need further study. For instance, how to design a helpful "novelty" index is crucial for a specific problem. In this study, an individual's novelty is measured by the average distance between the individual and its the *K*-nearest neighbors. However, it is not to say that the measurement is the optimal choice since different environments have their own distinct characteristics. Hence, it is more realistic that designing an appropriate measurement for novelty according to a specific problem. Moreover, it is another unavoidable issue that how to organically integrate the novelty-based driving force with the traditional fitness-based driving force, and then take full advantages of the both driving forces.

CRediT authorship contribution statement

Xuewen Xia: Conceptualization, Methodology, Writing - review & editing. **Lei Tong:** Data curation. **Yinglong Zhang:** Conceptualization, Writing - original draft. **Xing Xu:** Investigation, Software. **Honghe Yang:** Software. **Ling Gui:** Investigation. **Yuanxiang Li:** Supervision. **Kangshun Li:** Supervision.

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.

Acknowledgment

This study was funded by the National Natural Science Foundation of China (Grant Nos.: 61663009, 61762036, 61702239), the Science and Technology Plan Projects of Zhangzhou (Grant Nos: ZZ2020J06, ZZ2020J24), the National Natural Science Foundation of Education Department of Jiangxi Province (Grant No.: GJJ200629), and the Natural Science Foundation of Jiangxi Province (No. : 20192ACB21004).

References

[1] R.D. Al-Dabbagh, F. Neri, N. Idris, et al, Algorithmic design issues in adaptive differential evolution schemes: review and taxonomy, Swarm Evol. Comput. 43 (2018) 284–311.
[2] M.Z. Ali, N.H. Awad, P.N. Suganthan, et al, An adaptive multipopulation differential evolution with dynamic population reduction, IEEE Trans. Cybern. 47 (9) (2017) 2768–2779.
[3] N.H. Awad, M.Z. Ali, J.J. Liang, et al, Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization, Nanyang Technological Univ., Singapore, Tech. Rep. (2016).
[4] N.H. Awad, M.Z. Ali, P.N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems, in: Proceedings of IEEE Congress on Evolutionary Computation, CEC'17, Donostia, Spain, 2017, pp. 372–379.
[5] A. Channon, Passing the alife test: activity statistics classify evolution in geb as unbounded, in: Proceedings of the European conference on artificial life, ECAL'01, Springer, pp. 417–426.
[6] L.Z. Cui, G.H. Li, Z.X. Zhu, et al, Differential evolution algorithm with dichotomy-based parameter space compression, Soft Comput. 23 (2019) 3643–3660.
[7] W. Du, M. Zhang, Y. Wen, et al, The networked evolutionary algorithm: a network science perspective, Appl. Math. Comput. 338 (2018) 33–43.
[8] I. Fister, P.N. Suganthan, I. Fister, et al, Artificial neural network regression as a local search heuristic for ensemble strategies in differential evolution, Nonlinear Dyn. 84 (2) (2016) 895–914.
[9] A. Ghosh, S. Das, R. Mallipeddi, et al, A modified differential evolution with distance-based selection for continuous optimization in presence of noise, IEEE Access (2017) 26944–26964.
[10] J. Gomes, P. Urbano, A.L. Christensen, Evolution of swarm robotics systems with novelty search, Swarm Intell. 7 (2013) 115–144.
[11] B.X. Guan, Y.H. Zhao, Y. Yin, et al, A differential evolution based feature combination selection algorithm for high-dimensional data, Inf. Sci. 547 (2020) 870–886.
[12] A.A. Heidari, S. Mirjalili, H. Faris, et al, Harris hawks optimization: algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[13] B. Inden, Y.C. Jin, R. Haschke, et al, An examination of different fitness and novelty based selection methods for the evolution of neural networks, Soft Comput. 17 (2013) 753–767.
[14] J. Lehman, K.O. Stanley, Efficiently evolving programs through the search for novelty, in: Proceedings of Genetic and evolutionary computation conference, GECCO'10. Portland, Oreg, pp. 837–844..
[15] J. Lehman, K.O. Stanley, Abandoning objectives: evolution through the search for novelty alone, Evol. Comput. 19 (2) (2011) 189–223.
[16] J. Liu, J. Lampinen, A fuzzy adaptive differential evolution algorithm, Soft Comput. 9 (6) (2005) 448–462.
[17] Z.Z. Liu, Y. Wang, S.X. Yang et al., Differential evolution with a two-stage optimization mechanism for numerical optimization, in: Proceedings of IEEE Congress on Evolutionary Computation, CEC'16, Vancouver, BC, Canada, 2016, pp. 3170–3177.

[18] X. Liu, Z.H. Zhan, Y. Lin, et al, Historical and heuristic-based adaptive differential evolution, IEEE Trans. Sys. Man Cybern. Syst. 49 (12) (2019) 2623–2635.
[19] Q.X. Liu, S.Z. Du, B.J.V. Wyk, et al, Double-layer-clustering differential evolution multimodal optimization by speciation and self-adaptive strategies, Inf. Sci. 545 (2021) 465–486.
[20] N. Lynn, M.Z. Ali, P.N. Suganthan, Population topologies for particle swarm optimization and differential evolution, Swarm Evol. Comput. 39 (2018) 24–35.
[21] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[22] A.W. Mohamed, P.N. Suganthan, Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation, Soft Comput. 22 (10) (2018) 3215–3235.
[23] E.M. Montes, J. Velázquez-Reyes, C.A.C. Coello, A comparative study of differential evolution variants for global optimization, in: Proceedings of Genetic and Evolutionary Computation, GECCO'06, Seattle, Washington, USA, 2006, pp. 485–492.
[24] B. Morales-Castaneda, D. Zaldivar, E. Cuevas, et al, A better balance in metaheuristic algorithms: does it exist?, Swarm Evol Comput. 54 (2020) 100671.
[25] V.K. Patel, V.J. Savsani, Heat transfer search (HTS): a novel optimization algorithm, Inf. Sci. 324 (2015) 217–246.
[26] O. Patricia, C. Oscar, S. Jose, Optimization of fuzzy controller design using a differential evolution algorithm with dynamic parameter adaptation based on Type-1 and Interval Type-2 fuzzy systems, Soft Comput. 24 (1) (2020) 193–214.
[27] K.V. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer-Verlag, Berlin Heidelberg, 2005.
[28] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.
[29] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning based optimization: a novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (3) (2011) 303–315.
[30] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[31] N.R. Sabar, J. Abawajy, J. Yearwood, Heterogeneous cooperative co-evolution memetic differential evolution algorithm for big data optimization problems, IEEE Trans. Evol. Comput. 21 (2) (2017) 315–327.
[32] E. Segredo, E. Lalla-Ruiz, E. Hart, et al, A similarity-based neighbourhood search for enhancing the balance exploration-exploitation of differential evolution, Comput. Oper. Res. 117 (2020) 104871.
[33] R. Standish, Open-ended artificial evolution, Int. J. Comput. Intell. Appl. 3 (2) (2003) 167–175.
[34] R. Storn, K. Price, Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.
[35] Z.P. Tan, K.S Li, Y. Wang, Differential evolution with adaptive mutation strategy based on fitness landscape analysis, Inf. Sci. 549 (2021) 142–163..
[36] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, in: Proceedings of IEEE Congress on Evolutionary Computation, CEC'13, pp. 71–78.
[37] L. Tian, Z.C. Li, X.F. Yan, Differential evolution algorithm directed by individual difference information between generations and current individual information, Appl. Intell. 49 (2019) 628–649.
[38] M.N. Tian, X.B. Gao, Differential evolution with neighborhood-based adaptive evolution mechanism for numerical optimization, Inf. Sci. 478 (2019) 422–448.
[39] A. Viktorin, R. Senkerik, M. Pluhacek, et al, Distance based parameter adaptation for success-history based differential evolution, Swarm Evol. Comput. 50 (2019) 100462.
[40] Y. Wang, Z.X. Cai, Q.F. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (1) (2011) 55–67.
[41] S.L. Wang, F. Morsidi, T.F. Ng, et al, Insights into the effects of control parameters and mutation strategy on self-adaptive ensemble-based differential evolution, Inf. Sci. 514 (2020) 203–233.
[42] K. Wansaseub, N. Pholdee, S. Bureerat, Optimal U-shaped baffle square-duct heat exchanger through surrogate-assisted self-adaptive differential evolution with neighbourhood search and weighted exploitation-exploration, Appl. Therm. Eng. 118 (2017) 455–463.
[43] G.H. Wu, R. Mallipeddi, P.N. Suganthan et al., Differential evolution with multi population based ensemble of mutation strategies, Inf. Sci. 329 (C) (2016) 329–345.
[44] G.H. Wu, X. Shen, H.F. Li, et al, Ensemble of differential evolution variants, Inf. Sci. 423 (2018) 172–186.
[45] X.W. Xia, L. Gui, Y.L. Zhang, et al, A fitness-based adaptive differential evolution algorithm, Inf. Sci. 549 (2021) 116–141.
[46] M. Yang, C. Li, Z. Cai, J. Guan, Differential evolution with auto-enhanced population diversity, IEEE Trans. Cybern. 45 (2) (2014) 302–315.
[47] X.B. Yu, C.L. Li, J.F. Zhou, A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios, Knowl. Based Syst. 204 (2020) 106209.
[48] J. Zhang, A. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.
[49] S.X. Zhang, L.M. Zheng, K.S. Tang, et al, Multi-layer competitive-cooperative framework for performance enhancement of differential evolution, Inf. Sci. 482 (2019) 86–104.
[50] X.G. Zhou, C.X. Peng, J. Liu, et al, Underestimation-assisted global-local cooperative differential evolution and the application to protein structure prediction, IEEE Trans. Evol. Comput. 24 (3) (2020) 536–550.