Artif Intell Rev (2019) 51:219–253
https://doi.org/10.1007/s10462-017-9562-6

Differential evolution algorithm with strategy adaptation and knowledge-based control parameters

Qinqin Fan[1] · Weili Wang[1] · Xuefeng Yan[2]

Published online: 2 June 2017
© Springer Science+Business Media Dordrecht 2017

Abstract The search capability of differential evolution (DE) is largely affected by control parameters, mutation and crossover strategies. Therefore, choosing appropriate strategies and control parameters to solve different types of optimization problems or adapt distinct evolution phases is an important and challenging task. To achieve this objective, a DE with strategy adaptation and knowledge-based control parameters (SAKPDE) is proposed in the current study. In the proposed algorithm, a learning–forgetting mechanism is used to implement the adaptation of mutation and crossover strategies. Meanwhile, prior knowledge and opposition learning are utilized to supervise and guide the evolution of control parameters during the entire evolutionary process. SAKPDE is compared with eight improved DEs and four non-DE evolutionary algorithms using three well-known test suites (i.e., BBOB2012, IEEE CEC2005, and IEEE CEC2014). The results indicate that the average performance of SAKPDE is highly competitive among all compared algorithms.

Keywords Differential evolution · Learning–forgetting · Prior knowledge · Opposition learning

1 Introduction

Over the past decades, various evolutionary algorithms (EAs) have been developed to solve practical engineering optimization problems in different areas. A major difference of these

✉ Qinqin Fan
forever123fan@163.com

Xuefeng Yan
xfyan@ecust.edu.cn

1 Logistics Research Center, Shanghai Maritime University, No. 1550 Haigang Avenue, Shanghai 201306, People's Republic of China

2 Key Laboratory of Advanced Control and Optimization for Chemical Process of Ministry of Education, East China University of Science and Technology, Shanghai 200237, People's Republic of China

EAs is the offspring production operator. Generally, new candidate solutions in EAs are generated by two models (Li et al. 2014), namely, a distributed model and a centralized model. In EAs, differential evolution (DE) proposed by Storn and Price (1995) is a simple yet competitive stochastic search method. However, the search performance of DE is directly influenced by the setting of control parameters and use of strategies. Generally, DE has three main parameters (i.e., population size NP, mutation parameter F, and crossover parameter CR) and two strategies (i.e., mutation and crossover).

Previous studies demonstrate that a constant parameter setting or a single strategy may not be able to provide the best performance on all types of problems. Therefore, researchers developed many self-adaptive DEs to improve the performance and robustness of DE, such as jDE (Brest et al. 2006), self-adaptive DE (SaDE) (Qin et al. 2009), JADE (Zhang and Sanderson 2009), modified DE with p-best crossover (MDE_pBX) (Islam et al. 2012), ensemble of mutation strategies and control parameters with DE (EPSDE) (Mallipeddi et al. 2011), DE with self-adaptive mutation strategies and control parameters (SSCPDE) (Fan and Yan 2015), self-adaptive DE with zoning evolution of control parameters and adaptive mutation strategies (ZEPDE) (Fan and Yan 2016), and DE with dynamic parameters selection (DE-DPS) (Sarker et al. 2014). Although these DE variants can select appropriate control parameters, mutation strategies, or even both, some problems still exist. For instance, the adaptation of crossover strategies is not considered in most existing DE variants. Furthermore, some adaptation schemes may result in heavy computational burden (Islam et al. 2012) and may provide misleading or deceiving evolution information which would impact the search process of DE.

To tackle the above mentioned problems in DE, self-adaptive crossover operators and the knowledge-based control parameters are implemented in the current study. For the strategy adaptation, a pool of mutation and crossover strategies are used to find a promising solution. However, the actual situation may be that a strategy may suitable for searching a promising solution in the early evolution stages while it may perform poorly in the later phases of the evolution, and vice versa. Therefore, if the choice of strategy is relied entirely on the previous successful experience, it will result in greedy selection. In that case, the ensemble of strategy cannot perform effectively during the entire evolutionary process. To alleviate this problem, a learning–forgetting mechanism is used to select mutation and crossover strategies in this work. This is primarily due to the fact that the learning scheme is helpful in choosing an appropriate strategy based on the previous experience, and the forgetting mechanism can avoid the greedy selection effectively to some extent. For the adaptation of control parameters, prior knowledge and opposition learning are used to guide the evolution of F and CR. Das et al. (2005) stated that with regard to a powerful algorithm, the global search capability should be owned in the early stages of the search process and the local search capability should be encouraged in the later phases of the evolution. Therefore, this argument can be considered as the prior knowledge in the current study. In other words, the control method of parameters in this work should conform to this prior knowledge (Fan et al. 2016). However, for most cases, the prior knowledge may mislead the evolution of the control parameters in complex optimization environment. Therefore, the unconventional knowledge is also utilized to guide the evolution of the control parameters. The main goal is to reduce risk which is derived from the prior knowledge.

Based on these observations, we proposed a differential evolution algorithm with strategy adaptation and knowledge-based control parameters (SAKPDE). In SAKPDE, a learning–forgetting mechanism is employed to implement the adaptation of mutation and crossover strategies. Prior and non-traditional (obtained by opposition learning) knowledge is used to control the adjustment of parameters. Therefore, suitable mutation strategy, crossover strategy, and control parameters can be self-adaptively chosen in the proposed algorithm during the evolutionary process.

To demonstrate the search performance of the proposed algorithm, SAKPDE is systematically compared with eleven existing EAs (8 DE variants and 3 non-DE algorithms) on three sets of benchmark functions (Hansen et al. 2012; Liang et al. 2013; Suganthan et al. 2005), i.e., two IEEE CEC test suites: CEC2005 and CEC2014, and a real-parameter black-box optimization benchmark 2012: BBOB2012. Experimental results confirm that the proposed approaches can enhance the average performance of DE effectively.

The reminder of this paper is organized as follows. In Sect. 2, the basic DE algorithm is introduced. Section 3 presents the related studies on DE. Section 4 introduces the proposed algorithm. The results and parameter analysis are shown in Sect. 5. Finally, the conclusions are drawn in Sect. 6.

2 DE algorithm

A minimization problem can be expressed as follows:

$$f(\boldsymbol{x}^*) = \min_{x_i \in \Omega} f(\boldsymbol{x}_i), \boldsymbol{x}_i \in \boldsymbol{S} = \prod_{j=1}^{D} \left[L_j, U_j\right] \tag{1}$$

where f denotes the objective function, $\boldsymbol{x}_i = (x_{i,1}, \cdots, x_{i,D})$ is a D-dimensional vector, $\boldsymbol{x}^*$ is the global optimum solution of the optimization problem, $\Omega \subseteq R^D$. L_j and U_j $(j = 1, \cdots, D)$ are the lower and upper bounds of the j-th variable of $\boldsymbol{x}_i$, respectively, and $\boldsymbol{S}$ is the search space.

DE is a population-based stochastic optimization approach. Mutation, crossover, and selection are the three main operators in DE. $\boldsymbol{x}_i^G = (x_{i,1}^G, \cdots, x_{i,D}^G)$ and $\boldsymbol{X}^G = \{\boldsymbol{x}_1^G, \cdots, \boldsymbol{x}_{NP}^G\}$ denote the i-th target vector and the population at the G-th generation, respectively.

The procedure of DE is described as follows (Price et al. 2006):

(1) Initialization operator

Determine F, CR, NP, and maximum number of generations $G_{\max}$. Initial individuals $\boldsymbol{x}_i^0 (i = 1, \cdots, NP)$ are generated randomly in $\boldsymbol{S}$. The current generation G is set to be 0.

(2) Mutation operator

After initialization operation, for each $\boldsymbol{x}_i^G$ in the current population, the mutant vector $\boldsymbol{v}_i^G$ is produced by a mutation operator. Some of the most frequently used mutation strategies are listed as follows:

$$\text{“DE/rand/1”: } \boldsymbol{v}_i^G = \boldsymbol{x}_{r1}^G + F \cdot (\boldsymbol{x}_{r2}^G - \boldsymbol{x}_{r3}^G), \tag{2}$$

$$\text{“DE/rand/2”: } \boldsymbol{v}_i^G = \boldsymbol{x}_{r1}^G + F \cdot (\boldsymbol{x}_{r2}^G - \boldsymbol{x}_{r3}^G) + F \cdot (\boldsymbol{x}_{r4}^G - \boldsymbol{x}_{r5}^G), \tag{3}$$

$$\text{“DE/current-to-best/1”}: \boldsymbol{v}_i^G = \boldsymbol{x}_i^G + F \cdot (\boldsymbol{x}_{\text{best}}^G - \boldsymbol{x}_i^G) + F \cdot (\boldsymbol{x}_{r1}^G - \boldsymbol{x}_{r2}^G), \tag{4}$$

$$\text{“DE/current-to-best/2”}: \boldsymbol{v}_i^G = \boldsymbol{x}_i^G + F \cdot (\boldsymbol{x}_{\text{best}}^G - \boldsymbol{x}_i^G) + F \cdot (\boldsymbol{x}_{r1}^G - \boldsymbol{x}_{r2}^G + \boldsymbol{x}_{r3}^G - \boldsymbol{x}_{r4}^G), \tag{5}$$

$$\text{“DE/ best/2”}: \boldsymbol{v}_i^G = \boldsymbol{x}_{\text{best}}^G + F \cdot (\boldsymbol{x}_{r1}^G - \boldsymbol{x}_{r2}^G) + F \cdot (\boldsymbol{x}_{r3}^G - \boldsymbol{x}_{r4}^G), \tag{6}$$

where r_1, r_2, r_3, r_4, and r_5 are mutually exclusive integers randomly chosen within the range $[1, NP]$, and are also different from the index i (i.e., $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$); $\boldsymbol{x}_{\text{best}}^G$is the target vector with the best fitness value at the G-th generation.

(3) Crossover operator

In DE, the binomial and exponential crossover operators are two common strategies. For each $\boldsymbol{x}_i^G$, the trial vector $\boldsymbol{u}_i^G$ produced by the binomial crossover operator is generated as follows:

$$u_{ij}^G = \begin{cases} v_{ij}^G, & R_j \leq CR \text{ or } j = j_{rand} \\ x_{ij}^G, & \text{otherwise} \end{cases} \quad j = 1, 2, \cdots, D., \tag{7}$$

where R_j and j_{rand} are a uniform random number within the range [0, 1] and a randomly chosen integer within the range [1, D], respectively.

For each $\boldsymbol{x}_i^G$, the trial vector $\boldsymbol{u}_i^G$ generated by the exponential crossover operator can be defined as follows:

$$u_{ij}^G = \begin{cases} v_{ij}^G, & \text{if } j = \langle n \rangle_D, \langle n+1 \rangle_D, \cdots, \langle n+L-1 \rangle_D \\ x_{ij}^G, & \text{otherwise} \end{cases} \tag{8}$$

where $\langle\rangle_D$ denotes a modulo function with modulus D. n is a random number and L is an integer drawn from [1, D].

(4) Selection

The trial vector $\boldsymbol{u}_i^G$ competes with its target vector $\boldsymbol{x}_i^G$ and the better one will be selected for the next generation:

$$\boldsymbol{x}_i^{G+1} = \begin{cases} \boldsymbol{u}_i^G, & f(\boldsymbol{u}_i^G) \leq f(\boldsymbol{x}_i^G) \\ \boldsymbol{x}_i^G, & \text{otherwise} \end{cases} \tag{9}$$

(5) Implement

Steps 2–4 repeatedly until the number of generations is equal to $G_{\max}$.

3 Related work

Because the search performance of DE is significantly affected by mutation strategy, crossover strategy, and control parameters (Das and Suganthan 2011; Neri and Tirronen 2010), a large number of researches focused on the adaptation or self-adaptation of them to improve the optimization performance and robustness of DE.

To implement control parameter adaptation, in literature (Abbass 2002), a self-adaptive pareto DE (SPDE) is introduced. In SPDE, a normal distribution function is used to generate F and CR. Salman et al. (2007) proposed a self-adaptive DE (SDE), in which production of control parameters using a normal distribution function. Zaharie (2002) introduced an adaptive DE(ADE) wherein the adaptation of F and CR are based on the population diversity. A fuzzy adaptive DE (FADE) is proposed by Liu and Lampinen (2005). In FADE, a fuzzy logic controller is employed to generate appropriate control parameters. A self-adaptive DE is proposed by Tirronen and Neri (2009). In the proposed algorithm, three control parameters are dynamically updated through the fitness diversity. Brest et al. (2006) introduced a DE with self-adaptive parameter control (jDE), in which F and CR encoded into each individual are produced by a normal distribution function. In literature (Teo 2006), a dynamic self-adaptive population size is introduced. Self-adaptive NP is achieved by two encoding methodologies. An immune self-adaptive differential evolution algorithm (ISDE) is proposed by Hu and Yan (2009). In ISDE, a meta-algorithm (i.e., immune concepts and methods) is utilized to adjust the parameters. Ghosh et al. (2011) introduced a fitness-adaptive DE (FiADE), in which the fitness value of individuals in the current population is utilized to adjust F and CR during the evolutionary process. Sarker et al. (2014) introduced a DE with dynamic parameters selection (DE-DPS). In DE-DPS, a new technique is used to automatically choose the best performing combinations of F, CR, and NP. Their results show that the proposed mechanism can enhance the solution efficiency of DE. An adaptive population tuning scheme (APTS) is introduced in Zhu et al. (2013). The proposed approach can dynamically adjust NP and delete redundant individuals. Although various adaptation schemes have been proposed to adjust the control parameters of DE, most of them are based on the current population information or the fitness landscape which may provide misleading or deceptive evolution information.

In addition, since the performance of DE is significantly affected by the mutation and crossover strategies, DE researchers focus on the adaptation of both control parameter and strategy, and the improvement of strategy. For instance, In literature (Zhang and Sanderson 2009), a self-adaptive DE (JADE) is introduced and is evaluated on 20 benchmark functions. In JADE, normal and Cauchy distribution functions are employed to generate appropriate CR and F, respectively. Moreover, a novel mutation strategy is incorporated into DE. An improved DE with p-best crossover (MDE_pBX) is proposed by Islam et al. (2012). In MDE_pBX, a mutation strategy variant, a novel crossover strategy, and an adaptation mechanism of control parameters are used. Their results indicate that the performance of MDE_pBX is better than that of other existing DE variants on 25 IEEE CEC2005 benchmark functions. Gong and Cai (2013) introduced a ranking-based mutation operator, in which the selection probability of each individual is based on its rankings in the current population. In literature (Pan et al. 2011), a DE with self-adaptive mutation strategy and control parameters (SspDE) is proposed. In SspDE, both mutation strategy and its associated control parameters in each individual can be self-adaptively tuned by learning from their previous experience. An ensemble of mutation strategies and control parameters with DE (EPSDE) is introduced by Mallipeddi et al. (2011), in which a pool of mutation strategies and control parameters are used to provide different search capabilities. Similar to EPSDE, Wang et al. (2011) proposed a composite DE (CoDE) wherein three selected control parameter combinations are randomly combined with three mutation strategies. The experimental results indicate that CoDE has better performance than other competitors. Thereafter, Wang et al. (2014) introduced an improved DE (CoBiDE). In CoBiDE, the bimodal distribution parameter setting is used to balance global and local search capabilities, and the covariance matrix learning is utilized to enhance the search performance on optimization problems with high variable correlation. A DE with self-adaptive mutation strategy and control parameters (SSCPDE) is proposed by Fan and Yan (2015), in which appropriate mutation strategy and control parameters can be automatically obtained. Subsequently, Fan and Yan (2016) proposed a self-adaptive DE with zoning evolution of control parameters and adaptive mutation strategies (ZEPDE). In ZEPDE, a suitable mutation strategy can be self-adaptively achieved and the control parameters can be automatically generated in their own zoning. Yu et al. (2014) introduced an adaptive DE (ADE) wherein the proposed mutation strategy is a variant of DE/best/1 and two-level adaptive control approach is utilized to tune the control parameters. For the aforementioned DE variants, clearly, the adaptation scheme for the crossover strategy has seldom been considered.

Besides the above studies, a large number of methods have been proposed to enhance the solution efficiency and robustness of DE. For example, a crossover rate repair method (Gong et al. 2014) is proposed to improve the performance of other self-adaptive DEs. Recently, Gong et al. (2015) used a cheap surrogate model to select suitable candidate offspring solutions produced by different operators. Yang et al. (2015) proposed an auto-enhanced population diversity (AEPD) that is utilized to automatically improve the population diversity. The results indicate that AEPD can avoid population stagnation and premature convergence effectively. Guo and Yang (2015) introduced an eigenvector-based crossover strategy that

Table 1 Individual with control parameters, mutation, and crossover strategies

Individual	Control parameters	Mutation strategy	Crossover strategy
$\boldsymbol{x}_1^G$	(F_1^G, CR_1^G)	Mutation strategy$_1^G$	Crossover strategy$_1^G$
$\boldsymbol{x}_2^G$	(F_2^G, CR_2^G)	Mutation strategy$_2^G$	Crossover strategy$_2^G$
...	...	...	...
$\boldsymbol{x}_{NP}^G$	(F_{NP}^G, CR_{NP}^G)	Mutationstrategy$_{NP}^G$	Crossoverstrategy$_{NP}^G$

significantly enhance the search capability of DE on non-separable unimodal test functions. Recently, Guo et al. (2015) proposed a successful-parent selecting framework, in which an archive is used to store successful solutions. For more related works, interested readers can refer to Biswas et al. (2015), Cui et al. (2016), Gao et al. (2014), Kovačević et al. (2014), Li et al. (2015), Mokhtari and Salmasnia (2015), Shao and Pi (2016), Zhong et al. (2013) and X-g et al. (2016).

4 DE with strategy adaptation and knowledge-based control parameters

To further improve the performance of DE, a DE with strategy adaptation and knowledge-based control parameters (SAKPDE) is introduced to carry out the adaptation of strategies and supervise the evolution of the parameters. In SAKPDE, each individual has its own control parameters (i.e., F and CR), mutation strategy, and crossover strategy (see Table 1). Mutation Strategy$_i^G$ denotes a mutation strategy that involves DE/rand/1, DE/rand/2, DE/best/2, DE/current-to-best/1, or DE/ current-to-best/2. Crossover Strategy$_i^G$ denotes a crossover strategy that involves binomial, exponential, or eigenvector-based crossover strategy.

4.1 Self-adaptive mutation strategy

In DE algorithm, the mutation operator DE/rand/1 is the most widely used for its good global search ability, thus it is selected in the mutation strategy pool. Based on the studies (Qin et al. 2009; Wang et al. 2011), four other mutation strategies, specifically DE/rand/2, DE/best/2, DE/current-to-best/1, and DE/current-to-best/2 are also selected in the mutation strategy pool. Additionally, to encourage algorithm to execute global searching in the early stages of the evolution, the mutation operator used in the proposed algorithm is divided into two stages. In the first stage, DE/rand/1 is only used to produce the offspring in the early evolution stages; in the second stage, the self-adaptive mutation strategy is utilized to find the optimal solution. For the above operation steps, the advantages are twofold: (1) maintaining the population diversity in the early stages of the search, and (2) saving the computational resource.

The implement steps of self-adaptive mutation strategy can be described as follows:

(1) The maximum objective function value is computed as follows:

$$f_{\max} = \max f(\boldsymbol{u}_i^G), \tag{10}$$

(2) The difference between the fitness value of each individual and $f_{\max}$ is computed as follows:

$$\Delta f_i^G = \left| f(\boldsymbol{u}_i^G) - f_{\max} \right|, \tag{11}$$

(3) The sum of difference of each mutation strategy is calculated as follows:

$$S^{G}_{\text{str\_name}} = \sum_{k=1}^{N^{G}_{\text{str\_name}}} \Delta f^{G}_{\text{str\_name},k}, k = 1, 2, \cdots, N^{G}_{\text{str\_name}}, \tag{12}$$

where str_name denotes one mutation strategy in the mutation strategy pool, $N^{G}_{\text{str\_name}}$ denotes the number of individuals for the mutation strategy str_name.

(4) The maximum sum of difference of five selected mutation strategies is obtained as follows:

$$S^{G}_{\max} = \max\left(S^{G}_{\text{rand/1}}, S^{G}_{\text{rand/2}}, S^{G}_{\text{current-to-best/1}}, S^{G}_{\text{current-to-best/2}}, S^{G}_{\text{best/2}}\right) \tag{13}$$

Assuming that the performance of the mutation strategy DE/rand/1 is the best among five selected mutation strategies in the Gth generation; therefore, the sum of difference of the mutation strategy DE/rand/1 multiply by a forgetting factor φ. $S^{G}_{\text{rand/1}}$ can be updated as follows:

$$S^{G}_{\text{rand/1}} = \varphi \times S^{G}_{\text{rand/1}} \tag{14}$$

where φ is a forgetting factor.

(5) The sum of differences of all selected mutation strategies can be recalculated as follows:

$$S^{G} = S^{G}_{\text{rand/1}} + S^{G}_{\text{rand/2}} + S^{G}_{\text{current-to-best/1}} + S^{G}_{\text{current-to-best/2}} + S^{G}_{\text{best/2}} \tag{15}$$

(6) The Selection probability (sp) of each mutation strategy is calculated as follows:

$$sp^{G}_{\text{str\_name}} = S^{G}_{\text{str\_name}} / S^{G} \tag{16}$$

(7) The cumulative probability (cp) of each mutation strategy is computed as follows:

$$cp^{G}_{\text{rand/1}} = sp^{G}_{\text{rand/1}} \tag{17}$$

$$cp^{G}_{\text{rand/2}} = sp^{G}_{\text{rand/1}} + sp^{G}_{\text{rand/2}} \tag{18}$$

$$cp^{G}_{\text{current-to-best/1}} = sp^{G}_{\text{rand/1}} + sp^{G}_{\text{rand/2}} + sp^{G}_{\text{current-to-best/1}} \tag{19}$$

$$cp^{G}_{\text{current-to-best/2}} = sp^{G}_{\text{rand/1}} + sp^{G}_{\text{rand/2}} + sp^{G}_{\text{current-to-best/1}} + sp^{G}_{\text{current-to-best/2}} \tag{20}$$

$$cp^{G}_{\text{best/2}} = sp^{G}_{\text{rand/1}} + sp^{G}_{\text{rand/2}} + sp^{G}_{\text{current-to-best/1}} + sp^{G}_{\text{current-to-best/2}} + sp^{G}_{\text{best/2}} \tag{21}$$

Based on the cumulative probability (cp) of each mutation strategy, the number of each mutation strategy (i.e., $N^{G+1}_{\text{str\_rand}/1}$, $N^{G+1}_{\text{str\_rand}/2}$, $N^{G+1}_{\text{str\_rand}-\text{to}-\text{best}/1}$, $N^{G+1}_{\text{str\_rand}-\text{to}-\text{best}/2}$, and $N^{G+1}_{\text{str\_best}/2}$) in the mutation strategy pool is regenerated by a roulette wheel.

4.2 Self-adaptive crossover strategy

Similar to Sect. 4.1, the crossover operator is also divided into two stages. In the first stage, the binomial crossover strategy is used; in the second stage, a self-adaptive crossover strategy is utilized to implement the information exchange between two individuals. Moreover, the binomial, exponential, and eigenvector-based crossover strategies are chosen in the crossover strategy pool. In specific, the binomial crossover strategy has good exploration capability, the exponential crossover strategy can provide good local search ability, and the eigenvector-based crossover strategy can perform well on some non-separable unimodal functions.

The operation steps of self-adaptive crossover strategy can be implemented as follows:

(1) The sum of differences of each crossover strategy is computed as follows:

$$S_{\text{cstr\_name}}^{G} = \sum_{h=1}^{N_{\text{cstr\_name}}^{G}} \Delta f_{\text{cstr\_name},h}^{G}, h = 1, 2, \cdots, N_{\text{cstr\_name}}^{G} \tag{22}$$

where cstr_name denotes one of the three crossover strategies, $N_{\text{cstr\_name}}^{G}$ denotes the number of individuals for the crossover strategy cstr_name. $\Delta f_{\text{cstr\_name},h}^{G} = \left| f(\boldsymbol{u}_{\text{cstr\_name},h}^{G}) - \max f(\boldsymbol{u}^{G}) \right|$.

(2) The maximum sum of difference of crossover strategies is obtained as follows:

$$S_{\text{cmax}}^{G} = \max(S_{bin}^{G}, S_{\exp}^{G}, S_{\text{eig}}^{G}) \tag{23}$$

Assuming that the performance of the crossover strategy DE/bin is the best among three selected crossover strategies in the Gth generation, then the sum of difference of DE/bin multiply by φ. S_{bin}^{G} can be computed as follows:

$$S_{bin}^{G} = \varphi \times S_{bin}^{G} \tag{24}$$

(3) The sum of differences of three crossover strategies is calculated as follows:

$$S^{G} = S_{bin}^{G} + S_{\exp}^{G} + S_{eig}^{G} \tag{25}$$

(4) The selection probability (sp) of each crossover strategy is calculated as follows:

$$sp_{\text{cstr\_name}}^{G} = S_{\text{cstr\_name}}^{G} / S^{G} \tag{26}$$

(5) The cumulative probability (cp) of each crossover strategy is computed as follows:

$$cp_{bin}^{G} = sp_{bin}^{G} \tag{27}$$

$$cp_{\exp}^{G} = sp_{bin}^{G} + sp_{\exp}^{G} \tag{28}$$

$$cp_{eig}^{G} = sp_{bin}^{G} + sp_{\exp}^{G} + sp_{eig}^{G} \tag{29}$$

Based on the cumulative probability (cp) of each crossover strategy, the number of each crossover strategy (i.e., $N_{\text{cstr\_bin}}^{G+1}$, $N_{\text{cstr\_exp}}^{G+1}$, and $N_{\text{cstr\_eig}}^{G+1}$) is regenerated by a roulette wheel.

4.3 Self-adaptive control parameters

For most evolutionary algorithms, it is generally believed to be a good idea that encouraging the global search in the early evolutionary phases and ensuring the local search in the latter stages of the evolution (Das et al. 2005). For DE algorithm, a large value of F can provide good exploration capability, whereas a small value of F can accelerate the convergence speed (Das and Suganthan 2011). Moreover, a large value of CR can provide good local search capability (Montgomery and Chen 2010; Yu et al. 2014). Based on the above observations, two distribution functions incorporated with prior knowledge are used to supervise the adjustment of F and CR, specifically, the mean value of F gradually decreases (see Fig. 1) and the mean value of CR gradually increases (see Fig. 2) during the whole evolutionary process. Additionally, previous studies (Neri and Tirronen 2010; Storn and Price 1997) suggested that the value of F should be within the range of [0.4, 1]. Therefore, to produce new parameters as much as possible within the recommended range, the range of the local parameter in a Cauchy distribution function is [0.4, 0.1]. For the control parameter CR, its value should be within the range of [0.3, 0.9] (Gämperle et al. 2002; Ronkkonen et al. 2005). Therefore, to

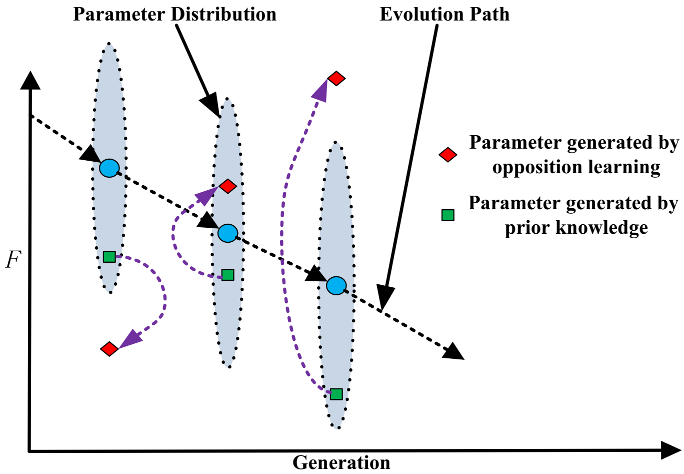

Fig. 1 Evolution of mutation control parameter

generate new values of *CR* as much as possible within the suggested range, the mean value of a normal distribution function is in [0.3, 1]. Note that the control parameters generated by prior knowledge is based on our previous study (Fan et al. 2016).

Although prior knowledge can guide the evolution of the control parameters effectively, it may be incorrect when the optimization environment is very complex. Therefore, opposition learning is used to generate the control parameters (see Figs. 1, 2). The main aim is that the control parameters generated by opposition learning can provide diverse search capabilities. From Figs. 1 and 2, we can observe that values of F and CR generated by opposition learning may be without or within the range of parameters generated by prior knowledge. Therefore, the control parameters produced by prior knowledge and opposition learning can provide different search capabilities for DE at each generation.

4.3.1 Evolution of F

If $rand > \rho$, F value of each individual is generated by opposition learning [see Eq. (30)]; otherwise, F value of the individual is produced by the prior knowledge [see Eq. (31)].

$$F_i^{G+1} = \left|1 - Cauchy((1 - 0.6 \times G/G_{\max}), \sigma_1)\right|, i = 1, \cdots, NP, \tag{30}$$

where *Cauchy* is a Cauchy distribution function and$\sigma_1 = 0.8 - 0.6 \times (1 - (G/G_{\max})^2)$.

$$F_i^{G+1} = \left|Cauchy((1 - 0.6 \times G/G_{\max}), \sigma_1)\right|, i = 1, \cdots, NP, \tag{31}$$

4.3.2 Evolution of CR

If $rand > \rho$, CR value of each individual is generated by opposition learning [see Eq. (32)]; otherwise, CR value of each individual is produced by prior knowledge (Gämperle et al. 2002; Ronkkonen et al. 2005) [see Eq. (33)].

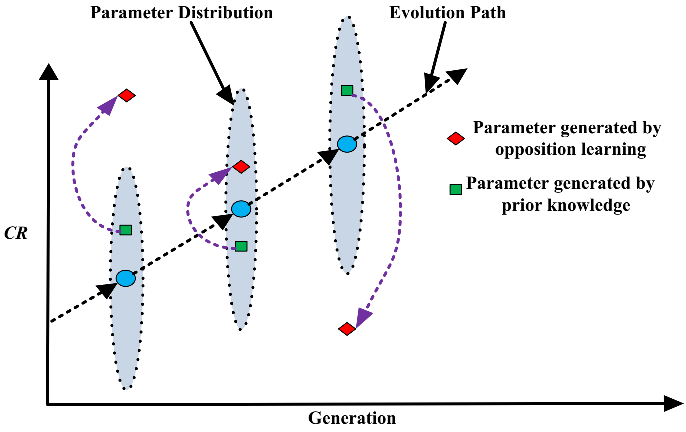

Fig. 2 Evolution of crossover control parameter

$$CR_i^{G+1} = 1 - N((1 - 0.7 \times (1 - G/G_{\max})), \sigma_2), i = 1, \cdots, NP, \tag{32}$$

where N is a normal distribution function and $\sigma_2 = 0.8 - 0.6 \times (1 - (G/G_{\max})^2)$.

$$CR_i^{G+1} = N((1 - 0.7 \times (1 - G/G_{\max})), \sigma_2), i = 1, \cdots, NP, \tag{33}$$

4.4 Overall implementation of SAKPDE

(1) Initialization operator
Determine the maximal number of generation $G_{\max}$ and population size. Initialize a population $\boldsymbol{P}_1^0$ that is generated randomly within the feasible search space $\boldsymbol{S}$. Set the current generation $G = 0, \varphi = 0.7, \rho = 0.8$ (opposition learning ratio), and $G_{\mathrm{s}} = 0.3 \times G_{\mathrm{s}}$ (a single mutation strategy is used before G_{s} generation). The initial number of individuals belonging to each mutation and crossover strategies are set as: $N_{\mathrm{str\_rand/1}}^{G_s} = N_{\mathrm{str\_rand/2}}^{G_s} = N_{\mathrm{str\_current\text{-}to\text{-}best/1}}^{G_s} = N_{\mathrm{str\_best/2}}^{G_s} = N_{\mathrm{str\_current\text{-}to\text{-}best/2}}^{G_s} = NP/5$, $N_{\mathrm{cstr\_bin}}^{G_s} = N_{\mathrm{cstr\_exp}}^{G_s} = N_{\mathrm{cstr\_eig}}^{G_s} = NP/3$.

(2) Population evolution
Mutation operation For each parent individual $\boldsymbol{x}_i^G$, if $G < G_{\mathrm{s}}$, then the mutation strategy DE/rand/1 is utilized to generate a trial individual for the next generation G+1; otherwise, a mutant vector $\boldsymbol{v}_i^G$ is produced by a mutation strategy which is chosen randomly from the mutation strategy pool.
Boundary operation If $v_{ij}^G < x_j^{\mathrm{low}}$ or $v_{ij}^G > x_j^{\mathrm{high}}$, then $v_{ij}^G = x_{r_1 j}^G$.
Crossover operation If $G < G_{\mathrm{s}}$, then the binomial crossover strategy is used to produce a trial individual $\boldsymbol{u}_i^G$; otherwise, $\boldsymbol{u}_i^G$ is produced by the corresponding crossover strategy that is selected randomly from the crossover strategy pool.

(3) Self-adaptive mutation strategy
The detailed implementation steps of self-adaptive mutation strategy are shown in Sect. 4.1.

(4) Self-adaptive crossover strategy

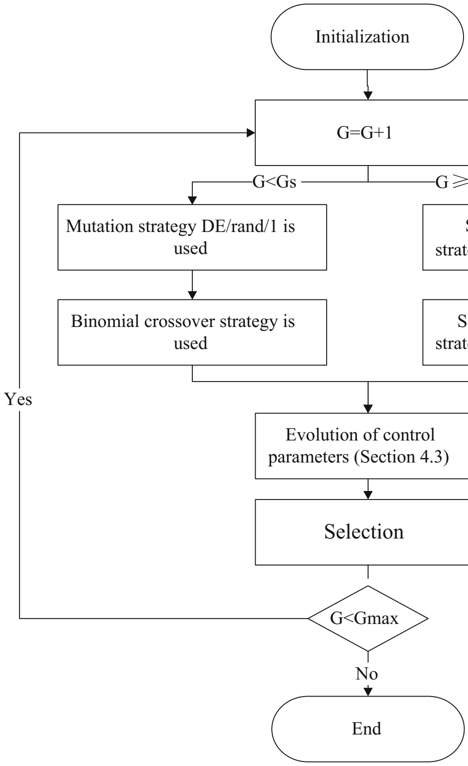

Fig. 3 Framework of the proposed algorithm

The detailed descriptions of self-adaptive crossover strategy are presented in Sect. 4.2.

(5) Self-adaptive control parameters
The detailed implementations of self-adaptive control parameters are given in Sect. 4.3.

(6) Generation updating
$G = G + 1$.

(7) Looping
Steps 2–6 are repeated until the number of generations is equal to $G_{\max}$.

The flowchart of the proposed algorithm is shown in Fig. 3.

5 Experimental results

In this part, three sets of benchmark test functions are utilized to comprehensively investigate the search capability of SAKPDE. The more detailed descriptions can be seen in Hansen et al. (2012), Liang et al. (2013) and Suganthan et al. (2005). Moreover, we compare the proposed algorithm with eight well-known improved DE algorithms [i.e., jDE (Brest et al. 2006), SaDE (Qin et al. 2009), JADE (Zhang and Sanderson 2009), CoDE (Wang et al. 2011), EPSDE (Mallipeddi et al. 2011), CoBiDE (Wang et al. 2014), L-AHADE (Tanabe and Fukunaga 2014), and GaAPADE (Mallipeddi et al. 2014)] and four non-DEs (i.e., UMOEAs (Elsayed et al. 2014), CMLSP (Chen et al. 2014), MVMO (Erlich et al. 2014), and rmalschcma (Molina et al. 2014)). All compared evolutionary algorithms are coded in Matlab (Matlab

R2012a) and run on a Windows 7 operating system (64 bit). The maximum number of fitness evaluations is set to be $10{,}000 \times D$ for each test function in all experiments. Because each evolutionary algorithm has its own appropriate *NP*, we use recommended *NP* from their papers, namely, 100 for jDE and JADE, 50 for SaDE and EPSDE, 60 for CoBiDE, and 30 for CoDE. The population size of SAKPDE is set as 100. Note that the results of six algorithms (i.e., L-AHADE, GaAPADE, UMOEAs, CMLSP, MVMO, and rmalschcma) introduced in IEEE CEC2014 are directly taken from their original literatures. Additionally, to evaluate the performances of all compared algorithms effectively; six non-parametric statistical tests [i.e., Wilcoxon's rank sum test (Wilcoxon 1945), Friedman's test (Friedman 1937), Iman-Davenport test (García et al. 2009), Bonferroni–Dunn's test (Dunn 1961), Holm's procedure, and Hochberg's procedure (García et al. 2009)] at the 0.05 significance level are used. For the Wilcoxon's rank sum test, the "+", "–", and "≈" signs denote that the performance of SAKPDE is significantly better than, worse than, and almost similar to that of other compared algorithms in a statistically significant way, respectively. The accuracy level of results for all experiments is set to be 1E-8 (i.e., the result less than 1E-8 is set as zero).

5.1 Comparison with six DE variants on 30-dimensional test functions

In this section, to demonstrate average performance of the proposed algorithm, SAKPDE is compared with six state-of-the-art DE variants on 25 IEEE CEC2005 and 24 BBOB2012 test functions in 30 dimensions. Each test function is independently run 50 times. The simulation and statistical analysis results are shown in Table 2. For unimodal test functions ($F1_{CEC2005} - F5_{CEC2005}, F1_{BBOB}, F2_{BBOB},$ and $F5_{BBOB} - F14_{BBOB}$), SAKPDE performs better than jDE, SaDE, JADE, CoDE, EPSDE, and CoBiDE on twelve, eleven, four, nine, six, and eight functions, respectively. The search capability of JADE, EPSDE, and CoBiDE is significantly better than that of the proposed algorithm on five, one, and one functions, respectively. However, jDE, SaDE, and CoDE cannot perform better than SAKPDE on any test functions. From Table 2, we can find that the exploitation capability of JADE is the best due to its greedy mutation strategy. For multimodal test functions ($F6_{CEC2005} - F14_{CEC2005}, F3_{BBOB}, F4_{BBOB},$ and $F15_{BBOB} - F24_{BBOB}$), SAKPDE significantly outperforms jDE, SaDE, JADE, CoDE, EPSDE, and CoBiDE on sixteen, sixteen, eleven, nine, fourteen, and seven test functions, respectively. The performances of jDE, SaDE, JADE, CoDE, EPSDE, and CoBiDE are significantly better than that of SAKPDE on four, one, six, four, three, and five functions, respectively. It can be observed from Table 2 that the exploration ability of SAKPDE is better than that of other compared algorithms. For hybrid composition functions $F15_{CEC2005} - F25_{CEC2005}$, as these test functions are more complex, the performance comparison on these functions is more comprehensive and convincing. The results in Table 2 show that SAKPDE outperforms jDE, SaDE, JADE, CoDE, EPSDE, and CoBiDE on five, three, five, five, five, and four test functions, respectively. However, all compared algorithms except for EPSDE cannot outperform SAKPDE on any test functions. The main reason may be that SAKPDE can balance the exploration and exploitation capabilities of DE, and automatically select appropriate strategies (mutation and crossover) and effectively guide the evolution of the control parameters (F and CR). Although EPSDE performs better than SAKPDE on five test functions, the overall performance of SAKPDE is much better than that of EPSDE on other test functions. Overall, the statistical analysis results (obtained by Wilcoxon's rank sum test) in Table 2 present that SAKPDE exhibits better search capability when compared with other competitors on these 30-dimensional test functions.

Table 2 Optimization results for 30-dimensional test functions

Function	jDE Mean(Std)	SaDE Mean(Std)	JADE Mean(Std)	CoDE Mean(Std)
$F1_{CEC2005}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F2_{CEC2005}$	4.78E−07(6.58E−07)+	1.12E−05(2.35E−05)+	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F3_{CEC2005}$	2.03E+05(1.32E+05)+	5.35E+05(2.30E+05)+	8.18E+03(6.86E+03)−	1.20E+05(7.11E+04)+
$F4_{CEC2005}$	3.44E−02(1.04E−01)+	1.45E+02(1.98E+02)+	0.00E+00(0.00E+00)≈	5.17E−03(1.06E−02)+
$F5_{CEC2005}$	4.46E+02(3.67E+02)+	3.13E+03(4.95E+02)+	4.86E−07(2.35E−06)−	3.64E+02(3.68E+02)+
$F6_{CEC2005}$	2.25E+01(2.41E+01)+	4.01E+01(3.54E+01)+	2.58E+00(7.46E+00)+	1.33E−01(7.28E−01)≈
$F7_{CEC2005}$	1.16E−02(9.04E−03)+	1.84E−02(1.80E−02)+	9.36E−03(9.02E−03)+	8.86E−03(9.57E−03)≈
$F8_{CEC2005}$	2.09E+01(7.51E−02)+	2.09E+01(3.67E−02)+	2.09E+01(1.43E−01)+	2.02E+01(1.48E−01)+
$F9_{CEC2005}$	0.00E+00(0.00E+00)≈	9.95E−02(3.04E−01)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F10_{CEC2005}$	5.79E+01(9.38E+00)+	4.48E+01(1.25E+01)≈	2.43E+01(5.23E+00)−	4.16E+01(1.15E+01)≈
$F11_{CEC2005}$	2.83E+01(1.25E+00)+	1.65E+01(2.94E+00)+	2.53E+01(1.29E+00)+	1.26E+01(3.43E+00)+
$F12_{CEC2005}$	1.20E+04(9.44E+03)+	2.17E+03(1.82E+03)≈	6.68E+03(4.17E+03)+	3.34E+03(4.51E+03)≈
$F13_{CEC2005}$	1.66E+00(1.28E−01)−	3.90E+00(4.64E−01)+	1.48E+00(1.26E−01)−	1.58E+00(3.43E−01)−
$F14_{CEC2005}$	1.30E+01(2.69E−01)+	1.26E+01(2.86E−01)+	1.23E+01(2.96E−01)≈	1.24E+01(3.76E−01)≈
$F15_{CEC2005}$	3.18E+02(1.19E+02)≈	3.74E+02(6.42E+01)≈	3.76E+02(8.97E+01)≈	4.03E+02(6.69E+01)+
$F16_{CEC2005}$	8.49E+01(3.59E+01)+	7.71E+01(2.74E+01)≈	9.63E+01(1.08E+02)+	6.39E+01(2.15E+01)≈
$F17_{CEC2005}$	1.39E+02(1.25E+01)+	8.70E+01(8.49E+01)≈	1.02E+02(6.68E+01)+	8.50E+01(7.21E+01)≈
$F18_{CEC2005}$	9.04E+02(7.96E−01)+	8.78E+02(6.06E+01)≈	9.04E+02(7.92E−01)+	9.05E+02(1.30E+00)+
$F19_{CEC2005}$	9.04E+02(7.69E−01)+	8.60E+02(6.13E+01)≈	9.04E+02(9.94E−01)+	9.04E+02(1.14E+00)+
$F20_{CEC2005}$	9.04E+02(9.216E−01)+	8.73E+02(6.14E+01)≈	9.04E+02(5.93E−01)+	9.04E+02(6.41E−01)+
$F21_{CEC2005}$	5.00E+02(1.96E−13)≈	5.43E+02(1.65E+02)+	5.10E+02(5.48E+01)≈	5.00E+02(1.95E−13)≈
$F22_{CEC2005}$	8.67E+02(1.87E+01)≈	9.36E+02(1.88E+01)+	8.64E+02(2.57E+01)≈	8.63E+02(2.68E+01)≈
$F23_{CEC2005}$	5.34E+02(2.98E−04)≈	5.69E+02(1.34E+02)+	5.34E+02(2.17E−04)≈	5.34E+02(4.32E−04)+
$F24_{CEC2005}$	2.00E+02(2.89E−14)≈	2.00E+02(2.89E−14)≈	2.00E+02(2.89E−14)≈	2.00E+02(2.89E−14)≈

Table 2 continued

Function	jDE Mean(Std)	SaDE Mean(Std)	JADE Mean(Std)	CoDE Mean(Std)
$F25_{CEC2005}$	2.11E+02(7.01E−01)+	2.13E+02(1.39E+00)+	2.11E+02(8.53E−01)+	2.11E+02(7.29E−01)+
$F1_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F2_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F3_{BBOB2012}$	0.00E+00(0.00E+00)−	5.97E−01(1.38E+00)+	0.00E+00(0.00E+00)−	9.94E−02(3.02E−01)≈
$F4_{BBOB2012}$	2.98E−01(5.77E−01)−	4.06E+00(2.68E+00)+	0.00E+00(0.00E+00)−	8.34E−01(9.72E−01)−
$F5_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F6_{BBOB2012}$	6.99E−07(9.51E−07)+	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F7_{BBOB2012}$	4.34E+00(1.59E+00)+	1.17E+01(3.50E+00)+	3.34E+00(1.53E+00)+	4.19E+00(1.67E+00)+
$F8_{BBOB2012}$	8.46E+00(1.17E+00)+	5.74E−01(1.53E+00)+	1.59E−01(7.89E−01)+	7.97E−02(5.64E−01)≈
$F9_{BBOB2012}$	1.91E+01(1.72E+00)+	2.40E+01(2.08E+00)+	7.97E−02(5.64E−01)−	4.20E+00(1.45E+00)+
$F10_{BBOB2012}$	5.09E+02(3.12E+02)+	1.80E+03(8.44E+02)+	2.05E+01(1.69E+01)−	3.08E+02(1.67E+02)+
$F11_{BBOB2012}$	2.62E−02(3.58E−02)+	8.43E+00(4.23E+00)+	9.72E+00(2.70E+01)+	7.08E−04(2.14E−03)+
$F12_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F13_{BBOB2012}$	1.00E+00(1.42E+00)+	2.71E+00(3.22E+00)+	1.26E+00(1.32E+00)+	1.23E+00(1.88E+00)+
$F14_{BBOB2012}$	5.83E−05(1.82E−05)+	2.20E−04(4.06E−05)+	7.24E−06(2.97E−06)−	4.22E−05(1.01E−05)+
$F15_{BBOB2012}$	6.46E+01(1.12E+01)+	5.55E+01(1.32E+01)+	2.78E+01(5.82E+00)−	3.80E+01(1.32E+01)≈
$F16_{BBOB2012}$	1.02E+01(1.53E+00)+	4.45E+00(3.30E+00)+	7.68E+00(9.90E−01)+	2.03E+00(1.30E+00)+
$F17_{BBOB2012}$	5.68E−03(1.22E−02)+	3.67E−01(2.04E−01)+	7.15E−04(8.24E−04)≈	3.19E−02(4.11E−02)+
$F18_{BBOB2012}$	1.36E−01(9.50E−02)+	1.71E+00(8.24E−01)+	5.32E−02(4.78E−02)≈	4.29E−01(2.31E−01)+
$F19_{BBOB2012}$	4.86E+00(4.16E−01)+	4.29E+00(3.70E−01)+	2.70E+00(5.10E−01)+	9.62E−01(5.46E−01)+
$F20_{BBOB2012}$	3.46E−01(9.36E−02)+	4.22E−01(1.29E−01)+	4.90E−01(1.00E−01)+	3.85E−01(1.50E−01)+
$F21_{BBOB2012}$	5.40E+00(5.88E+00)+	4.19E+00(5.35E+00)≈	6.16E+00(7.35E+00)+	3.01E+00(3.10E+00)+
$F22_{BBOB2012}$	1.28E+01(4.43E+00)−	1.34E+01(3.71E+00)−	1.21E+01(5.08E+00)−	1.13E+01(5.60E+00)−

Table 2 continued

Function	jDE Mean(Std)	SaDE Mean(Std)	JADE Mean(Std)	CoDE Mean(Std)
$F23_{BBOB2012}$	2.33E+00(2.48E−01)+	2.31E+00(2.32E−01)+	1.94E+00(6.01E−01)+	3.51E−01(1.85E−01)+
$F24_{BBOB2012}$	1.59E+02(1.38E+01)+	1.26E+02(3.68E+01)+	7.69E+01(7.81E+00)+	6.05E+01(1.02E+01)−
+	34	31	21	24
−	4	1	11	4
≈	11	17	17	21

Function	EPSDE Mean(Std)	CoBiDE Mean(Std)	SAKPDE Mean(Std)
$F1_{CEC2005}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
$F2_{CEC2005}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
$F3_{CEC2005}$	1.42E+05(3.80E+05)≈	7.80E+04(5.03E+04)+	4.14E+04(2.84E+04)
$F4_{CEC2005}$	5.27E+01(1.34E+02)+	8.88E−04(1.93E−03)+	0.00E+00(0.00E+00)
$F5_{CEC2005}$	1.40E+03(8.58E+02)+	3.71E+01(5.26E+01)+	2.56E+00(4.28E+00)
$F6_{CEC2005}$	2.66E−01(1.01E+00)≈	1.66E−01(7.23E−01)+	0.00E+00(0.00E+00)
$F7_{CEC2005}$	1.32E−02(1.55E−02)+	3.68E−03(8.30E−03)≈	3.86E−03(5.40E−03)
$F8_{CEC2005}$	2.10E+01(4.39E−02)+	2.07E+01(4.05E−01)+	2.01E+01(2.79E−01)
$F9_{CEC2005}$	3.31E−02(1.82E−01)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
$F10_{CEC2005}$	4.62E+01(7.43E+00)+	4.34E+01(1.40E+01)≈	4.03E+01(1.41E+00)
$F11_{CEC2005}$	3.43E+01(3.94E+00)+	5.67E+00(2.36E+00)−	1.08E+01(2.93E+00)
$F12_{CEC2005}$	3.81E+04(5.88E+03)+	2.96E+03(2.60E+03)≈	3.10E+03(2.92E+03)
$F13_{CEC2005}$	1.89E+00(2.06E−01)≈	2.64E+00(1.08E+00)≈	2.10E+00(6.08E−01)
$F14_{CEC2005}$	1.34E+01(3.75E−01)+	1.22E+01(5.32E−01)≈	1.23E+01(4.01E−01)
$F15_{CEC2005}$	2.17E+02(3.74E+01)−	4.10E+02(5.47E+01)+	3.60E+02(6.74E+01)
$F16_{CEC2005}$	1.50E+02(1.23E+02)+	8.42E+01(6.31E+01)≈	6.62E+01(1.22E+01)
$F17_{CEC2005}$	1.89E+02(1.01E+02)+	6.82E+01(2.05E+01)≈	6.70E+01(1.30E+01)
$F18_{CEC2005}$	8.21E+02(3.59E+00)−	9.04E+02(9.98E−01)+	9.00E+02(0.00E+00)

Table 2 continued

Function	EPSDE Mean(Std)	CoBiDE Mean(Std)	SAKPDE Mean(Std)
$F19_{CEC2005}$	8.20E+02(3.31E+00)−	9.04E+02(8.62E−01)+	9.00E+02(0.00E+00)
$F20_{CEC2005}$	8.21E+02(3.15E+00)−	9.04E+02(5.47E−01)+	9.00E+02(0.00E+00)
$F21_{CEC2005}$	8.52E+02(6.66E+01)+	5.00E+02(1.75E−13)≈	5.00E+02(2.31E−13)
$F22_{CEC2005}$	5.19E+02(6.36E+01)−	8.54E+02(2.46E+01)≈	8.61E+02(1.95E+01)
$F23_{CEC2005}$	8.58E+02(6.14E+01)+	5.34E+02(4.70E−07)≈	5.34E+02(3.10E−04)
$F24_{CEC2005}$	2.14E+02(2.21E+00)+	2.00E+02(2.89E−14)≈	2.00E+02(2.89E−14)
$F25_{CEC2005}$	2.13E+02(2.44E+00)+	2.10E+02(5.65E−01)≈	2.10E+02(8.85E−01)
$F1_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
$F2_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
$F3_{BBOB2012}$	3.98E−02(1.97E−01)≈	0.00E+00(0.00E+00)−	9.95E−02(3.02E−01)
$F4_{BBOB2012}$	9.35E−01(1.14E+00)−	5.97E−02(2.39E−01)−	1.59E+00(1.10E+00)
$F5_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
$F6_{BBOB2012}$	0.00E+00(0.00E+00)≈	1.00E−08(5.70E−09)+	0.00E+00(0.00E+00)
$F7_{BBOB2012}$	9.20E+00(4.27E+00)+	2.02E+00(9.89E−01)≈	2.17E+00(8.99E−01)
$F8_{BBOB2012}$	1.59E−01(7.89E−01)≈	1.12E−04(4.17E−03)+	0.00E+00(0.00E+00)
$F9_{BBOB2012}$	1.00E+01(1.90E+00)+	5.71E+00(8.73E+00)+	1.85E+00(8.23E−01)
$F10_{BBOB2012}$	3.44E+03(1.19E+04)≈	2.03E+02(1.26E+02)+	1.29E+02(8.79E+01)
$F11_{BBOB2012}$	9.11E+00(3.49E+01)+	6.05E−07(2.55E−06)−	1.30E−04(3.25E−04)
$F12_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
$F13_{BBOB2012}$	3.30E+00(4.34E+00)+	1.98E−03(3.35E−03)≈	1.76E−03(2.92E−03)
$F14_{BBOB2012}$	3.58E−05(1.91E−04)−	3.53E−05(1.36E−05)+	2.12E−05(8.42E−06)
$F15_{BBOB2012}$	5.49E+01(1.09E+01)+	4.11E+01(1.19E+01)+	3.42E+01(1.17E+01)
$F16_{BBOB2012}$	1.53E+01(4.47E+00)+	4.59E−01(4.13E−01)−	1.40E+00(9.03E−01)
$F17_{BBOB2012}$	3.77E−02(4.50E−02)+	1.32E−03(2.28E−03)≈	6.80E−04(5.84E−04)
$F18_{BBOB2012}$	3.94E−01(3.20E−01)+	8.58E−02(5.52E−02)+	3.35E−02(2.16E−02)

Table 2 continued

Function	EPSDE Mean(Std)	CoBiDE Mean(Std)	SAKPDE Mean(Std)
$F19_{BBOB2012}$	4.47E+00(3.54E−01)+	3.20E+00(2.10E+00)+	1.57E−02(4.39E−15)
$F20_{BBOB2012}$	3.37E−01(1.19E−01)+	3.86E−01(1.67E−01)+	2.77E−01(1.85E−01)
$F21_{BBOB2012}$	1.98E+00(4.15E+00)−	4.09E+00(4.77E+00)+	1.90E+00(2.23E+00)
$F22_{BBOB2012}$	1.18E+01(5.28E+00)−	9.79E+00(6.17E+00)−	1.46E+01(8.13E−15)
$F23_{BBOB2012}$	2.45E+00(3.05E−01)+	7.45E−01(9.97E−01)≈	1.67E−01(7.63E−02)
$F24_{BBOB2012}$	1.38E+02(3.02E+01)+	6.51E+01(1.11E+01)≈	6.80E+01(1.21E+01)
+	26	19	
−	9	6	
≈	14	24	

5.2 Comparison with six DE variants on 50-dimensional test functions

In this experiment, 25 IEEE CEC2005 and 24 BBOB2012 benchmark test functions in 50 dimensions are utilized to investigate the performance of SAKPDE. Similar to Sect. 5.1, the optimization performance of SAKPDE is compared with that of six improved DE algorithms and all obtained results are based on 50 independent runs for each function. The results are summarized in Table 3. For unimodal test functions ($F1_{CEC2005}$–$F5_{CEC2005}$, $F1_{BBOB}$, $F2_{BBOB}$, and $F5_{BBOB} - F14_{BBOB}$), SAKPDE outperforms jDE, SaDE, JADE, CoDE, EPSDE, and CoBiDE on eleven, twelve, three, six, eight, and seven functions, respectively. SaDE cannot perform better than the proposed algorithm on any test functions. The performance of jDE, JADE, CoDE, EPSDE, and CoBiDE is significantly better than that of SAKPDE on one, five, three, one, and two test functions, respectively. We can see from Table 3 that the local search capability of JADE is the best among all compared algorithms. However, Table 3 shows that the exploitation capability of SAKPDE is slightly worse than that of JADE. Moreover, SAKPDE can automatically select suitable strategies and control the evolution of parameters to improve the local search capability of DE. For multimodal test functions ($F6_{CEC2005}$–$F14_{CEC2005}$, $F3_{BBOB}$, $F4_{BBOB}$, and $F15_{BBOB}$–$F24_{BBOB}$), the performance of SAKPDE is significantly better than that of jDE, SaDE, JADE, CoDE, EPSDE, and CoBiDE on thirteen, sixteen, ten, ten, sixteen, and eight functions, respectively. Moreover, Table 3 indicates that jDE, SaDE, JADE, CoDE, EPSDE, and CoBiDE outperforms SAKPDE on six, two, eight, four, five, and six test functions, respectively. From Table 3, we can conclude that the overall performance of SAKPDE will decrease with the growth of optimization problem in dimensionality. For hybrid composition functions $F15_{CEC2005}$–$F25_{CEC2005}$, Table 3 shows that SAKPDE significantly performs better than jDE, SaDE, JADE, CoDE, EPSDE, and CoBiDE on eight, eleven, eight, seven, seven, and six test functions, respectively. Additionally, all compared DE variants except for EPSDE cannot significantly outperform the proposed algorithm on any test functions. Finally, the statistical analysis results shown in Table 3 indicate that the average performance of SAKPDE is the best among all compared algorithms on 49 50-dimensional test functions.

5.3 Comparison with six evolutionary algorithms on 30-dimensional CEC2014 functions

In this section, to further demonstrate the search performance of SAKPDE, 30 30-dimensional IEEE CEC2014 benchmark test functions are used. Moreover, the performance of SAKPDE is compared with that of six evolutionary algorithms introduced in IEEE CEC2014. Note that the results of six evolutionary algorithms are taken from their original literatures. In addition, because the mean values of six evolutionary algorithms are obtained based on 51 independent runs on each test function, times of independent runs for SAKPDE are also set to be 51 in this experiment. The mean values of all compared algorithms are shown in Table 4 and the best result is highlighted in bold. To detect the difference between SAKPDE and other compared algorithms, the average rankings obtained by Friedman's test are shown in Fig. 4. From Fig. 4, we can see that the average performance of L-SHADE is the best among all compared algorithms and the proposed algorithm is ranked the second. The main reason is that L-SHADE uses a greedy mutation strategy as well as JADE and a linear population size reduction mechanism. Therefore, the local capability of L-SHADE is better than that of SAKPDE on most of unimodal and basic multimodal test functions. However, for complex functions like hybrid ($F17_{CEC2014}$–$F22_{CEC2014}$) and composition ($F23_{CEC2014}$–$F30_{CEC2014}$) functions, the statistical analysis results achieved by Bonferroni–Dunn's, Holm's, Hochberg's

Table 3 Optimization results for 50-dimensional test functions

Function	jDE Mean(Std)	SaDE Mean(Std)	JADE Mean(Std)	CoDE Mean(Std)
$F1_{CEC2005}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F2_{CEC2005}$	8.35E−03(9.43E−03)+	8.03E−02(7.62E−02)+	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F3_{CEC2005}$	4.78E+05(1.99E+05)+	9.31E+05(2.63E+05)+	1.77E+04(9.43E+03)−	1.72E+05(5.43E+04)−
$F4_{CEC2005}$	4.33E+02(3.21E+02)+	5.88E+03(2.52E+03)+	2.21E+0(7.63E+00)+	4.29E+02(3.04E+02)+
$F5_{CEC2005}$	3.25E+03(7.18E+02)+	8.31E+03(1.56E+03)+	1.53E+03(4.76E+02)−	3.32E+03(5.21E+02)+
$F6_{CEC2005}$	4.32E+01(2.85E+01)+	9.48E+01(4.81E+01)+	6.64E−01(1.51E+00)≈	1.17E+00(2.40E+00)+
$F7_{CEC2005}$	4.18E−03(9.89E−03)+	5.08E−03(9.98E−03)+	5.33E−03(8.76E−03)+	5.00E−03(9.05E−03)≈
$F8_{CEC2005}$	2.11E+01(3.53E−02)+	2.11E+01(3.44E−02)+	2.10E+01(3.56E−01)+	2.01E+01(9.57E−02)+
$F9_{CEC2005}$	0.00E+00(0.00E+00)−	1.46E+00(1.25E+00)≈	0.00E+00(0.00E+00)−	6.63E−01(7.98E−01)≈
$F10_{CEC2005}$	1.05E+02(1.72E+01)+	1.30E+02(2.42E+01)+	4.94E+01(8.04E+00)−	8.01E+01(1.69E+01)≈
$F11_{CEC2005}$	5.42E+01(2.07E+00)+	3.92E+01(4.78E+00)+	5.16E+01(2.36E+00)+	3.05E+01(4.70E+00)+
$F12_{CEC2005}$	1.59E+04(1.34E+04)≈	1.58E+04(1.10E+04)≈	1.56E+04(1.14E+04)≈	1.35E+04(1.25E+04)≈
$F13_{CEC2005}$	3.05E+00(2.19E−01)−	9.90E+00(1.08E+00)+	2.75E+00(1.39E−01)−	3.11E+00(6.18E−01)−
$F14_{CEC2005}$	2.25E+01(2.13E−01)+	2.23E+01(2.88E−01)≈	2.17E+01(5.07E−01)−	2.20E+01(5.16E−01)≈
$F15_{CEC2005}$	3.20E+02(9.97E+01)+	3.64E+02(8.01E+01)+	3.49E+02(8.74E+01)+	3.47E+02(9.00E+01)+
$F16_{CEC2005}$	8.69E+01(2.07E+01)+	9.29E+01(5.96E+01)+	8.06E+01(9.15E+01)+	7.01E+01(1.81E+01)≈
$F17_{CEC2005}$	1.83E+02(4.92E+01)+	8.23E+01(1.49E+01)+	1.58E+02(1.27E+02)+	7.52E+01(3.03E+01)≈
$F18_{CEC2005}$	9.20E+02(2.45E+00)+	9.91E+02(1.58E+01)+	9.22E+02(2.74E+00)+	9.21E+02(3.87E+00)+
$F19_{CEC2005}$	9.20E+02(3.59E+00)+	9.79E+02(3.72E+01)+	9.24E+02(3.51E+00)+	9.23E+02(4.37E+00)+
$F20_{CEC2005}$	9.20E+02(3.17E+00)+	9.84E+02(1.49E+01)+	9.21E+02(4.01E+00)+	9.23E+02(3.99E+00)+
$F21_{CEC2005}$	6.85E+02(2.47E+02)+	6.39E+02(2.82E+02)+	5.51E+02(1.55E+02)≈	6.36E+02(2.29E+02)≈
$F22_{CEC2005}$	9.00E+02(9.42E+00)≈	9.85E+02(1.27E+01)+	9.00E+02(2.30E+01)≈	9.03E+02(2.04E+01)+
$F23_{CEC2005}$	8.39E+02(2.32E+02)+	7.36E+02(3.06E+02)+	6.34E+02(1.92E+02)+	7.29E+02(2.37E+02)+
$F24_{CEC2005}$	2.00E+02(1.56E−12)≈	6.29E+02(4.87E+02)+	2.00E+02(1.51E−12)≈	2.00E+02(5.99E−13)≈

Table 3 continued

Function	jDE Mean(Std)	SaDE Mean(Std)	JADE Mean(Std)	CoDE Mean(Std)
$F25_{CEC2005}$	2.16E+02(1.39E+00)≈	2.24E+02(3.94E+00)+	2.19E+02(2.76E+00)+	2.18E+02(1.95E+00)+
$F1_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F2_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F3_{BBOB2012}$	0.00E+00(0.00E+00)−	8.06E+00(5.71E+00)+	0.00E+00(0.00E+00)−	1.53E+00(1.27E+00)−
$F4_{BBOB2012}$	7.56E−01(7.67E−01)−	1.78E+01(1.17E+01)+	3.98E−02(1.97E−01)−	5.33E+00(2.14E+00)−
$F5_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F6_{BBOB2012}$	1.35E−03(2.35E−03)+	9.63E−05(3.83E−04)+	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F7_{BBOB2012}$	1.87E+01(5.89E+00)+	4.81E+01(1.29E+01)+	1.82E+01(4.62E+00)+	1.91E+01(4.62E+00)+
$F8_{BBOB2012}$	2.10E+01(1.75E+00)+	1.04E+00(1.77E+00)+	3.19E−01(1.09E+00)≈	2.83E−01(9.49E−01)+
$F9_{BBOB2012}$	4.36E+01(8.97E+00)+	4.42E+01(7.38E+00)+	5.10E−01(9.67E−01)≈	3.45E+01(7.86E+00)+
$F10_{BBOB2012}$	1.39E+03(5.33E+02)+	4.00E+03(1.15E+03)+	6.29E+01(2.56E+01)−	5.43E+02(2.42E+02)−
$F11_{BBOB2012}$	1.37E−01(2.12E−01)+	1.79E+01(6.63E+00)+	1.88E+01(4.40E+01)−	1.50E−03(2.26E−03)−
$F12_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈
$F13_{BBOB2012}$	1.17E+00(1.68E+00)−	5.49E+00(4.31E+00)+	2.09E+00(2.75E+00)+	1.86E+00(2.00E+00)+
$F14_{BBOB2012}$	1.32E−04(3.31E−05)+	4.02E−04(4.70E−05)+	1.81E−05(3.42E−06)−	9.01E−05(1.65E−05)≈
$F15_{BBOB2012}$	1.22E+02(2.02E+01)+	1.56E+02(2.94E+01)+	5.98E+01(1.08E+01)−	8.24E+01(2.23E+01)+
$F16_{BBOB2012}$	1.51E+01(1.51E+00)+	6.91E+00(1.77E+00)+	1.26E+01(1.47E+00)+	4.68E+00(1.94E+00)+
$F17_{BBOB2012}$	2.27E−02(3.52E−02)+	1.85E+00(3.86E−01)+	2.89E−02(2.40E−02)+	3.07E−01(1.90E−01)+
$F18_{BBOB2012}$	5.79E−01(3.04E−01)+	6.43E+00(1.49E+00)+	4.07E−01(1.66E−01)+	1.63E+00(6.06E−01)+
$F19_{BBOB2012}$	5.94E+00(3.55E−01)+	5.40E+00(3.15E−01)+	3.38E+00(7.16E−01)+	1.21E+00(5.55E−01)+
$F20_{BBOB2012}$	4.62E−01(7.73E−02)−	7.34E−01(1.12E−01)+	7.21E−01(7.30E−02)+	6.53E−01(1.58E−01)+
$F21_{BBOB2012}$	2.41E+00(3.63E−01)≈	2.36E+00(4.98E−01)−	2.76E+00(2.00E+00)≈	2.37E+00(4.51E−01)≈
$F22_{BBOB2012}$	8.52E+00(6.37E+00)−	8.52E+00(6.37E+00)−	8.78E+00(6.36E+00)−	4.99E+00(5.45E+00)−
$F23_{BBOB2012}$	3.01E+00(3.63E−01)+	3.08E+00(3.16E−01)+	1.92E+00(7.22E−01)+	9.06E−01(4.49E−01)+
$F24_{BBOB2012}$	2.74E+02(2.60E+01)+	1.40E+02(5.78E+01)+	1.40E+02(1.54E+00)+	1.13E+02(1.81E+01)≈

Table 3 continued

Function	jDE Mean(Std)	SaDE Mean(Std)	JADE Mean(Std)	CoDE Mean(Std)
+	32	39	21	23
−	7	2	13	7
≈	10	8	15	19

Function	EPSDE Mean(Std)	CoBiDE Mean(Std)	SAKPDE Mean(Std)
$F1_{CEC2005}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
$F2_{CEC2005}$	0.00E+00(0.00E+00)≈	1.39E−06(1.77E−06)+	0.00E+00(0.00E+00)
$F3_{CEC2005}$	1.09E+07(2.85E+07)≈	2.47E+05(1.00E+05)≈	2.35E+05(9.34E+04)
$F4_{CEC2005}$	3.21E+03(3.21E+03)+	2.30E+02(1.97E+02)+	1.55E+01(8.16E+01)
$F5_{CEC2005}$	4.64E+03(1.10E+03)+	2.60E+03(6.25E+02)+	2.14E+03(5.37E+02)
$F6_{CEC2005}$	1.19E+00(1.85E+00)+	2.89E+01(2.33E+01)+	1.66E−01(9.04E−01)
$F7_{CEC2005}$	8.27E−03(1.24E−02)+	4.92E−03(7.25E−03)+	6.57E−04(2.50E−03)
$F8_{CEC2005}$	2.11E+01(5.31E−01)+	2.08E+01(5.26E−01)+	2.00E+01(1.90E−01)
$F9_{CEC2005}$	3.31E−02(1.82E−01)−	8.52E−13(3.14E−12)−	7.29E−01(9.02E−01)
$F10_{CEC2005}$	1.56E+02(1.87E+01)+	8.51E+01(2.04E+01)≈	7.86E+01(1.96E+01)
$F11_{CEC2005}$	7.03E+01(3.79E+00)+	1.89E+01(4.43E+00)−	2.71E+01(5.22E+00)
$F12_{CEC2005}$	3.18E+05(4.93E+04)+	1.68E+04(1.79E+04)≈	1.53E+04(1.15E+04)
$F13_{CEC2005}$	6.14E+00(3.52E−01)+	4.36E+00(1.33E+00)≈	3.75E+00(7.82E−01)
$F14_{CEC2005}$	2.34E+01(2.61E−01)+	2.19E+01(4.06E−01)≈	2.20E+01(4.71E−01)
$F15_{CEC2005}$	2.62E+02(6.25E+01)+	3.87E+02(5.07E+01)+	2.46E+02(8.60E+01)
$F16_{CEC2005}$	1.59E+02(8.59E+01)+	8.48E+01(6.33E+01)≈	6.52E+01(1.27E+01)
$F17_{CEC2005}$	2.29E+02(8.18E+00)+	7.82E+01(2.97E+01)+	6.34E+01(1.40E+01)
$F18_{CEC2005}$	8.45E+02(2.44E+00)−	9.18E+02(2.80E+00)+	9.00E+02(0.00E+00)
$F19_{CEC2005}$	8.46E+02(2.82E+00)−	9.14E+02(2.17E+01)+	8.96E+02(1.82E+01)

Table 3 continued

Function	EPSDE Mean(Std)	CoBiDE Mean(Std)	SAKPDE Mean(Std)
$F20_{CEC2005}$	8.46E+02(3.53E+00)−	9.18E+02(2.73E+00)+	9.00E+02(0.00E+00)
$F21_{CEC2005}$	7.28E+02(2.28E+00)+	5.50E+02(1.53E+02)≈	5.00E+02(7.47E−13)
$F22_{CEC2005}$	5.00E+02(1.22E−01)−	8.86E+02(2.80E+01)≈	8.92E+02(1.34E+01)
$F23_{CEC2005}$	7.28E+02(3.62E+01)+	6.17E+02(1.79E+02)+	5.39E+02(1.82E−01)
$F24_{CEC2005}$	2.44E+02(1.75E+01)+	2.00E+02(1.56E−12)≈	2.00E+02(1.40E−12)
$F25_{CEC2005}$	3.00E+02(2.39E+02)+	2.16E+02(1.13E+00)≈	2.16E+02(1.54E+00)
$F1_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
$F2_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
$F3_{BBOB2012}$	3.78E−01(1.41E+00)−	1.98E−02(1.41E−01)−	2.19E+00(1.44E+00)
$F4_{BBOB2012}$	4.48E+00(4.47E+00)−	1.70E+00(1.84E+00)−	8.99E+00(2.81E+00)
$F5_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
$F6_{BBOB2012}$	0.00E+00(0.00E+00)≈	7.53E−07(3.30E−07)+	0.00E+00(0.00E+00)
$F7_{BBOB2012}$	4.16E+01(1.34E+01)+	1.07E+01(3.48E+00)−	1.42E+01(3.96E+00)
$F8_{BBOB2012}$	8.78E−01(1.67E+00)+	1.56E+01(7.81E+00)+	2.46E−01(9.56E−01)
$F9_{BBOB2012}$	3.55E+01(2.98E+00)+	3.50E+01(1.77E+00)+	7.77E+00(9.76E+00)
$F10_{BBOB2012}$	5.61E+04(9.74E+04)+	7.99E+02(3.33E+02)≈	8.12E+02(3.55E+02)
$F11_{BBOB2012}$	2.52E+01(1.05E+02)+	7.18E−04(8.65E−04)−	2.99E−03(2.95E−03)
$F12_{BBOB2012}$	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
$F13_{BBOB2012}$	6.02E+00(7.47E+00)+	1.25E+00(1.38E+00)≈	1.75E+00(1.65E+00)
$F14_{BBOB2012}$	2.77E−05(4.17E−06)−	1.16E−04(3.03E−05)+	9.13E−05(1.95E−05)
$F15_{BBOB2012}$	1.78E+02(2.68E+01)+	7.84E+01(2.30E+01)+	6.98E+01(2.35E+01)
$F16_{BBOB2012}$	2.54E+01(5.36E+00)+	1.66E+00(8.18E−01)−	3.36E+00(1.42E+00)
$F17_{BBOB2012}$	5.28E−01(3.93E−01)+	1.46E−02(1.38E−02)+	3.91E−03(3.79E−03)
$F18_{BBOB2012}$	2.50E+00(1.52E+00)+	4.77E−01(2.10E−01)+	1.70E−01(9.02E−02)
$F19_{BBOB2012}$	5.54E+00(2.88E−01)+	2.40E+00(2.00E+00)+	6.35E−02(8.28E−02)

Table 3 continued

Function	EPSDE Mean(Std)	CoBiDE Mean(Std)	SAKPDE Mean(Std)
$F20_{BBOB2012}$	7.04E−01(2.13E−01)+	5.52E−01(1.45E−01)≈	5.65E−01(1.43E−01)
$F21_{BBOB2012}$	2.26E+00(4.95E−01)−	2.48E+00(3.36E−01)≈	2.62E+00(3.73E−01)
$F22_{BBOB2012}$	4.73E+00(5.28E+00)−	5.49E+00(5.72E+00)−	1.46E+01(9.95E−15)
$F23_{BBOB2012}$	3.51E+00(3.85E−01)+	4.11E−01(7.06E−01)+	3.86E−01(1.87E−01)
$F24_{BBOB2012}$	3.01E+02(2.92E+01)+	1.04E+02(2.59E+01)≈	1.13E+02(1.58E+01)
+	31	21	
-	10	8	
≈	8	20	

Table 4 Mean values of different algorithms over 51 independent runs on 30 test functions

Function	UMOEAs Mean	L-SHADE Mean	CMLSP Mean	MVMO Mean	rmalschcma Mean	GaAPADE Mean	SAKPDE Mean
$F1_{CEC2014}$	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.76E+05	**0.00E+00**	**0.00E+00**	6.98E+03
$F2_{CEC2014}$	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
$F3_{CEC2014}$	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.62E+01	**0.00E+00**	**0.00E+00**
$F4_{CEC2014}$	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.04E+01	**0.00E+00**	**0.00E+00**	2.53E+00
$F5_{CEC2014}$	2.02E+01	2.01E+01	**2.00E+01**	2.05E+01	**2.00E+01**	**2.00E+01**	2.01E+01
$F6_{CEC2014}$	**0.00E+00**	1.38E−07	6.45E−04	1.29E+01	1.14E+00	6.13E−01	1.35E+00
$F7_{CEC2014}$	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.99E−03	1.93E−04	**0.00E+00**	**0.00E+00**
$F8_{CEC2014}$	1.95E+00	**0.00E+00**	9.40E+00	8.58E−01	1.95E−02	1.75E+00	**0.00E+00**
$F9_{CEC2014}$	8.45E+00	**6.78E+00**	8.56E+00	2.51E+01	1.79E+01	1.70E+01	4.25E+01
$F10_{CEC2014}$	1.36E+01	**1.63E−02**	2.23E+03	1.79E+01	8.12E+01	8.14E+00	3.35E+01
$F11_{CEC2014}$	1.58E+03	**1.23E+03**	3.16E+03	1.54E+03	1.55E+03	1.90E+03	2.21E+03
$F12_{CEC2014}$	2.15E−03	1.61E−01	**2.12E−03**	7.21E−02	1.60E−02	2.03E−01	1.16E−01
$F13_{CEC2014}$	5.68E−02	1.24E−01	**5.39E−02**	1.57E−01	1.38E−01	1.45E−01	1.87E−01
$F14_{CEC2014}$	2.11E−01	2.42E−01	3.32E−01	**1.99E−01**	2.22E−01	2.11E−01	2.19E−01
$F15_{CEC2014}$	3.09E+00	**2.15E+00**	3.32E+00	2.86E+00	2.45E+00	3.06E+00	3.18E+00
$F16_{CEC2014}$	1.05E+01	**8.50E+00**	1.20E+01	1.02E+01	9.65E+00	9.88E+00	1.05E+01
$F17_{CEC2014}$	1.01E+03	**1.88E+02**	8.02E+02	9.01E+02	6.98E+02	2.00E+02	2.17E+02
$F18_{CEC2014}$	2.32E+01	**5.91E+00**	2.48E+02	2.89E+01	5.67E+02	9.32E+00	1.32E+01
$F19_{CEC2014}$	3.81E+00	3.68E+00	5.56E+00	3.08E+00	5.82E+00	3.62E+00	**2.91E+00**
$F20_{CEC2014}$	1.17E+01	**3.08E+00**	5.73E+01	1.09E+02	1.99E+02	5.59E+00	7.03E+00
$F21_{CEC2014}$	3.18E+02	**8.68E+01**	5.95E+02	4.67E+02	5.73E+02	1.18E+02	1.22E+02
$F22_{CEC2014}$	9.59E+01	**2.76E+01**	1.59E+02	1.45E+02	1.59E+02	6.26E+01	3.31E+01
$F23_{CEC2014}$	3.14E+02	3.15E+02	3.14E+02	3.15E+02	3.15E+02	3.15E+02	**2.00E+02**

Table 4 continued

Function	UMOEAs Mean	L-SHADE Mean	CMLSP Mean	MVMO Mean	rmalschcma Mean	GaAPADE Mean	SAKPDE Mean
$F24_{CEC2014}$	2.24E+02	2.24E+02	2.28E+02	2.25E+02	2.22E+02	2.24E+02	**2.00E+02**
$F25_{CEC2014}$	**2.00E+02**	2.03E+02	2.00E+02	2.03E+02	2.06E+02	2.03E+02	**2.00E+02**
$F26_{CEC2014}$	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	1.02E+02	**1.00E+02**	**1.00E+02**
$F27_{CEC2014}$	3.01E+02	3.00E+02	3.65E+02	4.01E+02	3.28E+02	3.28E+02	**2.00E+02**
$F28_{CEC2014}$	3.68E+02	8.40E+02	3.70E+02	8.77E+02	8.23E+02	8.34E+02	**2.00E+02**
$F29_{CEC2014}$	2.06E+02	7.17E+02	2.05E+02	7.36E+02	1.15E+03	5.00E+05	**2.00E+02**
$F30_{CEC2014}$	4.30E+02	1.25E+03	3.47E+02	**2.00E+03**	2.04E+03	1.65E+03	**2.00E+02**

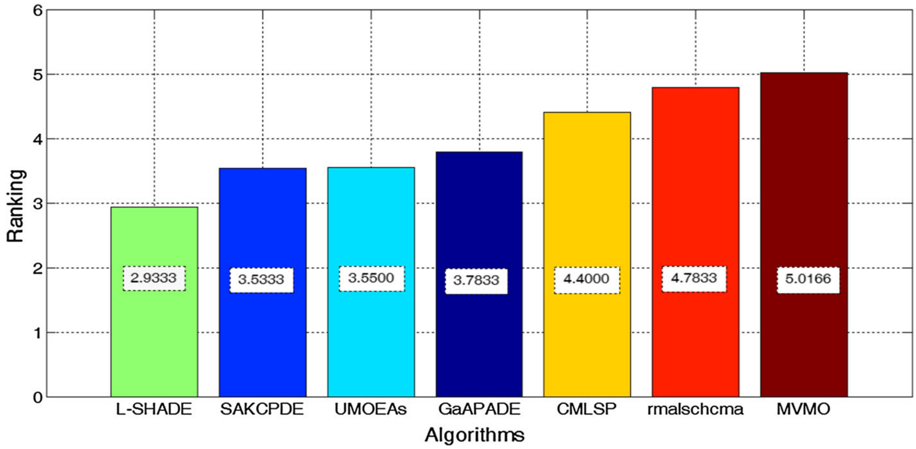

Fig. 4 Rankings of different evolutionary algorithms

Table 5 p-values obtained by Bonferroni-Dunn's, Holm's, and Hochberg's procedures for the compared DE algorithms on $F17_{CEC2014}$–$F30_{CEC2014}$

SAKPDE**v.s.**	z	Unadjusted p	Bonferroni-Dunn p	Holm p	Hochberg p
UMOEAs	2.74	6.14E-03	1.22E-02	1.22E-02	1.22E-02
L-SHADE	1.51	0.13	0.26	0.13	0.13

Table 6 Ranking obtained by Friedman's test on $F17_{CEC2014}$–$F30_{CEC2014}$

Algorithms	Ranking
UMOEAs	2.50
L-SHADE	2.03
SAKPDE	1.46

and Friedman's test are shown in Tables 5 and 6. We can observe from Table 5 that the average performance of SAKPDE is significantly better than that of UMOEAs (ranked the third among seven compared algorithms) and is better than that of L-SHADE on these complex test functions. Table 6 also shows that the search capability of SAKPDE is better than that of L-SHADE and UMOEAs on functions $F17_{CEC2014}$–$F30_{CEC2014}$. Overall, SAKPDE can exhibit highly competitive search performance among all compared algorithms, especially when the optimization problems are complex.

5.4 Sensitive analysis of parameters

In SAKPDE, four key parameters need be set. Because the performance of SAKPDE is not significantly influenced by G_s, the effects of other three key parameters are only analyzed in the present work. These three parameters are opposition learning ratio ρ, $sigma$, and forgetting factor φ. The times of independent runs and maximum number of function evaluations are set to be 30 and 300,000, respectively. Moreover, two non-parametric statistical tests (i.e., Friedman and Iman-Davenport tests) are utilized to investigate the sensitivity of these three parameters.

Table 7 Results of the Friedman and Iman-Davenport test with different ρ

Friedman value	χ^2 value	p-value	Iman-Davenport value	F_F value	p-value
81.274	18.307	2.823E–13	9.363	1.848	2.120E–14

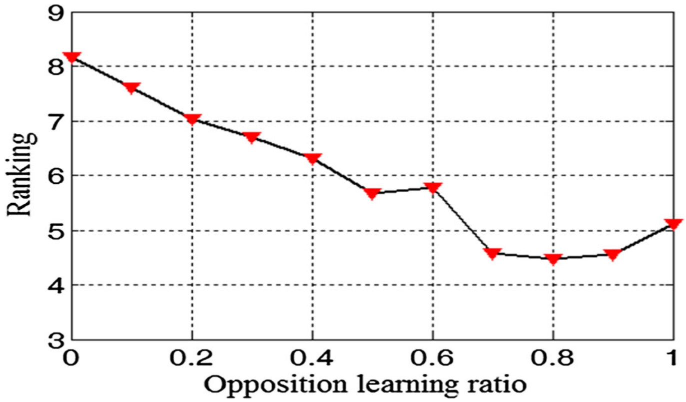

Fig. 5 Rankings of different opposition learning ratio

5.4.1 Impact of opposition learning ratio

A large value of opposition learning ratio encourages SAKPDE to use prior knowledge to generate the control parameters, and vice versa. Therefore, to investigate the effect of ρ, two sets of 30-dimensional test functions introduced in IEEE CEC2014 and CEC2005 are utilized in the present work. We test SAKPDE with different values of ρ, which varies from 0 to 1 with a step equal to 0.1. The statistical test results obtained by the Friedman and Iman-Davenport test are presented in Table 7. We can conclude that the value of ρ can significantly affect the performance of SAKPDE in a statistically way. However, the performance of SAKPDE is not significantly influenced by the value of ρ in the particular range. Overall, it is not difficult to choose a value of ρ in the proposed algorithm. Additionally, the rankings achieved by the Friedman's test are illustrated in Fig. 5, in which we can see $\rho = 0.8$ can provide the best performance. Therefore, $\rho = 0.8$ is set in the proposed algorithm.

5.4.2 Impact of sigma

In this section, the effect of *sigma* is analyzed and tested with different values varying from 0.1 to 0.8 with a step equal to 0.1. Moreover, two sets of benchmark test functions (IEEE CEC2005 and CEC2014) are used to investigate the sensitivity of *sigma*. The statistical test results are shown in Table 8. We can see from Table 8 that the performance of the proposed

Table 8 Results of the Friedman and Iman-Davenport test with different *sigma* (0.1–0.8)

Friedman value	χ^2value	p-value	Iman-Davenport value	$F_F$value	p-value
18.427	14.067	0.01018	2.7145	2.0338	0.009317

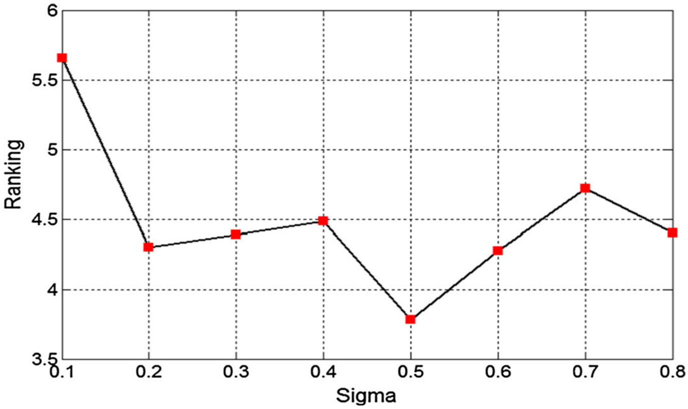

Fig. 6 Rankings of different values of sigma

Table 9 Results of the Friedman and Iman-Davenport test with different *sigma* (0.2-0.8)

Friedman value	χ^2 value	p-value	Iman-Davenport value	F_F value	p-value
5.509	12.592	0.4803	0.9167	2.1266	0.4829

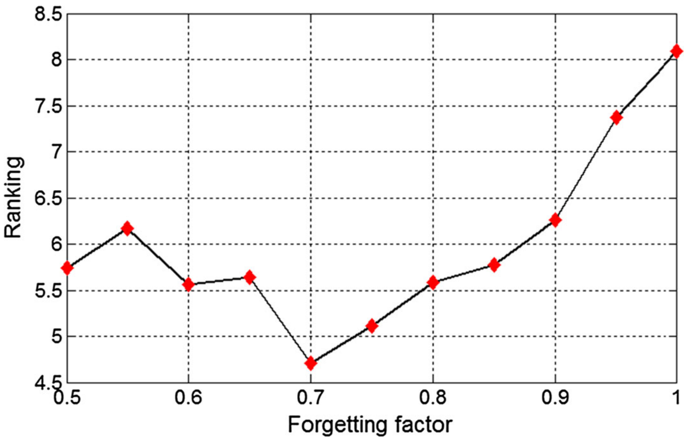

Fig. 7 Rankings of different values of forgetting factor

algorithm is significantly sensitive to the value of *sigma*. The rankings of different *sigma* are illustrated in Fig. 6. We can observe from Fig. 6 that the overall performance of SAKPDE is the best when *sigma* = 0.5. However, no single *sigma* value can perform well on all test functions and the constant parameter setting may not be suitable for dynamic evolutionary process. Additionally, we can also see the statistical test results shown in Table 9 that the *sigma* value cannot significantly affect the performance of SAKPDE when it is within the range of [0.2, 0.8]. Based on the above considerations, $sigma = 0.8 - 0.6 \times (1 - (G/G_{\max})^2)$ is chosen in the proposed algorithm.

Table 10 Results of the Friedman and Iman-Davenport test with different values of forgetting factor

Friedman value	χ^2value	p-value	Iman-Davenport value	$F_F$value	p-value
47.157	18.307	8.840E–7	5.064	1.848	4.510E–7

Table 11 Results of three SAKPDE variants

Function	SAKPDE-bin Mean(Std)	SAKPDE-exp Mean(Std)	SAKPDE-eig Mean(Std)
$F2_{CEC2005}$	4.73E−08(2.30E−07)	**4.71E−22(5.46E−22)**	2.03E+03(1.05E+03)
$F12_{CEC2005}$	**2.52E+03(3.76E+03)**	3.69E+03(5.31E+03)	1.28E+04(8.44E+03)
$F14_{CEC2005}$	1.25E+01(3.21E−01)	1.24E+01(4.06E−01)	**1.21E+01(5.90E−01)**

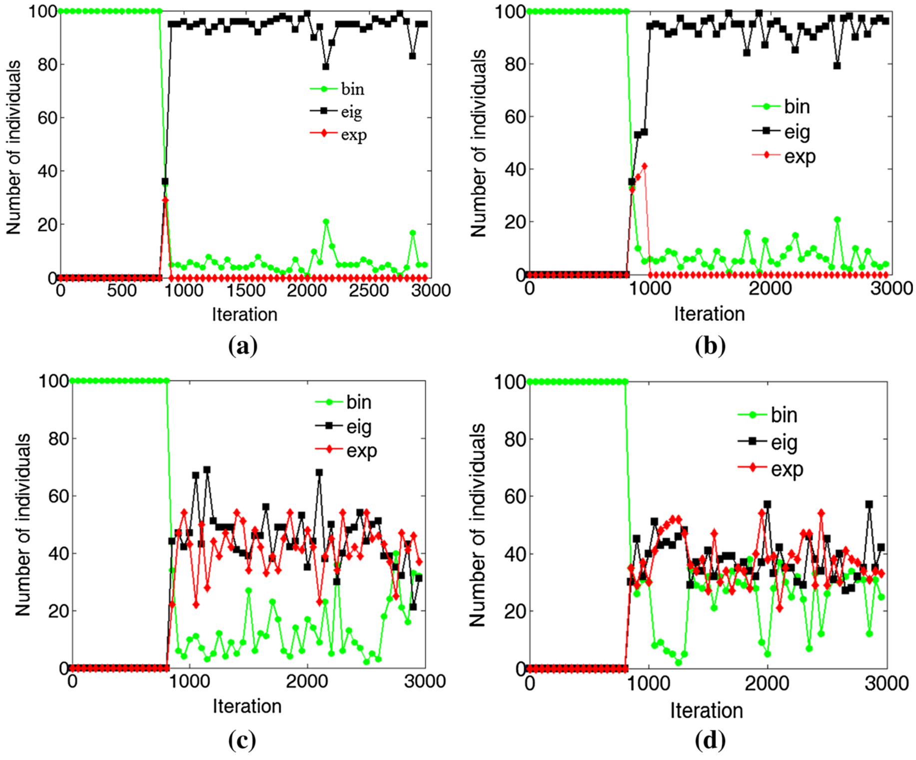

Fig. 8 Evolution curves of crossover strategies of SAKPDE for $F2_{CEC2005}$. **a** $\varphi = 1$. **b** $\varphi = 0.9$. **c** $\varphi = 0.8$. **d** $\varphi = 0.7$

5.4.3 Impact of forgetting factor

In general, a large value of φ may encourage learning, thus it may lead to greedy selection. However, if the value of φ is too small, the best strategy will be neglected in solving optimization problems. Therefore, it is important to select an appropriate value of forgetting factor in the proposed algorithm. In this experiment, two benchmark sets (IEEE CEC2005 and

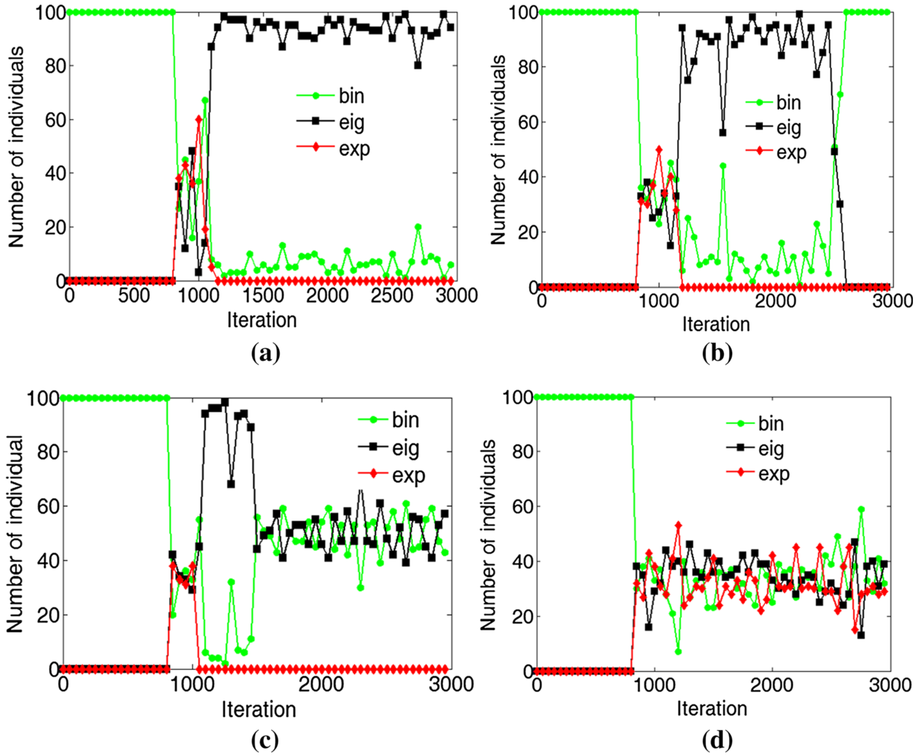

Fig. 9 Evolution curves of crossover strategies of SAKPDE for $F12_{CEC2005}$. **a** $\varphi = 1$. **b** $\varphi = 0.9$. **c** $\varphi = 0.8$. **d** $\varphi = 0.7$

CEC2014) are utilized to investigate the influences of different forgetting factor values on the performance of SAKPDE. φ varies from 0.5 to 1 with a step equal to 0.05. The statistical test results achieved by the Friedman and Iman-Davenport tests are presented in Table 10. From Table 10, we can find that the performance of SAKPDE is significantly sensitive to the value of forgetting factor. Additionally, for different φ values, the rankings achieved by the Friedman test are shown in Fig. 7, in which we can observe that $\varphi = 0.7$ can provide the best performance. Therefore, $\varphi = 0.7$ is set in SAKPDE.

5.5 Effectiveness of learning and forgetting mechanism

To demonstrate the effectiveness of learning and forgetting mechanism, three IEEE CEC2005 test functions are selected and the adaptation of crossover strategy is analyzed in this experiment. The maximum number of function evaluations is set to be 300,000 and the results are achieved by 30 independent runs for each SAKPDE variant in this experiment. These SAKPDE variants are denoted as SAKPDE-bin (using only binomial crossover strategy), SAKPDE-exp (using only exponential crossover strategy), and SAKPDE-eig (using only eigenvector-based crossover strategy). The results are shown in Table 11. The best results are highlighted in bold. From Table 11, we can find that SAKPDE-exp performs better than the other two SAKPDE variants on function $F2_{CEC2005}$, SAKPDE-bin outperforms SAKPDE-exp and SAKPDE-eig on function $F12_{CEC2005}$, and SAKPDE-eig performs the best on function $F14_{CEC2005}$. For function $F2_{CEC2005}$, we can see from Fig. 8a, b that the

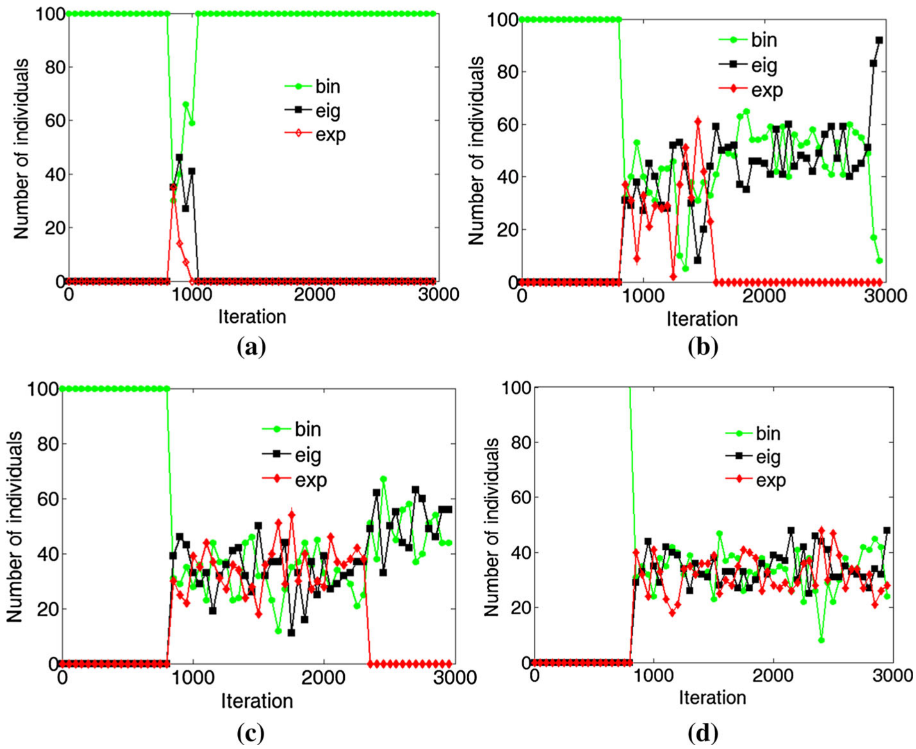

Fig. 10 Evolution curves of crossover strategies of SAKPDE for $F14_{CEC2005}$. **a** $\varphi = 1$. **b** $\varphi = 0.9$. **c** $\varphi = 0.8$. **d** $\varphi = 0.7$

eigenvector-based crossover strategy dominates the population evolution. The main reason may be that the eigenvector-based crossover strategy performs better than other crossover strategies on $F2_{CEC2005}$ in a few generations. Therefore, if $\varphi = 1$ or $\varphi = 0.9$, SAKPDE cannot find a real effective crossover strategy to solve optimization problem. If $\varphi = 0.8$, it can avoid greedy selection and can select a real effective crossover strategy. If $\varphi = 0.7$, it can not only avoid greedy selection and choose a real effective crossover strategy, but also select other suboptimal crossover strategy to participate in the population evolution. It may be more effective in improving the search performance of DE when actual optimization environment is very complex. For function $F12_{CEC2005}$, Fig. 9a indicates that the eigenvector-based crossover strategy performs better than other crossover strategies. However, we can see from Fig. 9b, c that the forgetting factor can avoid greedy selection and find out a real effective crossover strategy that may perform worse than other crossover strategies in a few generations. Additionally, Fig. 9d shows that a small value of forgetting factor ($\varphi = 0.7$) can select more crossover strategies to participate in the population evolution during the entire evolution process. From Fig. 10, we can also obtain the similar conclusion.

Based on the above observations and analysis, we can conclude that the pure learning may lead to greedy selection and the forgetting mechanism is a useful method to adapt dynamic and complex environment.

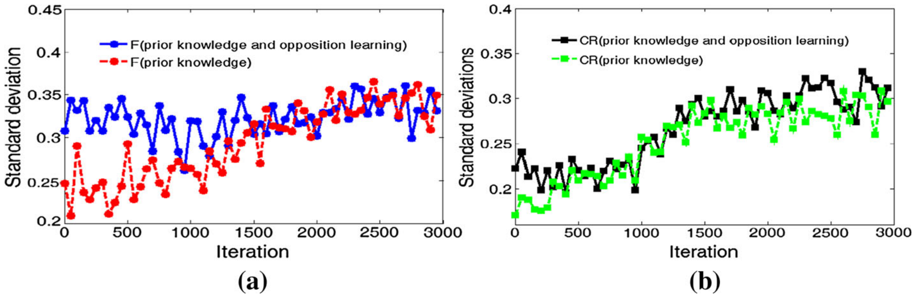

Fig. 11 Evolution curves of standard deviation of control parameters for $F12_{CEC2005}$. **a** Standard deviation of *F* generated by two methods. **b** Standard deviation of *CR* generated by two methods

5.6 Effectiveness of prior knowledge and opposition learning

In this section, the test function $F12_{CEC2005}$ is utilized to analyze the control parameters. The experiment is divided into two cases: (1) *F* and *CR* are generated by prior knowledge and opposition learning; (2) *F* and *CR* are only produced by prior knowledge. The maximum number of function evaluations is set to be 300,000. For these two cases, the standard deviation of the control parameters at each generation is illustrated in Fig. 11. We can observe from Fig. 11a that the standard deviation of *F* generated by prior knowledge and opposition learning is larger than that of *F* generated by prior knowledge. Fig. 11b also indicates that the standard deviation of *CR* produced by prior knowledge and opposition learning is slightly larger than that of *CR* produced by prior knowledge. Based on the above observations, we can conclude that prior knowledge incorporated with opposition learning can provide different search capabilities for DE during the evolutionary process. Therefore, it is an effective method to guide and supervise the production of the control parameters.

6 Conclusions

In this study, a differential evolution algorithm with strategy adaptation and knowledge-based control parameters (SAKPDE) is proposed to enhance the performance of DE. In SAKPDE, the learning and forgetting mechanism is utilized to implement the adaptation of mutation and crossover strategies, and prior knowledge and opposition learning are used to guide the evolution of the control parameters as well. We compare SAKPDE with eight DE variants and four non-DEs on three test suites. The statistical analysis results show that the overall performance of SAKPDE is very competitive among all compared algorithms, especially when the optimization problem is complex. Moreover, SAKPDE can automatically select suitable strategies in solving particular optimization problem during the entire evolution process, and prior knowledge and opposition learning can provide matching and different search capabilities for DE in different evolution stages.

The sensitivity of three key parameters is analyzed by statistical tests. The results show that the performance of SAKPDE is significantly influenced by these three parameter settings. However, it is not very difficult to set three values of additional parameters in the particular range. Moreover, the effectiveness of learning–forgetting mechanism is investigated in Sect. 5.5. The result indicates that a suitable value of forgetting factor can avoid the greedy selection effectively. Finally, the effectiveness of prior knowledge and opposition learning is studied in Sect. 5.6.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. 61603244).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflicts of interest in publishing this paper.

References

Abbass HA (2002) The self-adaptive pareto differential evolution algorithm. In: Proceedings of the 2002 congress on evolutionary computation, pp 831–836

Biswas S, Kundu S, Das S (2015) Inducing niching behavior in differential evolution through local information sharing. IEEE Trans Evolut Comput 19:246–263

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

Chen L, Zheng Z, Liu H-L, Xie S (2014) An evolutionary algorithm based on covariance matrix leaning and searching preference for solving CEC 2014 benchmark problems. In: Proceedings of the 2014 congress on evolutionary computation, pp 2672–2677

Cui L, Li G, Lin Q, Chen J, Lu N (2016) Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations. Comput Oper Res 67:155–173

Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evolut Comput 15:4–31

Das S, Konar A, Chakraborty UK (2005) Two improved differential evolution schemes for faster global search. In: Proceedings of the 2005 conference on genetic and evolutionary computation, 2005. ACM, pp 991–998

Dunn OJ (1961) Multiple comparisons among means. J Am Stat Assoc 56:52–64

Elsayed SM, Sarker RA, Essam DL, Hamza NM (2014) Testing united multi-operator evolutionary algorithms on the CEC2014 real-parameter numerical optimization. In: Proceedings of the 2014 congress on evolutionary computation, pp 1650–1657

Erlich I, Rueda JL, Wildenhues S, Shewarega F (2014) Evaluating the mean-variance mapping optimization on the IEEE-CEC 2014 test suite. In: Proceedings of the 2014 congress on evolutionary computation, pp 1625–1632

Fan Q, Yan X (2015) Differential evolution algorithm with self-adaptive strategy and control parameters for P-xylene oxidation process optimization. Soft Comput 19:1363–1391

Fan Q, Yan X (2016) Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies. IEEE Trans Cybern 46:219–232

Fan Q, Yan X, Xue Y (2016) Prior knowledge guided differential evolution. Soft Comput :1–18

Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32:675–701

Gämperle R, Müller SD, Koumoutsakos P (2002) A parameter study for differential evolution. Adv Intell Syst Fuzzy Syst Evolut Comput 10:293–298

Gao W, Yen GG, Liu S (2014) A cluster-based differential evolution with self-adaptive strategy for multimodal optimization. IEEE Trans Cybern 44:1314–1327

García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. J Heuristics 15:617–644

Ghosh A, Das S, Chowdhury A, Giri R (2011) An improved differential evolution algorithm with fitness-based adaptation of the control parameters. Inf Sci 181:3749–3765

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

Gong W, Cai Z, Wang Y (2014) Repairing the crossover rate in adaptive differential evolution. Appl Soft Comput 15:149–168

Gong W, Zhou A, Cai Z (2015) A multioperator search strategy based on cheap surrogate models for evolutionary optimization. IEEE Trans Evolut Comput 19:746–758

Guo S-M, Yang C-C (2015) Enhancing differential evolution utilizing eigenvector-based crossover operator. IEEE Trans Evolut Comput 19:31–49

Guo S-M, Yang C-C, Hsu P-H, Tsai JS-H (2015) Improving differential evolution with a successful-parent-selecting framework. IEEE Trans Evolut Comput 19:717–730

Hansen N, Auger A, Finck S, Ros R (2012) Real-parameter black-box optimization benchmarking 2012: experimental setup. In: Proceedings of the 14th annual conference on genetic and evolutionary computation

Hu C, Yan X (2009) An immune self-adaptive differential evolution algorithm with application to estimate kinetic parameters for homogeneous mercury oxidation. Chinese J Chem Eng 17:232–240

Islam SM, Das S, Ghosh S, Roy S, Suganthan PN (2012) An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. IEEE T Syst Man Cybern Part B 42:482–500

Kovačević D, Mladenović N, Petrović B, Milošević P (2014) DE-VNS: self-adaptive differential evolution with crossover neighborhood search for continuous global optimization. Comput Oper Res 52:157–169

Li Y-L, Zhan Z-H, Gong Y-J, Chen W-N, Zhang J, Li Y (2015) Differential evolution with an evolution path: a DEEP evolutionary algorithm. IEEE Trans Cybern 45:1798–1810

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

Liu J, Lampinen J (2005) A fuzzy adaptive differential evolution algorithm. Soft Comput 9:448–462

Mallipeddi R, Suganthan PN, Pan Q-K, Tasgetiren MF (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. Appl Soft Comput 11:1679–1696

Mallipeddi R, Wu G, Lee M, Suganthan P (2014) Gaussian adaptation based parameter adaptation for differential evolution. In: IEEE congress on evolutionary computation, pp 1760–1767

Mokhtari H, Salmasnia A (2015) A Monte Carlo simulation based chaotic differential evolution algorithm for scheduling a stochastic parallel processor system. Expert Syst Appl 42:7132–7147

Molina D, Lacroix B, Herrera F (2014) Influence of regions on the memetic algorithm for the CEC'2014 special session on real-parameter single objective optimisation. In: IEEE congress on evolutionary computation, pp 1633–1640

Montgomery J, Chen S (2010) An analysis of the operation of differential evolution at high and low crossover rates. In: IEEE congress on evolutionary computation, pp 1–8

Neri F, Tirronen V (2010) Recent advances in differential evolution: a survey and experimental analysis. Artif Intell Rev 33:61–106

Pan Q-K, Suganthan PN, Wang L, Gao L, Mallipeddi R (2011) A differential evolution algorithm with self-adapting strategy and control parameters. Comput Oper Res 38:394–408

Price K, Storn RM, Lampinen JA (2006) Differential evolution: a practical approach to global optimization. Springer, Berlin

Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evolut Comput 13:398–417

Ronkkonen J, Kukkonen S, Price KV (2005) Real-parameter optimization with differential evolution. In: IEEE congress on evolutionary computation, pp 506–513

Salman A, Engelbrecht AP, Omran MG (2007) Empirical analysis of self-adaptive differential evolution. Eur J Oper Res 183:785–804

Sarker R, Elsayed SM, Ray T (2014) Differential evolution with dynamic parameters selection for optimization problems. IEEE Trans Evolut Comput 18:689–707

Shao W, Pi D (2016) A self-guided differential evolution with neighborhood search for permutation flow shop scheduling. Expert Sys Appl 51:161–176

Storn R, Price K (1995) Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces, vol 3. ICSI, Berkeley

Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11:341–359

Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization KanGAL report 2005005

Tanabe R, Fukunaga AS (2014) Improving the search performance of SHADE using linear population size reduction. In: IEEE congress on evolutionary computation, pp 1658–1665

Teo J (2006) Exploring dynamic self-adaptive populations in differential evolution. Soft Comput 10:673–686

Tirronen V, Neri F (2009) Differential evolution with fitness diversity self-adaptation. In: Chiong R (ed) Nature-inspired algorithms for optimisation. Springer, pp 199–234

Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans Evolut Comput 15:55–66

Wang Y, Li H-X, Huang T, Li L (2014) Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. Appl Soft Comput 18:232–247

Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics 1:80–83

Yang M, Li C, Cai Z, Guan J (2015) Differential evolution with auto-enhanced population diversity. IEEE Trans Cybern 45:302–315

Yu W, Shen M, Chen WN, Zhan ZH, Gong YJ, Lin Y, Liu O, Zhang J (2014) Differential evolution with two-level parameter adaptation. IEEE Trans Cybern 44:1080–1099

Zaharie D (2002) Parameter adaptation in differential evolution by controlling the population diversity. In: Proceedings of the international workshop on symbolic and numeric algorithms for scientific computing. pp 385–397

Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evolut Comput 13:945–958

Zhong J-H, Shen M, Zhang J, Chung HS-H, Shi Y-H, Li Y (2013) A differential evolution algorithm with dual populations for solving periodic railway timetable scheduling problem. IEEE Trans Evolut Comput 17:512–527

Zhou X-G, Zhang G-J, Hao X-H, Yu L (2016) A novel differential evolution algorithm using local abstract convex underestimate strategy for global optimization. Comput Oper Res 75:132–149

Zhu W, Tang Y, Fang J-A, Zhang W (2013) Adaptive population tuning scheme for differential evolution. Inf Sci 223:164–191