

# New mutation strategies of differential evolution based on clearing niche mechanism

Yanan Li<sup>1,2</sup>  · Haixiang Guo<sup>1,2,3,4</sup> · Xiao Liu<sup>1,2</sup> · Yijing Li<sup>1,2</sup> · Wenwen Pan<sup>1,2</sup> · Bing Gong<sup>5</sup> · Shaoning Pang<sup>6</sup>

Published online: 29 August 2016  
© Springer-Verlag Berlin Heidelberg 2016

**Abstract** Although differential evolution (DE) algorithms have been widely proposed for tackling various of problems, the trade-off among population diversity, global and local exploration ability, and convergence rate is hard to maintain with the existing strategies. From this respective, this paper presents some new mutation strategies of DE by applying the clearing niche mechanism to the existing mutation strategies. Instead of using random, best or target individuals as base vector, the niche individuals are utilized in these strate-

gies. As the base vector is from a subpopulation, which is made up of the best individuals in each niche, the base vector can be guided by the global or local best ones. This mechanism is beneficial to the balance among population diversity, search capability, and convergence rate of DE, since it can both enhance the population diversity and search capability. Extensive experimental results indicate that the proposed strategies based on clearing niche mechanism can effectively enhance DE's performance.

---

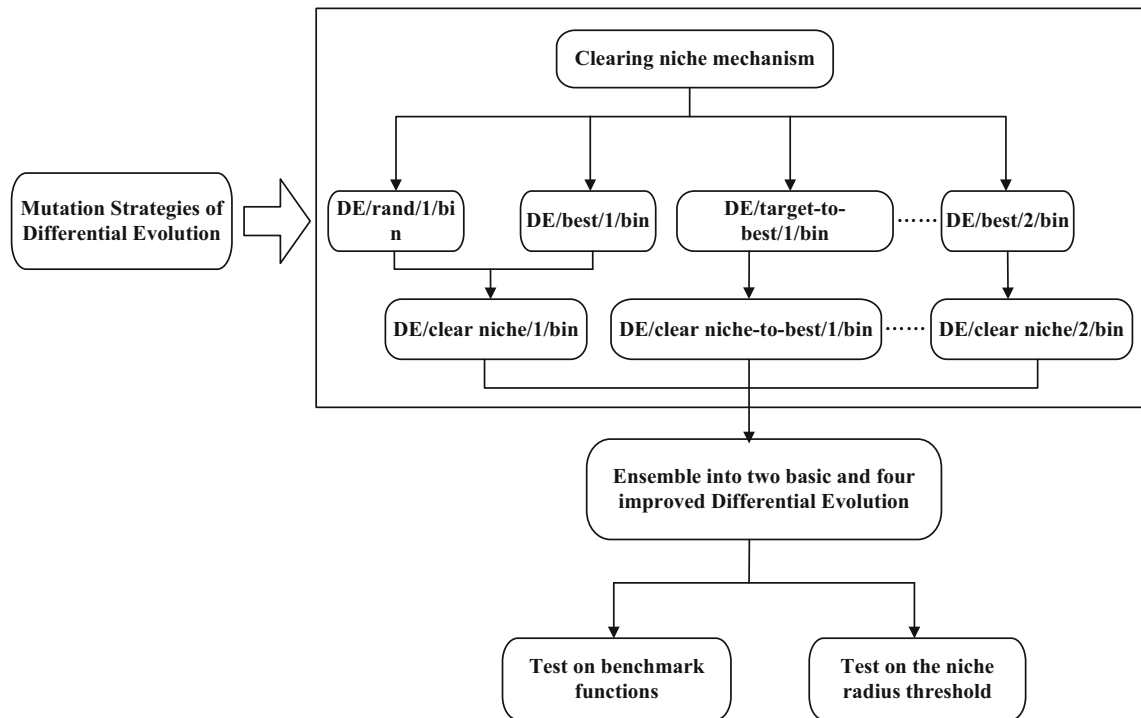
Communicated by Y. Jin.

---

✉ Haixiang Guo  
faterdumk0732@sina.com

- <sup>1</sup> College of Economics and Management, China University of Geosciences, Wuhan 430074, China
- <sup>2</sup> Research Center for Digital Business Management, China University of Geosciences, Wuhan 430074, China
- <sup>3</sup> Mineral Resource Strategy and Policy Research Center of China University of Geosciences, Wuhan 430074, China
- <sup>4</sup> The Joseph M. Katz Graduate School of Business, University of Pittsburgh, Pittsburgh, PA 15260, USA
- <sup>5</sup> Department of Industrial Engineering, Business Administration and Statistic E.T.S Industrial Engineering, Universidad Politécnica de Madrid C/José Gutiérrez Abascal 2, 20086 Madrid, Spain
- <sup>6</sup> Department of Computing, Unitec Institute of Technology, Private Bag 92025, Auckland, New Zealand

## Graphical Abstract



**Keywords** Differential evolution · Niche · Clearing mechanism · Niche radius

## 1 Introduction

The differential evolution (DE) (Storn and Price 1995, 1996) algorithm was proposed by Price and Storn in 1995 and has been a very competitive form of evolutionary computing afterward. As a simple powerful search technique, DE is always employed for solving complex continuous nonlinear functions. With a random population by initializing solutions, the DE algorithm employs mutation and crossover operators to generate new candidate solutions, and utilizes a simple selection operator to determine whether the offspring should replace their parents in the next generation. Compared with most other evolutionary algorithms (EAs), DE is simpler and much easier to be implemented. Moreover, the gross performance of DE in terms of accuracy, fast convergence speed and robustness makes it as an attractive algorithm to be applied on various real-world optimization problems. In addition, the number of control parameters in DE is very few (mutation control parameter, also called scaling factor, crossover control parameter, also called crossover rate and population size in classical DE). Therefore, the DE algorithm has gained much attention with successful applications in data mining (Zhu et al. 2012; Tvrdík and Křivý 2015), scheduling (Mokhtari and Salmasnia 2015), construction engineering (Ho-Huu et al. 2015), pattern recognition

(Secmen and Tasgetiren 2013), signal processing (Sheniha 2013), chemical engineering (Sharma and Rangaiah 2013), power system (Zhang et al. 2015), image processing (Ali et al. 2014), and in other domains (Al-Dabbagh et al. 2014; Rakshit and Konar 2015; Das and Prasad 2015).

Nevertheless, the performance (Liu and Lampinen 2002) of the DE algorithm is sensitive to the mutation strategy, which exists many different DE trial vector generation strategies and respective control parameters such as the population size ( $NP$ ), crossover rate ( $Cr$ ) and the scale factor ( $F$ ). In various search phases of the evolution process, these strategies often possess different searching capabilities. And, the best settings of the control parameters vary for different optimization problems, and for different requirements on consumption time and accuracy when the optimization problem is same. Therefore, it is necessary to find the most appropriate strategy and its corresponding parameters. To achieve this, however, a process of trial-and-error search is needed to be performed, which definitely will suffer a high computation cost. Computational costs may be induced by the fact that the population of DE is evolved through different regions in the search space, within which different strategies and respective different parameter settings (Qin et al. 2009). Recently, various offspring generation strategies and parameter adaptation mechanisms have been developed to enhance the reliability and robustness of DE. For example, Brest and Mernik (2008) presented the jDE algorithm, which owned self-adapting parameters and by encoding the parameters into each individual and adapting them by means of evo-

lution, and obtained the mutation strategy as DE/rand/1. [Qin and Suganthan \(2005\)](#) presented a self-adaptive variant of DE (SaDE), where trial vector generation strategies were gradually self-adapted by learning from their prior experiences in generating promising solutions. [Wang et al. \(2011\)](#) proposed a systematic framework for combining different trial vector generation strategies, called composite DE (CoDE), in which three well-studied offspring generation strategies are coupled with three-parameter settings randomly to generate trial vectors in the following way. [Mallipeddi et al. \(2011\)](#) proposed an ensemble of mutation strategies and parameter values for DE (EPSDE). Mutation strategies from mutation strategies pool can involve corresponding parameters from control parameters' values pool to produce their own offspring. Then, the optimized offspring can be obtained by further competing. [Zhang and Sanderson \(2009\)](#) presented JADE algorithm, which implemented a new mutation strategy DE/target-to-pbest/1 and updated control parameters in an adaptive manner. Based on JADE, [Brown et al. \(2015\)](#) proposed a new DE with small population, namely  $\mu$ JADE. In  $\mu$ JADE, a new mutation strategy DE/current-by-rand-to-pbest/1 is introduced. [Kundu et al. \(2014\)](#) presented a modified semi-adaptive DE, namely MSeDE. In MSeDE, a new mutation scheme DE/current-to-constr\_best/1 and a new crossover scheme p-BCX are used. [Yu et al. \(2014\)](#) presents an adaptive DE algorithm, namely ADE. In ADE, a new mutation strategy DE/lbest/1 and a two-level adaptive parameter control scheme are used. The new strategy is a variant of DE/best/1 strategy, which multiple locally best individuals instead of one globally and the two-level adaptive parameter control scheme includes population-level and individual-level parameter control. [Mohamed \(2015\)](#) presented an improved DE algorithm, namely IDE. In IDE, a new triangular mutation strategy based on the convex combination vector of the triplet, which is defined by three randomly chosen vectors and the difference vector between the best and the worst individuals among the three randomly selected vectors is introduced. [Liu et al. \(2014\)](#) proposed a random-based differential evolution with neighborhood mutation, namely NRDE. In NRDE, two mutation schemes are used. The mutation schemes are random-based mutation scheme and neighborhood mutation scheme, the best vector of which will be replace the target vector. [Han et al. \(2013\)](#) presented a differential evolution with local information, namely DELI. In DELI, considering both global information and local information, a new mutation operation is applied to generate a mutated individual. In order to get an appropriate combination of strategies and control parameters for different problems, many other adaption techniques have been developed ([Tang et al. 2015](#); [Wang et al. 2014](#); [Cai and Wang 2015](#); [Mallipeddi and Lee 2015](#)). Although different partial adaptation schemes have been proposed to overcome the trial-and-error procedure, many

strategies are hard to maintain the balance between the global exploration and the local exploitation. Here, we demonstrate the superior performance of the proposed mutation strategy with various DE algorithms based on niche on maintaining the balance between the global exploration and the local exploitation.

In the paper, the clearing niche method ([Petrowski 1996](#)) is integrated with mutation strategies to enhance population diversity, improve the search ability, and accelerate the convergence rate of DE algorithms. The niche techniques ([Petrowski 1996](#); [Mahfoud 1995](#); [Yin and Gernay 1993](#); [Petrowski and Genet 1999](#); [Sareni and Krahenbuhl 1998](#)) are regarded as effective methods to maintain the balance between both population diversity and the search domain. They aim at gathering the individuals on several peaks of fitness function in the population according to genetic likeness, and then permit DE to investigate those peaks in parallel. The individual with a high fitness in the niche is keeping its fitness, while the others in the niche are changed to reduce their fitness values sharply. Hence, the individuals in the population may be dispersed into the whole search space. Thus, some diversity can be maintained effectively during the generations in the population. Several niche techniques have been proposed, such as crowding methods ([Mahfoud 1995](#)), clustering-based methods ([Yin and Gernay 1993](#)), speciation tree methods ([Petrowski and Genet 1999](#)), fitness-sharing methods ([Sareni and Krahenbuhl 1998](#)), and clearing methods ([Petrowski 1996](#)). Also, some niche-based DE algorithms have been presented. [Thomsen \(2004\)](#) presented a DE algorithm based on crowding and fitness-sharing scheme to tackle multimodal optimization, namely CrowdingDE and SharingDE. Different from conventional DE, CrowdingDE modifies the selection operation, in which the offspring replaces the most similar individual among a (the crowding factor) subset of the population. In the selection operation of SharingDE, all parents and offsprings are added to the population enlarging the population, and all the individuals are rescaled using the sharing function, then the individuals are sorted with the new fitness and the worst half of the population will be removed. [Li \(2005\)](#) extended DE with the notion of speciation for solving multimodal optimization problems, namely SDE. SDE locates multiple global optima simultaneously through the adaptive formation of multiple species. Each species is evolved by its own DE process, which tries to successively improve itself. [Efitropakis et al. \(2012\)](#) proposed two new DE mutation strategies, namely DE/nrand/1 and DE/nrand/2. In DE/nrand/1 and DE/nrand/2, the local information from the current population is incorporated into the mutation schemes, when each individual is evolved by applying its nearest neighbor individual as a base vector. Based on DE/nrand/1, [Efitropakis et al. \(2013\)](#) proposed a new niching DE algorithm with dynamic archive to overcome the population size influence and produce good

performance almost independently of its population size, namely dADE/ncrand/1 algorithm, which involves in incorporation between a control parameter adaptation technique and an external dynamic archive along with a re-initialization mechanism. The control parameter adaptation technique, proposed in the context of JADE algorithm (Zhang and Sanderson 2009), is designed to efficiently adapt the control parameters of the algorithm. Meanwhile, the external dynamic archive along with a re-initialization mechanism aims to alleviate the problem, that is, have to tune the population size and allow the algorithm to maintain good performance regardless of the population size value. The control parameter adaptation technique incorporates a dynamic archive proposed in Zhai and Li (2011), aiming to record good solutions found along with a re-initialization procedure to continue searching for additional good solutions in unexplored regions of the search space. Biswas et al. (2014) presents a DE variant with parent centric mutation that makes use of normalized search neighborhood and integrates with the proximity-based crowding technique, namely PNPCE. The PNPCE does not make use of the problem-dependent niching parameters (like niche radius), which are hardly determined the values. The mutation operator helps to maintain the population diversity at an optimum level by using well-defined local neighborhoods. Zhang et al. (2015) presented a DE with dynamic niche radius strategy, namely DNRDE. In DNRDE, the niche radius is adjusted dynamically to make the algorithm navigate from global exploration to local exploitation by a new two-stage annealing schedule. At first stage, exploration dominates the search process, and the radius decrease dynamically. When the niche radius reaches to a cutoff value, it will be stable. At second stage, exploitation takes over to enhance the quality of the acquired optima.

In this paper, the clearing niche method is used in the mutation strategy. Compared with other niche techniques, such as speciation tree and sharing fitness methods, the clearing niche method may maintain the population diversity effectively with a lower population size, and is also simpler to implement. However, in contrast to the speciation tree method, the clearing niche method needs a problem-dependent parameter, namely, the niche radius. In such case, the DE algorithm will be sensitive to niche radius, since its performance will be altered by changing the setting of niche radius, this impact of radius changes on algorithms will be demonstrated by numerical experiment in this paper.

The paper is organized as follows. In Sect. 2, we introduce the basic differential evolution and several well performance DE variants. In Sect. 3, we describe the proposed mutation strategy in details, and we introduce the change mutation strategies in four state-of-the-art DE variants. In Sect. 4, the paper lists the functions, and their corresponding simulated diagrams. In Sect. 5, numerical experiments are presented. In Sect. 6, the concluding remarks are contained.

## 2 Differential evolution algorithms

In this section, we describe the basic differential evolution and several variants of DE.

### 2.1 Brief description of basic differential evolution

There are several strategies of DE that are proposed in the literature (Zaharie 2009), one of which DE/rand/1/bin is used widely. Accordingly, we choose DE/rand/1/bin as example to introduce DE. Just like other Evolution Algorithms (EAs), DE uses mutation and crossover to generate new individuals. DE has four basic operations involving in initialization, mutation, crossover, and selection. The whole flow chart of DE is shown in Fig. 1. In DE-literature, a parent vector from the current generation is called target vector, a mutant vector obtained through the differential mutation operation is known as donor vector, and finally an offspring formed by recombining the donor with the target vector is called trial vector. The details of DE operations are described as follows.

(1) Coding. DE is a global optimization algorithm, and individuals in population are encoded using real number.

(2) Individual.  $NP$  denotes size of the population in DE. The  $i$ th individual at  $G$ th generation is denoted by  $X_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \dots, x_{D,i,G}]$ , where  $D$  is dimension.

(3) Initializing population. The initial population (at  $G = 0$ ) should cover the entire search space as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum bounds:  $x_{j,\min} \in X_{\min} = \{x_{1,\min}, x_{2,\min}, \dots, x_{D,\min}\}$  and  $x_{j,\max} \in X_{\max} = \{x_{1,\max}, x_{2,\max}, \dots, x_{D,\max}\}$ . Hence the  $j$ th component of the  $i$ th individual should be initialized as  $x_{j,i,0} = x_{j,\min} + rand_{i,j}(0, 1) \times (x_{j,\max} - x_{j,\min})$ .

(4) Mutation. Differential idea is embodied into mutation operation. According to the strategy DE/rand/1/bin, the gen-

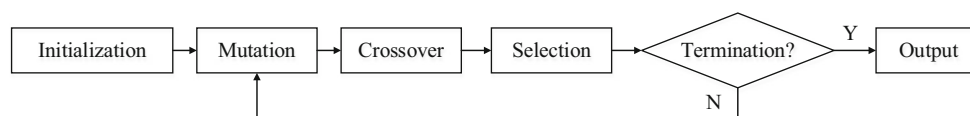


Fig. 1 Basic flowchart of DE

eral process of mutation is expressed by Eq. (1):

$$V_{i,G} = X_{r_1,G} + F \times (X_{r_2,G} - X_{r_3,G}) \tag{1}$$

where  $i$  is the  $i$ th individual vector of current generation.  $X_{r_1,G}$ ,  $X_{r_2,G}$  and  $X_{r_3,G}$  are other three individuals, which are from current generation, where  $i \neq r_1 \neq r_2 \neq r_3$ .  $V_{k,G}$  is donor vector.  $F \in [0, 1]$  is mutation control parameter.

There are usually following differential evolution strategies when crossover operation is bin:

- DE/rand/1/bin:  $V_{i,G} = X_{r_1,G} + F \times (X_{r_2,G} - X_{r_3,G})$
- DE/best/1/bin:  $V_{i,G} = X_{best,G} + F \times (X_{r_1,G} - X_{r_2,G})$
- DE/target-to-best/1/bin:  $V_{i,G} = X_{i,G} + F \times (X_{best,G} - X_{i,G}) + F \times (X_{r_1,G} - X_{r_2,G})$
- DE/best/2/bin:  $V_{i,G} = X_{best,G} + F \times (X_{r_1,G} - X_{r_2,G}) + F \times (X_{r_3,G} - X_{r_4,G})$
- DE/rand/2/bin:  $V_{i,G} = X_{r_1,G} + F \times (X_{r_2,G} - X_{r_3,G}) + F \times (X_{r_4,G} - X_{r_5,G})$

where  $X_{best,G}$  means the best individual in the  $G$ th generation.  $X_{i,G}$  means the  $i$ th individual in the  $G$ th generation.  $X_{r_1,G}$ ,  $X_{r_2,G}$ ,  $X_{r_3,G}$ ,  $X_{r_4,G}$ , and  $X_{r_5,G}$  are other five individuals, which are from current generation, where  $i \neq r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5$ .

(5) Crossover. Crossover probability is  $Cr \in [0, 1]$ . Crossover means to swap the dimensions between the donor vector and the target vector controlled by crossover parameter  $Cr$ . Binomial crossover and exponential crossover are two different crossover strategies. Mahfoud (1995) analyzed the influence of crossover on the behavior of DE. The numerical experiments illustrate that main difference between them is the different distributions of the number of mutated components. Theoretical analysis shows that the behavior of exponential crossover variants is more sensitive to the problem dimension than binomial crossover variants'. Usually the binomial crossover is accepted, which is described as Eq. (2):

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if } (rand_{i,j}(0, 1) \leq Cr \text{ or } j = j_{rand}) \\ x_{j,i,G}, & \text{otherwise} \end{cases} \tag{2}$$

where  $j$  is dimension,  $i$  is individual,  $G$  is generation.  $v_{j,i,G}$  is coming from donor vector  $V_{k,G}$ ,  $x_{j,i,G}$  is coming from target vector  $X_{k,G}$ .  $j_{rand} \in [1, 2, \dots, D]$  is a randomly chosen index.

(6) Selection. The offspring or trial vector  $X_{i,G+1}$  can be obtained through comparing the fitness value of trial vector  $U_{i,G}$  and target vector  $X_{i,G}$  according to Eqs. (3) and (4).

$$X_{i,G+1} = U_{i,G} \quad \text{if } f(U_{i,G}) \leq f(X_{i,G}) \tag{3}$$

$$X_{i,G+1} = X_{i,G} \quad \text{if } f(U_{i,G}) > f(X_{i,G}) \tag{4}$$

## 2.2 Some variants of DE

(1) The jDE algorithm

Brest and Mernik (2008) proposed the jDE algorithm, in which the control parameters  $F$  and  $Cr$  are encoded into the individual  $X_{i,G} = (\bar{x}_{i,G}, F_{i,G}, Cr_{i,G})$  and adjusted by two new arguments  $\tau_1$  and  $\tau_2$ . They are calculated independently, as shown in Eqs. (5), (6)

$$F_{i,G+1} = \begin{cases} F_{l,G} + rand_1 \times F_{u,G}, & \text{if } rand_2 < \tau_1 \\ F_{i,G}, & \text{otherwise} \end{cases} \tag{5}$$

$$Cr_{i,G+1} = \begin{cases} rand_3, & \text{if } rand_4 < \tau_2 \\ Cr_{i,G}, & \text{otherwise} \end{cases} \tag{6}$$

where  $rand_k, k \in \{1, 2, 3, 4\}$  are uniformly distributed random values belonging to the range  $[0, 1]$ ;  $\tau_1$  and  $\tau_2$  are constant values that represent the probabilities of parameters being adjusted;  $F_{u,G}$  and  $F_{l,G}$  are also constant values, which denote the upper and lower bounds of the parameters, respectively. The newly generated  $F_{i,G+1}$  and  $Cr_{i,G+1}$  are procured before the mutation is implemented. Thus, the scheme influences the mutation, crossover and selection operations of the new vector.

(2) The JADE algorithm

Zhang and Sanderson (2009) proposed the JADE algorithm, in which a novel mutation strategy and an optional external archive are utilized to provide information of progress direction. This DE/target-to-best strategy uses multiple best solutions to balance the greediness of the mutation and the diversity of the population, which is generated using Eq. (7):

$$V_{i,G} = X_{i,G} + F_{i,G} \times (X_{best,G}^P - X_{i,G}) + F_{i,G} \times (X_{r_1,G} - \tilde{X}_{r_2,G}) \tag{7}$$

where  $X_{best,G}^P$  is the individual that is randomly selected from the top  $100p\%$  of the current population, with  $p \in (0, 1]$ .  $X_{i,G}$ ,  $X_{best,G}^P$  and  $X_{r_1,G}$  are chosen from the current population  $P$ .  $\tilde{X}_{r_2,G}$  is randomly selected from the union,  $P \cup A$ , while  $A$ , an archive, is employed to store the recently explored inferior solutions. The archive operation is made very simply to avoid significant computation overhead. The archive is initiated to be empty. Then, after each generation, the parent solutions that fail in the selection process are added to the archive. If the archive size exceeds a certain threshold, say  $NP$ , then some solutions are randomly removed from the archive to keep the archive size at  $NP$ .  $F_{i,G}$  denotes the scaling factor and  $Cr_{i,G}$  denotes the crossover rate associated with the  $i$ th individual and they are updated dynamically in



each generation, as shown in Eqs. (8), (9).

$$F_{i,G} = \text{rand}c_i(u_F, 0.1) \quad (8)$$

$$Cr_{i,G} = \text{rand}n_i(u_{Cr}, 0.1) \quad (9)$$

$F_{i,G}$  and  $Cr_{i,G}$  are generated according to a Normal distribution and a Cauchy distribution with associated mean values  $u_F$  and  $u_{Cr}$ . The proposed two location parameters are initialized to be 0.5 and then updated at the end of each generation according to Eqs. (10), (11).

$$u_F = (1 - c) \times u_F + c \times \text{mean}_L(S_F) \quad (10)$$

$$u_{Cr} = (1 - c) \times u_{Cr} + c \times \text{mean}_A(S_{Cr}) \quad (11)$$

where  $c$  is a positive constant in the range(0,1);  $S_F$  and  $S_{Cr}$  denote the set of all successful mutation/crossover rates;  $\text{mean}_A(\cdot)$  indicates the usual arithmetic mean and  $\text{mean}_L(\cdot)$  returns the Lehmer mean shown as Eq. (12).

$$\text{mean}_L(S_F) = \frac{\sum_{i=1}^{|S_F|} F_i^2}{\sum_{i=1}^{|S_F|} F_i} \quad (12)$$

How to get archive  $A$ ,  $S_F$  and  $S_{Cr}$ .

In the selection process,  $A$ ,  $S_F$  and  $S_{Cr}$  will be got.

If  $f(X_{i,G}) \leq f(U_{i,G})$   $X_{i,G+1} = X_{i,G}$

Else  $X_{i,G+1} = U_{i,G}$ ;  $X_{i,G} \rightarrow A$ ;  $Cr_i \rightarrow S_{Cr}$ ,  $F_i \rightarrow S_F$

### (3) The SaDE algorithm

Qin and Suganthan (2005) presented the SaDE, where one trial vector generation strategy was chosen from the candidate pool(“DE/rand/1”, “DE/rand/2”, “DE/target-to-best/2” and “DE/target-to-rand/1”), according to the probability learned from its success rate in generating promising solutions within a certain number of previous generations, called the learning period( $LP$ ). More specifically, these probabilities are initially equal and then gradually self-adapted upon Eq. (13)

$$P_{k,G} = \frac{S_{k,G}}{\sum_{k=1}^K S_{k,G}} \quad (13)$$

where  $P_{k,G}$ ,  $k = 1, 2, \dots, K$ , denotes the probability of applying the  $k$ th strategy. Here  $K$  is the total number of strategies contained in the pool.  $S_{k,G}$  is the success rate of the trial vector, which is generated by the  $k$ th strategy and successfully enters the next generation according to Eq. (14):

$$S_{k,G} = \frac{\sum_{g=G-LP}^{G-1} nS_{k,g}}{\sum_{g=G-LP}^{G-1} nS_{k,g} + \sum_{g=G-LP}^{G-1} nF_{k,g}} + \varepsilon \quad (14)$$

where  $nS_{k,g}$  and  $nF_{k,g}$  record the number of trial vectors generated by the  $k$ th strategy that are retained or discarded in the selection operation in the last  $LP$  generations. The small constant value  $\varepsilon = 0.01$  is used to avoid the possible null success rates. At each generation, for each solution in the current population, the parameters  $F_{i,k}$  and  $Cr_{i,k}$  are independently calculated upon Eqs. (15), (16).

$$F_{i,k} = \text{rand}n_i(0.5, 0.3) \quad (15)$$

$$Cr_{i,k} = \text{rand}n_i(Cr_{m_k}, 0.1) \quad (16)$$

where coefficients are respectively generated for each individual by sampling their values from a normal distribution. Nevertheless, the mean value of  $Cr$  ( $Cr_{m_k}$ ) is gradually self-adapted on the basis of a success rule.

### (4) The EPSDE algorithm

Mallipeddi et al. (2011) provided the EPSDE involving a pool of mutation strategies along with various combinations of parameters, which are employed for competing in order to produce successful offspring population. More concretely, the pool of strategies is formed with three schemes with diverse characteristics. The pool of  $Cr$  values is taken in the range [0.1, 0.9] in steps of 0.1, and the pool of  $F$  values is assigned in the range [0.4, 0.9] in steps of 0.1 as follows:

A pool of mutation strategies includes DE/best/2, DE/rand/1 and DE/target-to-rand/1;

A pool of  $F$  values includes  $F = [0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$

A pool of  $Cr$  values includes  $Cr = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$ .

In EPSDE, each individual of the initial population is randomly adopted with a mutation strategy and associated parameter values taken from the respective pools. When the trial vector performs better than the target vector, the combination of the mutation strategy and parameter values can be survived in the next generation. And the combination is also stored. Afterward, the target vector should be randomly reinitialized with a new mutation strategy from the respective pools or from the successful combinations stored with equal probability when the trial vector performs poorer.

## 3 Differential evolution algorithm based on niche

In this section, we propose some new mutation strategies which are based on clearing niche mechanism. It will be explained in the following sections.

### 3.1 The niche operation based on clearing mechanism

The main idea of the niche algorithm based on clearing mechanism is that the population is divided into a number of niches. And each subpopulation contains a dominant individual: the one that has the best fitness. If the dissimilarity between an individual and the dominant one in a given subpopulation is less than a threshold  $\delta$ : the clearing radius, this individual, then, belongs to this subpopulation. The basic clearing algorithm preserves the fitness of the dominant individual, but resets the fitness of all the other individuals of the same subpopulation. Consequently, the clearing procedure makes the whole resource of a niche belong to a single individual: the winner. The winner takes all rather than sharing resources with the other individuals of the same niche, and other individuals as losers will take a punishment mechanism, this mechanism differs from the fitness-sharing methods. It is also possible to generalize the clearing algorithm by accepting several winners in each niche. The capacity of a niche is defined as the maximum number of winners that this niche can accept. We assume that the population size is  $NP$ , the individuals which perform better in each niche are winners, in contrast, perform worse are loser. Finally, process of flow is shown in Fig. 2.

### 3.2 Mutation strategies based on clearing niche

We all know that the mutation strategies play an important role in the search capability and convergence rate of DE. However, many mutation strategies are hard to maintain a balance among a good population diversity, a good global exploration ability, a good local exploitation ability, and a fast convergence rate. Clearly, from the equation of the basic mutation strategy DE/rand/1/bin, it can be seen that three vectors are randomly chosen, one of which is the base vector. As a result, DE/rand/1/bin is able to maintain the population diversity and the global search ability, but it cannot guarantee the local search ability and the convergence rate. From the equation of the basic mutation strategy DE/best/1/bin, it can be observed that the base vector is the globally best vector and the other two vectors are randomly chosen. In this way, all the vectors are guided by the best vector. Such greedy strategy is helpful for the local search ability and the convergence rate. However, the greedy strategy may lose its population diversity and global search ability. In order to maintain a balance among the population diversity, the global exploration ability, the local exploitation ability, and the convergence rate, we present some new mutation strategies, which combine with the clearing niche mechanism.

In the proposed strategies, the population is divided into some niches according to the sort of the fitness values. Each niche contains several winners whose fitness values are better than others. And all the winners will make up a new subpop-

ulation. Note that the number of niches may be change and the number of winners in each niche will be kept unchanged during the evolution of algorithms. As aforementioned, the mutation strategy DE/best/1 obtains a best vector as the base vector. In the new mutation strategies, the base vector is selected from the group which the subpopulation instead of the entire population. Therefore, all the vectors are guided by the subpopulation which is made of several locally best vectors rather than the randomly vector or the single globally best vector. At the same time, the difference vectors involved in the mutation strategies are selected from the entire population. In such way can maintain a balance among the population diversity, the global exploration ability, the local exploitation ability, and the convergence rate.

Differing from the existing clearing niche mechanisms, we adopt a new idea of dealing with the distance. Here, it calculates the distances from the best individual and other individuals in current status respectively by using Euclidean distance. Moreover, a different normalization method is proposed to calculate the relative distance based on maximum and minimum distances for each niche. In addition, different niches vary maximum distance and minimum distance. Consequently, the clearing radius is the same whereas the ranges of niches are different. Hence, the number of each niche is not only related to individual density but also related to the maximum distance and minimum distance.

For example, if the population size is set as 50, the dimension is set as 2, the clearing radius is set as 0.3. And the fitness is the minimum value of sum of squares. We can get the two-dimensional drawing of population. Then, we can see the range of each niche in Fig. 3. The figure can be found that the range of each niche is different.

When we combine the clearing niche mechanism with several classical mutation strategies, we set a changed solution. In the changed solution, the target individual will have priority, and the best individual will be considered next. At last, if the mutation strategy do not contain the target or best individual, the rand individual will be changed which does not multiply the scaling factor  $F$ . After modifying, the usual differential evolution strategies DE/rand/1, DE/best/1, DE/target-to-best/1, DE/best/2, and DE/rand/2 can be seen in the following.

DE/rand/1/bin:

$$V_{i,G} = X_{r1,G} + F \times (X_{r2,G} - X_{r3,G}) \quad (17)$$

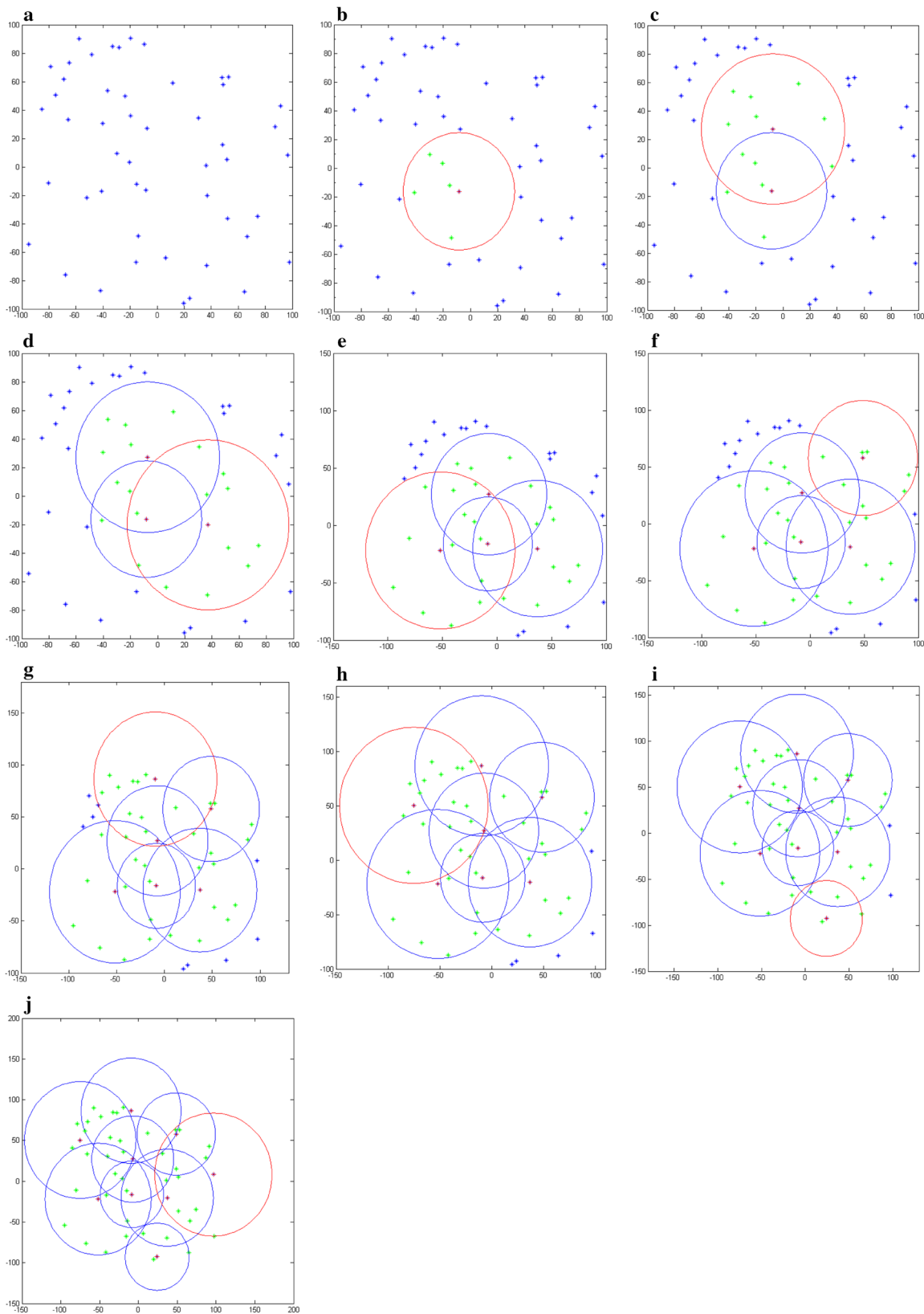
DE/best/1/bin:

$$V_{i,G} = X_{\text{best},G} + F \times (X_{r2,G} - X_{r3,G}) \quad (18)$$

DE/rand/1 based on clearing niche mechanism, we name it DE/clear niche/1. And DE/best/1 based on clearing niche mechanism, we also name it DE/clear niche/1. As illustrated







**Fig. 3** The range of niches. *Note: a–j* are the process which a population is divided into several niches based on the clearing niche mechanism. *Blue “+”* denotes the individuals that have not been divided into niches; *Green “+”* denotes the individuals that have been divided

into niches; *Red “+”* denotes the best individual in each niche; *Red circle* denotes the current niche; *Blue circle* denotes the existing niches (color figure online)

DE/best/2 based on clearing niche mechanism, we name it DE/clear niche /2. And DE/rand/2 based on clearing niche mechanism, we also name it DE/clear niche /2. In DE/clear niche /2, a mutation vector is generated in the following manner:

$$V_{i,G} = X_{c,G} + F \times (X_{r1,G} - X_{r2,G} + X_{r3,G} - X_{r4,G}) \quad (24)$$

where  $X_{c,G}$  is randomly chosen individual from subpopulation which preserves the dominant individual by clearing niche mechanism.  $X_{best,G}$  is the best individual in the current generation.  $X_{r1,G}$ ,  $X_{r2,G}$ ,  $X_{r3,G}$ ,  $X_{r4,G}$ , and  $X_{r5,G}$  are randomly chosen five individuals from the current generation, where  $r1 \neq r2 \neq r3 \neq r4 \neq r5 \neq i$ .  $G$  is the current generation.  $V_{i,G}$  is the  $i$ th donor vector.  $F$  is the scaling factor.

We can see the flowchart of the new DE combined with clearing niche mechanism in Fig. 5 when the mutation strategy is DE/rand/1.

### 3.3 DE variants based on clearing niche

We can see the flowchart of DE with the new mutation strategy based on clearing niche mechanism in Fig. 5. And when the new mutation strategy based on clearing niche mechanism is applied to variants of DE, the corresponding mutation strategy of DE will change. It can be seen in the following section.

#### (1) The jDE algorithm

In jDE algorithm, the mutation strategy is DE/rand/1. We can see change of the mutation strategy as following.

The initial mutation strategy:

$$V_{i,G} = X_{r1,G} + F \times (X_{r2,G} - X_{r3,G}) \quad (17)$$

The mutation strategy based on clearing niche mechanism:

$$V_{i,G} = X_{c,G} + F \times (X_{r1,G} - X_{r2,G}) \quad (19)$$

#### (2) The JADE algorithm

In JADE algorithm, the mutation strategy is DE/target-to-best/1. We can see change of the mutation strategy as following.

The initial mutation strategy:

$$V_{i,G} = X_{i,G} + F \times (X_{best,G}^p - X_{i,G} + X_{r1,G} - \tilde{X}_{r2,G}) \quad (25)$$

The mutation strategy based on clearing niche mechanism:

$$V_{i,G} = X_{c,G} + F \times (X_{best,G}^p - X_{c,G} + X_{r1,G} - \tilde{X}_{r2,G}) \quad (26)$$

#### (3) The SaDE algorithm

In SaDE algorithm, the mutation strategies are DE/rand/1, DE/rand/2, DE/target-to-best/2, DE/target-to-rand/1. We can see change of the mutation strategy as following.

The initial mutation strategy:

$$V_{i,G} = X_{r1,G} + F \times (X_{r2,G} - X_{r3,G}) \quad (17)$$

$$V_{i,G} = X_{r1,G} + F \times (X_{r2,G} - X_{r3,G} + X_{r4,G} + X_{r5,G}) \quad (23)$$

$$V_{i,G} = X_{i,G} + F \times (X_{best,G} - X_{i,G} + X_{r1,G} - X_{r2,G} + X_{r3,G} - X_{r4,G}) \quad (27)$$

$$V_{i,G} = X_{i,G} + F \times (X_{r3,G} - X_{i,G} + X_{r1,G} - X_{r2,G}) \quad (20)$$

The mutation strategy based on clearing niche mechanism:

$$V_{i,G} = X_{c,G} + F \times (X_{r1,G} - X_{r2,G}) \quad (19)$$

$$V_{i,G} = X_{c,G} + F \times (X_{r2,G} - X_{r3,G} + X_{r4,G} + X_{r5,G}) \quad (24)$$

$$V_{i,G} = X_{i,G} + F \times (X_{c,G} - X_{i,G} + X_{r1,G} - X_{r2,G} + X_{r3,G} - X_{r4,G}) \quad (28)$$

$$V_{i,G} = X_{c,G} + F \times (X_{r3,G} - X_{c,G} + X_{r1,G} - X_{r2,G}) \quad (21)$$

#### (4) The EPSDE algorithm

In EPSDE algorithm, the mutation strategies are DE/rand/1, DE/best/2, DE/target-to-rand/1. We can see changes of the mutation strategy as following.

The initial mutation strategy:

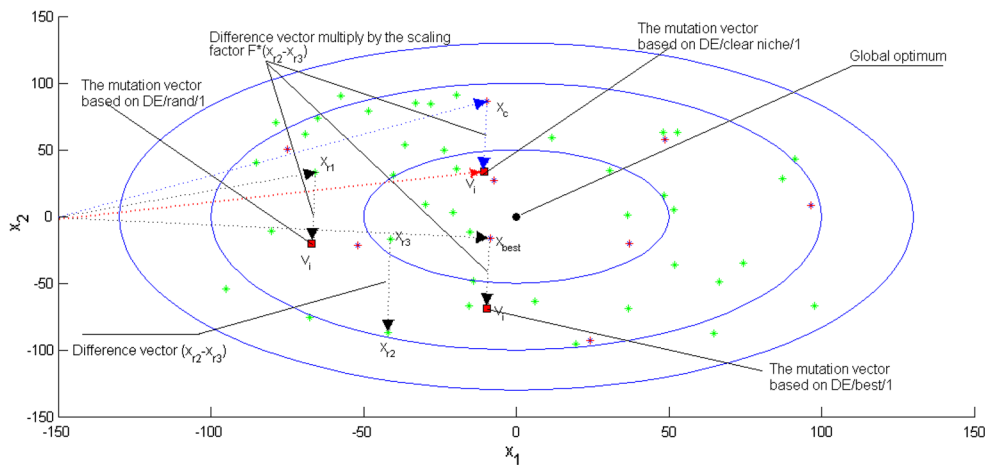
$$V_{i,G} = X_{r1,G} + F \times (X_{r2,G} - X_{r3,G}) \quad (17)$$

$$V_{i,G} = X_{best,G} + F \times (X_{r1,G} - X_{r2,G} + X_{r3,G} - X_{r4,G}) \quad (22)$$

$$V_{i,G} = X_{i,G} + F \times (X_{r3,G} - X_{i,G} + X_{r1,G} - X_{r2,G}) \quad (27)$$

The mutation strategy based on clearing niche mechanism:

$$V_{i,G} = X_{c,G} + F \times (X_{r1,G} - X_{r2,G}) \quad (19)$$



**Fig. 4** Illustrations of the new and basic mutation strategies in two-dimensional parametric space. Green “+” denotes the individuals in the population; Red “+” denotes the best individual in each niche; Red “□” denotes the mutation vector, and they are based on different mutation strategies, such as DE/rand/1, DE/best/1, and DE/clear niche/1. For

$$V_{i,G} = X_{c,G} + F \times (X_{r1,G} - X_{r2,G} + X_{r3,G} - X_{r4,G}) \tag{24}$$

$$V_{i,G} = X_{c,G} + F \times (X_{r3,G} - X_{c,G} + X_{r1,G} - X_{r2,G}) \tag{28}$$

### 4 Test functions

#### 4.1 Minimization optimization benchmark functions

This section lists the global minimization benchmark functions which are used to evaluate the performance of DE variants. In the section, 17 global minimization benchmark functions are chosen as test functions. The 17 test functions ( $f_1 - f_{17}$ ) are dimension-wise scalable (Liang et al. 2005).

(1) Shifted Sphere function, defined as

$$f_1(x) = \sum_{i=1}^D z_i^2, \quad z = x - o \tag{29}$$

With  $o = (o_1, o_2, \dots, o_D)$  is the shifted global optimum, global optimum is  $x^* = o$  and  $f(x^*) = 0$  for  $-100 \leq x_i \leq 100$ . Three-dimensional graph corresponding to this function is shown in Fig. 6.

(2) Shifted Schwefel’s Problem 1.2, defined as

$$f_2(x) = \sum_{i=1}^D \left( \sum_{j=1}^i z_j \right)^2, \quad z = x - o \tag{30}$$

With  $o = (o_1, o_2, \dots, o_D)$  is the shifted global optimum, global optimum is  $x^* = o$  and  $f(x^*) = 0$  for  $-100 \leq x_i \leq$

example, we know the generated manner of the mutation vector based on DE/clear niche/1, it likes that “ $V_{i,G} = X_{c,G} + F \times (X_{r2,G} - X_{r3,G})$ ”. The blue lines mean the vector  $X_{c,G}$  and the vector  $F \times (X_{r2,G} - X_{r3,G})$ , adding two vectors, the red line is got which represents the mutation vector (color figure online)

100. Three-dimensional graph corresponding to this function is shown in Fig. 6.

(3) Shifted Rotated High Conditioned Elliptic function, defined as

$$f_3(x) = \sum_{i=1}^D \left( 10^6 \right)^{\frac{i-1}{D-1}} z_i^2, \quad z = (x - o) * M \tag{31}$$

With  $o = (o_1, o_2, \dots, o_D)$  is the shifted global optimum,  $M$  is a orthogonal rotation matrix, global optimum is  $x^* = o$  and  $f(x^*) = 0$  for  $-100 \leq x_i \leq 100$ . Three-dimensional graph corresponding to this function is shown in Fig. 6.

(4) Shifted Schwefel’s Problem 1.2 with Noise in Fitness, defined as

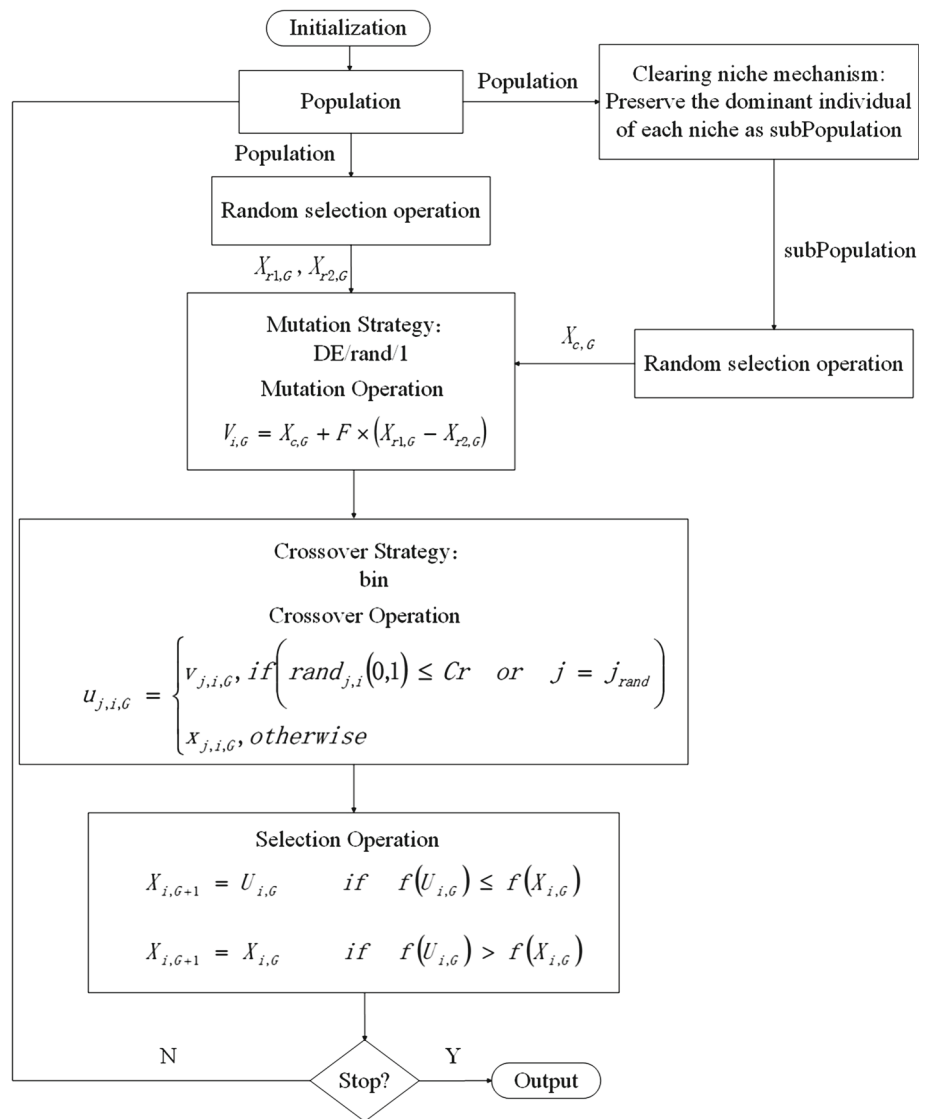
$$f_4(x) = \left( \sum_{i=1}^D \left( \sum_{j=1}^i z_j \right)^2 \right) * (1 + 0.4 * |N(0, 1)|), \quad z = x - o \tag{32}$$

With  $o = (o_1, o_2, \dots, o_D)$  is the shifted global optimum, global optimum is  $x^* = o$  and  $f(x^*) = 0$  for  $-100 \leq x_i \leq 100$ . Three-dimensional graph corresponding to this function is shown in Fig. 6.

(5) Shifted Rosenbrock’s Function, defined as

$$f_5(x) = \sum_{i=1}^{D-1} \left( 100 * (z_i^2 - z_{i+1})^2 + (z_i - 1)^2 \right), \quad z = x - o \tag{33}$$

**Fig. 5** Flowchart of DE combined with clearing niche mechanism



With  $o = (o_1, o_2, \dots, o_D)$  is the shifted global optimum, global optimum is  $x^* = o$  and  $f(x^*) = 0$  for  $-100 \leq x_i \leq 100$ . Three-dimensional graph corresponding to this function is shown in Fig. 6.

(6) Shifted Rotated Griewank's Function without Bounds, defined as

$$f_6(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1, \quad z = (x - o) * M \tag{34}$$

With  $o = (o_1, o_2, \dots, o_D)$  is the shifted global optimum,  $M$  is a orthogonal rotation matrix, global optimum is  $x^* = o$  and  $f(x^*) = 0$  for  $0 \leq x_i \leq 600$ . Three-dimensional graph corresponding to this function is shown in Fig. 6.

(7) Shifted Rotated Ackley's Function with Global Optimum on Bounds, defined as

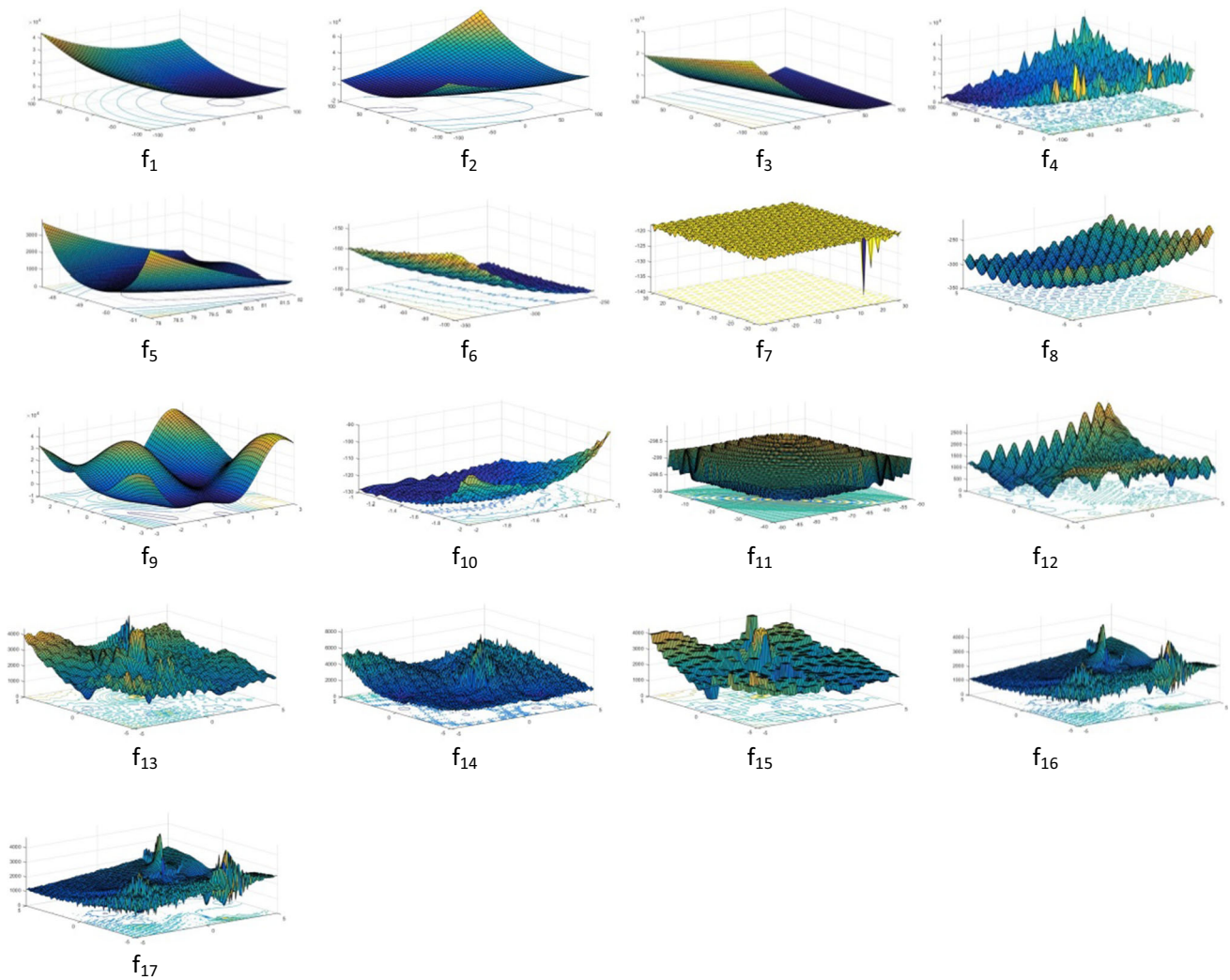
$$f_7(x) = -20 * \exp\left(-0.2 * \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2 * \pi * z_i)\right) + 20 + e, \tag{35}$$

$$z = (x - o) * M$$

With  $o = (o_1, o_2, \dots, o_D)$  is the shifted global optimum,  $M$  is a orthogonal rotation matrix, global optimum is  $x^* = o$  and  $f(x^*) = 0$  for  $-32 \leq x_i \leq 32$ . Three-dimensional graph corresponding to this function is shown in Fig. 6.

(8) Shifted Rastrigin's Function, defined as

$$f_8(x) = \sum_{i=1}^D \left(z_i^2 - 10 * \cos(2 * \pi * z_i) + 10\right), \quad z = x - o \tag{36}$$



**Fig. 6** Three-dimensional graph of 17 benchmark functions

With  $o = (o_1, o_2, \dots, o_D)$  is the shifted global optimum, global optimum is  $x^* = o$  and  $f(x^*) = 0$  for  $-5 \leq x_i \leq 5$ . Three-dimensional graph corresponding to this function is shown in Fig. 6.

(9) Schwefel’s Problem2.13, defined as

$$\begin{aligned}
 f_9 &= \sum_{i=1}^D (A_i - B_i(x))^2 \\
 A_i &= \sum_{j=1}^D (a_{ij} \sin \alpha_j + b_{ij} \cos \alpha_j), \\
 B_i(x) &= \sum_{j=1}^D (a_{ij} \sin x_j + b_{ij} \cos x_j)
 \end{aligned}
 \tag{37}$$

With  $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_D]$ ,  $\alpha_j \in [-\pi, \pi]$ ; With global optimum  $x^* = \alpha$  and  $f(x^*) = 0$  for  $-\pi \leq x_i \leq \pi$ . Three-dimensional graph corresponding to this function is shown in Fig. 6.

(10) Shifted Expanded Griewank’s plus Rosenbrock’s Function, defined as

$$\begin{aligned}
 f_{10}(x) &= f_7(f_2(z_1, z_2)) + f_7(f_2(z_2, z_3)) + \dots \\
 &\quad + f_7(f_2(z_{D-1}, z_D)) + f_7(f_2(z_D, z_1)) \\
 z &= x - o + 1
 \end{aligned}
 \tag{38}$$

With  $o = (o_1, o_2, \dots, o_D)$  is the shifted global optimum, global optimum is  $x^* = o$  and  $f(x^*) = 0$  for  $-3 \leq x_i \leq 1$ . Three-dimensional graph corresponding to this function is shown in Fig. 6.

(11) Shifted Rotated Expanded Scaffer’s F6 Function, defined as

$$\begin{aligned}
 F(x, y) &= 0.5 + \frac{(\sin^2(\sqrt{x^2 + y^2}) - 0.5)}{(1 + 0.001 \times (x^2 + y^2))^2} \\
 f_{11} &= F(z_1, z_2) + F(z_2, z_3) + \dots + F(z_{D-1}, z_D) \\
 &\quad + F(z_D, z_1), \quad z = (x - o) * M
 \end{aligned}
 \tag{39}$$



**Table 1** Global optimum and initialization ranges for the benchmark functions

Functions	Dimension	Global optimum	$f(x^*)$	Initialization range
$f_1$	30	$o$	0	$[-100, 100]^D$
$f_2$		$o$	0	$[-100, 100]^D$
$f_3$		$o$	0	$[-100, 100]^D$
$f_4$		$o$	0	$[-100, 100]^D$
$f_5$		$o$	0	$[-100, 100]^D$
$f_6$		$o$	0	$[0, 600]^D$
$f_7$		$o$	0	$[-32, 32]^D$
$f_8$		$o$	0	$[-5, 5]^D$
$f_9$		$\alpha$	0	$[-\pi, \pi]^D$
$f_{10}$		$o$	0	$[-3, 1]^D$
$f_{11}$		$o$	0	$[-100, 100]^D$
$f_{12}$		$o_l$	0	$[-5, 5]^D$
$f_{13}$		$o_l$	0	$[-5, 5]^D$
$f_{14}$		$o_l$	0	$[-5, 5]^D$
$f_{15}$		$o_l$	0	$[-5, 5]^D$
$f_{16}$		$o_l$	0	$[-5, 5]^D$
$f_{17}$		$o_l$	0	$[2, 5]^D$

$o$  is the shifted vector.  $o_l$  is the shifted vector for the first basic function in the composition function

With  $o = (o_1, o_2, \dots, o_D)$  is the shifted global optimum, global optimum is  $x^* = o$  and  $f(x^*) = 0$  for  $-100 \leq x_i \leq 100$ . Three-dimensional graph corresponding to this function is shown in Fig. 6.

(12) Hybrid Composition Function, defined as

The functions  $f_{12}$  (CF1),  $f_{13}$  (CF7),  $f_{14}$  (CF8),  $f_{15}$  (CF9),  $f_{16}$  (CF10) and  $f_{17}$  (CF11) are composed by using 10 different functions respectively. Their global optimums are easy to find once the global basins are found. The details of constructing such functions are presented in Liang et al. (2005). And three-dimensional graph corresponding to these six functions is shown in Fig. 6.

Among the above 17 benchmark problems, functions  $f_1 - f_4$  are unimodal and functions  $f_5 - f_9$  are basic multimodal functions,  $f_{10}$  and  $f_{11}$  are expanded multimodal functions, and  $f_{12} - f_{17}$  are hybrid composition functions. The optimum value, position of the global optima, and initialization ranges for these 17 benchmark problems are provided in Table 1.

## 4.2 Multimodal optimization benchmark functions

This section lists the multimodal optimization functions which are used to evaluate the performance of DE variants. In the section, 8 multimodal optimization functions are chosen as test functions. The eight test functions ( $F_1 - F_8$ ) has diverse characteristics, and the functions used in the exper-

iment are presented in Simionescu (2014), Surjanovic and Bingham (2015).

(1) Beale function, defined as

$$F_1 = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2 \quad (41)$$

With global optimum is  $x^* = (3, 0.5)$  and  $F(x^*) = 0$  for  $-4.5 \leq x_i \leq 4.5$  ( $i = 1, 2$ ). Three-dimensional graph corresponding to this function is shown in Fig. 7.

(2) Goldstein-price function, defined as

$$F_2 = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \quad (42)$$

With global optimum is  $x^* = (0, -1)$  and  $F(x^*) = 3$  for  $-2 \leq x_i \leq 2$  ( $i = 1, 2$ ). Three-dimensional graph corresponding to this function is shown in Fig. 7.

(3) Matyas function, defined as

$$F_3 = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2 \quad (43)$$

With global optimum is  $x^* = (0, 0)$  and  $F(x^*) = 0$  for  $-10 \leq x_i \leq 10$  ( $i = 1, 2$ ). Three-dimensional graph corresponding to this function is shown in Fig. 7.

(4) Cross-in-tray function, defined as

$$F_4 = -0.0001 \left( \left| \sin(x_1) \sin(x_2) \exp \left( \left| 100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right| \right) \right| + 1 \right)^{0.1} \quad (44)$$

With global optimum is  $x^* = (1.3491, -1.3491), (-1.3491, -1.3491), (-1.3491, 1.3491), (1.3491, 1.3491)$  and  $F(x^*) = -2.06261$  for  $-10 \leq x_i \leq 10$  ( $i = 1, 2$ ). Three-dimensional graph corresponding to this function is shown in Fig. 7.

(5) Eggholder function, defined as

$$F_5 = -(x_2 + 47) \sin \left( \sqrt{\left| x_2 + \frac{x_1}{2} + 47 \right|} \right) - x_1 \sin \left( \sqrt{|x_1 - (x_2 + 47)|} \right) \quad (45)$$

With global optimum is  $x^* = (512, 404.2319)$  and  $F(x^*) = -959.6407$  for  $-512 \leq x_i \leq 512 (i = 1, 2)$ . Three-dimensional graph corresponding to this function is shown in Fig. 7.

(6) Simionescu function, defined as

$$F_6 = 0.1x_1x_2$$

$$\text{subjected to: } x_1^2 + x_2^2 \leq \left( r_T + r_S \cos \left( n \tan^{-1} \left( \frac{x_1}{x_2} \right) \right) \right)^2 \quad (46)$$

With global optimum is  $x^* = (-0.85586, 0.85586), (0.85586, 0.85586), (0.85586, -0.85586), (-0.85586, -0.85586)$  and  $F(x^*) = -0.072625$  for  $-1.25 \leq x_i \leq 1.25 (i = 1, 2)$ . Three-dimensional graph corresponding to this function is shown in Fig. 7.

(7) Holder Table function, defined as

$$F_7 = - \left| \sin(x_1) \cos(x_2) \exp \left( \left| 1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right| \right) \right| \quad (47)$$

With global optimum is  $x^* = (8.05502, -9.66459), (-8.05502, -9.66459), (-8.05502, 9.66459), (8.05502, 9.66459)$  and  $F(x^*) = -19.2085$  for  $-10 \leq x_i \leq 10 (i = 1, 2)$ . Three-dimensional graph corresponding to this function is shown in Fig. 7.

(8) Shubert function, defined as

$$F_8 = \left( \sum_{i=1}^5 i \cos((i+1)x_1 + i) \right) \times \left( \sum_{i=1}^5 i \cos((i+1)x_2 + i) \right) \quad (48)$$

With global optimum is  $x^* = (-7.0835, -1.4250), (-7.0835, 4.8580), (-7.0835, -7.7082), (-7.7082, 5.4829), (-7.7082, -0.8003), (-7.7082, -7.0835), (5.4829, -1.4250), (5.4829, 4.8580), (5.4829, -7.7082), (4.8580, -0.8003), (4.8580, 5.4829), (4.8580, -7.0835), (-0.8003, -1.4250), (-0.8003, 4.8580), (-0.8003, -7.7082), (-1.4250, -0.8003), (-1.4250, 5.4829), (-1.4250, -7.0835)$  and  $F(x^*) = -186.7309$  for  $-5.12 \leq x_i \leq 5.12 (i = 1, 2)$ . Three-dimensional graph corresponding to this function is shown in Fig. 7.

## 5 Experimental studies

In this section, comprehensive experiments are carried out to evaluate the effectiveness of new mutation strategies, including two basic DE algorithms and four advanced DE variants. It includes three studies, one of which is the accuracy study, one is the multimodal study, and another is a multiobjective problem study.

### 5.1 Accuracy study

In this section, 17 benchmark functions are selected as the test functions, which are from the CEC 2005 special session. The objective is to minimize values of the functions. Here, the experimental setup is shown first. Then, the comparisons between DE changed with new mutation strategies and the corresponding DE algorithms are proposed.

#### 5.1.1 Experimental setup

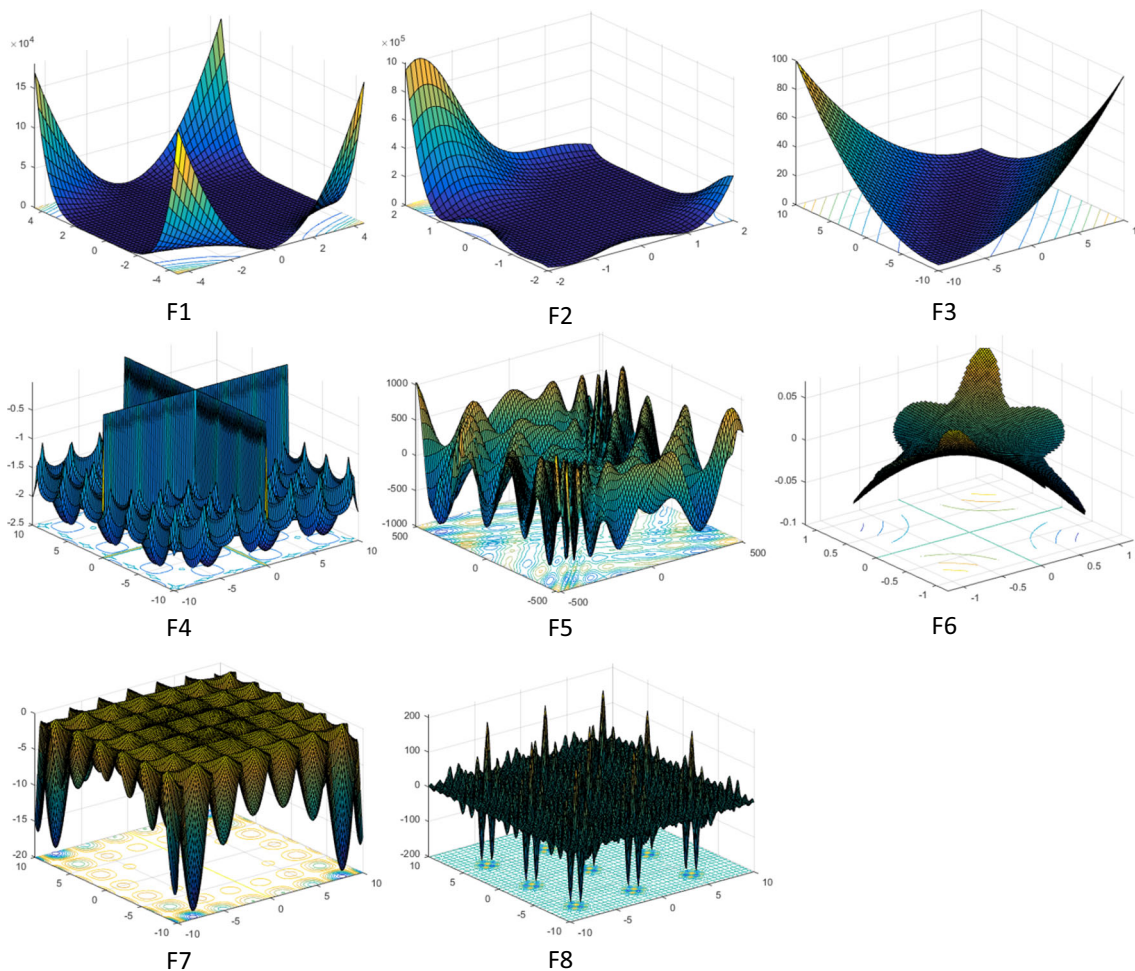
To make a fair comparison, the parameters for all the DE algorithms are set as follows unless a change is mentioned.

- (1) Dimension of the functions ( $D$ ): 30.
- (2) Population size ( $NP$ ): 50.
- (3) Maximal number of fitness function evaluations: 10000.
- (4) Number of runs: 25.
- (5) Scaling Factor ( $F$ ) and Crossover Control Parameter ( $Cr$ ): For two basic DE algorithms,  $F$  and  $Cr$  are (0.5, 0.9) and (0.5, 0.3), respectively. For four advanced DE variants,  $F$  and  $Cr$  are set as mentioned in [Brest and Mernik \(2008\)](#), [Qin and Suganthan \(2005\)](#), [Mallipeddi et al. \(2011\)](#), [Zhang and Sanderson \(2009\)](#).

In the experiments, comparisons between two basic DE algorithms whose mutation strategies are set as DE/rand/1, DE/best/1 and their corresponding DE algorithms with new mutation strategies are made first. Then, the performance of several advanced DE variants with the corresponding DE variants with new mutation strategies are compared, including jDE ([Brest and Mernik 2008](#)), JADE ([Zhang and Sanderson 2009](#)), SaDE ([Qin and Suganthan 2005](#)), and EPSDE ([Mallipeddi et al. 2011](#)). All the parameters of these DE variants are set as their basic papers. Due to space limitation, only mean and standard deviation of the best error values obtained by algorithms are shown in this paper.

#### 5.1.2 Comparison with basic DE algorithms

In this section, the DE algorithms with new mutation strategies (call DE-Niche for short) are compared with two basic DE algorithms whose mutation strategies are set as DE/rand/1 and DE/best/1 to test the effectiveness of new



**Fig. 7** Three-dimensional graph of 8 multimodal functions

mutation strategies when whose  $F$  and  $Cr$  respectively are 0.5, 0.9 and 0.5, 0.3. As the changed rule which is described in Sect. 3.2, the new mutation strategies of DE/rand/1 and DE/best/1 are same. The results for the functions at  $30D$  are shown in Table 2. Wilcoxon's rank sum test at a 0.05 significance is performed between DE-Niche and two basic DE algorithms with different control parameters.

From Table 2, we can find that the DE-Niche significantly outperforms the corresponding DE algorithm with respect to the overall performance. Specifically, when  $F = 0.5$  and  $Cr = 0.9$ , DE-Niche significantly improves the performance of DE/best/1 on 15 out of 17 functions and loses on only one functions. For  $F = 0.5$  and  $Cr = 0.3$ , DE-Niche improves the performance of DE/rand/1 on 7 out of 17 functions, without losing on any functions. Table 3 presents the result of comparison of DE-Niche with two basic DE algorithms. It is clear that on average the proposed DE-Niche algorithm performs better than the two basic DE algorithms.

From the comparison between DE-Niche and two basic DE algorithms, it can be shown that the advantages of DE-Niche are obviously and the solution quality of DE-Niche

is better than DE/rand/1 and DE/best/1. The improvements from DE-Niche may be contributed by the fact that the new mutation strategies are combined with the clearing niche method can balance the population diversity and exploration capability.

### 5.1.3 Comparison with advanced DE variants

In order to further evaluate the effectiveness of the new mutation strategies, the clearing niche method is incorporated into several advanced DE variants, namely, jDE (Brest and Mernik 2008), JADE (Zhang and Sanderson 2009), SaDE (Qin and Suganthan 2005), and EPSDE (Mallipeddi et al. 2011) including jDE-Niche, JADE-Niche, SaDE-Niche and EPSDE-Niche. The comparisons between DE variants with new mutation strategies and the corresponding DE variants are made on 17 functions at  $30D$ . The results are shown in Table 4. Wilcoxon's rank sum test at a 0.05 significance is performed between jDE-Niche, JADE-Niche, SaDE-Niche and EPSDE-Niche and their corresponding DE variants.

**Table 2** Comparison results between the basic DE algorithms and their corresponding DE variants

Function			$F = 0.5, Cr = 0.9$			$F = 0.5, Cr = 0.3$			
			DE/rand/1	DE/best/1	DE-Niche	DE/rand/1	DE/best/1	DE-Niche	
Unimodal	$f_1$	Mean	2.31E-28-	1.56E+04+	9.09E-12	0.00E+00≈	1.78E-28+	0.00E+00	
		SD	1.77E-28	4.77E+03	1.70E-11	0.00E+00	1.69E-28	0.00E+00	
	$f_2$	Mean	7.46E-23-	2.33E+04+	4.02E-11	3.65E+02+	4.44E-10-	2.96E+00	
		SD	1.65E-22	7.23E+03	2.52E-11	9.84E+01	1.05E-09	5.56E+00	
	$f_3$	Mean	1.03E+05+	7.86E+07+	1.89E+04	4.92E+07+	1.08E+07-	3.57E+07	
		SD	5.76E+04	5.97E+07	1.22E+04	7.48E+07	4.94E+06	8.82E+06	
	$f_4$	Mean	6.17E-09+	4.18E+03+	6.25E-14	1.47E+03+	9.98E-03-	2.43E+02	
		SD	2.97E-08	3.16E+03	2.67E-13	4.00E+02	1.51E-02	1.85E+02	
Basic multimodal	$f_5$	Mean	1.89E+01+	2.82E+09+	1.27E+00	2.59E+01≈	2.10E+01≈	2.05E+01	
		SD	1.48E+01	1.27E+09	1.89E+00	1.67E+01	2.10E+01	2.69E+00	
	$f_6$	Mean	4.69E+03≈	7.26E+03+	4.69E+03	4.69E+03≈	4.70E+03≈	4.69E+03	
		SD	8.50E-13	4.78E+02	1.77E-12	1.36E-12	2.12E-12	3.23E-12	
	$f_7$	Mean	2.09E+01≈	2.09E+01≈	2.09E+01	2.09E+01≈	2.09E+01≈	2.09E+01	
		SD	4.50E-02	5.47E-02	5.34E-02	5.62E-02	6.32E-02	6.02E-02	
	$f_8$	Mean	1.37E+01≈	1.62E+02+	1.44E+01	1.26E+00≈	1.65E+01+	6.36E-01	
		SD	3.11E+00	3.41E+01	4.03E+00	3.74E+00	4.10E+00	8.06E-01	
	$f_9$	Mean	4.93E+03+	2.73E+05+	1.84E+03	1.00E+05+	9.87E+03-	6.94E+04	
		SD	6.46E+03	1.57E+05	2.32E+03	2.41E+04	7.20E+03	2.65E+04	
	Expanded multimodal	$f_{10}$	Mean	2.81E+00+	1.73E+01+	2.21E+00	7.48E+00≈	5.60E+00-	7.59E+00
			SD	7.24E-01	7.30E+00	5.38E-01	5.42E-01	6.66E-01	5.35E-01
$f_{11}$		Mean	1.28E+01≈	1.24E+01-	1.27E+01	1.32E+01≈	1.29E+01≈	1.32E+01	
		SD	3.19E-01	5.01E-01	2.40E-01	1.56E-01	3.52E-01	1.48E-01	
Hybrid composition	$f_{12}$	Mean	4.00E+02≈	6.71E+02+	3.76E+02	2.72E+02+	2.99E+02+	2.42E+02	
		SD	8.17E+01	1.16E+02	8.23E+01	4.51E+01	1.06E+02	5.63E+01	
	$f_{13}$	Mean	5.12E+02≈	1.22E+03+	5.00E+02	5.00E+02≈	5.67E+02+	5.00E+02	
		SD	6.00E+01	6.25E+01	1.82E-10	1.16E-13	1.67E+02	1.70E-13	
	$f_{14}$	Mean	8.74E+02≈	1.11E+03+	8.71E+02	9.01E+02+	8.97E+02≈	8.91E+02	
		SD	1.81E+01	6.93E+01	1.73E+01	1.27E+01	1.60E+01	1.37E+01	
	$f_{15}$	Mean	5.34E+02≈	1.23E+03+	5.40E+02	5.34E+02+	6.16E+02+	5.34E+02	
		SD	2.31E+00	3.90E+01	1.61E+01	2.51E-04	1.80E+02	7.83E-03	
	$f_{16}$	Mean	2.00E+02≈	1.26E+03+	2.00E+02	2.00E+02≈	2.12E+02≈	2.00E+02	
		SD	2.90E-14	5.05E+01	2.90E-14	2.90E-14	6.00E+01	6.20E-13	
$f_{17}$	Mean	1.62E+03≈	1.74E+03+	1.62E+03	1.63E+03≈	1.63E+03≈	1.62E+03		
	SD	5.25E+00	2.89E+01	3.77E+00	3.52E+00	5.36E+00	2.65E+00		

“+”, “-”, and “≈” denote that the performance of the DE-Niche algorithm is better than, worse than, and similar to that of two basic DE algorithm, respectively. “Mean” and “SD” indicate the average and standard deviation of the function error values obtained in 25 runs, respectively. Wilcoxon’s rank sum test at a 0.05 significance is performed between DE-Niche and two basic DE algorithms

From the results of functions in Table 4, it is clear that the new mutation strategies with clearing niche mechanism outperform most of the DE variants. For jDE, jDE-Niche is significantly better on 8 out of 17 functions and is outperformed by jDE on only one function. For JADE, JADE-Niche improves the performance of DE/rand/1 on 9 functions and loses on 2 functions. For SaDE, SaDE-Niche can significantly outperform SaDE on 10 functions, without being outperformed on any functions. For EPSDE, EPSDE-Niche

is better on 6 functions and is worse on 3 functions. Table 5 presents the result of comparison of jDE, JADE, SaDE, and EPSDE algorithms with their corresponding DE variants. It is clear that on average the DE variants with new mutation strategies perform better than their corresponding DE algorithms.

From the comparison between jDE, JADE, SaDE, and EPSDE and their corresponding DE variants jDE-Niche, JADE-Niche, SaDE-Niche, and EPSDE-Niche, it can be



**Table 3** Comparison of DE-Niche with two basic DE algorithms

DE-Niche	$F = 0.5, Cr = 0.9$		$F = 0.5, Cr = 0.3$	
	DE/rand/1	DE/best/1	DE/rand/1	DE/best/1
DE-Niche better	5	15	7	5
DE-Niche worse	2	1	0	5
DE-Niche equal	10	1	10	7
Success Rate	88.23%	94.11%	100%	70.58%

“DE-Niche better” means the value of DE-Niche is better than other algorithm, “DE-Niche worse” means the value of DE-Niche is worse than other algorithm, “DE-Niche equal” means the value of DE-Niche is equal than other algorithm, and “Success Rate” means the successful rate of DE-Niche (include the better and equal ones)

shown that the solution quality of DE whose mutation strategy is based on the clearing niche mechanism is the best. Therefore, the new mutation strategies can improve the performance of most DE variants studied.

#### 5.1.4 Discussion about the choices the clearing radius for DE algorithms with new mutation strategies

As mentioned in Sect. 3.1, the clearing radius threshold  $\delta$  is used to determine the size of the niches. If the value of  $\delta$  is large, most individuals would belong to one niche. On the contrary, if the value of  $\delta$  is small, the number of niches would be large. In order to determine the setup of  $\delta$  for DE algorithms with new mutation strategies, namely, DE-Niche, jDE-Niche, JADE-Niche, SaDE-Niche and EPSDE-Niche, the comparisons of DE variants with different  $\delta$  are made. The  $\delta$  value is set as 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9, respectively. Due to space limitation, only highly distinct functions are shown in this paper. In general, if the value for  $\delta$  obtains a poor performance we will give it up, if the value for  $\delta$  obtains a good performance we will give priority to it, if the value for  $\delta$  obtains a mild performance, at the same times, other values obtain both a good and poor performance for different functions, we will choose a mild value for  $\delta$ , and if the performance of all values are close, we can also consider the convergence of functions.

In Fig. 8 and Table 6, the comparisons of DE-Niche with different  $\delta$  are shown when  $F = 0.5, Cr = 0.9$ . It is clear that too big  $\delta$  will affect the performance of DE-Niche. For function  $f_1, f_2, f_4, f_6, f_8, f_{10}, f_{13}, f_{15}$  and  $f_{16}$ , 0.8 and 0.9 for  $\delta$  are the worst values, while the best one is 0.8 for function  $f_5$  and  $f_{17}$  and the second is 0.9 for function  $f_{11}$ . Similarly, for function  $f_1, f_2, f_4, f_5, f_{10}, f_{12}, f_{14}$ , and  $f_{16}$ , 0.7 for  $\delta$  are the worst value, while 0.7 is best for function  $f_{11}$  and the second best for function  $f_7, f_9$ , and  $f_{17}$ . Likewise, 0.6 for  $\delta$  are the best value for function  $f_7, f_9, f_{10}$ , and  $f_{16}$ , the second and third best for function  $f_3, f_5, f_{12}$ , and  $f_{16}$ , while 0.6 for  $\delta$  are the second and third worst value for function  $f_8, f_{15}$ , and  $f_{17}$ . We can also analyze the results when the  $\delta$  is set as 0.1, 0.2, 0.3, 0.4, and 0.5. Apparently, 0.2 for  $\delta$  are the

best one for function  $f_{14}$ , the second and third best value for function  $f_1, f_2, f_5, f_6, f_9, f_{10}, f_{13}, f_{15}$ , and  $f_{17}$ , while 0.2 for  $\delta$  are the third worst value for function  $f_{12}$ . For function  $f_1, f_2, f_3, f_4, f_6, f_8, f_{10}, f_{12}$ , and  $f_{16}$ , 0.3 for  $\delta$  are the first, second or third best value, while 0.3 for  $\delta$  is the first, second or third worst value for function  $f_9, f_{11}$ , and  $f_{17}$ . In general, considering all functions,  $\delta = 0.2$  can most enhance the performance of DE-Niche when  $F = 0.5, Cr = 0.9$ . And the result can be seen in Table 12.

In Fig. 9 and Table 7, the comparisons of DE-Niche with different  $\delta$  are shown when  $F = 0.5, Cr = 0.3$ . It is clear that too small  $\delta$  will affect the performance of DE-Niche and mild values just show medium performance. For function  $f_1, f_6, f_8, f_{10}, f_{13}, f_{15}$ , and  $f_{16}$ , DE-Niche obtains the similar performance for all values of  $\delta$ . 0.9 for  $\delta$  obtains the best value for function  $f_2, f_4$ , and  $f_9$ , and the second best value for function  $f_{11}$ , while the worst value for function for function  $f_{13}$ . For function  $f_2, f_3, f_4, f_5, f_{11}$ , and  $f_{17}$ , 0.8 for  $\delta$  obtains the first or second best value, while for function  $f_9, f_{12}$ , and  $f_{14}$ , 0.8 for  $\delta$  performs worse. For function  $f_2, f_4, f_5, f_7, f_{12}$ , and  $f_{17}$ , 0.7 for  $\delta$  obtains the second or third best value. In the same way, we can analyze the other values of  $\delta$  and it can be found that the other values of  $\delta$  cannot obtain a better result than 0.9, 0.8, or 0.7. In general, considering all functions,  $\delta = 0.7$  can most enhance the performance of DE-Niche when  $F = 0.5, Cr = 0.3$ . And the result can be seen in Table 12.

In Fig. 10 and Table 8, the comparisons of jDE-Niche with different  $\delta$  are shown. The results indicate that a mild value of  $\delta$  for unimodal and basic multimodal functions is better, and a small value of  $\delta$  for expanded multimodal and hybrid composition functions is better. For function  $f_7$ , 0.9 for  $\delta$  is best value, 0.1 for  $\delta$  is worst value, and the other values for  $\delta$  is medium. However, when we consider from  $f_1$  to  $f_9$  in Table 8, it can be found 0.5 for  $\delta$  is the better choice for unimodal and basic multimodal functions. For function  $f_{11}$ , the best value for  $\delta$  is 0.8, the worst value for  $\delta$  is 0.7, and the other values for  $\delta$  are close. For function  $f_{12}, f_{14}$  and  $f_{15}$ , 0.1 for  $\delta$  is the best value and the other values are close. For function  $f_{15}$ , the

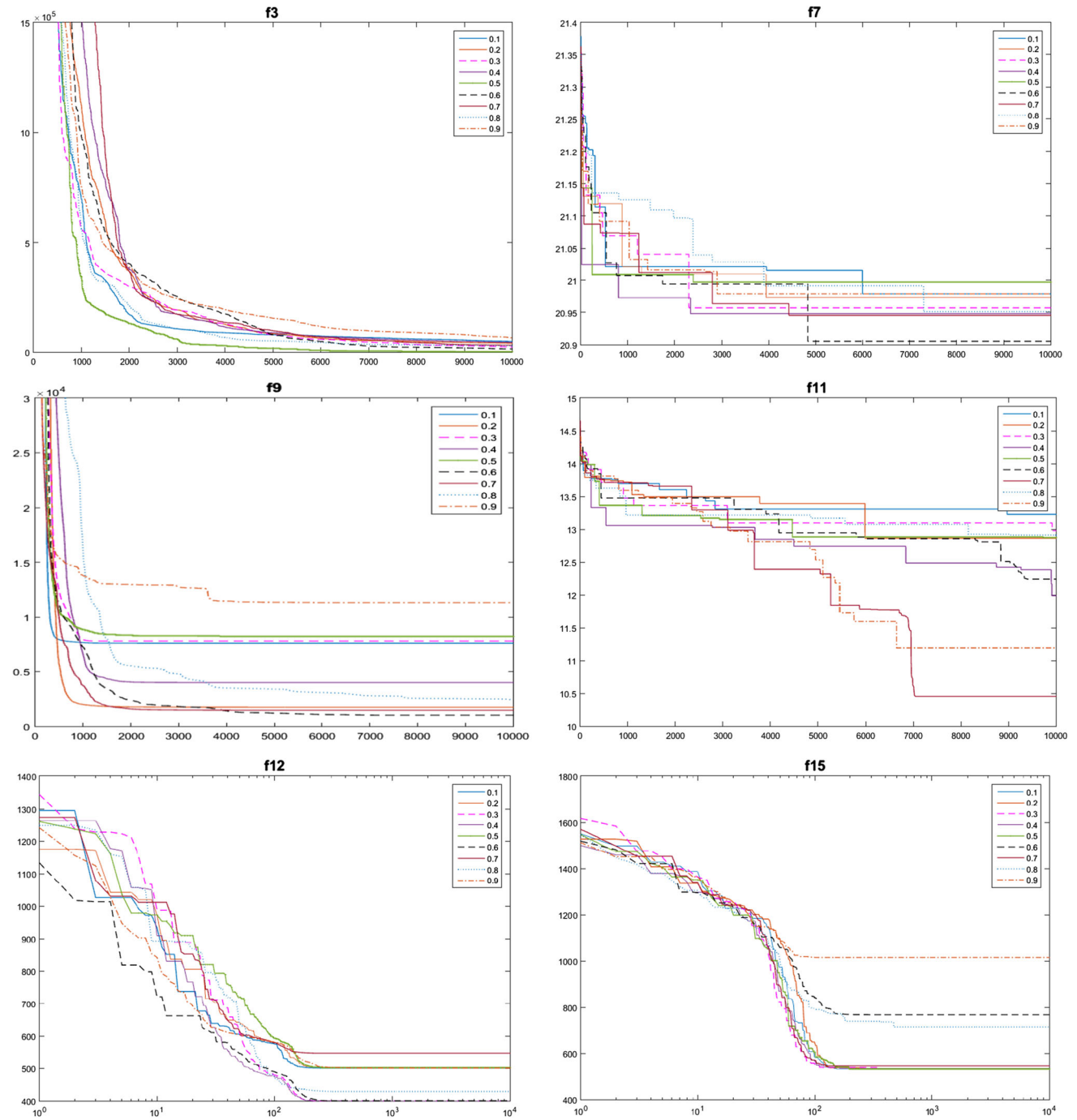


**Table 4** Comparison results between the jDE, JADE, SaDE and EPSDE algorithms and their corresponding DE variants with new mutation strategies

Function		jDE	jDE_Niche	JADE	JADE_Niche	SaDE	SaDE_Niche	EPSDE	EPSDE_Niche
Unimodal	$f_1$	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
		SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	$f_2$	Mean	1.20E-19+	8.49E-25	6.78E-28-	2.86E-27	4.68E-12	1.65E-12	6.32E-26+
		SD	3.79E-19	9.09E-25	3.15E-28	1.04E-27	1.28E-11	1.75E-12	1.25E-25
	$f_3$	Mean	4.74E+04+	1.07E+04	8.38E+02	7.50E+02	2.95E+05+	2.15E+05	1.37E+06+
		SD	2.91E+04	3.84E+03	6.77E+02	4.55E+02	1.44E+05	7.41E+04	5.48E+06
Basic multimodal	$f_4$	Mean	1.14E-03+	2.99E-10	8.86E-13+	4.85E-19	1.46E+01	1.29E+01	4.92E-01-
		SD	5.13E-03	2.64E-10	3.98E-12	6.01E-19	3.42E+01	1.57E+01	1.55E+00
	$f_5$	Mean	4.78E+01	1.04E-22	1.22E+00+	9.89E-25	4.28E+01	3.71E+01	4.78E-01
		SD	1.32E+00	1.19E-22	3.87E+00	8.58E-25	3.48E+01	3.09E+01	1.32E+00
	$f_6$	Mean	4.69E+03+	4.69E+03	4.69E+03+	4.69E+03	4.69E+03	4.69E+03	3.78E+00+
		SD	1.24E-12	0.00E+00	1.25E-12	1.65E+00	2.59E-12	0.00E+00	2.26E+00
Expanded multimodal	$f_7$	Mean	2.09E+01+	2.08E+01	2.07E+01+	2.05E+01	2.09E+01	2.09E+01	2.09E+01+
		SD	4.86E-02	5.47E-02	3.81E-01	4.58E-01	5.00E-02	4.79E-02	6.65E-02
	$f_8$	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
		SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	$f_9$	Mean	3.33E+03	3.26E+03	3.56E+03+	3.52E+02	2.83E+03+	2.66E+03	2.56E+04
		SD	3.63E+03	3.55E+03	4.43E+03	5.73E+02	2.97E+03	2.40E+03	5.62E+03
Hybrid composition	$f_{10}$	Mean	1.01E+00	1.04E+00	1.04E+00+	9.30E-01	3.06E+00+	2.66E+00	1.47E+00
		SD	9.82E-02	1.74E-01	1.05E-01	9.78E-02	4.64E-01	4.85E-01	1.36E-01
	$f_{11}$	Mean	1.27E+01+	1.23E+01	1.16E+01	1.20E+01	1.19E+01	1.19E+01	1.34E+01+
		SD	1.91E-01	1.62E-01	1.11E+00	4.76E-01	5.82E-01	4.37E-01	2.24E-01
	$f_{12}$	Mean	3.60E+02	3.03E+02	3.38E+02+	2.00E+02	3.92E+02+	3.21E+02	2.19E+02
		SD	1.00E+02	1.58E+02	1.07E+02	1.22E+02	7.02E+01	8.49E+01	4.65E+01
	$f_{13}$	Mean	5.00E+02	5.00E+02	5.36E+02+	5.00E+02	5.25E+02	5.00E+02	8.33E+02-
		SD	1.06E-12	0.00E+00	9.94E+01	0.00E+00	1.29E+02	0.00E+00	1.00E+02
	$f_{14}$	Mean	8.95E+02+	8.85E+02	8.71E+02-	8.99E+02	9.54E+02+	9.29E+02	5.06E+02
		SD	1.67E+01	1.21E+01	2.11E+01	7.07E+00	1.65E+01	9.01E+00	6.46E+00
	$f_{15}$	Mean	5.36E+02+	5.34E+02	5.34E+02	5.34E+02	5.34E+02+	5.34E+02	8.69E+02-
		SD	5.36E+00	1.66E+00	1.90E+00	3.77E-02	7.00E-03	6.17E+04	2.68E+00
	$f_{16}$	Mean	2.00E+02	2.00E+02	2.00E+00	2.00E+00	2.00E+02	2.00E+02	2.12E+02
		SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.21E+00
	$f_{17}$	Mean	1.62E+03-	1.65E+02	1.63E+03+	1.62E+03	1.64E+03+	1.63E+03	2.12E+02
		SD	4.06E+00	4.36E+00	4.18E+00	4.10E+00	5.01E+00	4.24E+00	2.07E+00

**Table 5** Comparison of the jDE, JADE, SaDE and EPSDE algorithms with their corresponding DE variants

	jDE	JADE	SaDE	EPSDE
Corresponding algorithm better	8	9	10	6
Corresponding algorithm worse	1	2	0	3
Corresponding algorithm equal	8	6	7	8
Success rate	94.11 %	88.23 %	100 %	82.35 %



**Fig. 8** DE-Niche with different clearing radius when  $F = 0.5$ ,  $Cr = 0.9$

**Table 6** Results on DE-Niche with different clearing radius when  $F$  and  $Cr$  are set as 0.5 and 0.9

Rank	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$	$f_{16}$	$f_{17}$
<i>Best</i>																	
1	0.1	0.1	0.5	0.1	0.8	0.3	0.6	0.3	0.6	0.6	0.7	0.3	0.4	0.2	0.1	0.6	0.8
2	0.2	0.2	0.6	0.5	0.6	0.4	0.7	0.1	0.7	0.2	0.9	0.4	0.2	0.6	0.5	0.3	0.7
3	0.3	0.3	0.3	0.3	0.2	0.2	0.4	0.4	0.2	0.3	0.4	0.6	0.1	0.1	0.2	0.4	0.2
<i>Worst</i>																	
1	0.9	0.9	0.9	0.7	0.9	0.9	0.5	0.9	0.9	0.9	0.1	0.7	0.9	0.9	0.9	0.9	0.3
2	0.8	0.8	0.1	0.9	0.1	0.8	0.1	0.8	0.5	0.7	0.3	0.5	0.8	0.5	0.6	0.8	0.6
3	0.7	0.7	0.4	0.8	0.7	0.1	0.9	0.6	0.3	0.8	0.8	0.2	0.5	0.7	0.8	0.7	0.4

“Best” means that for each function the DE algorithm obtains the best performance when  $\delta$  is set as the corresponding value, and here we give out the value of  $\delta$  which are in the top three; similarly, “Worst” means for each function the DE algorithm obtains the worst performance when  $\delta$  is set as the corresponding value, and here we give out the value of  $\delta$  which are in the last three. For example, the corresponding value of  $\delta$  is 0.1 in the second row and the third column, it means 0.1 for  $\delta$  obtains the best performance for function  $f_1$ ; the corresponding value of  $\delta$  is 0.9 in the fifth row and the third column, it means 0.9 for  $\delta$  obtains the worst performance for function  $f_1$

In Fig. 11 and Table 9, the comparisons of JADE-Niche with different  $\delta$  are shown. It is clear that too small or too large value for  $\delta$  shows an unstable performance on different functions. For function  $f_4$ , the worst and second worst value for  $\delta$  is 0.2 and 0.8, the other values are close, the convergence of 0.4 for  $\delta$  performs best. For function  $f_7$ , 0.8 and 0.9 for  $\delta$  are signally better than other values and other values for  $\delta$  are close, however, 0.8 and 0.9 for  $\delta$  are signally worse than other values for functions  $f_{12}$ ,  $f_{14}$  and  $f_{17}$ . For function  $f_{11}$ , 0.3 for  $\delta$  are signally better than other values and other values for  $\delta$  are close, however, 0.3 for  $\delta$  are signally worse than other values for functions  $f_7$ . For functions  $f_{14}$  and  $f_{17}$ , all the values for  $\delta$  are close, but 0.4 for  $\delta$  performs better than other values. It is shown in Table 9, synthesizing each kind of situation, for JADE-Niche, the value of  $\delta$  is set as 0.4 from  $f_1$  to  $f_{17}$ . 0.4 for  $\delta$  can most enhance the performance of JADE-Niche. And the result is shown in Table 12.

In Fig. 12 and Table 10, the comparisons of SaDE-Niche with different  $\delta$  are shown. We can see that 0.5, 0.6, 0.8, and 0.9 show an unstable performance on different functions. For function  $f_4$ , the worst value for  $\delta$  is 0.5, the other values are close, and the convergence of 0.9 and 0.1 for  $\delta$  perform better. For function  $f_7$ , 0.6 and 0.3 for  $\delta$  are signally better than other values, the third best value for  $\delta$  is 0.2, and the worst and second worst values for  $\delta$  is 0.9 and 0.8 while 0.9 and 0.8 for  $\delta$  perform unstable for functions  $f_{11}$ ,  $f_{12}$ . For function  $f_{11}$ , all the values for  $\delta$  are close, 0.9 for  $\delta$  performs best, 0.2 performs worst, while 0.8 performs third worst. For function  $f_{12}$ , 0.8 for  $\delta$  performs signally best than other values. For function  $f_{15}$  we can find that 0.5 for  $\delta$  performs signally worse than other values, the other values perform similar, and the convergence of 0.2 for  $\delta$  perform better. For function  $f_{17}$ , 0.2 for  $\delta$  performs signally best than other values. Considering both performance and convergence in Table 10, 0.2 for  $\delta$  performs better on functions  $f_1$ ,  $f_2$ ,  $f_7$ ,  $f_9$ ,  $f_{13}$ ,  $f_{14}$ ,  $f_{15}$ , and

$f_{17}$ . Therefore, for SaDE-Niche, the value of  $\delta$  is set as 0.2. And the result is shown in Table 12.

In Fig. 13 and Table 11, the comparisons of EPSDE-Niche with different  $\delta$  are shown. It can be seen that most values for  $\delta$  perform unstably on different functions. 0.1 for  $\delta$  performs better on functions  $f_4$  and  $f_7$ , while it performs worse on functions  $f_1$ ,  $f_2$ ,  $f_3$ ,  $f_6$ ,  $f_8$ ,  $f_{10}$ ,  $f_{11}$  and  $f_{16}$ . 0.2 for  $\delta$  performs better on functions  $f_8$ ,  $f_{10}$ , and  $f_{17}$ , while it performs worse on functions  $f_2$ ,  $f_9$ ,  $f_{11}$ ,  $f_{14}$ ,  $f_{15}$ , and  $f_{16}$ . 0.3 for  $\delta$  performs better on functions  $f_1, f_2, f_4, f_5, f_7, f_{11}, f_{13}, f_{15}$ , and  $f_{17}$  while it performs worse on functions  $f_6$  and  $f_{12}$ . 0.4 for  $\delta$  performs better on functions  $f_3, f_5, f_6, f_{12}, f_{13}$ , and  $f_{14}$ , while it performs worse on functions  $f_2, f_7, f_8, f_9$  and  $f_{15}$ . 0.5 for  $\delta$  performs better on functions  $f_1, f_2, f_3, f_{10}, f_{12}$ , and  $f_{16}$ , while it performs worse on functions  $f_4, f_5, f_7, f_9$  and  $f_{14}$ . 0.6 for  $\delta$  performs better on functions  $f_1, f_9, f_{11}$ , and  $f_{15}$ , while it performs worse on functions  $f_4, f_5, f_7, f_{10}, f_{12}, f_{13}$ , and  $f_{16}$ . 0.7 for  $\delta$  performs better on functions  $f_4, f_5, f_6, f_8, f_{14}$  and  $f_{18}$ , while it performs worse on functions  $f_3, f_{12}, f_{13}$  and  $f_{17}$ . 0.8 for  $\delta$  performs better on functions  $f_3, f_7, f_{12}$  and  $f_{16}$ , while it performs worse on functions  $f_1, f_5, f_6, f_8, f_{10}, f_{11}, f_{13}, f_{14}$ , and  $f_{16}$ . 0.9 for  $\delta$  performs better on functions  $f_2, f_6, f_8, f_9, f_{10}, f_{11}, f_{13}, f_{14}$ , and  $f_{16}$ , while it performs worse on functions  $f_1, f_4, f_{15}$ , and  $f_{17}$ . Comprehensively considered with every factor, it can be found that the value for  $\delta$  should be set as 0.3 for EPSDE-Niche. And the result is shown in Table 12.

We all know that the bigger  $F$  or the smaller  $Cr$  is, the higher exploration capability of DE is. Regarding to the clearing niche mechanism, the smaller the value of clearing radius  $\delta$  is, the higher exploration capability of DE is. And the larger the value of clearing radius  $\delta$  is, the higher exploitation capability of DE is. The recommendation of cleaing radiuses is 0.2 when  $F = 0.5, Cr = 0.9$ , while  $\delta$  is 0.7 when  $F = 0.5, Cr = 0.3$ . It shows that when the values

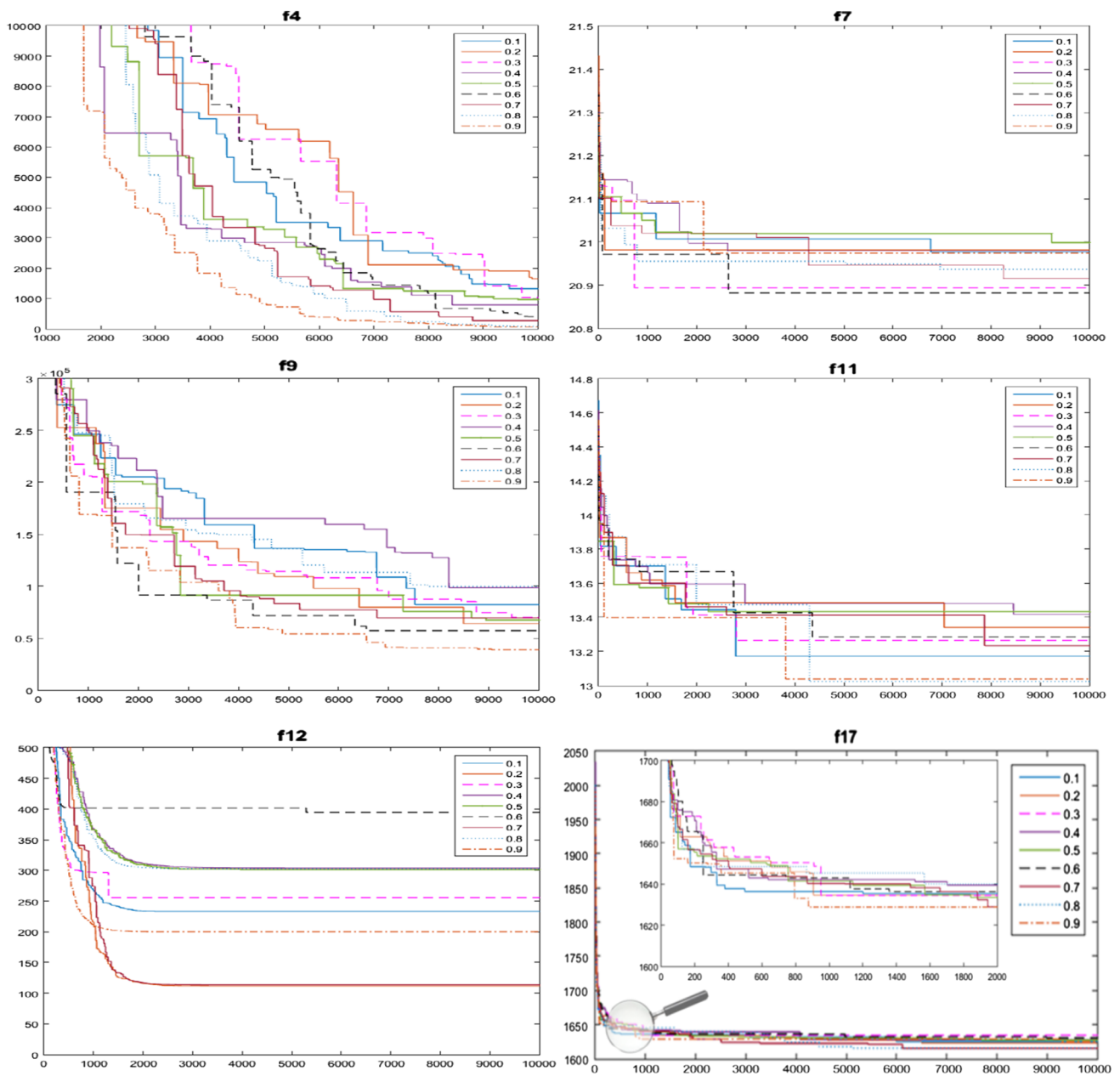


Fig. 9 DE-Niche with different clearing radius when  $F = 0.5, Cr = 0.3$

Table 7 Results on DE-Niche with different clearing radius when  $F$  and  $Cr$  are set as 0.5 and 0.3

Rank	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$	$f_{16}$	$f_{17}$
<i>Best</i>																	
1	0.4	0.9	0.2	0.9	0.8	0.9	0.6	0.6	0.9	0.5	0.8	0.2	0.3	0.6	0.2	0.9	0.8
2	0.2	0.8	0.8	0.8	0.7	0.4	0.3	0.5	0.6	0.1	0.9	0.7	0.1	0.1	0.3	0.8	0.7
3	0.5	0.7	0.1	0.7	0.6	0.5	0.7	0.2	0.2	0.3	0.1	0.9	0.5	0.4	0.6	0.7	0.2
<i>Worst</i>																	
1	0.7	0.3	0.9	0.2	0.2	0.1	0.5	0.9	0.8	0.6	0.5	0.6	0.8	0.5	0.9	0.5	0.3
2	0.9	0.4	0.6	0.1	0.5	0.7	0.2	0.7	0.4	0.8	0.4	0.4	0.9	0.3	0.5	0.2	0.6
3	0.3	0.2	0.4	0.3	0.3	0.6	0.1	0.8	0.1	0.7	0.2	0.8	0.7	0.8	0.8	0.1	0.5

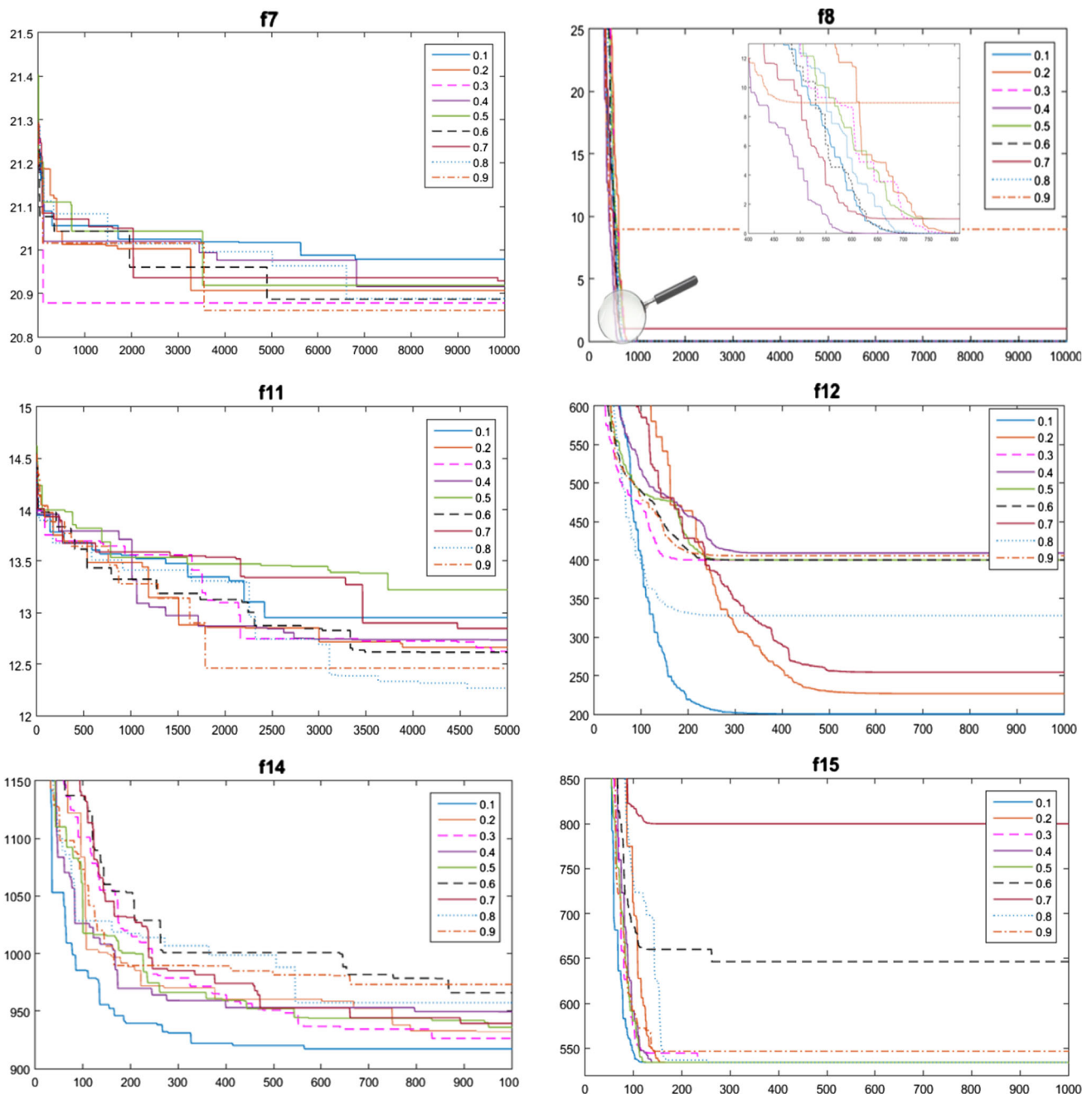
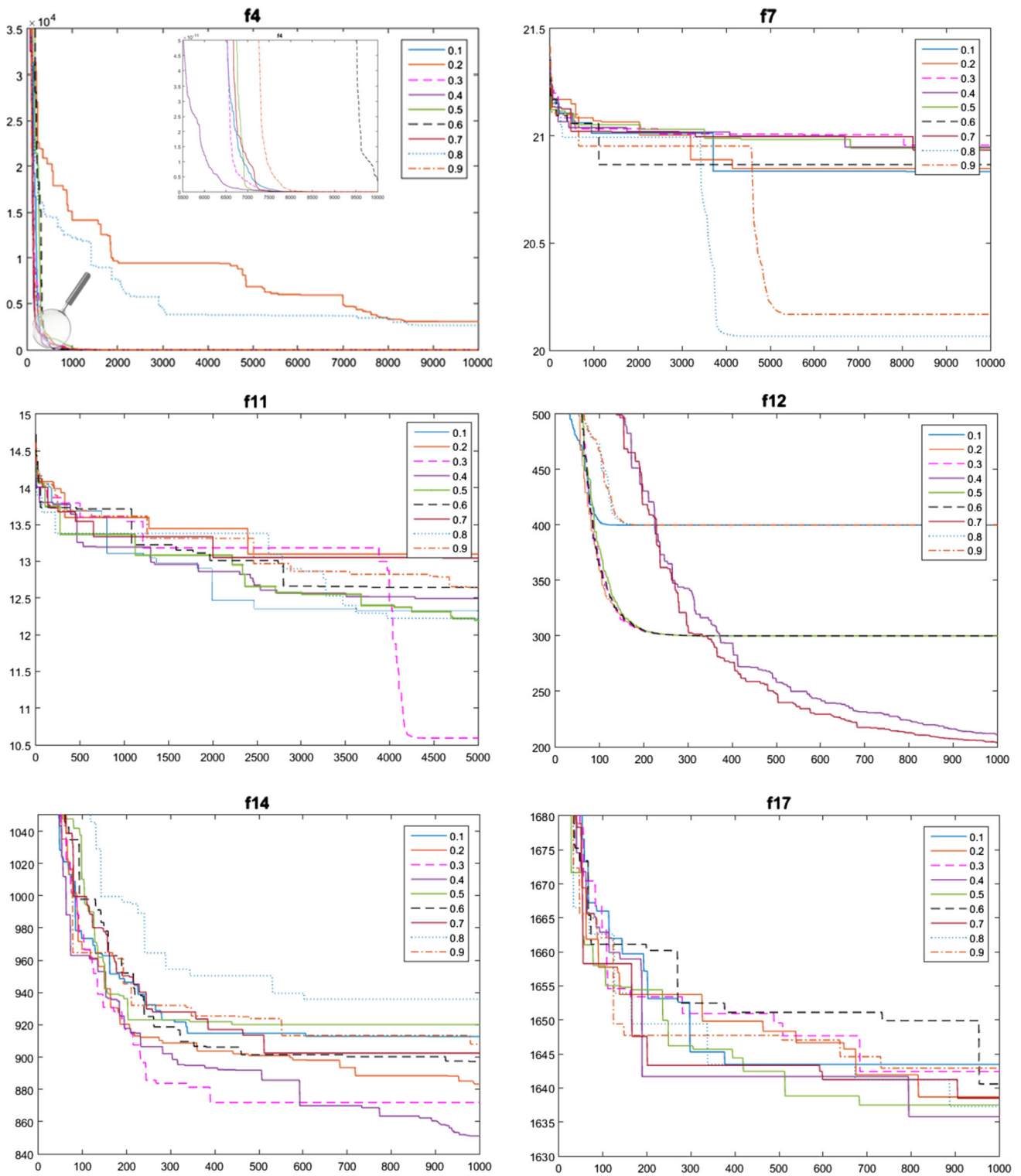


Fig. 10 jDE-Niche with different clearing radius

Table 8 Results on jDE-Niche with different clearing radius

Rank	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$	$f_{16}$	$f_{17}$
<i>Best</i>																	
1	0.9	0.1	0.7	0.3	0.8	0.4	0.9	0.6	0.7	0.8	0.8	0.1	0.2	0.1	0.1	0.2	0.3
2	0.8	0.6	0.5	0.1	0.6	0.5	0.3	0.1	0.6	0.7	0.9	0.2	0.1	0.3	0.5	0.1	0.1
3	0.3	0.3	0.6	0.8	0.7	0.3	0.6	0.2	0.1	0.6	0.6	0.7	0.3	0.2	0.4	0.3	0.2
<i>Worst</i>																	
1	0.2	0.8	0.1	0.9	0.3	0.7	0.1	0.9	0.8	0.2	0.7	0.4	0.8	0.9	0.7	0.4	0.9
2	0.5	0.9	0.2	0.4	0.4	0.8	0.7	0.5	0.9	0.9	0.1	0.9	0.7	0.6	0.6	0.8	0.6
3	0.1	0.5	0.3	0.6	0.5	0.9	0.5	0.7	0.2	0.1	0.5	0.5	0.9	0.8	0.9	0.6	0.8





**Fig. 11** JADE-Niche with different clearing radius

**Table 9** Results on JADE-Niche with different clearing radius

Rank	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$	$f_{16}$	$f_{17}$
<i>Best</i>																	
1	0.3	0.1	0.1	0.4	0.2	0.9	0.8	0.2	0.3	0.6	0.3	0.7	0.3	0.4	0.1	0.1	0.4
2	0.6	0.4	0.4	0.3	0.3	0.2	0.9	0.6	0.4	0.1	0.8	0.4	0.1	0.3	0.2	0.5	0.8
3	0.1	0.3	0.9	0.1	0.5	0.8	0.1	0.7	0.5	0.4	0.5	0.6	0.8	0.2	0.7	0.8	0.5
<i>Worst</i>																	
1	0.2	0.8	0.5	0.2	0.8	0.5	0.3	0.1	0.9	0.9	0.7	0.9	0.6	0.8	0.9	0.6	0.1
2	0.9	0.2	0.8	0.8	0.6	0.7	0.5	0.3	0.6	0.3	0.2	0.8	0.9	0.5	0.6	0.7	0.9
3	0.5	0.7	0.7	0.6	0.7	0.3	0.4	0.8	0.7	0.7	0.9	0.1	0.4	0.1	0.8	0.3	0.3

of  $F$  is equal, the smaller the value of  $Cr$  is, the bigger the value of clearing radius  $\delta$  is. Therefore, it maintains a balance between the exploitation capability and the exploration capability. Since jDE-Niche, JADE-Niche, SaDE-Niche and EPSEDE-Niche are self-adaptive algorithms, it is hard to analyze the relationship between  $F$ ,  $Cr$  and clearing radius  $\delta$ . Thus, we do not know how it maintains a balance between the exploitation capability and the exploration capability.

In addition, do a longitudinal analysis, it is also hard to distinguish which value of clearing radius  $\delta$  suits which function for different algorithms own different value of clearing radius  $\delta$  when the function is same. So the values of clearing radius  $\delta$  are not only based on the functions but also the algorithms.

### 5.2 Multimodal study

In this section, eight multimodal functions are selected as the test functions. The study focuses on finding all the global optimal solutions. Here, the experimental setup is shown first. Then, the comparisons between DE changed with new mutation strategies and the corresponding DE algorithms are made.

#### 5.2.1 Experimental setup

To make a fair comparison, the parameters for all the DE algorithms are set as follows unless a change is mentioned.

- (1) Dimension of the functions ( $D$ ): 2.
- (2) Population size ( $NP$ ): 50.
- (3) Maximal number of fitness function evaluations: 10000.
- (4) Number of runs: 25.
- (5) Scaling Factor ( $F$ ) and Crossover Control Parameter ( $Cr$ ): For two basic DE algorithms,  $F$  and  $Cr$  are (0.5, 0.9) and (0.5, 0.3), respectively. For four advanced DE variants,  $F$  and  $Cr$  are set as mentioned in [Brest and Mernik \(2008\)](#), [Qin and Suganthan \(2005\)](#), [Mallipeddi et al. \(2011\)](#), [Zhang and Sanderson \(2009\)](#).

- (6) Clearing radius ( $\delta$ ): The clearing radiuses for the algorithms are shown in [Table 12](#). And, for  $F1$ ,  $F2$ ,  $F3$ , and  $F5$ , the clearing radiuses for jDE-Niche is set as 0.5, while for  $F4$ ,  $F6$ ,  $F7$ , and  $F8$ , the clearing radiuses for jDE-Niche is set as 0.1.
- (7) Level of accuracy ( $\epsilon$ ): For functions  $F1$ ,  $F2$ ,  $F3$  and  $F7$ , the level of accuracy is set as {1.0E-03, 1.0E-04, 1.0E-05, 1.0E-06}. For function  $F4$ , the level of accuracy is set as {1.0E-01, 1.0E-02, 1.0E-03, 1.0E-04}. For functions  $F5$ , and  $F8$ , the level of accuracy is set as {1.0E-02, 1.0E-03, 1.0E-04, 1.0E-05}. For function  $F6$ , the level of accuracy is set as {5.0E-01, 3.0E-01, 1.0E-01, 1.0E-02}.

#### 5.2.2 Evaluation criterions

The performance of all multimodal algorithms is measured by the following two criterions:

(1) Peak Ratio (PR): the average percentage of all known global optima found over multiple runs. The high peak ratio indicates the versatility of the algorithm against variety of the functional landscapes.

$$PR = \frac{\sum_{i=1}^{NR} NPF_i}{NKP * NR} \tag{49}$$

where  $NPF_i$  denotes the number of global optima found in the end of the  $i$ th run,  $NKP$  denotes the number of known global optima, and  $NR$  the number of runs.

(2) Success Rate (SR): the percentage of successful runs (a successful run is defined as a run where all known global optimal solutions are found) out of all runs. The high values of success rate indicate the fine search performed by the algorithms as well as the tendency to preserve the detected peaks against the drifting forces.

$$SR = \frac{NSR}{NR} \tag{50}$$

where  $NSR$  denotes the number of successful runs.

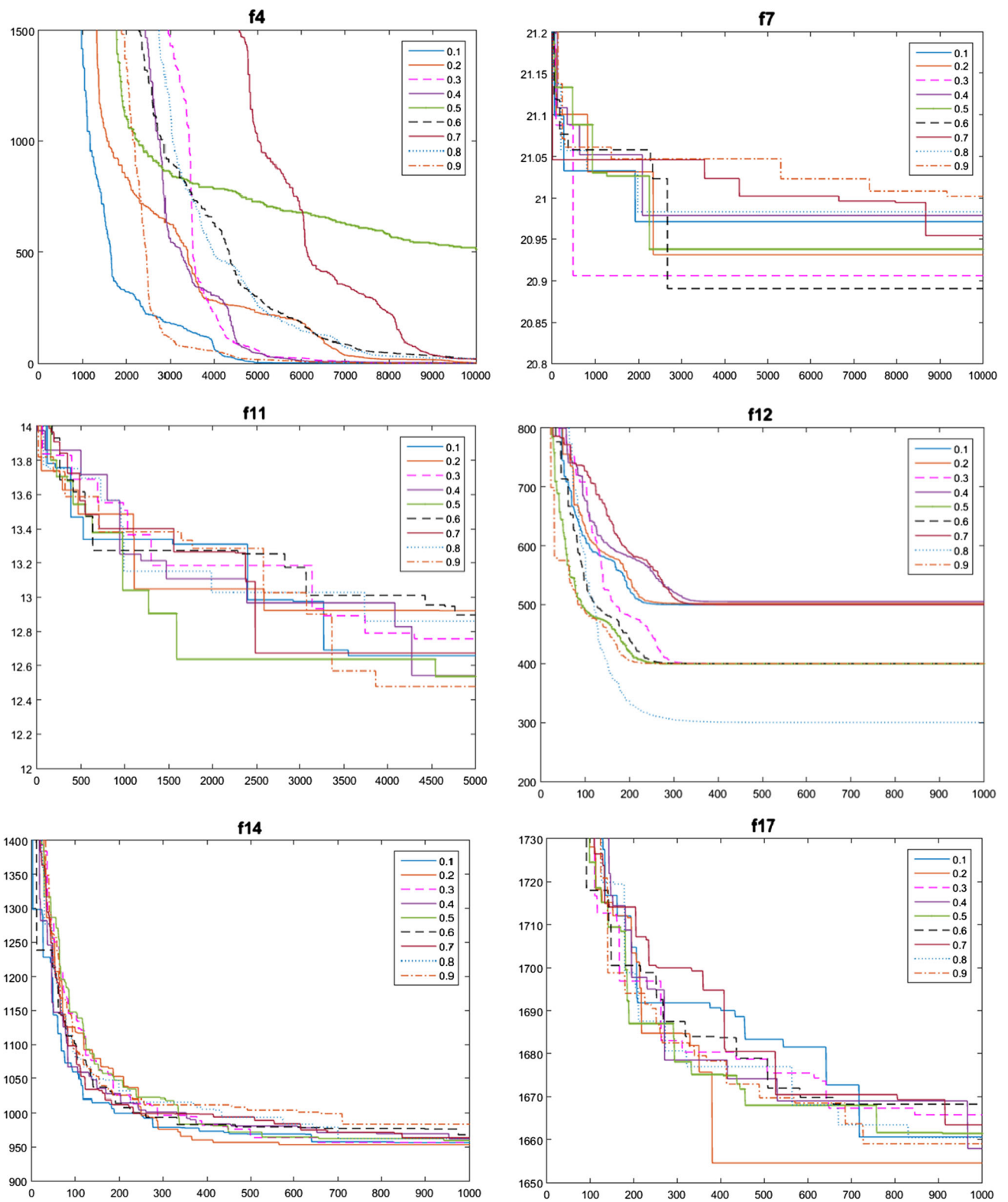


Fig. 12 SaDE-Niche with different clearing radius

**Table 10** Results on SaDE-Niche with different clearing radius

Rank	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$	$f_{16}$	$f_{17}$
<i>Best</i>																	
1	0.2	0.6	0.7	0.1	0.3	0.8	0.6	0.3	0.4	0.7	0.9	0.8	0.4	0.2	0.3	0.1	0.2
2	0.5	0.5	0.5	0.9	0.7	0.7	0.3	0.1	0.2	0.4	0.5	0.9	0.7	0.3	0.2	0.7	0.4
3	0.8	0.2	0.6	0.4	0.6	0.5	0.2	0.6	0.8	0.9	0.4	0.5	0.2	0.1	0.9	0.6	0.9
<i>Worst</i>																	
1	0.4	0.9	0.4	0.5	0.4	0.2	0.9	0.8	0.1	0.3	0.2	0.7	0.1	0.9	0.5	0.8	0.6
2	0.1	0.1	0.3	0.7	0.8	0.1	0.8	0.9	0.6	0.1	0.6	0.4	0.8	0.6	0.8	0.3	0.3
3	0.9	0.3	0.9	0.6	0.9	0.4	0.4	0.5	0.3	0.8	0.8	0.2	0.6	0.4	0.1	0.9	0.7

5.2.3 Result and analysis

In this section, DE-Niche, jDE-Niche, JADE-Niche, SaDE-Niche, and EPSDE-Niche comparing with DE, jDE, JADE, SaDE, and EPSDE are evaluated when they are used for handling multimodal optimization problems. To verify the effectiveness of the new mutation strategies, we employ eight classic multimodal benchmark functions.

In Tables 13, 14, and 15, the success rate and peak ratio of 10 algorithms are reported respectively. The high values of success rate indicate the fine searches are performed by the algorithms, as well as the tendency is observed to preserve the detected peaks against the drifting forces. The high peak ratio indicates the versatility of the algorithm against variety of the functional landscapes. Here, we mark the results that exhibits better performance measured by SR or PR with **bold-face** in Tables 13, 14 and 15. Based on the experimental results, for the majority of the considered functions, the DE algorithms with the new mutation strategies based on clearing niche method show better or similar performances measured by both SR and PR, compared with the DE algorithms with common mutation strategies. Specifically, in the first three functions, all algorithms behave similarly, exhibiting best performance independently of the accuracy level, except the basic DE with DE/best/1 strategy performs worse on function  $F_1$ . This behavior changes in the next functions, where the DE algorithms with the new mutation strategies based on clearing niche method clearly outperform the DE algorithms with the common mutation strategies on both PR and SR measures for functions  $F_4, F_6, F_7,$  and  $F_8,$  but underperform for functions  $F_5$ . It is worth noting that the DE algorithms with the new mutation strategies based on clearing niche method exhibits significantly great performance gains for the majority of the functions aforementioned for almost all accuracy levels.

In Table 16, it can be observed that DE-Niche achieves the best total rank of 57.5 in SR and 52 in PR when  $F$  and  $Cr$  are 0.5, 0.9. It achieves the best total rank of 58 in SR and 53 in PR when  $F$  and  $Cr$  are 0.5 and 0.3, respectively. jDE-Niche achieves the best total rank of 46.5 in SR and 43.5 in PR. SaDE-Niche achieves the best total rank of 46.5 in SR and 46 in PR. EPSDE-Niche achieves the best total rank of 46.5 in

SR and 43.5 in PR. However, JADE-Niche achieves the best total rank of 47.5 in SR, while JADE achieves the best total rank of 47 in PR. Overall the DE algorithms with the new mutation strategies based on clearing niche method perform better than the DE algorithms with the common mutation strategies when it maintains both PR and SR measure.

5.3 Study for EED problem

In this section, the present formulation treats economic environmental dispatch (EED) problem as a multi-objective mathematical programming problem is selected to test the performance of the new strategies. Here, the introduction of the multiobjective problem is given first. Then, the experimental setup is shown. Lastly, the experimental result and analysis are made.

5.3.1 Problem formulation

The EED problem attempts to optimize both cost and emission simultaneously, while satisfying both equality and inequality constraints. The following objectives and constraints are taken into account in the formulation of EED problem.

5.3.1.1 Objectives

(1) Cost

The fuel cost function of each fossil fuel-fired generator, considering the valve-point effect (Walters and Sheble 1993), is expressed as the sum of a quadratic and a sinusoidal function.

The total fuel cost in terms of real power output can be expressed as Eq. (51).

$$F = \sum_{i=1}^N \left[ a_i + b_i P_i + c_i P_i^2 + \left| d_i \sin \left\{ e_i \left( P_i^{\min} - P_i \right) \right\} \right| \right] \tag{51}$$

where  $P_i$  is the power output of  $i$ th unit,  $P_i^{\min}$  is the lower generation limits for  $i$ th unit.  $a_i, b_i, c_i, d_i,$  and  $e_i$  are the cost coefficients of  $i$ th unit.  $N$  is the number of generating units.

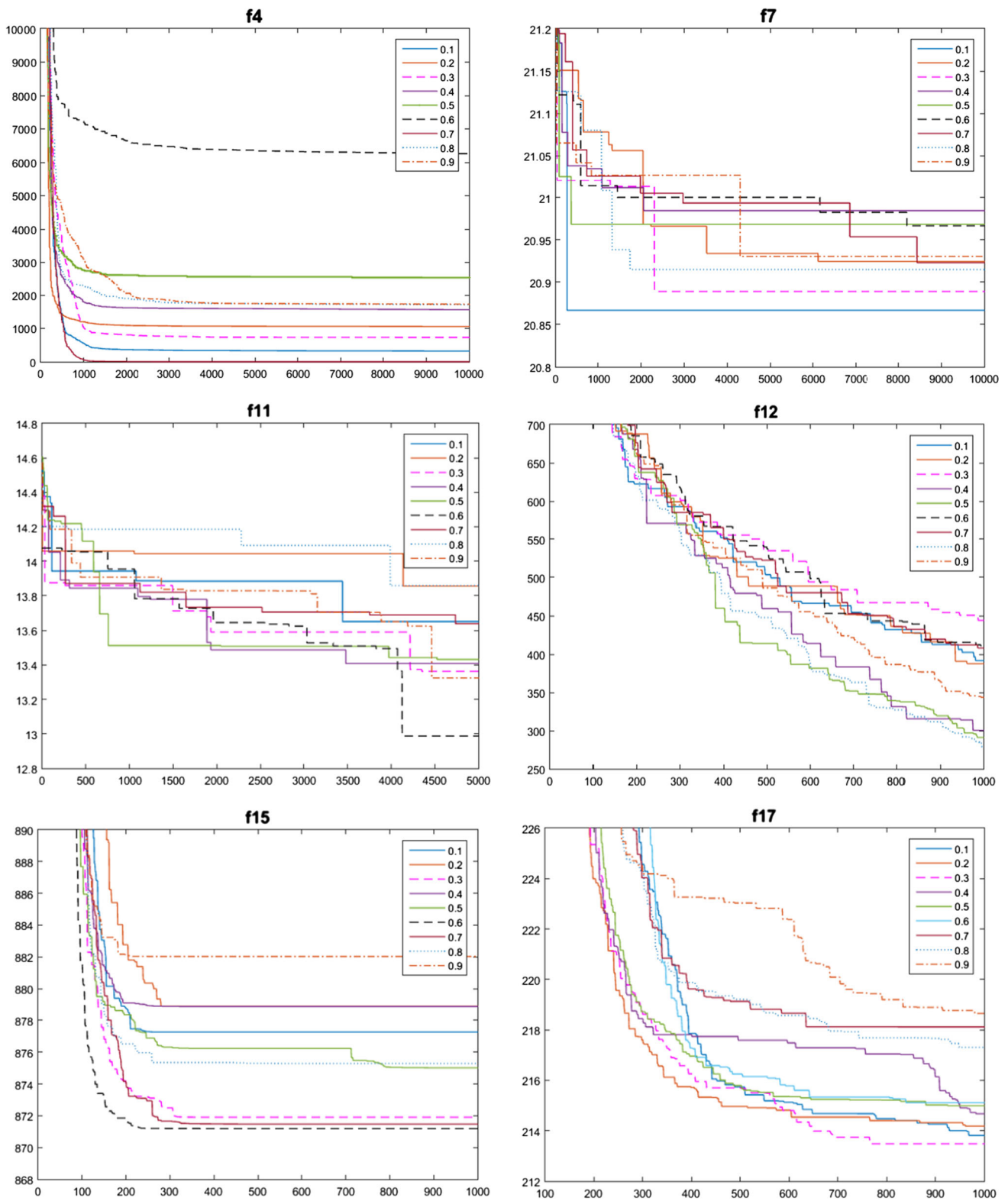


Fig. 13 EPSDE-Niche with different clearing radius



**Table 11** Results on EPSDE-Niche with different clearing radius

Rank	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$	$f_{16}$	$f_{17}$
<i>Best</i>																	
1	0.5	0.9	0.4	0.7	0.7	0.4	0.1	0.2	0.1	0.2	0.6	0.8	0.4	0.9	0.6	0.5	0.3
2	0.3	0.5	0.8	0.1	0.3	0.9	0.3	0.9	0.9	0.9	0.9	0.5	0.9	0.4	0.7	0.9	0.1
3	0.6	0.3	0.5	0.3	0.4	0.7	0.8	0.7	0.6	0.5	0.3	0.4	0.3	0.7	0.3	0.8	0.2
<i>Worst</i>																	
1	0.9	0.1	0.7	0.6	0.5	0.1	0.4	0.8	0.2	0.1	0.8	0.3	0.8	0.5	0.9	0.1	0.9
2	0.8	0.2	0.1	0.5	0.6	0.3	0.5	0.1	0.4	0.6	0.2	0.6	0.6	0.8	0.2	0.6	0.7
3	0.1	0.4	0.6	0.9	0.8	0.8	0.6	0.4	0.5	0.8	0.1	0.7	0.7	0.2	0.4	0.2	0.8

**Table 12** Recommended clearing radiuses for the algorithms

Algorithms	Function	Parameters	$\delta$
DE-Niche	$f_1 - f_{17}$	$F = 0.5, Cr = 0.9$	0.2
		$F = 0.5, Cr = 0.3$	0.7
jDE-Niche	$f_1 - f_9$ $f_{10} - f_{17}$	Self-adaptive	0.5
		Self-adaptive	0.1
JADE-Niche	$f_1 - f_{17}$	Self-adaptive	0.4
SaDE-Niche	$f_1 - f_{17}$	Self-adaptive	0.2
EPSDE-Niche	$f_1 - f_{17}$	Self-adaptive	0.3

(2) Emission

The atmospheric pollutants such as sulfur oxides (SOx), nitrogen oxides (NOx) and carbon dioxide (CO2) caused by fossil fuel-fired generator can be modeled separately. However, for comparison purposes, the total emission of these pollutants which is the sum of a quadratic and an exponential function (Basu 2011) can be expressed as Eq. (52).

$$E = \sum_{i=1}^N [\alpha_i + \beta_i P_i + \gamma_i P_i^2 + \eta_i \exp(\delta_i P_i)] \tag{52}$$

where  $P_i$  is the power output of  $i$ th unit.  $\alpha_i, \beta_i, \gamma_i, \eta_i,$  and  $\delta_i$  are the emission coefficients of  $i$ th unit.  $N$  is the number of generating units.

5.3.1.2 Constraints

(1) Real power balance constraint

The total real power generation must balance the predicted power demand plus the real power losses in the transmission lines.

$$\sum_{i=1}^N P_i - P_D - P_L = 0 \tag{53}$$

where  $P_D$  is the load demand.  $P_L$  is calculated by using  $B$  coefficients (the transmission loss coefficient) which can be expressed in the quadratic form as follows:

$$P_L = \sum_{i=1}^N \sum_{j=1}^N P_i B_{ij} P_j \tag{54}$$

(2) Real power operating limits

$$P_i^{min} \leq P_i \leq P_i^{max}, \quad i \in N$$

where  $P_i^{min}$  and  $P_i^{max}$  are the lower and upper generation limits for  $i$ th unit.

5.3.2 Experimental setup

To make a fair comparison, the parameters for all the DE algorithms are set as follows unless a change is mentioned.

- (1) Dimension of the functions ( $D$ ): 10.
- (2) Population size ( $NP$ ): 50.
- (3) Maximal number of fitness function evaluations: 1000.
- (4) Scaling Factor ( $F$ ) and Crossover Control Parameter ( $Cr$ ): For two basic DE algorithms,  $F$  and  $Cr$  are (0.5, 0.9) and (0.5, 0.3), respectively. For four advanced DE variants,  $F$  and  $Cr$  are set as mentioned in Brest and Mernik (2008), Qin and Suganthan (2005), Mallipeddi et al. (2011), Zhang and Sanderson (2009).
- (5) Clearing radius ( $\delta$ ): The clearing radiuses for the algorithms are shown in Table 12. And, the clearing radiuses for jDE-Niche is set as 0.5.
- (6) Load demand:  $P_D = 2000$  MW.

**Table 13** Peak ratios, success rates and rank of the basic DE algorithms and DE-Niche algorithm

Function	Accuracy level $\varepsilon$	$F = 0.5, Cr = 0.3$											
		$F = 0.5, Cr = 0.9$						$F = 0.5, Cr = 0.3$					
		DE/rand/1		DE/best/1		DE-Niche		DE/rand/1		DE/best/1		DE-Niche	
		SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
F1	1.0E-03	1.0 (1.5)	1.0 (1.5)	0.96 (3)	0.96 (3)	1.0 (1.5)	1.0 (1.5)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-04	1.0 (1.5)	1.0 (1.5)	0.92 (3)	0.92 (3)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	0.92 (3)	0.92 (3)	1.0 (1.5)	1.0 (1.5)
	1.0E-05	1.0 (1.5)	1.0 (1.5)	0.72 (3)	0.72 (3)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	0.8 (3)	0.8 (3)	1.0 (1.5)	1.0 (1.5)
	1.0E-06	1.0 (1.5)	1.0 (1.5)	0.72 (3)	0.72 (3)	1.0 (1.5)	1.0 (1.5)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-03	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-04	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
F2	1.0E-05	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-06	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-03	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-04	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-05	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-06	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
F3	1.0E-03	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-04	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-05	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-06	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-03	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
	1.0E-04	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)
F4	1.0E-01	0 (2.5)	0.39 (2)	0 (2.5)	0.25 (3)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	0.51 (2)	0 (3)	0.25 (3)	1.0 (1)
	1.0E-02	0 (2.5)	0.39 (2)	0 (2.5)	0.25 (3)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	0.63 (2)	0 (3)	0.25 (3)	1.0 (1)
	1.0E-03	0 (2.5)	0.33 (2)	0 (2.5)	0.25 (3)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	0.48 (2)	0 (3)	0.25 (3)	1.0 (1)
	1.0E-04	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)
	1.0E-02	0.60 (1)	0.60 (1)	0.12 (3)	0.12 (3)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	0.84 (1)	0.24 (3)	0.24 (3)	0.44 (2)
	1.0E-03	0.88 (1)	0.88 (1)	0.16 (3)	0.16 (3)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	0.76 (1)	0.28 (3)	0.28 (3)	0.48 (2)
F5	1.0E-04	0.60 (1)	0.60 (1)	0.16 (3)	0.16 (3)	1.0 (2)	1.0 (2)	1.0 (2)	1.0 (2)	0.84 (1)	0.32 (3)	0.32 (3)	0.44 (2)
	1.0E-05	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)
	5.0E-01	0 (2.5)	0.27 (3)	0.24 (1)	0.67 (1)	0 (2.5)	0.50 (2)	0 (2.5)	0 (2.5)	0.08 (1)	0.53 (1)	0 (2.5)	0.49 (2)
	3.0E-01	0 (2)	0.35 (3)	0 (2)	0.51 (1)	0 (2)	0.50 (2)	0 (2)	0.26 (3)	0 (2)	0.34 (2)	0 (2)	0.48 (1)
	1.0E-01	0 (2)	0.30 (2)	0 (2)	0.29 (3)	0 (2)	0.50 (1)	0 (2)	0.28 (2)	0 (2)	0.26 (3)	0 (2)	0.48 (1)
	1.0E-02	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)
F6	1.0E-03	0 (2)	0.28 (2)	0 (2)	0.25 (3)	0 (2)	0.40 (1)	0.04 (2)	0.40 (2)	0 (2)	0.25 (3)	0.12 (1)	0.45 (1)
	1.0E-04	0 (1.5)	0.28 (2)	0 (1.5)	0.25 (3)	0.04 (1)	0.40 (1)	0 (1.5)	0.35 (2)	0 (1.5)	0.25 (3)	0.24 (1)	0.55 (1)
	1.0E-05	0 (1.5)	0.28 (2)	0 (1.5)	0.25 (3)	0.08 (1)	0.42 (1)	0.04 (1.5)	0.43 (1)	0 (3)	0.25 (3)	0.04 (1.5)	0.37 (2)
	1.0E-06	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)
	1.0E-02	0 (2)	0.06 (2)	0 (2)	0.05 (3)	0 (2)	0.09 (1)	0 (2)	0.10 (2)	0 (2)	0.05 (3)	0 (2)	0.14 (1)
	1.0E-03	0 (2)	0.07 (2)	0 (2)	0.05 (3)	0 (2)	0.10 (1)	0 (2)	0.08 (2)	0 (2)	0.05 (3)	0 (2)	0.13 (1)
F7	1.0E-04	0 (2)	0.02 (1.5)	0 (2)	0.02 (1.5)	0 (2)	0.03 (1)	0 (2)	0.04 (2)	0 (2)	0.01 (3)	0 (2)	0.08 (1)
	1.0E-05	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)
	1.0E-03	0 (2)	0.02 (1.5)	0 (2)	0.02 (1.5)	0 (2)	0.03 (1)	0 (2)	0.04 (2)	0 (2)	0.01 (3)	0 (2)	0.08 (1)
	1.0E-04	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)
	1.0E-05	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)
	1.0E-06	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)	0 (2)

**Table 14** Peak ratios, success rates and rank of the jDE, JADE, and SaDE algorithms and their corresponding DE variants with new mutation strategies

Function	Accuracy level $\varepsilon$	jDE		jDE_Niche		JADE		JADE_Niche		SaDE		SaDE_Niche	
		SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR
F1	1.0E-03	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-04	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-05	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-06	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-03	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-04	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
F2	1.0E-05	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-06	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-03	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-04	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-05	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-06	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
F3	1.0E-05	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-06	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-03	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-04	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-05	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
	1.0E-06	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
F4	1.0E-01	0.08 (2)	0.45 (2)	<b>0.88 (1)</b>	<b>0.94 (1)</b>	0.04 (2)	0.48 (2)	<b>0.48 (1)</b>	<b>0.74 (1)</b>	0 (1.5)	0 (1.5)	0 (1.5)	0.25 (1.5)
	1.0E-02	0.08 (2)	0.44 (2)	<b>0.86 (1)</b>	<b>0.98 (1)</b>	0.12 (2)	0.52 (2)	<b>0.6 (1)</b>	<b>0.77 (1)</b>	0 (1.5)	0 (1.5)	0 (1.5)	0.25 (1.5)
	1.0E-03	0.16 (2)	0.52 (2)	<b>0.88 (1)</b>	<b>0.94 (1)</b>	0.16 (2)	0.54 (2)	<b>0.6 (1)</b>	<b>0.81 (1)</b>	0 (1.5)	0 (1.5)	0 (1.5)	0.25 (2)
	1.0E-04	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
	1.0E-02	<b>0.80 (1)</b>	<b>0.80 (1)</b>	0.32 (2)	0.32 (2)	<b>0.88 (1)</b>	<b>0.88 (1)</b>	0.64 (2)	0.64 (2)	<b>0.8 (1)</b>	<b>0.8 (1)</b>	0.52 (2)	0.52 (2)
	1.0E-03	<b>0.88 (1)</b>	<b>0.88 (1)</b>	0.32 (2)	0.32 (2)	<b>0.76 (1)</b>	<b>0.76 (1)</b>	0.72 (2)	0.72 (2)	0.72 (2)	<b>0.76 (1)</b>	<b>0.76 (1)</b>	<b>0.76 (1)</b>
F5	1.0E-04	<b>0.56 (1)</b>	<b>0.56 (1)</b>	0.32 (2)	0.32 (2)	<b>0.76 (1)</b>	<b>0.76 (1)</b>	0.72 (2)	0.72 (2)	0.72 (2)	<b>0.76 (1)</b>	<b>0.76 (1)</b>	0 (1.5)
	1.0E-05	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
	5.0E-01	0 (1.5)	0.27 (2)	0 (1.5)	<b>0.39 (1)</b>	0 (1.5)	0.25 (2)	0 (1.5)	0.27 (1)	0 (1.5)	0 (1.5)	0 (1.5)	<b>0.39 (1)</b>
	3.0E-01	0 (1.5)	0.25 (2)	0 (1.5)	<b>0.43 (1)</b>	0 (1.5)	0.25 (2)	0 (1.5)	0.26 (1)	0 (1.5)	0 (1.5)	0 (1.5)	<b>0.33 (1)</b>
	1.0E-01	0 (1.5)	0.26 (2)	0 (1.5)	<b>0.38 (1)</b>	0 (1.5)	0.25 (2)	0 (1.5)	0.29 (1)	0 (1.5)	0 (1.5)	0 (1.5)	<b>0.36 (1)</b>
	1.0E-02	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
F6	1.0E-03	0.04 (2)	0.38 (2)	<b>0.92 (1)</b>	<b>0.97 (1)</b>	0 (1.5)	<b>0.31 (1)</b>	0 (1.5)	0.26 (2)	0.08 (1.5)	<b>0.43 (1)</b>	0.40 (2)	0.40 (2)
	1.0E-04	0.08 (2)	0.44 (2)	<b>1.0 (1)</b>	<b>1.0 (1)</b>	0 (2)	0.30 (1.5)	<b>0.04 (1)</b>	0.30 (1.5)	0 (2)	<b>0.04 (1)</b>	0.36 (1.5)	0.36 (1.5)
	1.0E-05	0.04 (2)	0.37 (2)	<b>0.92 (1)</b>	<b>0.96 (1)</b>	0.04 (1.5)	<b>0.33 (1)</b>	0.04 (1.5)	0.30 (2)	0.04 (2)	<b>0.08 (1)</b>	<b>0.08 (1)</b>	<b>0.39 (1)</b>
	1.0E-06	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
	1.0E-02	0 (1.5)	0.08 (2)	0 (1.5)	<b>0.17 (1)</b>	0 (1.5)	<b>0.15 (1)</b>	0 (1.5)	0.09 (2)	0 (1.5)	0 (1.5)	0 (1.5)	<b>0.06 (1)</b>
	1.0E-03	0 (1.5)	0.08 (2)	0 (1.5)	<b>0.16 (1)</b>	0 (1.5)	<b>0.16 (1)</b>	0 (1.5)	0.1 (2)	0 (1.5)	0 (1.5)	0 (1.5)	0.06 (1.5)
F7	1.0E-04	0 (1.5)	0.02 (2)	0 (1.5)	<b>0.05 (1)</b>	0 (1.5)	<b>0.07 (1)</b>	0 (1.5)	0.05 (2)	0 (1.5)	0 (1.5)	0 (1.5)	0.02 (1.5)
	1.0E-05	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
	1.0E-03	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
	1.0E-04	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
	1.0E-05	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
	1.0E-06	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
F8	1.0E-02	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
	1.0E-03	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
	1.0E-04	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
	1.0E-05	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
	1.0E-06	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
	1.0E-03	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)

**Table 15** Peak ratios, success rates and rank of the EPSDE algorithms and EPSDE\_Niche algorithms

Function	F1			F2			F3			
	Accuracy level $\varepsilon$			Accuracy level $\varepsilon$			Accuracy level $\varepsilon$			
	SR	PR	EPSDE	SR	PR	EPSDE	SR	PR	EPSDE	
1.0E-03	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
1.0E-04	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
1.0E-05	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
1.0E-06	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)	1.0 (1.5)
Function	F4			F5			F8			
1.0E-01	0 (2)	0.27 (2)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
1.0E-02	0 (2)	0.30 (2)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
1.0E-03	0 (2)	0.27 (2)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
1.0E-04	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
1.0E-05	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)
Function	F7			Accuracy level $\varepsilon$			F6			
1.0E-03	0.04 (2)	0.31 (2)	<b>1.0 (1)</b>	<b>1.0 (1)</b>	<b>1.0 (1)</b>	5.0E-01	0.32 (2)	0 (1.5)	<b>0.49 (1)</b>	0.32 (2)
1.0E-04	0 (2)	0.29 (2)	<b>1.0 (1)</b>	<b>1.0 (1)</b>	<b>1.0 (1)</b>	3.0E-01	0.36 (2)	0 (1.5)	<b>0.48 (1)</b>	0.36 (2)
1.0E-05	0 (2)	0.27 (2)	<b>1.0 (1)</b>	<b>1.0 (1)</b>	<b>1.0 (1)</b>	1.0E-01	0.31 (2)	0 (1.5)	<b>0.50 (1)</b>	0.31 (2)
1.0E-06	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)	1.0E-02	0 (1.5)	0 (1.5)	0 (1.5)	0 (1.5)

**Table 16** Total ranks for all algorithms

Algorithms		SR	PR	Algorithms		SR	PR
DE	$F = 0.5, Cr = 0.9$	DE/rand/1	60	60.5	jDE-Niche	46.5	43.5
		DE/best/1	70.5	77.5	JADE	48.5	47
		DE-Niche	57.5	52	JADE- Niche	47.5	49
DE	$F = 0.5, Cr = 0.3$	DE/rand/1	59.5	61	SaDE	49.5	49
		DE/best/1	72.5	78	SaDE-Niche	46.5	46
		DE-Niche	58	53	EPSDE	49.5	52.5
jDE		49.5	52.5	EPSDE-Niche	46.5	43.5	

**Table 17** Power output, cost, and emission of ten-unit system by the basic DE algorithms and DE-Niche algorithm

Power output of <i>i</i> th unit	$F = 0.5, Cr = 0.9$					$F = 0.5, Cr = 0.3$		
	DE/rand/1		EED	DE/best/1	DE-Niche	DE/rand/1	DE/best/1	DE-Niche
	Economic dispatch	Emission dispatch		EED	EED	EED	EED	EED
P1	55	55	54.72209	54.90951	54.69379	54.98127	54.9912	54.85888
P2	80	80	79.80728	78.93928	79.0844	79.52119	79.48729	79.40966
P3	106.9396	81.13417	87.3326	94.38051	89.80176	90.84672	80.02748	84.61223
P4	100.5743	81.36374	84.69496	92.84528	85.10693	87.09652	81.7519	83.85574
P5	81.50925	160	114.2762	120.2293	116.2629	128.4365	131.7568	140.3557
P6	83.01569	240	151.5787	146.4247	149.0805	162.3995	162.948	150.034
P7	300	294.4851	295.0466	289.299	298.6134	295.0677	295.6985	298.1719
P8	340	297.2701	329.8471	322.306	326.337	304.5769	292.6717	307.7047
P9	470	396.7657	451.7851	439.9297	445.6074	437.2558	451.9699	445.3863
P10	470	395.5763	435.7389	445.3095	440.2065	444.0994	453.5933	440.1272
Cost(10 <sup>5</sup> \$)	1.11497	1.16412	1.12622	1.12621	1.12606	1.13084	1.13132	1.13075
Emission (lb)	4612.18	3972.24	4264.56	4267.28	4263.24	4200.36	4205.82	4197.79

5.3.3 Result and analysis

This system consists of ten generating units with non-smooth fuel cost and emission level functions. Unit data and loss coefficients are given in ‘‘Appendix’’.

Solutions by using basic DE and DE-Niche on EED problem are shown in Table 17. During cost minimization, fuel cost is  $1.11497 \times 10^5$ \$ and emission is 4612.18 lb. But cost increases to  $1.16412 \times 10^5$ \$ and emission decreases to 3972.24 lb in case of emission minimization. Results obtained from DE/rand/1, DE/best/1, and DE-Niche, are summarized in Table 17. In case of EED by using DE-Niche when  $F = 0.5$  and  $Cr = 0.9$ , fuel cost is  $1.12606 \times 10^5$ \$ which is more than  $1.11497 \times 10^5$ \$ and less than  $1.16412 \times 10^5$ \$, and emission is 4263.24 lb which is less than 4612.18 lb and more than 3972.24 lb. By using DE/rand/1, fuel cost is  $1.12622 \times 10^5$ \$ which is more than  $1.12606 \times 10^5$ \$ and emission is 4264.56 lb which is more than 4263.24 lb. By using DE/best/1, fuel cost is  $1.12621 \times 10^5$ \$ which is more than  $1.12606 \times 10^5$ \$ and emission is 4267.28 lb which is

more than 4263.24 lb. It can be found that, the values of both fuel cost and emission by using DE-Niche are less than the values by using DE/rand/1 and DE/best/1. Likewise, it can be found that the conclusion is applicable to the results when  $F = 0.5$  and  $Cr = 0.3$ .

Solutions by using jDE, JADE, SaDE, EPSDE, jDE-Niche, JADE-Niche, SaDE- Niche, and EPSDE-Niche on EED problem are shown in Table 18. In case of EED by using jDE, fuel cost is  $1.12612 \times 10^5$ \$, and emission is 4263.96 lb. When by using jDE-Niche, fuel cost is  $1.12625 \times 10^5$ \$ which is more than  $1.12612 \times 10^5$ \$, but emission is 4262.36 lb which is less than 4263.96 lb. These two solutions do not dominate each other. Similarly, the solutions by using JADE and JADE-Niche do not dominate each other. At the same time, in case of EED by using SaDE, fuel cost is  $1.13763 \times 10^5$ \$, and emission is 4291.64 lb. When by using SaDE-Niche, fuel cost is  $1.13543 \times 10^5$ \$ which is less than  $1.13763 \times 10^5$ \$, and emission is 4198.39 lb which is less than 4291.64 lb. It shows that, the values of both fuel cost and emission by using SaDE-Niche are less than the val-



**Table 18** Power output, cost, and emission of ten-unit system by jDE, JADE, SaDE, and EPSDE algorithms and their corresponding DE variants with new mutation strategies

Power output of <i>i</i> th unit	jDE	jDE_Niche	JADE	JADE_Niche	SaDE	SaDE_Niche	EPSDE	EPSDE_Niche
	EED	EED	EED	EED	EED	EED	EED	EED
P1	54.84462	54.94983	54.97065	54.95015	29.40508	46.1734	36.17975	46.85479
P2	78.99054	79.24224	79.82228	79.48305	72.79229	71.09816	59.6031	71.79953
P3	83.60589	96.30204	85.04759	83.97245	102.2589	85.7957	93.19733	105.1295
P4	83.75415	87.41277	81.56497	85.5831	77.95818	80.60412	114.4035	83.52687
P5	124.5532	127.1113	120.6808	128.4333	158.7933	125.385	135.8526	133.1242
P6	143.7592	139.2127	148.6987	136.4341	151.0108	181.0122	193.5948	201.308
P7	294.4362	298.132	294.7467	297.2601	271.6371	295.5609	261.4637	269.4317
P8	324.6998	318.79	322.2707	325.436	339.3171	325.5314	286.1742	296.2601
P9	432.7897	446.5679	444.229	454.0236	425.3794	446.4012	448.4764	454.4499
P10	463.724	436.895	453.0648	439.4411	455.6692	426.5131	454.6876	421.4889
Cost(10 <sup>5</sup> \$)	1.12612	1.12625	1.12628	1.12580	1.13763	1.13543	1.14286	1.14283
Emission (lb)	4263.96	4262.36	4257.48	4265.05	4291.64	4198.39	4290.01	4173.79

ues by using SaDE. Similarly, fuel cost and emission by using EPSDE are both less than fuel cost and emission by using.

From the comparison of solutions between jDE, JADE, SaDE, and EPSDE and their corresponding DE variants jDE-Niche, JADE-Niche, SaDE-Niche, and EPSDE-Niche, it shows that the solutions of DE whose mutation strategy is based on the clearing niche mechanism are better as a whole. Therefore, the new mutation strategies can improve the performance of most DE variants on solving EED.

## 6 Conclusion

In this work, the clearing niche mechanism has been incorporated into several mutation strategies, which are used on two basic DE algorithms, and four improved DE variants. Through evaluating and comparing the effectiveness of DE variants with new mutation strategies with basic DE algorithms, four improved DE variants, it can confirmed that new mutation strategies based on the clearing niche mechanism can improve the performance of most DE algorithms studied. In addition, through the discussion of adjusting the clearing radius threshold, the exploitation capability and the exploration capability are maintained.

In the future, the present work could be extended in multiple directions. First of all, an adaptive clearing radius can be used in the algorithms. Secondly, other strategies based on the clearing niche mechanism will guide the search of DE. Thirdly, more niche mechanisms will be introduced to study the mutation strategies. Lastly, the DE variants with new strategies will be used to solve real-world prob-

lems. For instance, it can be applied to petroleum reservoir identification, location problem, feature selection and rule extraction.

**Acknowledgements** This research has been supported by National Natural Science Foundation of China under Grant Nos. 71103163, 71573237; New Century Excellent Talents in University of China under Grant No. NCET-13-1012; Research Foundation of Humanities and Social Sciences of Ministry of Education of China No. 15YJA630019; Special Funding for Basic Scientific Research of Chinese Central University under Grant Nos. CUG120111, CUG110411, G2012002A, CUG140604, CUG160605; Open Foundation for the Research Center of Resource Environment Economics in China University of Geosciences (Wuhan); Structure and Oil Resources Key Laboratory Open Project of China under Grant No. TPR-2011-11.

### Compliance with ethical standards

**Conflict of interest** Author Yanan Li declares that she has no conflict of interest. Author Haixiang Guo declares that he has no conflict of interest. Author Xiao Liu declares that she has no conflict of interest. Author Yijing Li declares that she has no conflict of interest. Author Wenwen Pan declares that she has no conflict of interest. Author Bing Gong declares that he has no conflict of interest. Author Shaoning Pang declares that he has no conflict of interest.

**Human and animals participants** This article does not contain any studies with human participants or animals performed by any of the authors.

**Informed consent** Informed consent was obtained from all individual participants included in the study.

## Appendix

See Table 19.

**Table 19** Ten-unit generator characteristics

Unit	$P_i^{max}$ (MW)	$P_i^{min}$ (MW)	$a_i$ (\$/h)	$b_i$ (\$/MWh)	$c_i$ (\$/ (MW) <sup>2</sup> h)	$d_i$ (\$/h)	$e_i$ (rad/ MW)	$\alpha_i$ (lb/ h)	$\beta_i$ (lb/ MWh)	$\gamma_i$ (lb/ (MW) <sup>2</sup> h)	$\eta_i$ (lb/ h)	$\delta_i$ (l/ MW)
1	55	10	1000.403	40.5407	0.12951	33	0.0174	360.0012	-3.9864	0.04702	0.25475	0.01234
2	80	20	950.606	39.5804	0.10908	25	0.0178	350.0056	-3.9524	0.04652	0.25475	0.01234
3	120	47	900.705	36.5104	0.12511	32	0.0162	330.0056	-3.9023	0.04652	0.25163	0.01215
4	130	20	800.705	39.5104	0.12111	30	0.0168	330.0056	-3.9023	0.04652	0.25163	0.01215
5	160	50	756.799	38.5390	0.15247	30	0.0148	13.8593	0.3277	0.00420	0.24970	0.01200
6	240	70	451.325	46.1592	0.10587	20	0.0163	13.8593	0.3277	0.00420	0.24970	0.01200
7	300	60	1243.531	38.3055	0.03546	20	0.0152	40.2669	-0.5455	0.00680	0.24800	0.01290
8	340	70	1049.998	40.3965	0.02803	30	0.0128	40.2669	-0.5455	0.00680	0.24990	0.01203
9	470	135	1658.569	36.3278	0.02111	60	0.0136	42.8955	-0.5112	0.00460	0.25470	0.01234
10	470	150	1356.659	38.27041	0.01799	40	0.0141	42.8955	-0.5112	0.00460	0.25470	0.01234

The transmission loss formula coefficients of ten-unit system are:

$$B = \begin{bmatrix} 0.000049 & 0.000014 & 0.000015 & 0.000015 & 0.000016 & 0.000017 & 0.000017 & 0.000018 & 0.000019 & 0.000020 \\ 0.000014 & 0.000045 & 0.000016 & 0.000016 & 0.000017 & 0.000015 & 0.000015 & 0.000016 & 0.000018 & 0.000018 \\ 0.000015 & 0.000016 & 0.000039 & 0.000010 & 0.000012 & 0.000012 & 0.000014 & 0.000014 & 0.000016 & 0.000016 \\ 0.000015 & 0.000016 & 0.000010 & 0.000040 & 0.000014 & 0.000010 & 0.000011 & 0.000012 & 0.000014 & 0.000015 \\ 0.000016 & 0.000017 & 0.000012 & 0.000014 & 0.000035 & 0.000011 & 0.000013 & 0.000013 & 0.000015 & 0.000016 \\ 0.000017 & 0.000015 & 0.000012 & 0.000010 & 0.000011 & 0.000036 & 0.000012 & 0.000012 & 0.000014 & 0.000015 \\ 0.000017 & 0.000015 & 0.000014 & 0.000011 & 0.000013 & 0.000012 & 0.000038 & 0.000016 & 0.000016 & 0.000018 \\ 0.000018 & 0.000016 & 0.000014 & 0.000012 & 0.000013 & 0.000012 & 0.000016 & 0.000040 & 0.000015 & 0.000016 \\ 0.000019 & 0.000018 & 0.000016 & 0.000014 & 0.000015 & 0.000014 & 0.000016 & 0.000015 & 0.000042 & 0.000019 \\ 0.000020 & 0.000018 & 0.000016 & 0.000015 & 0.000016 & 0.000015 & 0.000018 & 0.000016 & 0.000019 & 0.000044 \end{bmatrix}$$

**References**

Al-Dabbagh RD, Kinsheel A, Mekhilef S et al (2014) System identification and control of robot manipulator based on fuzzy adaptive differential evolution algorithm. *Adv Eng Softw* 78:60–66

Ali M, Ahn CW, Siary P (2014) Differential evolution algorithm for the selection of optimal scaling factors in image watermarking. *Eng Appl Artif Intell* 31:15–26

Basu M (2011) Economic environmental dispatch using multi-objective differential evolution. *Appl Soft Comput* 11(2):2845–2853

Biswas S, Kundu S, Das S (2014) An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution. *IEEE Trans Cybern* 44(10):1726–1737

Brest J, Mernik M (2008) Population size reduction for the differential evolution algorithm. *Appl Intell* 29(3):228–247

Brown C, Jin Y, Leach M, et al (2015)  $\mu$ JADE: adaptive differential evolution with a small population. *Soft Comput*. doi:10.1007/s00500-015-1746-x

Cai Y, Wang J (2015) Differential evolution with hybrid linkage crossover. *Inf Sci* 320:244–287

Das R, Prasad DK (2015) Prediction of porosity and thermal diffusivity in a porous fin using differential evolution algorithm. *Swarm Evol Comput* 23:27–39

Epitropakis MG, Li X, Burke EK (2013) A dynamic archive niching differential evolution algorithm for multimodal optimization. In: *IEEE Congress on Evolutionary Computation (CEC 2013)*, pp 79–86

Epitropakis MG, Plagianakos VP, Vrahat MN (2012) Multimodal optimization using niching differential evolution with index-based neighborhoods. In: *IEEE Congress on Evolutionary Computation (CEC 2012)*, pp 1–8

Han MF, Lin CT, Chang JY (2013) Differential evolution with local information for neuro-fuzzy systems optimization. *Knowl Based Syst* 44:78–89

Ho-Huu V, Nguyen-Thoi T, Nguyen-Thoi MH et al (2015) An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures. *Expert Syst Appl* 42(20):7057–7069

Kundu S, Das S, Vasilakos AV et al (2014) A modified differential evolution-based combined routing and sleep scheduling scheme for lifetime maximization of wireless sensor networks. *Soft Comput* 19(3):637–659

Li X (2005) Efficient differential evolution using speciation for multimodal function optimization. In: *The 7th annual conference on genetic and evolutionary computation, ACM*, pp 873–880

Liang JJ, Suganthan PN, Deb K (2005) Novel composition test functions for numerical global optimization. In: *Swarm Intelligence Symposium*, pp 68–75

- Liu JH, Lampinen J (2002) On setting the control parameter of the differential evolution method. In: The 8th international conference on soft computing (MENDEL 2002), pp 11–18
- Liu G, Xiong C, Guo Z (2014) Enhanced differential evolution using random-based sampling and neighborhood mutation. *Soft Comput* 19(8):1–20
- Mahfoud SW (1995) Niching methods for genetic algorithms, Ph.D. dissertation, Univ. of Illinois, Urbana-Champaign
- Mallipeddi R, Suganthan PN, Pan QK et al (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl Soft Comput* 11(2):1679–1696
- Mallipeddi R, Lee M (2015) An evolving surrogate model-based differential evolution algorithm. *Appl Soft Comput* 34:770–787
- Mohamed AW (2015) An improved differential evolution algorithm with triangular mutation for global numerical optimization. *Comput Ind Eng* 85:359–375
- Mokhtari H, Salmasnia A (2015) A monte carlo simulation based chaotic differential evolution algorithm for scheduling a stochastic parallel processor system. *Expert Syst Appl* 42(20):7132–7147
- Petrowski A (1996) A clearing procedure as a niching method for genetic algorithms. In: The 1996 IEEE international conference on evolutionary computation, pp 798–803
- Petrowski A, Genet MG (1999) A classification tree for speciation. In: IEEE Congress on Evolutionary Computation (CEC 1999), vol 1, pp 204–211
- Qin AK, Suganthan PN (2005) Self-adaptive differential evolution algorithm for numerical optimization. In: IEEE congress on evolutionary computation (CEC 2005), vol 2, IEEE Press, pp 1785–1791
- Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans Evolut Comput* 13(2):398–417
- Rakshit P, Konar A (2015) Differential evolution for noisy multi-objective optimization. *Artif Intell* 227:165–189
- Sareni B, Krahenbuhl L (1998) Fitness sharing and niching methods revisited. *IEEE Trans Evolut Comput* 3(2):97–106
- Secmen M, Tasgetiren MF (2013) Ensemble of differential evolution algorithms for electromagnetic target recognition problem. *IET Radar Sonar Navig* 7(7):780–788
- Sharma S, Rangaiah GP (2013) An improved multi-objective differential evolution with a termination criterion for optimizing chemical processes. *Comput Chem Eng* 56:155–173
- Sheniha SF, Priyadharsini SS, Rajan SE (2013) Removal of artifact from EEG signal using differential evolution algorithm. In: The 2th international conference on communication and signal processing, pp 134–138
- Simionescu PA (2014) Computer-aided graphing and simulation tools for AutoCAD User, (1st Ed.) CRC Press, Boca Raton
- Storn R, Price KV (1995) Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. ICSI, USA, Tech. Rep. TR-95-012, [Online]. Available:<http://icsi.berkeley.edu/~storn/litera.html>
- Storn R, Price KV (1996) Minimizing the real functions of the ICEC 1996 contest by differential evolution. In: Proceedings: 1996 IEEE international conference on evolutionary computation, pp 842–844
- Surjanovic S, Bingham D (2015) Virtual library of simulation experiments: test functions and datasets. <http://www.sfu.ca/~ssurjano/optimization.html>
- Tang L, Dong Y, Liu J (2015) Differential evolution with an individual-dependent mechanism. *IEEE Trans Evolut Comput* 19(4):560–574
- Thomsen R (2004) Multimodal optimization using crowding-based differential evolution. In: IEEE Congress on Evolutionary Computation (CEC 2004) vol 2, pp 1382–1389
- Tvrđík J, Křivý I (2015) Hybrid differential evolution algorithm for optimal clustering. *Appl Soft Comput* 35:502–512
- Walters DC, Sheble GB (1993) Genetic algorithm solution of economic dispatch with valve point loading. *IEEE Trans Power Syst* 8(3):1325–1332
- Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans Evolut Comput* 15(1):55–66
- Wang J, Liao J, Zhou Y et al (2014) Differential evolution enhanced with multiobjective sorting-based mutation operators. *IEEE Trans Cybern* 44(12):2792–2805
- Yin X, Gernay N (1993) A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In: Artificial Neural Networks and Genetic Algorithms, pp 450–457
- Yu WJ, Shen M, Chen WN et al (2014) Differential evolution with two-level parameter adaptation. *IEEE Trans Cybern* 44(7):1080–1099
- Zaharie D (2009) Influence of crossover on the behavior of differential evolution algorithm. *Appl Soft Comput* 9(3):1126–1138
- Zhai Z, Li X (2011) A dynamic archive based niching particle swarm optimizer using a small population size. In: Proceedings of the Australian Computer Science Conference (ACSC 2011), ACM, pp 1–7
- Zhang H, Yue D, Xie X et al (2015) Multi-elite guide hybrid differential evolution with simulated annealing technique for dynamic economic emission dispatch. *Appl Soft Comput* 34:312–323
- Zhang G, Li D, Zhou X, et al (2015) Differential evolution with dynamic niche radius strategy for multimodal optimization. In: The 27th international conference on control and decision conference (CCDC 2015), IEEE, pp 3059–3064
- Zhang JQ, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. *IEEE Trans Evolut Comput* 13(5):945–958
- Zhu W, Fang JA, Tang Y et al (2012) Digital IIR filters design using differential evolution algorithm with a controllable probabilistic population size. *PLoS One* 7(7):e40549