

Preprints are preliminary reports that have not undergone peer review. They should not be considered conclusive, used to inform clinical practice, or referenced by the media as validated information.

Dual Mutations Collaboration Mechanism with Elites Guiding and Inferiors Eliminating Techniques for Differential Evolution

Libao Deng (denglibao_paper@163.com)

Harbin Institute of Technology Weihai https://orcid.org/0000-0002-5102-0330

Chunlei Li

Harbin Institute of Technology School of Electronics and Information Engineering

Haili Sun

Harbin Institute of Technology Weihai

Liyan Qiao

Harbin Institute of Technology School of Electronics and Information Engineering

Xiaodong Miao

Nanjing Tech University

Research Article

Keywords: Differential evolution, Dual mutations, Elites guiding technique, Inferior eliminating technique

Posted Date: September 7th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-855741/v1

License: (a) This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Version of Record: A version of this preprint was published at Soft Computing on November 9th, 2021. See the published version at https://doi.org/10.1007/s00500-021-06454-1.

6 7

8 9 10

11

18 19 20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

Dual Mutations Collaboration Mechanism with Elites Guiding and Inferiors Eliminating Techniques for Differential Evolution

Libao Deng 1 · Chunlei Li 1,2 · Haili Sun 1 · Liyan Qiao 2 · Xiaodong Miao³

Received: date / Accepted: date

Abstract Differential Evolution (DE) is a powerful evolutionary algorithm for global optimization problems. Generally, appropriate mutation strategies and proper equilibrium between global exploration and local exploitation are significant to the performance of DE. From this consideration, in this paper, we present a novel DE variant, abbreviated to DMIE-DE, to further enhance the optimization capacity of DE by developing a dual mutations collaboration mechanism with elites guiding and inferiors eliminating techniques. More specifically, an explorative mutation strategy DE/current-to-embest with an elite individual serving as part of the difference vector and an exploitative mutation strategy DE/e best-to-rand with selecting an elite individual as the base vector are employed simultaneously to achieve the balance between local and global performance of the whole population instead of only one mutation strategy in classical DE algorithm. The control parameters F and CR for above mutation strategies are updated adaptively to supplement the optimization ability of DMIE-DE based on a rational probability distribution model and the successful experience from the previous iterations. Moreover, an inferior solutions eliminating technique is embedded to enhance the convergence speed and compensate cost of the fitness evaluation times during the evaluation process. To evaluate the performance of DMIE-DE, experiments are

🖂 Libao Deng 52

E-mail: denglibao_paper@163.com 53

- 58 3 College of Mechanical and Power Engineering, Nanjing Tech University, Nanjing, China.
- 59
- 60
- 61 62
- 63
- 64 65

conducted by comparing with five state-of-the-art DE variants on solving 29 test functions in CEC2017 benchmark set. The experimental results indicate that the performance of DMIE-DE is significantly better than, or at least comparable to the considered DE variants.

Keywords Differential evolution · Dual mutations · Elites guiding technique · Inferior eliminating technique

1 Introduction

Differential evolution (DE), first proposed by Storn and Price (Storn and Price, 1997), is a simple yet powerful evolutionary algorithm. DE has exhibited notable performance due to its simple structure, rapid convergence speed as well as strong robustness and has been applied successfully in many domains of science and engineering such as neural network (Su et al, 2019; Baioletti et al, 2020), power system (Sakr et al, 2017; Reddy and Bijwe, 2019), medical aspect (Nunes et al, 2017; Song et al, 2019; Hosny et al, 2020), image processing (Paul and Das, 2015; Tarkhaneh and Shen, 2019) and many other practical optimization problems (Balamurugan and Muthukumar, 2019; Huang et al, 2020).

The research directions of DE mainly consist of four categories. The first direction is to implement rational mechanisms to adjust core parameters in DE. For example, The fitness-adaptive parametric scheme for the scale factor F and crossover rate CR is developed in (Ghosh et al, 2011) to maintain the exploration ability and help the population escape from the premature convergence situations. Zhu et al. (Zhu et al, 2013) proposed an adaptive population correction method to improve the optimization ability of DE. Draa et al (Draa et al, 2015) employed a cosine distribution

⁵⁴ 1 School of Information Science and Engineering, Harbin 55 Institute of Technology, Weihai, China.

⁵⁶ 2 School of Electronics and Information Engineering, Harbin 57 Institute of Technology, Harbin, China.

and a sine distribution to generate F and CR, respectively. AGDE (Mohamed and Mohamed, 2019) utilized a pre-determined specific candidate pool with a novel and effective adaptation scheme for generating appropriate values for CR. The second direction involves improving the mutation operator, which is one of the most significant operators in DE. Fan and Lampinen (Ghosh et al, 2011) proposed a trigonometric mutant strategy to accelerate the convergence speed. Zhang and Sanderson (Zhang and Sanderson, 2009) developed a DE variant with an effective mutation strategy called DE/current-pbest/1, which is a milestone for the development of DE algorithm. EFADE (Mohamed and Suganthan, 2018) provided a new triangular mutation operator which was based on the convex combination vector of the triplet defined by three randomly chosen vectors. The third improvement direction is to hybridize DE algorithm with other heuristic algorithms including particle swarm optimization (Thangaraj et al, 2011), artificial bee colony algorithm (Abraham et al, 2012), grey wolf optimization algorithm (Luo and Liu, 2020), whale optimization algorithm (Luo and Shi, 2019) and so on. The main purpose of the hybridization is to combine the advantages of DE and other algorithms to enhance the optimization capacity. The fourth direction is executed by adding an extra framework to DE. For example, Guo et al. (Guo et al, 2014) proposed a framework to detect the stagnation intelligently. When the population was stagnant, the vectors involved in mutation were selected from the archive and through this operation, the quality of solutions were significantly improved. In ADE-ALC algorithm (Fu et al, 2017), an aging mechanism is introduced into the framework of DE to maintain diversity of the population. Deng et al. (Deng et al, 2020a) introduced a regeneration framework at the dimension level to alleviate the stagnation problem in DE.

No matter which improvement strategy mentioned above is utilized, an appropriate tradeoff between the local exploitation and global exploration ability is an important guideline for the algorithm design, especially for the mutation operator, and excessive emphasis on one of them will adversely influence another. Based on this consideration, we propose a novel DE variant with double mutation strategies (DE/current-to-embest and DE/ebest-to-rand) and an inferior solution eliminating technique for further enhancing DE's optimization ability. More specifically, in the strategy DE/currentto-embest, the current individual is taken as the base vector while an randomly selected top ranking elite individual and middle ranking individual serve as part of the difference vector to guide the current individual to explore the promising area, which is propitious to the population diversity but will not lead to the premature

convergence. The strategy DE/ebest-to-rand chooses an elite individual as the base vector and two randomly selected individuals to guide the elite individual to exploit it's neighbour area. These two mutation strategies are employed simultaneously to generate the mutant vectors. Furthermore, the parameter adaptation scheme utilizes the Cauchy distribution model and a sinusoidal formula which is associated with successful experience from the previous iterations to automatically update scale factor F and crossover rate CR respectively. The inferior solution eliminating technique is introduced to reduce the population size at certain generations by eliminating several bad-performing individuals to enhance the convergence speed and compensate the cost for fitness evaluation times during the evaluation process. In a summary, the main contributions of the proposed algorithm can be listed as follows:

- (1) Dual mutation strategies including DE/current-toembest and DE/ebest-to-rand are proposed and employed simultaneously to achieve an appropriate equilibrium between the global exploration and local exploitation abilities.
- (2) The core control parameters F and CR are updated adaptively to supplement optimization ability of algorithm on basis of rational probability distribution and the successful experience from the previous iterations.
- (3) An inferior solution eliminating technique is utilized based on the principle of survival of the fittest to compensate the cost for fitness evaluation times and speed up the population convergence.

To verify the effectiveness of proposed DMIE-DE, it's compared with five state-of-the-art DE variants. Extensive experiments are carried out based on 30D, 50D and 100D benchmark functions from CEC2017 test suite. The results indicate that the proposed DMIE-DE algorithm is superior and competitive to the considered DE variants.

The remainder of this paper is arranged as follows. Section 2 studies the basic DE. Section 3 briefly reviews some advanced methods related to our research. Section 4 outlines our approach for the proposed DMIE-DE algorithm. The simulation results including the algorithm performance comparison with several DE variants on empirical functions, effectiveness certification for dual mutation strategies and inferior solution eliminating technique are described in Section 5. Conclusion is provided in Section 6.

2 Basic DE Algorithm

In this section, four basic operations in DE will be demonstrated including initialization, mutation, crossover and selection. The global minimization task of D dimensions is considered in this paper and can be defined as follows:

$$f(\vec{X}^*) = Min f(\vec{X}_i), \vec{X}_i \in \boldsymbol{S}$$
(1)

where $f(\cdot)$ denotes the objective function, and $\vec{X}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ is called the target vector, \vec{X}^* means the global optima of the objective function. The variable \boldsymbol{S} means the searching space, and the target vectors will be restricted in the searching space by a predefined lower bound $\vec{X}_{min} = (x_{min,1}, x_{min,2}, \dots, x_{min,D})$ and a upper bound $\vec{X}_{max} = (x_{max,1}, x_{max,2}, \dots, x_{max,D})$.

2.1 Initialization

To begin with, an initial population consisted of NP individuals will be produced which can be expressed as D-dimensional vectors:

$$\vec{X}_{i}^{0} = \left\{ x_{i,1}^{0}, x_{i,2}^{0}, \cdots, x_{i,D}^{0} \right\}, \quad i = 1, 2, \cdots, NP.$$
(2)

The initial population can be produced randomly by a uniform distribution within the search space via the formula as follows.

$$x_{i,j}^{0} = x_{\min,j} + rand[0,1] \cdot (x_{\max,j} - x_{\min,j})$$
(3)

where rand[0, 1] denotes a uniformly distributed number in the interval [0,1].

2.2 Mutation

At each iteration, the mutation operator is employed for each target vector to yield corresponding mutant vector $\overrightarrow{V}_{i}^{g} = (v_{i,1}^{g}, v_{i,2}^{g}, \cdots, v_{i,D}^{g})$. The five most frequently used mutation strategies are listed below.

1) DE/rand/1:

$$\overrightarrow{V}_{i}^{g} = \overrightarrow{X}_{r_{1}}^{g} + F \cdot (\overrightarrow{X}_{r_{2}}^{g} - \overrightarrow{X}_{r_{3}}^{g}) \tag{4}$$

2) DE/best/1:

$$\overrightarrow{V}_{i}^{g} = \overrightarrow{X}_{best}^{g} + F \cdot (\overrightarrow{X}_{r_{1}}^{g} - \overrightarrow{X}_{r_{2}}^{g}) \tag{5}$$

$$\overrightarrow{V}_{i}^{g} = \overrightarrow{X}_{i}^{g} + F \cdot (\overrightarrow{X}_{best}^{g} - \overrightarrow{X}_{i}^{g}) + F \cdot (\overrightarrow{X}_{r_{1}}^{g} - \overrightarrow{X}_{r_{2}}^{g}) \quad (6)$$

4) DE/rand/2:

64 65

$$\overrightarrow{V}_{i}^{g} = \overrightarrow{X}_{r_{1}}^{g} + F \cdot (\overrightarrow{X}_{r_{2}}^{g} - \overrightarrow{X}_{r_{3}}^{g}) + F \cdot (\overrightarrow{X}_{r_{4}}^{g} - \overrightarrow{X}_{r_{5}}^{g})$$
(7)

5) DE/best/2:

$$\overrightarrow{V}_{i}^{g} = \overrightarrow{X}_{best}^{g} + F \cdot (\overrightarrow{X}_{r_{1}}^{g} - \overrightarrow{X}_{r_{2}}^{g}) + F \cdot (\overrightarrow{X}_{r_{3}}^{g} - \overrightarrow{X}_{r_{4}}^{g})$$
(8)

where the indices r_1, r_2, r_3, r_4 and $r_5 \in \{1, 2, \dots, NP\}$ are mutually exclusive integers and are different form the index *i*. The vector $\overrightarrow{X}_{best}^g$ represents the optimal individual in the population at the *g*th generation. The parameter *F* is called the scaling factor, which is a positive constant within the range [0,1].

2.3 Crossover

The population diversity is enhanced by the crossover operator. The trial vector $\vec{U}_i^g = (u_{i,1}^g, u_{i,2}^g, \dots, u_{i,D}^g)$ is produced by recombining the variables of target vector \vec{X}_i^g and mutation vector \vec{V}_i^g . The most generally implemented binomial crossover operator is formulated as follow:

$$u_{i,j}^g = \begin{cases} v_{i,j}^g & \text{if } rand[0,1] \leqslant CR \text{ or } j = j_{rand} \\ x_{i,j}^g & \text{otherwise} \end{cases}$$
(9)

where rand[0, 1] is a random constant complying with the uniform distribution in [0,1]. The parameter CRcontrols the fraction of components copied from the mutant vector and $j_{rand} \in (1, 2, \dots, D)$ is a stochastic integer to ensure that at least one component of trial vector is inherited from the mutant vector.

2.4 Selection

The population is rebuilt by employing the selection operation between the target vector \vec{X}_i^g and the trial vector \vec{U}_i^g . Individual vectors with better fitness values survive as the offsprings. For a minimization problem, the selection operator is performed as follows:

$$\vec{X}_{i}^{g+1} = \begin{cases} \vec{U}_{i}^{g} & \text{if } f(\vec{U}_{i}^{g}) \leqslant f(\vec{X}_{i}^{g}) \\ \vec{X}_{i}^{g} & \text{otherwise} \end{cases}$$
(10)

To sum up, the pseudo code of the classical DE algorithm is written in Algorithm 1.

3 Related Works

Researches on DE have reached an impressive state over the past two decades, and many variants have been proposed by researchers to enhance DE's optimization capability. In this section, we will briefly review some 4

1

2

3 4

5

6

Algorithm 1: The classical DE algorithm with
DE/rand/1 strategy.
1 Randomly produce the initial population through:
$x_{i,j}^0 = x_{min,j} + rand[0,1] \cdot (x_{max,j} - x_{min,j});$
2 Calculate the initial fitness value of individuals and
find the initial $\overrightarrow{X}_{best}$ and $f(\overrightarrow{X}_{best})$;
3 while (The stopping criterion is not met) do
4 for $(i = 1 : NP)$ do
5 Generate mutation vector \vec{V}_i^g by:
$\overrightarrow{V}_{g}^{g} = \overrightarrow{X}_{g1}^{g} + F \cdot (\overrightarrow{X}_{g2}^{g} - \overrightarrow{X}_{g2}^{g})$
6 Execute the crossover operator formulated by
Eq.(9) to generate the trial vector \vec{U}_i^g .
7 Employ the selection operator depicted by
Eq.(10) to generate the offspring \vec{X}_i^{g+1} .
8 end
9 Recalculate the fitness value of individuals and
update $\overrightarrow{X}_{best}$ and $f(\overrightarrow{X}_{best})$;
10 end
11 Output \vec{X}_{best} and $f(\vec{X}_{best})$.

modified methods related to our research. For a comprehensive survey on DE, please refer to literatures (Das et al, 2016) and (Opara and Arabas, 2019).

An appropriate trial vector generation strategy is of great significance to the performance of DE, and many scholars have proposed various improvement strategies. To modify the DE/current-to-best/1 approach, Zhang and Sanderson (Zhang and Sanderson, 2009) introduced a novel mutation strategy DE/current-to-pbest in their research of JADE. The innovation of this strategy lies in an optional external archive which could provide information about the evolution direction. Tanabe and Fukunaga (Tanabe and Fukunaga, 2013) proposed an advanced JADE variant named success history based DE, which applied historical memory archives to store a set of well-performed values of CR and F. In order to alleviate the premature convergence and stagnation problems, Islam et al. (Islam et al. 2012) developed a DE variant, called MDE_pBX with a new mutation strategy DE/current-to-gr_best/1 and a novel crossover scheme *p*-best crossover operation. Otani et al. (Otani et al, 2013) introduced a DE/isolated/1 mutation operator to generates new individuals close to an isolated individual which was expected to evenly allocate the search resources. Mohamed and Suganthan (Mohamed and Suganthan, 2018) presented an enhanced fitnessadaptive DE with applying a new triangular mutation operator to improve the global and local search capabilities and increase the convergence speed of the basic DE. Zheng et al. (Zheng et al, 2017) proposed a novel mutation strategy, referred to as DE/current-to $ci_mbest/1$, which utilized the collective information of the m best individuals to form the difference vector in mutation.

Efforts have also been made to utilize multiple mutation strategies during the evolution process. For example, Li et al. (Li et al, 2017a) introduced three mutation strategies including DE/current to cbest, DE/current-to-rbest/1 and DE/current-tof best. Meanwhile, the whole population is divided into three subpopulations and individuals in each subpopulation utilize different mutation based on their fitness value. Mohamed et al. (Mohamed et al, 2019) introduced two novel mutation operators DE/currentto-ord_pbest and DE/current-to-ord_best to enhance DE's performance. Liu et al. (Li et al, 2020) presented two variants of the classical DE/rand/2 and DE/best/2 strategies, referred to as DE/e-rand/2 and DE/e-best/2 respectively, and these two mutations were used to achieve a balance between global exploration and local exploitation. Deng et al. (Deng et al, 2020b) employed an explorative mutation technique DE/seeds-to-seeds and an exploitative mutation strategy DE/seeds-to-rand to enhance the optimization ability of DE.

The control parameters are also extremely important to DE's optimization capability, especially for the scale factor F and crossover rate CR. However, the parameters are always sensitive to solve different optimization problems, therefore, researchers have introduced various adaptive or self-adaptive parameter adjustment methods to alleviate this problem. For instance, Draa et al. (Draa et al, 2015) introduced a sinusoidal differential evolution namely SinDE, which utilized two sinusoidal formulas to adjust the values of scaling factor and the crossover rate. Recently, Draa et al. (Draa et al, 2019) presented a new variant of SinDE, which utilized a compound sine-based formula to adjust F and CR. Compared with SinDE, the new version could make parameters variation less monotone. Yu et al. (Yu et al, 2013) proposed a two-level adaptive parameters control strategy. More specifically, the individual-level parameters are updated according to not only individuals fitness value but also its distance from the optimal individual, and the population-level parameters were updated adaptively during the evolution process. Meng et al. (Meng et al, 2018) introduced a DE variant called PALM-DE to generate new parameters with an adaptive learning mechanism for the inconvenience in selecting control parameters. However, this variant was heavily relied on the number of individuals to generate a suitable value for CR. Thus, on the basis of (Meng et al, 2018), Meng et al. (Meng et al, 2019) introduced a new parameter adaptive DE variant called

PaDE to resolve inappropriate adaptation schemes with a novel grouping strategy.

To summarize, the improvement methods on mutation strategies and control parameters are promising research area, which therefore provides the references for our proposed DMIE-DE variant in this paper.

4 Description of DMIE-DE

In this section, we will provide an explicit elaboration of proposed DMIE-DE algorithm. Firstly, we will introduce the designing motivation of DMIE-DE. Then, we will describe the new mutation method with two strategies including DE/current-to-*em*best and DE/*e*best-torand as well as main parameters adaptation schemes. Lastly, the inferior solution eliminating technique will be described.

4.1 Motivation

The conspicuous flaw for DE variants lies in the fact that it is difficult to coordinate the global exploration ability and the local exploitation ability. Globality and locality are ambivalent but existing in solving the optimization problems. The mutation operator in DE is generally composed of base vectors and difference vectors. The base vector is used to determine the searching reference point while the difference vectors are utilized to provide the searching direction and perturb the base vector. The classical mutation strategy DE/rand/1 is said to be the most frequently used scheme (Das and Suganthan, 2011) with maintaining outstanding global exploration ability but it tends to cause the problem of stagnation because of its random manner. Studies indicate that greedy strategies, like DE/best/1 and DE/best/2, have higher convergence speed and strong exploitive ability (Li et al, 2017b) but it is easy to cause premature convergence problem. Therefore, if an algorithm could employ dual mutation strategies with different emphasis that one focuses on exploitation work while the other on exploration work at the same time to generate offsprings and select a better one to next generation, it may promote the accuracy and robustness because their explorative and exploitive advantages are combined. However, the problem with this scheme is that the fitness evaluation times will also be doubled. To overcome this problem, we can eliminate some individuals with bad performance at certain time during the evolution process.

4.2 Dual Mutations Collaboration Mechanism with Elites Guiding Technique

In order to achieve a proper balance of the global exploration and local exploitation, we propose dual mutation strategies called DE/current-to-embest and DE/ebestto-rand.

The first mutation DE/current-to-embest is relatively explorative to maintain the population diversity, and is formulated as follows:

$$\overrightarrow{V}_{i} = \overrightarrow{X}_{i} + F_{i} \cdot (\overrightarrow{X}_{ebest} - \overrightarrow{X}_{i}) + F_{i} \cdot (\overrightarrow{X}_{mbest} - \overrightarrow{X}_{r}) \quad (11)$$

From Eq. (11), we can see that this strategy selects the current individual \overline{X}_i as the base vector and two difference vectors are utilized. More specifically, \overline{X}_{ebest} is called elite individual which is randomly selected from the top 10% individuals with better fitness values in the current population to guide the evolution direction. The vector X_{mbest} is called the medium individual which is randomly selected from a group centred on the middle ranking individual according to the fitness values, and the group size is D/5, where the parameter D means the problem dimension. \overline{X}_r is a random individual from the union $P \cup A$, where P denotes the current population, and A represents an external archive used to store the inferior parents that fail in the selection process during last iteration. Fig. 1 illustrates this mutation strategy in a 2-D plane picture in which the marker a means the vector $\overrightarrow{X}_{mbest} - \overrightarrow{X}_{r2}$, and markers b and c denote the vectors $F_i \cdot (\overrightarrow{X}_{mbest} - \overrightarrow{X}_{r2})$ and $F_i \cdot (\overrightarrow{X}_{ebest} - \overrightarrow{X}_i)$ respectively. From the figure, we can draw two conclusions. On the one hand, the current individual serves as the searching centre, which will be helpful to get out of the current poor area. On the other hand, it combines the relatively better solution, i.e., elite individual, and medium solution in the current population to guide the current individual to a more promising area which is helpful to a faster convergence speed.

The second mutation strategy DE/ebest-to-rand is designed to be relatively exploitive and its calculation formula can be written as follows:

$$\overrightarrow{V}_{i} = \overrightarrow{X}_{ebest} + F_{i} \cdot (\overrightarrow{X}_{r1} - \overrightarrow{X}_{r2})$$
(12)

where the vectors $X_{\underline{ebest}}$ is defined consistently as Eq. (11). The vector \overrightarrow{X}_{r1} means a random individual selected from the current population P and \overrightarrow{X}_{r2} is a random individual from the union $P \cup A$. The schematic diagram for this strategy can be illustrated as Fig. 2 in which the marker e denotes the difference vector $\overrightarrow{X}_{r1} - \overrightarrow{X}_{r2}$. From the figure, we can observe that this mutation strategy selects the elite individual as base vector instead of the best individual like DE/best/1,

65



Fig. 1: Illustration of the mutation operator DE/current-to-*em*best.



Fig. 2: Illustration of the mutation operator DE/ebest-to-rand.

which is expected to have better solutions around it and meanwhile alleviate the premature convergence problems. What's more, the difference vector is beneficial to maintain the population diversity because of its random characteristic.

At each generation, these two mutation strategies, i.e., DE/current-to-*em*best and DE/*e*best-to-rand, are applied simultaneously to generate two trial vectors. Fitness values of these two trial vectors are calculated and compared to select a better one to against with the target vector in the selection operation. This scheme is expected to enhance the optimization capacity and robustness of DMIE-DE because the exploration and exploitation strengthes are complementary.

4.3 Control Parameters Adaptation

The control parameters also paly an important role in the effectiveness of DE. The scale factor F is utilized to determine the searching radius centred on the base vector, and the crossover rate CR controls the components inherited from mutant vector. Different parameter adaptation schemes may be suitable for different optimization problems. In our work, we design a modification of the parameter adaptation scheme proposed in (Zhang and Sanderson, 2009).

At each generation, the scale factor F_i for each individual is generated independently according to a Cauchy distribution model:

$$F_i = Cauchyrand(\mu_F, \theta_F) \tag{13}$$

where μ_F is called the location parameter which determines the peak location of the distribution and θ_F denotes the scale parameter which decides half width value at half of the maximum distribution value. In our work, we set the parameter θ_F to a fixed value 0.1, while the parameter μ_F is updated at each generation according to the previous successful experience and its initial value is set to be 0.3. Then, the update formula of μ_F can be written as follows:

$$\mu_F = (1 - \rho) \cdot \mu_F + \rho \cdot mean_{Lm}(S_F) \tag{14}$$

In Eq. (14), the parameter ρ is a dynamically adjusted by an normal distribution:

$$\rho = abs(0.5 \cdot Normalrand(0,1)) \tag{15}$$

The function $abs(\cdot)$ means the absolute value and *Normalrand*(0,1) is a random number subject to the standard normal distribution. S_F is a set of successful F values during last iteration, and $mean_{Lm}(\cdot)$ denotes the Lehmer mean function:

$$mean_{Lm}(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F}$$
(16)

According to (Zheng et al, 2017), Cauchy distribution is beneficial for the diversify of F and can avoid premature convergence to some extent. Moreover, the Lehmer mean is utilized to generate larger F values to enhance the population diversity.

For the crossover rate CR, we employ a sine distribution formula which can be illustrated as follow:

$$CR_i = \mu_{CR} + \delta \cdot sin(\pi \cdot (rand[0, 1] - 0.5))$$
 (17)

The parameter μ_{CR} is initialized as 0.8, and is updated by the following equation:

$$\mu_{CR} = (1 - \rho) \cdot \mu_{CR} + \rho \cdot mean_{Lm}(S_{CR}) \tag{18}$$

The parameter ρ is the same definition as Eq. (15). The successful CR are also stored for the generation of μ_{CR} by the Lehmer mean:

$$mean_{Lm}(S_{CR}) = \frac{\sum_{CR \in S_{CR}} CR^2}{\sum_{CR \in S_{CR}} CR}$$
(19)

Algorithm 2: Pseudo code of the inferior so-
lutions eliminating technique.
1 Input:(1) Current population size: NP; (2) Current index of iteration: g;(3) The generation interval: gip;
2 if $(Mod(g,gip) == 0)$ then
3 Sorting the current population in an ascending
order according to the fitness values;
4 Decide the number of inferior solutions needing
to be eliminated: $EP = round(NP/10);$
5 Eliminate the last <i>EP</i> solutions;
6 Recount the population size: $NP = NP - EP$;
7 end
s else
9 $g = g + 1;$
10 end
11 Output: The current population and its size NP.

In Eq. (17), the δ is set to 0.1 to control rate of change caused by sine distribution and rand[0, 1] is a random constant complying with the uniform distribution in [0,1]. The design philosophy of the adaptation of CRis that better CR values are more likely to generate individuals that will have more chances to survive and these better values should be inherited to the next generation.

4.4 Inferiors Eliminating Technique

During the selection process, the fitness value of each individual need to be evaluated for the comparison with the target vector for surviving to next generation which is time-consuming. When the proposed dual mutation strategy is operated, function evaluation times will be doubled. In order to ease the reduction of practical evaluation rate for function evaluation, inferior solution eliminating technique is provided to improve the situation. More specifically, some individuals with bad fitness values will be removed from the population for a certain generation interval, which is presented by parameter gip in our algorithm. In fact, the value of gipis important for the performance of proposed inferiors eliminating technique. On the one hand, if the value of gip is too large, the inferiors eliminating technique will be seldom utilized so the function evaluations will be wasted. On the other hand, if *gip* is set to a small value, individuals will be discarded frequently and the diversity of population might be affected, which is against the optimization capacity. Therefore, it's necessary to provide a suitable value for gip. In our work, we set gipto a constant, i.e., 400, which will be discussed in the experimental part. During the elimination process, the eliminating size EP is associated with the initial population size. and the population size will be reduced by a

tenth for every *gip* iterations. For a clearer illustration, this technique is presented as Algorithm.2, in which the function $Mod(\cdot)$ means the modular arithmetic and the function $round(\cdot)$ denotes the rounding function.

The detailed description of dual mutation strategies and control parameters adaptation scheme as well as the inferior solution eliminating technique have been provided. Then, an overall implementation of DMIE-DE is presented in Algorithm 3.

5 Experimental Results

In this section, the comparisons results on the adopted experimental platform between proposed DMIE-DE and five state-of-the-art DE variants will be presented. Moreover, the parameter sensitivity analysis as well as the effectiveness certification for dual mutation strategies and inferior solution eliminating technique will be analysed in detail.

5.1 Benchmark Functions and Compared Algorithms

To evaluate the performance of the proposed DMIE-DE algorithm, comparative experiments are carried out based on 29 benchmark functions provided by CEC2017 platform, and the *i*th function is denoted by f_i in this paper. These functions can be divided into the following four groups with different characteristics: f_1, f_3 : unimodal functions; $f_4 - f_{10}$: simple multimodal functions; $f_{11} - f_{20}$: hybrid functions; $f_{21} - f_{30}$: composition functions. It should be pointed out that function f_2 has been removed from the test suite because of its unstable characteristic according to the original reference (Wu et al, 2017), which also provides more detail information about other functions in this test suite.

The performance of DMIE-DE is compared with five state-of-the-art DE variants including two classical DEs: JADE (Zhang and Sanderson, 2009) and SinDE (Draa et al, 2015); three recently proposed DE variants: TSDE (Liu et al, 2016), AGDE (Mohamed and Mohamed, 2019) and EFADE (Mohamed and Suganthan, 2018). For a convincing comparison, the associated parameters of the five compared DE variants are configured as recommended in the corresponding original references and the detail information is listed in Table 1. In DMIE-DE, the initial population size NP is set to $5 \cdot D$ for all different problems and Drepresents the problem dimension. This setting for the value of NP is recommended by DE's original reference (Storn and Price, 1997) and is also adopted by

65

Algorithm 3: Pseudo code of proposed DMIE-DE algorithm. Input: (1) The initial population size: NP; (2) The maximum of function evaluation: FES_{max} ; (3) The problem dimension: D. Set $\mu_F = 0.3$; $\mu_{CR} = 0.8$; Generation index g=1; Archive $A = \emptyset$; FES = 0; Generate an initial population P^0 by Eq. (3); з Calculate each individual's fitness value and find the initial \vec{X}_{best} and its fitness value $f(\vec{X}_{best})$; FES = FES + NP;while $FES \leq FES_{max}$ do Set $S_F = \emptyset$; $S_{CR} = \emptyset$; for (i = 1 : NP) do Generate F_i and CR_i by Eq. (13) and Eq. (17) respectively; Randomly select \overline{X}_{ebest}^{g} form the top 10% best individuals; Randomly select \vec{X}_{mbest}^{g} form the medium group controlled by the dimension D; Randomly select \vec{X}_{r1}^g and \vec{X}_r^g , \vec{X}_{r2}^g form the current population P and the union $P \cup A$ respectively; Generate $\overrightarrow{V}_{i1}^g$ and $\overrightarrow{V}_{i2}^g$ using the DE/current-to-*em*best and DE/*e*best-to-rand strategy as Eq. (11) and Eq. (12) respectively; Execute the crossover strategy as Eq.(9) to generate \vec{U}_{i1}^g and \vec{U}_{i2}^g for \vec{V}_{i1}^g and \vec{V}_{i2}^g respectively; Calculate the fitness value $f(\vec{U}_{i1}^g)$ and $f(\vec{U}_{i2}^g)$ and select a better one as \vec{U}_{i}^g ; FES = FES + 2;if $(f(\overrightarrow{U}_i^g) < f(\overrightarrow{X}_i^g))$ then $| \overrightarrow{X}_i^{g+1} = \overrightarrow{U}_i^g; F_i \text{ and } CR_i \text{ are stored in } S_F \text{ and } S_{CR} \text{ respectively; } \overrightarrow{X}_i^g \text{ is stored in archive } A;$ end else $\overrightarrow{X}_{i}^{g+1} = \overrightarrow{X}_{i}^{g};$ end end Update the optimal individual \vec{X}_{best} and fitness value f_{best} . Update μ_F and μ_{CR} according to Eq. (14) and Eq. (18); Execute the inferior solution eliminating technique in Algorithm 2. 27 end **28** Output: $\overrightarrow{X}_{best}$ and $f(\overrightarrow{X}_{best})$.

some advanced DE variants, such as Refs. (Guo et al, 2014; Guo and Yang, 2014). According to the requirements for these test functions in Ref. (Wu et al, 2017), the maximum number of function evaluations FES_{max} is set to $10,000 \cdot D$, which is also the stopping criteria. For the fairness of the comparison, the computational results are all obtained on a PC with Intel(R) Xeon(R) Gold 5115, 2.39GHz CPU, 64.0 GB Memory and MAT-LAB R2018b on Windows 10 system.

5.2 Performance Metric

In our simulation, all the compared algorithms are conducted 50 independent runs to obtain the function error value which is calculated by $f(\vec{X}_{best}) - f(\vec{X}^*)$, where $f(\vec{X}_{best})$ is the global optimal fitness value obtained by each algorithm and $f(\vec{X}^*)$ is the fitness value of the actual global optimal solution. The average value(denoted by "Mean") and the standard deviation(denoted by "Std.") of the function error values are utilized to reflect the effectiveness of test algorithms for searching the solution within limited evaluations. Convergence graphics

Table 1: Parameters setting for DE variants

Algorithm	Parameter setting
JADE	$\mu F_0 = \mu C R_0 = 0.5; p = 0.05$
SinDE	F = 0.5; CR = 0.9; freq = 0.25
TSDE	[F=1.0, CR=0.1]; [F=1.0, CR=0.9]; [F=0.8, CR=0.2]
AGDE	$CR_1 \in [0.05, 0.15]; CR_2 \in [0.9, 1]; p_j = 1/j$
EFADE	F = 0.7; p = 0.5

of the function error value are also utilized to compare the convergence characteristic of each algorithm in the respective experiment.

Moreover, to draw a statistically sound conclusion. we employ the single-problem analysis based on Wilcoxon test (García et al, 2009) at the 0.05 significance level to test whether there is significant difference between DMIE-DE and each compared DE variant and the symbols "+/=/-" represent that the performance of the DMIE-DE is significantly better than, similar to, or inferior to each competitor, respectively. Moreover, the multiple-problem analysis based on Wilcoxon test and the Friedman test are also conducted at the 0.05 significance level. The Friedman test is a non-parametric test

		JADE			SinDE			TSDE AGDE			EFADE			DMI	E-DE		
Func.	Mean	Std.	+/-/=	Mean	Std.	+/-/=	Mean	Std.	+/-/=	Mean	Std.	+/-/=	Mean	Std.	+/-/=	Mean	Std.
f_1	2.48E+03	3.15E+03	+	5.12E+03	3.74E+03	+	6.89E+09	1.24E+09	+	3.23E+03	3.72E+03	+	8.07E+04	$6.51E{+}04$	+	2.33E-14	8.00E-15
f3	8.52E+04	7.20E+04	+	1.00E+05	$1.56E{+}04$	+	1.75E+05	1.94E+04	+	4.02E-07	8.74E-07	+	3.68E + 04	$8.35E{+}03$	+	7.71E-12	1.94E-11
f_4	1.27E+02	4.05E+01	+	1.25E+02	2.45E+01	+	1.10E+03	1.28E+02	+	8.43E+01	4.62E + 01	=	2.07E+02	$2.53E{+}01$	+	7.35E+01	4.73E+01
f_5	2.33E+02	1.15E+01	+	1.04E+02	$2.76E{+}01$	+	4.66E+02	1.82E+01	+	2.08E+02	3.37E + 01	+	3.78E+02	$1.62E{+}01$	+	5.45E+01	1.70E+01
f_6	4.92E-02	3.41E-02	+	2.54E-05	3.39E-06	+	3.29E+01	2.91E+00	+	2.38E-02	3.96E-02	+	7.58E+00	$1.01E{+}00$	+	1.04E-04	6.84E-04
f7	2.82E+02	1.37E+01	+	1.98E+02	$3.43E{+}01$	+	7.16E+02	4.11E+01	+	2.89E+02	1.66E + 01	+	$4.23E{+}02$	$1.62E{+}01$	+	9.31E+01	9.23E+00
f_8	2.28E+02	1.10E+01	+	1.09E+02	2.62E+01	+	4.63E+02	2.10E+01	+	1.97E+02	4.74E+01	+	3.75E+02	$1.66E{+}01$	+	5.25E+01	2.12E+01
f_9	4.71E-02	1.37E-01	-	3.59E-07	2.27E-07	-	1.07E+04	1.46E+03	+	4.75E+01	7.15E+01	+	3.75E+01	$1.75E{+}01$	+	4.84E-01	5.52E-01
f_{10}	9.17E+03	4.03E+02	+	8.28E + 03	7.08E+02	+	1.25E+04	3.63E+02	+	9.24E+03	4.10E + 02	+	$1.12E{+}04$	$2.93E{+}02$	+	6.65E+03	1.86E+03
f11	3.82E+02	5.01E+02	+	4.39E + 01	5.82E+00	-	1.90E+03	3.22E+02	+	5.38E + 01	1.63E+01	-	$2.19E{+}02$	$1.91E{+}01$	+	8.14E+01	2.32E+01
f12	5.53E+04	3.97E+04	+	2.34E+06	1.23E+06	+	1.07E+09	2.06E+08	+	7.36E+04	$4.23E{+}04$	+	$1.46E{+}06$	5.72E + 05	+	6.41E+03	5.26E+03
f13	5.61E+02	1.76E+02	+	1.96E+03	1.83E+03	+	7.69E+07	1.65E+07	+	4.46E+03	5.21E + 03	+	2.13E+03	$1.34E{+}03$	+	7.83E+01	3.76E+01
f_{14}	2.17E+04	9.20E+04	+	6.48E + 03	5.66E + 03	+	8.58E+04	2.84E+04	+	6.81E+02	7.24E+02	+	1.67E+02	$1.35E{+}01$	+	5.27E+01	1.21E+01
f ₁₅	2.35E+02	4.94E+01	+	8.56E + 02	$9.28E{+}02$	+	2.46E+06	8.27E + 05	+	2.42E+03	2.69E+03	+	$6.99E{+}02$	$6.97E{+}02$	+	6.05E+01	1.97E+01
f_{16}	2.09E+03	2.16E+02	+	5.71E + 02	$2.25E{+}02$	-	3.38E+03	2.14E+02	+	6.83E+02	2.90E+02	=	2.62E+03	$2.11E{+}02$	+	7.80E+02	2.44E+02
f17	1.42E+03	1.64E+02	+	3.42E + 02	$1.41E{+}02$	-	2.12E+03	1.69E+02	+	5.97E+02	2.38E+02	+	1.54E+03	$1.61E{+}02$	+	4.47E+02	2.13E+02
f ₁₈	2.56E+02	1.01E+02	+	2.37E+05	1.34E+05	+	3.83E+06	1.33E+06	+	5.74E + 03	4.04E+03	+	9.36E + 03	$6.40E{+}03$	+	6.68E+01	3.52E+01
f_{19}	1.00E+02	2.55E+01	+	8.41E + 03	3.64E + 03	+	1.46E+06	4.53E+05	+	4.23E+03	5.72E + 03	+	$2.03E{+}02$	$2.57E{+}02$	+	4.84E+01	1.75E+01
f_{20}	1.25E+03	1.63E+02	+	2.79E + 02	$1.12E{+}02$	=	1.62E+03	1.83E+02	+	4.64E+02	1.85E+02	+	$1.30E{+}03$	$1.31E{+}02$	+	3.57E + 02	2.32E+02
f21	4.33E+02	1.33E+01	+	3.05E+02	$2.61E{+}01$	+	6.55E+02	1.85E+01	+	3.53E+02	7.53E+01	+	5.82E + 02	$1.29E{+}01$	+	2.49E+02	1.04E+01
f_{22}	7.91E+03	2.86E+03	+	7.74E+03	$2.41E{+}03$	+	1.23E+04	1.80E + 03	+	5.24E + 03	4.66E + 03	=	$9.05E{+}03$	3.05E+03	+	6.14E+03	2.40E+03
f_{23}	6.51E+02	1.70E+01	+	5.00E + 02	$1.92E{+}01$	+	9.04E + 02	1.93E+01	+	4.91E+02	3.61E + 01	+	8.26E + 02	$1.76E{+}01$	+	4.72E+02	2.26E+01
f_{24}	7.22E+02	1.74E+01	+	5.63E + 02	$1.41E{+}01$	+	9.62E+02	1.86E+01	+	5.84E + 02	1.93E+01	+	$8.93E{+}02$	$1.99E{+}01$	+	5.50E+02	2.16E+01
f_{25}	5.19E+02	3.54E+01	+	4.85E+02	1.75E+00	=	1.41E+03	1.21E+02	+	5.35E+02	3.91E+01	+	5.97E + 02	$1.77E{+}01$	+	5.06E+02	3.50E+01
f_{26}	3.16E+03	1.52E+02	+	1.82E+03	$1.67E{+}02$	+	5.94E+03	1.89E+02	+	2.09E+03	2.64E+02	+	$4.80E{+}03$	$1.90E{+}02$	+	1.64E+03	1.39E+02
f_{27}	5.31E+02	1.14E+01	-	$5.07E \pm 02$	$5.16E{+}00$	-	1.17E+03	5.79E + 01	+	6.23E+02	4.02E+01	+	$9.58E{+}02$	$3.91E{+}01$	+	5.38E + 02	2.41E+01
f_{28}	4.86E+02	2.43E+01	+	$4.59E{+}02$	1.46E-02	+	1.44E+03	1.58E+02	+	4.92E+02	1.90E+01	+	$6.12E{+}02$	$2.60\mathrm{E}{+}01$	+	4.72E+02	2.14E+01
f29	1.00E+03	1.25E+02	+	4.23E+02	$3.29E{+}01$	+	2.83E+03	2.21E+02	+	4.14E+02	1.14E+02	-	1.74E+03	$1.29E{+}02$	+	4.43E+02	1.27E+02
f ₃₀	6.57E+05	6.63E+04	=	7.10E+05	$5.09E{+}04$	+	6.94E+07	1.23E+07	+	8.36E+05	1.04E+05	+	1.73E+06	$1.58\mathrm{E}{+}05$	+	6.51E+05	5.73E+04
	Total Num	ber of $(+/=)$	-) 26/1/2	Total Numb	er of $(+/=/-)$) 22/2/5	Total Num	ber of $(+/=/$	-) 29/0/0	Total Numb	er of (+/=/-) 24/3/2	Total Num	ber of $(+/=/$	/-) 29/0/0		

Table 2: Mean and standard deviation of function error values obtained by JADE, SinDE, TSDE, AGDE, EFADE and DMIE-DE on 50D test functions in CEC2017 over 50 independent runs.

methods for multiple comparison to detect significant difference and calculate the average ranking between all the compared algorithms.

5.3 Results and Analysis

The average and standard deviation of the function error values obtained by DMIE-DE and five compared variants for solving 50D benchmark functions are presented in Table 2, and the results of 30D and 100Dfunctions are summarized in Table A1 and Table A2 respectively at the supplementary due to the page limit. In these tables, The smallest average of function error values obtained for each function are highlighted in boldface. For the sake of convenience, the results of the single-problem analysis by the Wilcoxon test are listed in Table 3 and the comparison about the number of the smallest error values obtained by each algorithm is plotted in Fig. 3.

For 30D test functions, in terms of the smallest fitness error values, DMIE-DE ranks second by getting the smallest function error value on 10 functions and is defeated by SinDE with obtaining smallest function error value on 12 functions. For other algorithms, AGDE, TSDE, JADE and EFADE are the winners on 6, 5, 4, 4 functions in turn. From the perspective of statistical results based on Wilcoxon test, DMIE-DE is significantly better than EFADE on 23 functions, and is beaten by EFADE on only 2 function (i.e., f_{15} and f_{18}). For JADE, SinDE, TSDE and AGDE, DMIE-DE performs better on 16, 16, 17 and 18 functions, meanwhile it shows equal performance on 10, 9, 5 and 6 functions, respectively.

Table 3: Results of the single-problem Wilcoxon test between DMIE-DE and other compared variants at the 0.05 significance level.

	0		
DMIE-DE VS.	30D(+/=/-)	50D(+/=/-)	100D(+/=/-)
JADE	17/10/3	27/1/2	20/4/6
SinDE	17/9/4	23/2/5	28/1/1
TSDE	18/5/7	30/0/0	30/0/0
AGDE	19/6/5	25/3/2	28/1/1
EFADE	24/4/2	30/0/0	27/2/1

Considering 50D problems, DMIE-DE still shows its outstanding performance although it's more difficult to deal with higher dimensional problems. In terms of the average function error, DMIE-DE takes the first-ranking position compared with other algorithms by yielding the smallest function error values on 18 test functions, which is increased by 8 functions



Fig. 3: The number of the smallest error values obtained by each algorithm on CEC2017 with different dimensions.

compared with that of 30*D* problems. SinDE ranks the second place by winning on 9 functions and is followed by AGDE on 2 functions. For other algorithms, their performance is slightly worse without obtaining smallest function error values. Based on the single-problem Wilcoxon test, it's noticeable the DMIE-DE obtains 29 significantly better solutions without getting worse or equal results compared with TSDE and EFADE algorithms. For JADE, SinDE and AGDE, DMIE-DE yields better results on 26, 22 and 24 functions, meanwhile, it shows equal performance on 1, 2 and 3 functions, respectively.

For 100D test functions, DMIE-DE obtains the smallest fitness error for 21 functions, much more than other algorithms. In detail, JADE wins on 7 functions and EFADE on 1 functions while SinDE, TSDE and AGDE on zero function with yielding the optimal average results. In terms of the Wilcoxon test, DMIE-DE still outperforms TSDE on all the 29 test functions, which remains the same as the performance of 50D. For SinDE and AGDE, DMIE-DE shows the similar performance by yielding better solutions on 27 functions, equal solutions on 1 function, and worse results on 1 function respectively. JADE performs best among the 5 compared DE variants by beating DMIE-DE on 5 functions and obtaining equal results on 4 functions. For EFADE, it is beaten by DMIE-DE on 26 functions except for f_9 , f_{11} and f_{14} , on which it shows equal or better performance.

The multi-problem analysis results based on the Wilcoxon test between DMIE-DE and other compared variants at the 0.05 significance level are presented in Table 4, in which R^+ and R^- mean the sum of ranks that DMIE-DE performs significantly better and worse

Table 4: Results of the multiple-problem Wilcoxon test between DMIE-DE and other compared variants at the 0.05 significance level.

		,			
Dimension	DMIE-DE VS.	\mathbf{R}^+	\mathbf{R}^{-}	Asymptotic p -value	$\alpha = 0.05$
	JADE	336.5	98.5	0.009376	YES
	SinDE	303.0	132.0	0.062233	NO
30D	TSDE	261.0	145.0	0.177829	NO
	AGDE	290.0	116.0	0.046316	YES
	EFADE	374.5	60.5	0.000660	YES
	JADE	430.0	5.0	0.000004	YES
	SinDE	352.5	82.5	0.003305	YES
50D	TSDE	435.0	0.0	0.000002	YES
	AGDE	388.5	46.5	0.000200	YES
	EFADE	435.0	0.0	0.000002	YES
	JADE	379.0	56.0	0.000460	YES
	SinDE	433.0	2.0	0.000003	YES
100D	TSDE	435.0	0.0	0.000002	YES
	AGDE	422.0	13.0	0.000009	YES
	EFADE	426.0	9.0	0.000006	YES

Table 5: Average ranking of DMIE-DE and other compared variants by Friedman test at the 0.05 significance level.

Algorithms	Ranking on $30D$	Ranking on $50D$	Ranking on $100D$
JADE	3.5345	3.3103	2.4655
SinDE	3.3621	2.4828	3.7931
TSDE	3.2241	6.0000	5.8621
AGDE	3.5172	3.1207	3.4138
EFADE	4.6379	4.5172	4.1034
DMIE-DE	2.7241	1.5690	1.3621

than each competitor, respectively. The asymptotic pvalue indicates the difference level between each pair of algorithms, and if the asymptotic *p*-value is less than the significance level (i.e., 0.05), it will be assigned the marker "YES" which means there is 95% probability to ensure that there is significant difference between DMIE-DE and its competitor and vice versa for "NO". From the results, we can draw two conclusions. On the one hand, R^+ value is much higer than R^- compared with all the other five variants in 30D, 50D and 100Dfunctions, which proves that DMIE-DE performs significantly better than other algorithms in all the three dimensions. On the other hand, except for SinDE and TSDE when solving 30D problems, the asymptotic *p*value is much lower than the significance level 0.05 for other cases. What's more, compared with 30D functions, DMIE-DE provides much lower asymptotic pvalue than 50D and 100D functions, which indicates the outstanding performance of DMIE-DE for higher dimensional optimization problems.

The results in Table 5 present the average ranking of DMIE-DE and other compared variants by Friedman test at the 0.05 significance level for solving 30D, 50D

63 64 65



Fig. 4: The convergence process of the mean fitness errors derived from JADE, SinDE, TSDE, AGDE, EFADE and DMIE-DE on 30D benchmark functions f_1 , f_7 , f_8 , f_{12} , f_{21} , f_{30} .



Fig. 5: The convergence process of the mean fitness errors derived from JADE, SinDE, TSDE, AGDE, EFADE and DMIE-DE on 50D benchmark functions f_1 , f_5 , f_7 , f_8 , f_{14} , f_{21} .

and 100D functions in CEC2017, and the first ranking values are presented in boldface. From the results, we can see that DMIE-DE is the best among the compared algorithms on lower or higher dimensional problems. More specifically, the average ranking of DMIE-DE remains the first place for these three dimensional cases. Moreover, along with the increase of dimension, the average ranking value of DMIE-DE becomes lower and lower, which also proves that the performance of DMIE-DE becomes better when solving higher dimensional problems.

DMIE-DE and compared DE variants are also contrasted from the perspective of convergence speed. Figs.4-6 respectively depict the convergence progress of DMIE-DE and other DE variants for solving several sample test functions selected from CEC 2017 bench-



Fig. 6: The convergence process of the mean fitness errors derived from JADE, SinDE, TSDE, AGDE, EFADE DMIE-DE on 100D benchmark functions f_1 , f_5 , f_7 , f_8 , f_{16} , f_{21} .

mark test suite with different characteristics. The rest convergence figures for solving other functions could be found in Figs. A1-A3 of supplementary material file. From these figures, we can see that DMIE-DE has the best performance among these competitors in terms of the convergence speed and the final solution accuracy in most cases, which proves the efficiency of balancing global exploration and local exploitation with dual mutation strategies and inferior solution eliminating technique.

5.4 Technique Validity and Parameter Sensitivity Analysis

In order to testify the validity of the proposed dual mutation strategies (i.e., DE/current-to-embest and DE/ebest-to-rand) and the inferior solution eliminating technique in DMIE-DE, comparative experiments are carried out by designing three versions of DMIE-DE, which are listed as follows:

- DMIE-DE1: DMIE-DE with only the mutation DE/current-to-*em*best and the inferiors eliminating technique;
- DMIE-DE2: DMIE-DE with only the mutation DE/ebest-to-rand and the inferiors eliminating technique;
- DMIE-DE3: DMIE-DE with both the mutation
 DE/current-to-embest and DE/ebest-to-rand but
 without the inferiors eliminating technique.

The experiment is also based on the 29 functions from CEC 2017 test suite, and each function is tested for 50 independent runs to obtain the solutions. Detail statistic results of 50D functions are presented in Table 6, meanwhile the results of 30D and 100D functions are listed in the supplementary as Table A3 and Table A4 respectively. In these tables, the best solutions obtained by DMIE-DE and three versions are shown in boldface. From the statistic results, we can observe that DMIE-DE is superior to its three versions by obtaining the smallest function error values in most cases.

The multi-problem analysis results based on the Wilcoxon test at the 0.05 significance level are shown in Table 7. The results indicate that DMIE-DE outperforms other versions with obtaining larger R^+ than R^- compared to all versions in 30D, 50D and 100D functions. What's more, based on the asymptotic *p*-values, we have 95% probability to ensure that there is significant difference between DMIE-DE and its three versions except for 30D DMIE-DE1. The average ranking results by Friedman test is listed in Table 8, from which we can also observe that DMIE-DE remains the first place for three kinds of dimensions.

To compare the convergence rate of three versions with DMIE-DE, several typical functions are selected here from different dimensions for the test, and their convergence graphs are plotted in Fig. 7. The rest convergence figures are attached in Figs.A4-A6 of supplementary material file. From Fig. 7, we can observe that DMIE-DE2 has the poorest performance in terms of the accuracy of final solutions. The reason for this phe-

	50D test	function	ons in Cl	EC2017	over 50	maeper	ident rui	is.	
	DMIE	-DE1	DMIE	-DE2	DMIE	-DE3	DMIE-DE		
Func.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	
f_1	2.98E-14	1.12E-14	1.04E-13	1.30E-13	2.33E-14	8.00E-15	2.33E-14	8.00E-15	
f_3	4.05E-12	9.32E-12	1.12E-12	7.61E-13	6.16E-11	1.09E-10	7.71E-12	1.94E-11	
f_4	$8.73E{+}01$	3.96E + 01	5.56E+01	5.66E + 01	$8.80E{+}01$	4.54E + 01	7.35E+01	4.73E+0	
f_5	5.67E + 01	$1.59E{+}01$	6.98E+01	1.67E+01	6.22E + 01	1.52E + 01	5.45E+01	1.70E+0	
f_6	3.08E-02	2.83E-02	8.61E-02	9.75E-02	4.86E-03	6.99E-03	1.04E-04	6.84E-04	
f_7	1.03E+02	6.40E + 00	1.37E+02	2.09E+01	9.86E + 01	1.27E + 01	9.31E+01	9.23E+0	
f_8	4.29E + 01	$6.72E{+}00$	7.38E+01	1.60E+01	5.36E + 01	$1.59E{+}01$	5.25E+01	2.12E+0	
f_9	5.33E+00	3.28E + 00	3.57E+01	2.15E+01	$3.05E{+}00$	4.37E + 00	4.84E-01	5.52E-01	
f_{10}	5.33E + 03	$4.41E{+}02$	5.46E+03	1.07E+03	5.23E + 03	$1.35E{+}03$	6.65E+03	1.86E+0	
f_{11}	1.24E+02	$2.40E{+}01$	1.21E+02	3.64E+01	9.88E + 01	$2.30E{+}01$	8.14E+01	2.32E+0	
f_{12}	7.59E+03	6.12E + 03	1.35E+04	1.14E+04	$1.12E{+}04$	$1.14E{+}04$	6.41E+03	5.26E+0	
f_{13}	1.07E+02	$5.11E{+}01$	1.36E+02	8.54E+01	9.07E + 01	$4.50E{+}01$	7.83E+01	3.76E+0	
f_{14}	$6.06E{+}01$	1.70E + 01	1.06E+02	4.20E+01	6.02E + 01	$1.50E{+}01$	5.27E+01	1.21E+0	
f_{15}	1.12E+02	$6.02E{+}01$	1.26E+02	6.99E+01	7.39E+01	2.64E + 01	6.05E+01	1.97E+0	
f_{16}	5.46E + 02	1.84E + 02	9.31E+02	3.41E+02	9.23E + 02	2.03E+02	7.80E+02	2.44E+0	
f_{17}	4.55E+02	$1.45E{+}02$	5.87E+02	2.76E+02	4.42E + 02	$1.93E{+}02$	4.47E+02	2.13E+0	
f_{18}	2.01E+02	1.31E + 02	8.94E+01	5.52E+01	8.41E + 01	3.28E + 01	6.68E+01	3.52E+0	
f_{19}	1.00E+02	3.44E+01	8.22E+01	3.89E+01	5.75E + 01	2.05E+01	4.84E+01	1.75E+0	
f_{20}	2.92E + 02	8.56E + 01	4.94E+02	2.19E+02	3.07E + 02	1.60E + 02	3.57E+02	2.32E+0	
f_{21}	2.52E+02	9.04E + 00	2.74E+02	1.41E+01	2.53E+02	1.36E + 01	2.49E+02	1.04E+0	
f_{22}	6.74E + 03	1.73E+03	5.60E+03	1.59E+03	5.06E + 03	1.66E + 03	6.14E+03	2.40E+0	
f_{23}	4.59E + 02	1.08E+01	5.05E+02	2.19E+01	4.78E + 02	$1.51E{+}01$	4.72E+02	2.26E+0	
f_{24}	5.52E + 02	2.05E+01	5.75E+02	1.81E+01	5.56E + 02	1.64E + 01	5.50E + 02	2.16E+0	
f_{25}	5.21E + 02	$4.12E{+}01$	5.08E+02	3.18E+01	5.14E + 02	$3.30E{+}01$	5.06E+02	3.50E+0	
f_{26}	1.42E + 03	$9.05E{+}01$	1.88E+03	1.97E+02	1.66E + 03	1.66E + 02	1.64E+03	1.39E+0	
f_{27}	5.60E + 02	$3.48E{+}01$	5.58E+02	3.56E+01	5.40E + 02	$2.75E{+}01$	5.38E+02	2.41E+0	
f_{28}	4.87E + 02	$2.60E{+}01$	4.79E+02	2.24E+01	4.77E + 02	$2.39E{+}01$	4.72E+02	2.14E+0	
f_{29}	4.62E+02	$6.39E{+}01$	6.57E+02	2.06E+02	4.92E+02	1.34E+02	4.43E+02	1.27E+0	
f30	6.91E+05	1.14E + 05	6.56E+05	6.77E+04	6.52E + 05	7.86E+04	6.51E+05	5.73E+0	

Table 6: Mean and standard deviation of function error values obtained by DMIE-DE and its three versions on 50D test functions in CEC2017 over 50 independent runs.

Table 7: Results of the multiple-problem Wilcoxon test between DMIE-DE and its three versions at the 0.05 significance level.

Dimension	DMIE-DE VS.	\mathbf{R}^+	\mathbf{R}^{-}	Asymptotic p -value	$\alpha = 0.05$
30D	DMIE-DE1	272.5	162.5	0.228845	NO
	DMIE-DE2	371.0	35.0	0.000119	YES
	DMIE-DE3	300.0	106.0	0.025977	YES
50D	DMIE-DE1	312.0	123.0	0.000004	YES
	DMIE-DE2	373.0	62.0	0.000719	YES
	DMIE-DE3	322.5	83.5	0.005992	YES
	DMIE-DE1	341.0	65.0	0.001562	YES
100D	DMIE-DE2	291.0	115.0	0.043876	YES
	DMIE-DE3	311.5	123.5	0.039389	YES

nomenon is that the designed DE/ebest-to-rand is a relatively exploitative strategy to help DMIE-DE2 have a faster convergence speed, however, this strategy is against the population diversity and is easy to cause premature convergence problem. On the contrary, the mutation DE/current-to-embest in DMIE-DE1 is an explorative strategy which is helpful to maintain the population diversity and keep the population active to find better solutions than DMIE-2, but it's easy to lead to the stagnation problem and fall into local optimum. By contrast, DMIE-DE3 has the best performance among the three versions of DMIE-DE since it combines DE/*e*best-to-rand and DE/current-to-*em*best and the exploitation and exploration strengthes are complementary. However, DMIE-DE3 still needs to be improved due to the dual cost of evaluation times. Thus, the inferior solution eliminating technique plays an important role and helps DMIE-DE has better performance than DMIE-DE3. In summary, the dual mutation strategies and inferior solution eliminating technique play indispensable role in balancing the exploration and exploitation abilities and help DMIE-DE have an outstanding performance.

Table 8: Average ranking of DMIE-DE and its three versions by Friedman test at the 0.05 significance level.

Algorithms	Ranking on $30D$	Ranking on $50D$	Ranking on $100D$
DMIE-DE1	2.2414	2.6724	2.9483
DMIE-DE2	3.5690	3.3966	2.8793
DMIE-DE3	2.4828	2.3966	2.3103
DMIE-DE	1.7069	1.5345	1.8621

As mentioned previously, the inferior solution eliminating technique is proposed for avoiding the waste of function evaluations by eliminating some poorperforming individuals at a certain generation interval.



Fig. 7: The convergence process of the mean fitness errors derived from DMIE-DE and its three versions on several typical 30D, 50D and 100D functions.

The effectiveness of inferior solution eliminating technique is controlled by the parameter gip. In order to confirm the best choice of qip, the experiment is carried out based on CEC2017 functions by setting gip to different values, i.e., 200, 300, 400, 500 and 600 respectively. The experimental results obtained by DMIE-DE with different qip values for solving 50D functions are presented in Table 9, and results of 30D and 100D functions are listed in Table A5 and Table A6 respectively of the supplementary material file. The smallest functions error values are highlighted in bold to indicate the best-performing gip value. Based on these function error values, Friedman test method is employed to draw a comprehensive comparison conclusion. The results are shown in Table 10, from which we can draw the conclusion that gip equaling to 400 is the most appropriate choice for DMIE-DE with obtaining the best ranking results for 30D, 50D and 100D functions. Therefore, based on a comprehensive consideration, we confirm that gip equaling to 400 is the most suitable for the superior performance of DMIE-DE.

6 Conclusion

The performance of DE algorithm is highly depends on the mutation strategies and associated control parameters which are expected to balance the global exploration and local exploitation abilities. Based on this consideration, we propose a dual mutations collaboration mechanism with elites guiding and inferiors eliminating techniques for DE. The dual mutation strategies are guided by elite individuals with different functional emphasis. More specifically, the mutation strategy DE/current-to-embest is designed relatively explorative while the mutation DE/ebest-to-rand is relatively exploitative. During the evolution process, these two mutation strategies are employed simultaneously to achieve the balance between local and global performance. Moreover, a newly designed parameter adaptation method is applied to automatically adjust the parameters F and CR. They are updated according to a Cauchy distribution model and a sine formula respectively, and the updating process is associated with

	on $50D$ test functions in CEC2017 over 50 independent runs.										
	gip =	200	gip =	300	gip =	400	gip =	500	gip =	600	
Func.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	
f_1	3.07E-14	7.79E-15	2.79E-14	8.59E-15	2.33E-14	8.00E-15	2.33E-14	8.50E-15	2.47E-14	8.02E-15	
f_3	2.72E+03	$1.92E{+}04$	2.05E-12	3.73E-12	7.71E-12	1.94E-11	1.20E+02	$8.50E{+}02$	1.67E-11	8.22E-11	
f_4	5.37E + 01	$4.89E{+}01$	6.12E + 01	$5.11E{+}01$	7.35E+01	$4.73E{+}01$	6.57E+01	$5.06E{+}01$	6.39E+01	5.08E+01	
f_5	4.91E+01	$1.60E{+}01$	5.01E + 01	$1.22E{+}01$	5.45E+01	$1.70E{+}01$	5.24E+01	$1.56E{+}01$	4.88E+01	1.44E+01	
f_6	1.10E-04	6.93E-04	4.27E-06	1.21E-05	1.04E-04	6.84E-04	9.57E-05	6.12E-04	4.97E-06	1.79E-05	
f_7	1.04E+02	$2.71E{+}01$	1.06E + 02	$2.85E{+}01$	9.31E+01	$9.23E{+}00$	1.05E+02	$2.93E{+}01$	9.98E+01	2.71E+01	
f_8	4.75E+01	$1.20E{+}01$	5.06E + 01	$1.12E{+}01$	5.25E+01	2.12E+01	5.01E+01	1.41E+01	4.81E+01	1.31E+01	
f_9	4.45E-01	5.68E-01	6.27E-01	8.79E-01	4.84E-01	5.52E-01	5.07E-01	6.35E-01	7.19E-01	9.35E-01	
f_{10}	6.46E + 03	$1.65E{+}03$	6.83E + 03	$1.54E{+}03$	6.65E+03	$1.86E{+}03$	6.39E+03	1.83E+03	6.77E+03	1.77E+03	
f_{11}	8.73E+01	$2.30E{+}01$	8.07E + 01	$2.50E{+}01$	8.14E+01	2.32E+01	8.64E+01	2.21E+01	8.22E+01	2.30E+01	
f_{12}	7.09E+03	4.73E+03	6.62E + 03	$6.18E{+}03$	6.41E+03	5.26E + 03	7.58E+03	$9.32E{+}03$	6.01E+03	5.19E+03	
f_{13}	8.52E+01	$3.37E{+}01$	9.17E + 01	$4.03E{+}01$	7.83E+01	$3.76E{+}01$	8.18E+01	$2.99E{+}01$	8.24E+01	3.06E+01	
f_{14}	5.22E+01	$1.33E{+}01$	5.06E + 01	$9.79E{+}00$	5.27E+01	$1.21E{+}01$	4.81E+01	$8.95E{+}00$	5.11E+01	1.31E+01	
f_{15}	6.86E + 01	$2.61E{+}01$	$6.89E{+}01$	$2.23E{+}01$	6.05E+01	1.97E+01	7.14E+01	$2.83E{+}01$	6.90E+01	2.57E+01	
f_{16}	7.44E+02	3.04E+02	7.07E+02	$2.61E{+}02$	7.80E+02	2.44E+02	7.03E+02	$3.11E{+}02$	7.21E+02	2.49E+02	
f_{17}	4.74E+02	$2.13E{+}02$	4.27E + 02	$1.96E{+}02$	4.47E+02	$2.13E{+}02$	4.57E+02	1.87E+02	4.92E+02	2.09E+02	
f_{18}	7.08E+01	$3.46E{+}01$	$7.19E{+}01$	$4.44E{+}01$	6.68E + 01	$3.52E{+}01$	7.74E+01	$3.96E{+}01$	8.39E+01	3.95E+01	
f_{19}	5.01E+01	$2.35E{+}01$	$4.56E{+}01$	$1.81E{+}01$	4.84E+01	1.75E+01	4.92E+01	$2.38E{+}01$	4.41E+01	1.56E+01	
f_{20}	3.37E + 02	2.02E+02	3.54E + 02	$2.09E{+}02$	3.57E+02	2.32E+02	3.70E+02	$2.17E{+}02$	3.81E+02	2.12E+02	
f_{21}	2.48E + 02	$1.53E{+}01$	2.48E + 02	$1.03E{+}01$	2.49E+02	1.04E+01	2.53E+02	2.24E+01	2.50E+02	2.00E+01	
f_{22}	5.88E + 03	2.48E+03	5.57E + 03	$2.62E{+}03$	6.14E+03	$2.40E{+}03$	5.76E+03	3.01E+03	6.23E+03	2.71E+03	
f_{23}	4.78E+02	$1.96E{+}01$	$4.73E{+}02$	$1.45E{+}01$	4.72E+02	$2.26E{+}01$	4.74E+02	$1.28E{+}01$	4.70E+02	1.60E+01	
f_{24}	5.48E + 02	$1.89E{+}01$	$5.51E{+}02$	$1.98E{+}01$	5.50E+02	$2.16E{+}01$	5.53E+02	$2.33E{+}01$	5.51E+02	2.33E+01	
f_{25}	5.15E+02	$3.54E{+}01$	$5.09E{+}02$	$3.45E{+}01$	5.06E+02	$3.50E{+}01$	5.05E+02	$3.15E{+}01$	5.07E+02	3.35E+01	
f_{26}	1.58E + 03	$1.69E{+}02$	$1.60E{+}03$	$1.47E{+}02$	1.64E+03	$1.39E{+}02$	1.60E+03	1.44E+02	1.60E+03	1.50E+02	
f_{27}	5.39E+02	$1.96E{+}01$	5.40E + 02	$1.73E{+}01$	5.38E+02	2.41E+01	5.38E+02	1.94E+01	5.45E+02	2.72E+01	
f_{28}	4.77E+02	$2.33E{+}01$	$4.79E{+}02$	$2.36E{+}01$	4.72E+02	$2.14E{+}01$	4.82E+02	$2.47E{+}01$	4.79E+02	2.40E+01	
f_{29}	4.51E+02	$1.15E{+}02$	4.27E + 02	$7.90E{+}01$	4.43E+02	1.27E+02	4.61E+02	1.18E+02	4.26E+02	1.22E+02	
f_{30}	6.51E + 05	6.12E + 04	6.57E+05	5.80E + 04	6.51E+05	5.73E+04	6.62E+05	6.83E+04	6.54E+05	7.81E+04	

Table 9: Mean and standard deviation of function error values obtained by DMIE-DE with different gip values on 50D test functions in CEC2017 over 50 independent runs.

Table 10: Average ranking of DMIE-DE with differentgip settings by Friedman test at the 0.05 significancelevel.

Dimension	$gip{=}200$	$gip{=}300$	$gip{=}400$	$gip{=}500$	$gip{=}600$
30D	3.1207	3.3448	2.1552	2.6897	3.6897
50D	2.9138	2.8448	2.7759	3.3966	3.0690
100D	3.0172	2.8276	2.7759	3.0345	3.3448

the successful experience from the previous generations. The inferior solution eliminating technique is supplemented to enhance the convergence speed and compensate the cost for fitness evaluations in the evaluation process. The experiments between DMIE-DE and its three versions prove that the dual mutation strategies and inferior solution eliminating technique play indispensable role in the performance of DMIE-DE. The comparative experiments with five state-ofthe-art DE variants, i.e., JADE, SinDE, TSDE, AGDE, EFADE, on solving 30D, 50D and 100D test functions in CEC2017 benchmark set, demonstrate that the proposed DMIE-DE algorithm is superior to the considered DE variants, especially in higher dimension. Acknowledgements This work is supported by National Natural Science Foundation of China (62176075 and 61906088).

Conflict of interest

The authors declare that they have no conflict of interest.

References

- Abraham A, Jatoth R, Anguluri R (2012) Hybrid differential artificial bee colony algorithm. Journal of Computational and Theoretical Nanoscience 9:249–
- Baioletti M, Di Bari G, Milani A, Poggioni V (2020) Differential evolution for neural networks optimization. Mathematics 8(1):69
- Balamurugan K, Muthukumar K (2019) Differential evolution algorithm for contingency analysis-based optimal location of facts controllers in deregulated electricity market. Soft Computing 23(1):163–179
- Das S, Suganthan PN (2011) Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation 15(1):4–31

- 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
- 56 57 58 59 60 61
- 62
- 62 63
- 64 65

- Das S, Mullick SS, Suganthan PN (2016) Recent advances in differential evolution–an updated survey. Swarm and Evolutionary Computation 27:1–30
- Deng L, Li C, Sun G (2020a) An adaptive dimension level adjustment framework for differential evolution. Knowledge-Based Systems p 106388
- Deng L, Zhang L, Sun H, Qiao L (2020b) DSM-DE: a differential evolution with dynamic speciation-based mutation for single-objective optimization. Memetic Computing 12(1):73–86
- Draa A, Bouzoubia S, Boukhalfa I (2015) A sinusoidal differential evolution algorithm for numerical optimisation. Applied Soft Computing 27:99–126
- Draa A, Chettah K, Talbi H (2019) A compound sinusoidal differential evolution algorithm for continuous optimization. Swarm and Evolutionary Computation 50:100450
- Fu C, Jiang C, Chen G, Liu Q (2017) An adaptive differential evolution algorithm with an aging leader and challengers mechanism. Applied Soft Computing 57:60-73
- García S, Molina D, Lozano M, Herrera F (2009) A
 study on the use of non-parametric tests for analyzing
 the evolutionary algorithms behaviour: a case study
 on the CEC 2005 special session on real parameter
 optimization. Journal of Heuristics 15(6)
- Ghosh A, Das S, Chowdhury A, Giri R (2011) An improved differential evolution algorithm with fitness based adaptation of the control parameters. Information Sciences 181(18):3749 3765
- Guo S, Yang C, Hsu P, Tsai JSH (2014) Improving differential evolution with a successful-parentselecting framework. IEEE Transactions on Evolutionary Computation 19(5):717–730
- Guo SM, Yang CC (2014) Enhancing differential evolution utilizing eigenvector-based crossover operator.
 IEEE Transactions on Evolutionary Computation 19(1):31–49
- Hosny KM, Khalid AM, Mohamed ER (2020) Efficient
 compression of volumetric medical images using legendre moments and differential evolution. Soft Computing 24(1):409–427
- Huang H, Dong K, Yan T, Han B (2020) Tactical maneuver trajectory optimization for unmanned combat aerial vehicle using improved differential evolution.
 Soft Computing 24(8):5959–5970
- Islam SM, Das S, Ghosh S, Roy S, Suganthan PN (2012)
 An adaptive differential evolution algorithm with
 novel mutation and crossover strategies for global
 numerical optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)
 42(2):482–500

- Li X, Ma S, Hu J (2017a) Multi-search differential evolution algorithm. Applied Intelligence 47(1):231–256
- Li Y, Guo H, Liu X, Li Y, Pan W, Gong B, Pang S (2017b) New mutation strategies of differential evolution based on clearing niche mechanism. Soft Computing 21(20):5939–5974
- Li Y, Wang S, Yang B (2020) An improved differential evolution algorithm with dual mutation strategies collaboration. Expert Systems with Applications p 113451
- Liu Z, Wang Y, Yang S, Cai Z (2016) Differential evolution with a two-stage optimization mechanism for numerical optimization. 2016 IEEE Congress on Evolutionary Computation (CEC) pp 3170–3177
- Luo J, Liu Z (2020) Novel grey wolf optimization based on modified differential evolution for numerical function optimization. Applied Intelligence 50(2):468–486
- Luo J, Shi B (2019) A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems. Applied Intelligence 49(5):1982–2000
- Meng Z, Pan JS, Kong L (2018) Parameters with adaptive learning mechanism (PALM) for the enhancement of differential evolution. Knowledge-Based Systems 141:92–112
- Meng Z, Pan JS, Tseng KK (2019) PaDE: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization. Knowledge-Based Systems 168:80–99
- Mohamed AW, Mohamed AK (2019) Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. International Journal of Machine Learning and Cybernetics 10(2):253–277
- Mohamed AW, Suganthan PN (2018) Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation. Soft Computing 22(10):3215–3235
- Mohamed AW, Hadi AA, Jambi KM (2019) Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization. Swarm and Evolutionary Computation 50:100455
- Nunes UM, Faria DR, Peixoto P (2017) A human activity recognition framework using max-min features and key poses with differential evolution random forests classifier. Pattern Recognition Letters 99:21 -31
- Opara KR, Arabas J (2019) Differential evolution: A survey of theoretical analyses. Swarm and evolutionary computation 44:546–558
- Otani T, Suzuki R, Arita T (2013) DE/isolated/1: a new mutation operator for multimodal optimization with differential evolution. International Journal of Machine Learning and Cybernetics 4(2):99–105

16

- Paul S, Das S (2015) Simultaneous feature selection and weighting c an evolutionary multi-objective optimization approach. Pattern Recognition Letters 65:51-59
- Reddy SS, Bijwe P (2019) Differential evolution-based efficient multi-objective optimal power flow. Neural Computing and Applications 31(1):509–522
- Sakr WS, El-Schiemy RA, Azmy AM (2017) Adaptive differential evolution algorithm for efficient reactive power management. Applied Soft Computing 53:336– 351
- Song Y, Ma B, Gao W, Fan S (2019) Medical image edge detection based on improved differential evolution algorithm and prewitt operator. Acta Microscopica 28(1)
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4):341–359
- Su J, Vargas DV, Sakurai K (2019) Attacking convolutional neural network using differential evolution.
 IPSJ Transactions on Computer Vision and Applications 11(1):1
- Tanabe R, Fukunaga A (2013) Success-history based
 parameter adaptation for differential evolution. In:
 2013 IEEE congress on evolutionary computation,
 IEEE, pp 71–78
- Tarkhaneh O, Shen H (2019) An adaptive differential
 evolution algorithm to optimal multi-level thresholding for mri brain image segmentation. Expert Systems with Applications 138:112820
- Thangaraj R, Pant M, Abraham A, Bouvry P (2011)
 Particle swarm optimization: hybridization perspectives and experimental illustrations. Applied Mathematics and Computation 217(12):5208–5226
- Wu G, Mallipeddi R, Suganthan PN (2017) Prob-40 lem definitions and evaluation criteria for the cec 41 42 2017 competition on constrained real-parameter opti-43 mization. National University of Defense Technology, 44 Changsha, Hunan, PR China and Kyungpook Na-45 tional University, Daegu, South Korea and Nanyang 46 Technological University, Singapore, Technical Re-47 port 48
- Yu WJ, Shen M, Chen WN, Zhan ZH, Gong YJ, Lin
 Y, Liu O, Zhang J (2013) Differential evolution with
 two-level parameter adaptation. IEEE Transactions
 on Cybernetics 44(7):1080–1099
- Zheng LM, Zhang SX, Tang KS, Zheng SY (2017) Dif ferential evolution powered by collective information.

Information Sciences 399:13–29

Zhu W, Tang Y, an Fang J, Zhang W (2013) Adaptive population tuning scheme for differential evolution. Information Sciences 223:164 – 191

64 65

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

• SupportingDocument.pdf