Applied Soft Computing 48 (2016) 329–346

Contents lists available at ScienceDirect

Applied Soft Computing

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

Utilizing cumulative population distribution information in differential evolution

累积种群分布信息

Yong Wang [a,b,*], Zhi-Zhong Liu [a], Jianbin Li [c], Han-Xiong Li [d,e], Gary G. Yen [f]

[a] *School of Information Science and Engineering, Central South University, Changsha 410083, PR China*
[b] *School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK*
[c] *Institute of Information Security and Big Data, Central South University, Changsha 410083, China*
[d] *Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong*
[e] *State Key Laboratory of High Performance Complex Manufacturing, Central South University, Changsha 410083, PR China*
[f] *School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA*

ARTICLE INFO

Article history:
Received 1 October 2015
Received in revised form 30 May 2016
Accepted 5 July 2016
Available online 15 July 2016

Keywords:
Cumulative population distribution information
Differential evolution
Eigen coordinate system
Evolutionary algorithms

ABSTRACT

Differential evolution (DE) is one of the most popular paradigms of evolutionary algorithms. In general, DE does not exploit distribution information provided by the population and, as a result, its search performance is limited. In this paper, cumulative population distribution information of DE has been utilized to establish an Eigen coordinate system by making use of covariance matrix adaptation. The crossover operator of DE implemented in the Eigen coordinate system has the capability to identify the features of the fitness landscape. Furthermore, we propose a cumulative *p*opulation distribution *i*nformation based *DE* framework called CPI-DE. In CPI-DE, for each target vector, two trial vectors are generated based on both the original coordinate system and the Eigen coordinate system. Then, the target vector is compared with these two trial vectors and the best one will survive into the next generation. CPI-DE has been applied to two classic versions of DE and three state-of-the-art variants of DE for solving two sets of benchmark test functions, namely, 28 test functions with 30 and 50 dimensions at the 2013 IEEE Congress on Evolutionary Computation, and 30 test functions with 30 and 50 dimensions at the 2014 IEEE Congress on Evolutionary Computation. The experimental results suggest that CPI-DE is an effective framework to enhance the performance of DE.

1. Introduction

Differential Evolution (DE), proposed by Storn and Price [1,2] in 1995, is a very popular evolutionary algorithm (EA) paradigm. During the past two decades, DE has attracted a lot of attention and has been successfully applied to solve a variety of numerical and real-world optimization problems [3–5].

The remarkable advantages of DE are its simple structure and ease of implementation. In DE, each individual in the population is called a target vector. DE contains three basic operators: mutation, crossover and selection. During the evolution, DE generates a trial vector for each target vector through the mutant and crossover operators. Afterward, the trial vector competes with its target vector for survival according to their fitness. DE also involves three control parameters: the population size, the scaling factor, and the crossover control parameter. The performance of DE is dependent mainly on these three operators and three control parameters. In order to further improve the performance of DE, a lot of DE variants have been designed, such as JADE [6], jDE [7], SaDE [8], EPSDE [9], CoDE [10], and so on.

DE is a population-based optimization algorithm; however, population distribution information has not yet been widely utilized in the DE community, which makes DE inefficient especially when solving some optimization problems with complex characteristics. Very recently, two attempts have been made along this line [11,12]. However, the methods proposed in Refs. [11,12] only utilize the distribution information from a single population of one generation, and the cumulative distribution information of the population over the course of evolution has been ignored. Moreover, these methods introduce some extra parameters. Therefore, new insights into the usage of the population distribution information in DE are quite necessary.

* Corresponding author at: School of Information Science and Engineering, Central South University, Changsha 410083, PR China.

E-mail addresses: ywang@csu.edu.cn (Y. Wang), zzliucsu@126.com (Z.-Z. Liu), lijianbin@csu.edu.cn (J. Li), mehxli@cityu.edu.hk (H.-X. Li), gyen@okstate.edu (G.G. Yen).

http://dx.doi.org/10.1016/j.asoc.2016.07.012
1568-4946/

In 2001, Hansen and Ostermeier [13] proposed the well-known covariance matrix adaptation evolution strategy, called CMA-ES. CMA-ES generates offspring by sampling a multivariate normal distribution, which includes three main elements: mean vector of the search distribution, covariance matrix, and step-size. Indeed, covariance matrix reflects the population distribution information to a certain degree [12]. In CMA-ES, the covariance matrix is self-adaptively updated according to the information from the previous and current generations.

In this paper, we make use of the cumulative distribution information of the population to establish an Eigen coordinate system in DE, by considering CMA as an effective tool. Furthermore, we suggest a *c*umulative *p*opulation distribution *i*nformation based *DE* framework called CPI-DE. In CPI-DE, for each target vector, the crossover operator of DE is implemented in both the original coordinate system and the Eigen coordinate system and, as a result, two trial vectors are generated. Subsequently, the target vector is compared with these two trial vectors and the best one will enter the next population. CPI-DE is applied to two classic DE versions as well as three state-of-the-art DE variants. Extensive experiments across two benchmark test sets from the 2013 IEEE Congress on Evolutionary Computation (IEEE CEC2013) [14] and the 2014 IEEE Congress on Evolutionary Computation (IEEE CEC2014) [15] have been implemented to verify the effectiveness of CPI-DE.

The main contributions of this paper can be summarized as follows:

- Due to the fact that single population fails to contain enough information to estimate the covariance matrix reliably, this paper updates the covariance matrix in DE by an adaptation procedure, which makes use of the cumulative distribution information of the population.
- CPI-DE provides a simple yet efficient synergy of two kinds of crossover: the crossover in the Eigen coordinate system and the crossover in the original coordinate system. The former aims at identifying the properties of the fitness landscape and improving the efficiency and effectiveness of DE by producing the offspring toward the promising directions. In addition, the purpose of the latter is to maintain the superiority of the original DE. Moreover, no extra parameters are required in CPI-DE.
- Our experimental studies have shown that CPI-DE is capable of enhancing the performance of several classic DE versions and advanced DE variants.

The rest of this paper is organized as follows. Section 2 describes the basic procedure of DE. Section 3 briefly reviews the recent developments of DE in the last five years. The proposed CPI-DE is presented in Section 4. The experimental results and the performance comparison are given in Section 5. Finally, Section 6 concludes this paper.

2. Differential evolution (DE)

Similar to other EA paradigms, DE starts with a population of NP individuals, i.e., $\boldsymbol{P}^{(g)} = \{\vec{x}_i^{(g)} = (x_{i,1}^{(g)}, \ldots, x_{i,D}^{(g)}),\ i = 1, \ldots, NP\}$, where g is the generation number, D is the dimension of the decision space, and NP is the population size. In $\boldsymbol{P}^{(g)}$, each individual is also called a target vector. At $g = 0$, the jth decision variable of the ith target vector is initialized as follows:

$$x_{i,j}^{(0)} = L_j + rand(0,1) * (U_j - L_j), i = 1, \ldots, NP, j = 1, \ldots, D \quad (1)$$

where $rand(0,1)$ represents a uniformly distributed random number between 0 and 1, and L_j and U_j are the lower and upper bounds of the jth decision variable, respectively.

After the initialization, DE repeatedly implements three basic operators, i.e., mutation, crossover, and selection, to search for the optimal solution of an optimization problem. Note that in DE, a combination of the mutation operator and the crossover operator is called a trial vector generation strategy.

2.1. Mutation operator

At each generation, a mutant vector is generated for each target vector by the mutation operator. The following are four commonly used mutation operators in the DE community:

- DE/rand/1

$$\vec{v}_i^{(g)} = \vec{x}_{r1}^{(g)} + F * (\vec{x}_{r2}^{(g)} - \vec{x}_{r3}^{(g)}) \quad (2)$$

- DE/rand/2

应用于两个经典DE和三个变体

$$\vec{v}_i^{(g)} = \vec{x}_{r1}^{(g)} + F * (\vec{x}_{r2}^{(g)} - \vec{x}_{r3}^{(g)}) + F * (\vec{x}_{r4}^{(g)} - \vec{x}_{r5}^{(g)}) \quad (3)$$

- DE/current-to-best/1

$$\vec{v}_i^{(g)} = \vec{x}_i^{(g)} + F * (\vec{x}_{best}^{(g)} - \vec{x}_i^{(g)}) + F * (\vec{x}_{r1}^{(g)} - \vec{x}_{r2}^{(g)}) \quad (4)$$

- DE/current-to-rand/1

$$\vec{v}_i^{(g)} = \vec{x}_i^{(g)} + F * (\vec{x}_{r1}^{(g)} - \vec{x}_i^{(g)}) + F * (\vec{x}_{r2}^{(g)} - \vec{x}_{r3}^{(g)}) \quad (5)$$

In the above equations, the indices r_1, r_2, r_3, r_4, and r_5 are distinct integers randomly selected from $[1, \ldots, NP]$ and are also different from i, $\vec{x}_{best}^{(g)}$ is the best target vector in the current population, F is the scaling factor, and $\vec{v}_i^{(g)}$ is the mutant vector.

2.2. Crossover operator

After mutation, the crossover operation is applied to each pair of $\vec{x}_i^{(g)}$ and $\vec{v}_i^{(g)}$ to generate a trial vector $\vec{u}_i^{(g)} = (u_{i,1}^{(g)}, \ldots, u_{i,D}^{(g)})$. The binomial crossover can be expressed as follows:

$$u_{i,j}^{(g)} = \begin{cases} v_{i,j}^{(g)}, & \text{if } rand(0,1) \le CR \text{ or } j = j_{rand} \\ x_{i,j}^{(g)}, & \text{otherwise} \end{cases}, j = 1, \ldots, D \quad (6)$$

where j_{rand} is a random integer between 1 and D, $rand(0,1)$ is a uniformly distributed random number between 0 and 1, and CR is the crossover control parameter. The condition $j = j_{rand}$ makes the trial vector different from the corresponding target vector by at least one dimension.

2.3. Selection operator

The selection operator of DE adopts a one-to-one competition between the target vector and its trial vector. For a minimization problem, if the objective function value of the trial vector is less than or equal to that of the target vector, then the trial vector will survive into the next generation; otherwise, the target vector will enter the next generation:

$$\vec{x}_i^{(g+1)} = \begin{cases} \vec{u}_i^{(g)}, & \text{if } f(\vec{u}_i^{(g)}) \le f(\vec{x}_i^{(g)}) \\ \vec{x}_i^{(g)}, & \text{otherwise} \end{cases} \quad (7)$$

where $f(\bullet)$ is the objective function.

It is evident that NP, F, and CR are three main control parameters of DE. The setting of NP is related to the dimension of the decision space. In general, the higher the dimension of the decision space, the larger the value of NP. In addition, F is always chosen from the range [0.4, 1.0] and *CR* is usually set to a value close to 0.1 or 1.0 depending on the characteristics of an optimization problem [10].

3. The related work

Recent two decades have witnessed significant progress in the developments of DE. In 2011, Das and Suganthan [16] presented a comprehensive survey on DE, including the basic concepts and major variants of DE, as well as the applications and theoretical studies of DE. Next, we will briefly introduce the recent developments of DE in the last five years.

3.1. *Introduction of new trial vector generation strategies*

Zhou et al. [17] proposed an intersect mutation operator, in which the individuals in the population are divided into two parts according to their fitness: the worse part and the better part. Zhang and Yuen [18] presented a directional mutation operator, in which a differential vector pool is established once the best-so-far fitness of the population has been improved at one generation. Subsequently, this differential vector pool is utilized to create the trial vectors in the next generation. Hu et al. [19] introduced a subspace clustering mutation operator which selects an elite individual as the base vector and employs the difference between two randomly generated boundary individuals as the perturbation vector. Gong and Cai [20] proposed the ranking-based mutation operators. In this kind of mutation operators, the individuals are selected based on their rankings, which means the individuals with better ranking have more opportunity to be selected. Cai and Wang [21] incorporated the neighborhood and direction information into the mutation operator. They also proposed two strategies. In the first strategy the neighborhood information is used to select the base and differential vectors, and in the second strategy the direction information is incorporated into the mutation operator. Wang et al. [22] proposed a multiobjective sorting-based mutation operator. In this operator, the fitness and diversity information are simultaneously considered as two objective functions in DE, with the aim of selecting those individuals with both high fitness and better diversity for mutation. Guo et al. [23] proposed a successful-parent-selecting framework to select individuals for mutation and crossover. In this framework, successful solutions are stored into an archive and some individuals in the archive are chosen to implement mutation and crossover when stagnation is happening. Yu et al. [24] designed an adaptive greedy mutation strategy, in which one of the vectors in mutation is randomly selected from the top k individuals in the current population. Moreover, in order to adjust the greediness degree, the parameter k is set by an adaptive scheme. Wang et al. [25] proposed Gaussian bare-bones DE, the core technique of which is a Gaussian mutation operator. Zhao and Suganthan [26] empirically investigated the performance of the exponential crossover operator of DE, and suggested a linearly scalable exponential crossover operator. In classic DE, the trial vector generated by the crossover operator is usually a vertex of the hyper-rectangle defined by the mutant and target vectors. In order to alleviate this drawback, Wang et al. [27] exploited orthogonal crossover to probe the hyper-rectangle defined by the mutant and target vectors, thus enhancing the search ability of DE.

3.2. *Adapting the control parameter settings*

Gong et al. [28] analyzed the behavior of the crossover operator and proposed a crossover rate repair technique for adaptive DE variants. Tanabe and Fukunaga [29] suggested a success-history based parameter adaptation scheme to revise the settings of both the scaling factor *F* and the crossover control parameter *CR* in JADE [6]. Under the framework in Ref. [29], Tanabe and Fukunaga [30] further incorporated linear population size reduction. By using the correlation coefficient, Takahama and Sakai [31] improved the settings of *F* and *CR* in JADE [6]. In Ref. [32], *F* and *CR* are determined by a mechanism based on exponentially weighting moving average. Sarker et al. [33] defined three sets for *F*, *CR*, and the population size *NP*, respectively. During the evolution, dynamic selection is executed for these three control parameters of DE. He and Yang [34] controlled *F* by taking advantage of Lévy distribution. Yu and Zhang [35] introduced an adaptive parameter control scheme based on optimization state estimation. Zhu et al. [36] proposed an adaptive population tuning scheme to dynamically adjust *NP* according to a status monitor. Zamuda et al. [37] proposed a population reduction DE with multiple mutation operators, in which *NP* is reduced with the increase of the generation number. In adaptive or self-adaptive DE variants, it is always expected that the crossover control parameter *CR* which induces a larger amount of replacements can generate solutions of higher quality. Segura et al. [38] carried out an empirical investigation on this issue by analyzing the correlation between the quality of the obtained solutions and the probability of replacement induced by different *CR* values. In Ref. [39], Segura et al. studied on the effectiveness of incorporating feedback information from the search process to guide the adaptation of *F*, and pointed out that further research is required to successfully adapt *F*.

单种群多个参数生成多个试验个体

3.3. *Integrating multiple trial vector generation strategies with multiply control parameter settings in a single population*

Tang et al. [40] made use of the fitness information to tune the control parameters and choose the mutation operators. Moreover, they designed an individual-dependent parameter setting and an individual-dependent mutation operator which are associated with each individual in the population. Yi et al. [41] proposed a novel DE, called HSDE. HSDE combines two mutation operators to balance the exploration and exploitation abilities of DE and integrates them with a self-adaptive parameter control strategy introduced in Ref. [7]. Fan and Yan [42] presented a self-adaptive DE in which each individual has its own *F*, *CR*, and mutation operator. Moreover, five mutation operators have been adopted and *F* and *CR* are automatically adjusted. Very recently, Fan and Yan [43] proposed zoning evolution of control parameters, in which suitable combinations of *F* and *CR* can be generated by zoning evolution. Moreover, adaptive mutation operator is also employed. In Ref. [44], two mutation operators are combined by a linear increment rule. In addition, *F* is generated according to two Gaussian distributions and *CR* is produced by two uniform distributions based on the success ratio. Takahama and Sakai [45] proposed a novel method to detect the modality of landscape being searched, i.e., unimodality or multimodality. Afterward, a mutation operator is selected according to the modality of landscape. Moreover, *F* and *CR* are tuned dynamically in Ref. [45]. Zhou et al. [46] dynamically divided the population into three groups by considering the position and fitness information of each individual. Moreover, these three groups are assigned with different roles and equipped with different mutation operators and control parameter values.

3.4. *Multi-populated DE*

多种群

Bujok and Tvrdik [47] developed a parallel DE, which contains a parallel migration model employing various adaptive DE variants. Huo et al. [48] proposed a multi-swarm DE with swarm sharing management. In this method, each swarm explores the search space independently and the sharing management is applied to adjust swarm size. Kushida et al. [49] designed an island-based DE. This method allocates different control parameters to each island, performs migration among islands, and dynamically varies subpopulation size by individual transfer. Zhou et al. [50] proposed a two-layer hierarchical DE, in which the population in the top layer consists of the best individuals obtained from the several populations in the bottom layer. In Ref. [51], an island based distributed DE framework has been proposed. Cheng et al. [52] presented a distributed DE with multicultural migration, which makes use of two migration selection approaches to maintain the diversity in the subpopulations and an affinity based replacement strategy to control the diversity among the individuals. Peng and Wu [53] presented a heterozygous DE. In this method, the population is firstly divided into four sub-swarms, and then each sub-swarm corresponds to a parameter adjustment scheme.

3.5. *Combining DE with other techniques*

During the past five years, combining DE with other techniques has attracted considerable attention. For example, DE has been combined with opposition-based learning [54,55], restart technique [56], adaptive disturbance mechanism [57], and Taguchi local search [53]. Recently, Yang et al. [58] proposed an auto-enhanced population diversity mechanism, which firstly identifies whether the population is converging or stagnant, and then rediversifies the population at the dimensional level. Li et al. [59] presented a novel idea, i.e., the cumulatively learned evolution path. In this method, after a trial vector has been created, an additional differential vector is added to this trial vector based on the evolution path information. In addition, DE has also been combined with surrogate models to deal with computationally expensive global numerical optimization problems in Refs. [60–62].

3.6. *Hybridized DE*

与其他算法结合

At present, DE has been hybridized with a lot of meta-heuristic methods, such as artificial bee colony algorithm [63], variable neighborhood search [64], simulated annealing [65], estimation of distribution algorithm [66], genetic programming [67], and Cuckoo search [68]. Moreover, several DE variants have been developed under the memetic framework [69,70].

Our work in this paper falls in the first category, i.e., introducing new crossover operator by utilizing cumulative population distribution information in DE.

4. Proposed approach

4.1. *Motivation*

种群分布，（2）协方差矩阵来估计特征坐标系，（3）

Based on the above introduction, it is clear that population distribution information has seldom been involved in the current state-of-the-art DE.

Very recently, Guo and Yang [11] and Wang et al. [12] made the first attempt to exploit the population distribution information in DE. The methods proposed in Refs. [11,12] share some similar ideas. More specifically, these two methods firstly compute the covariance matrix of the population. Subsequently, the Eigenvectors obtained from the Eigen decomposition are used to establish an Eigen coordinate system. Finally, the crossover operator of DE is implemented in the Eigen coordinate system to generate the trial vectors. Compared with the crossover operator in the original coordinate system, the crossover operator in the Eigen coordinate system makes the recombination process of DE rotationally invariant [11]. However, these two methods only make use of the distribution information of the current population and, consequently, the estimation of the covariance matrix is unreliable due to insufficient information. Besides, some problem-dependent parameters have been introduced, such as the number of individuals adopted to compute the covariance matrix and the frequency of the crossover operator being executed in the Eigen coordinate system.

The above discussion motivates us to carry out an in-depth investigation on the utilization of population distribution information in DE. Actually, the methods in Refs. [11,12], and this paper are all inspired by CMA-ES [13]. However, unlike Refs. [11,12], this paper adapts the covariance matrix according to the information of the previous and current generations to increase the probability of producing successful search distribution for the subsequent evolution, which results in a more reasonable search behavior. Besides, this paper proposes a DE framework called CPI-DE, which eliminates the problem-dependent parameters in Refs. [11,12].

In CMA-ES, the individuals in the population are generated by the following equation [13]:

$$\vec{x}_i^{(g+1)} = \vec{m}^{(g)} + \sigma^{(g)} N(\mathbf{0}, \mathbf{C}^{(g)}), \quad i = 1,$$

根据前代和当前代的信息调整协方差矩阵，增加后续进化产生成功搜索分布的概率，使搜索行为更加合理。此外，本文还提出了一种CPI-DE框架来消除参考文献中与问题相关的参数。

where $\vec{m}^{(g)}$ is the mean vector of the search distribution, $\sigma^{(g)}$ is the step-size, $\mathbf{C}^{(g)}$ is the covariance matrix, and $N(\mathbf{0}, \mathbf{C}^{(g)})$ is a multivariate normal distribution with zero mean and covariance matrix $\mathbf{C}^{(g)}$. The main aim of CMA-ES is to calculate $\vec{m}^{(g+1)}$, $\sigma^{(g+1)}$, and $\mathbf{C}^{(g+1)}$ for the next generation $(g+1)$. Moreover, the step-size and the covariance matrix in CMA-ES are self-adaptively updated as the search goes on. It is noteworthy that ES and DE have different search patterns. In ES, the offspring is produced according to a predefined probabilistic distribution. However, in DE the offspring is generated by the arithmetic operation of the base and differential vectors and by the information exchange between the target vector and the mutant vector. Therefore, it is unnecessary to update the step-size for DE and we only focus on the adaptation of the covariance matrix (i.e., CMA).

与ES不同，DE不需要步长信息

4.2. *Rank-NP-update of the covariance matrix in DE*

In CMA-ES, two updating strategies have been introduced to adapt the covariance matrix: rank-μ-update and rand-one-update. This paper only employs rank-μ-update based on the following considerations: (1) rank-μ-update plays a primary role when the population is large. Compared with ES, DE usually adopts a relatively larger population size; and (2) rank-one-update exploits correlations between consecutive generations and constructs an evolution path to update the covariance matrix, which inevitably adds computational complexity. By eliminating the rank-one-update, CMA in DE becomes simpler. In rank-μ-update, μ represents the population size. Since the population size is equal to NP in DE, rank-μ-update in CMA-ES is called rank-NP-update in this paper.

The covariance matrix $\mathbf{C}^{(g)}$ is initialized as $\mathbf{C}^{(0)} = \boldsymbol{I}$, where $\boldsymbol{I} \in \Re^{D \times D}$ is a unity matrix. In addition, the mean vector of the search distribution $\vec{m}^{(g)}$ is initialized as a randomly generated point in the search space. At generation $(g+1)$, $\vec{m}^{(g+1)}$ is updated as follows:

$$\vec{m}^{(g+1)} = \sum_{i=1}^{NP} w_i \vec{x}_{i:2*NP}^{(g+1)} \tag{9}$$

where $\vec{x}_{i:2*NP}^{(g+1)}$ is the ith best individual in the offspring population (note that in CPI-DE, the offspring population consists of $2*NP$ individuals), i.e., $f(\vec{x}_{1:2*NP}^{(g+1)}) \le f(\vec{x}_{2:2*NP}^{(g+1)}) \le \ldots \le f(\vec{x}_{NP:2*NP}^{(g+1)})$, w_i is the ith positive weight coefficient, and $\sum_{i=1}^{NP} w_i = 1$. It is evident that $\vec{m}^{(g+1)}$ is the weighted average of the NP best individuals in the offspring population. In order to introduce a search bias toward the promising area, the value of the weight coefficient depends on the quality of the individual, i.e., $w_1 \geq w_2 \ldots \geq w_{NP} > 0$. According to the suggestion in Ref. [13], w_i is set as follows:

$$w_i = \frac{w'_i}{\sum_{j=1}^{NP} w'_j},\ i \in 1, \ldots, NP \tag{10}$$

and

$$w'_i = ln(NP + 0.5) - ln(i),\ i \in 1, \ldots, NP \tag{11}$$

In order to update the covariance matrix, firstly an estimator of $\boldsymbol{C}^{(g)}$ is computed:

$$\boldsymbol{C}_{NP}^{(g+1)} = \sum_{i=1}^{NP} w_i \left(\vec{x}_{i:2*NP}^{(g+1)} - \vec{m}^{(g)}\right)\left(\vec{x}_{i:2*NP}^{(g+1)} - \vec{m}^{(g)}\right)^T \tag{12}$$

Afterward, $\boldsymbol{C}^{(g+1)}$ at generation $(g+1)$ is updated as follows:

$$\boldsymbol{C}^{(g+1)} = (1 - c_{NP})\boldsymbol{C}^{(g)} + c_{NP}\left(\sigma^{(g)2}\right)^{-1}\boldsymbol{C}_{NP}^{(g+1)}$$

当前种群的信息和以往种群的信息都用来更新

where $c_{NP} \approx \min(1, NP_{eff}/D^2)$ is the learning rate and $NP_{eff} = \left(\sum_{i=1}^{NP} w_i^2\right)^{-1}$ is the variance effective selection mass.

It is clear from (13) that the information from both the previous and current generations are used to update the covariance matrix, which means that the cumulative distribution information of the population has been utilized to adapt the search distribution. It is necessary to note that (13) includes the step-size $\sigma^{(g)}$ at generation g. As pointed out previously, it does not make sense to adjust the step-size for DE. Therefore, for the sake of simplicity, $\sigma^{(g)}$ is set to 1 in this paper. Indeed, $\sigma^{(g)} = 1$ resembles the covariance matrix from the estimation of multivariate normal algorithm [71]. Moreover, $\sigma^{(g)} = 1$ implies that the covariance matrix at each generation is of equal importance [72].

4.3. Crossover in the Eigen coordinate system

The main idea of the crossover in the Eigen coordinate system is the following [12]. Firstly, by implementing the Eigen decomposition on the covariance matrix, an orthonormal basis of Eigenvectors can be obtained, which forms an Eigen coordinate system. Then the target vector and its mutant vector are transformed into the Eigen coordinate system. Afterward, the crossover of DE is executed on the transformed target and mutant vectors, and thus, a trial vector is produced in the Eigen coordinate system. Finally, this trial vector is transformed back into the original coordinate system. Next, we will give the details of the above procedure.

The Eigen decomposition of the covariance matrix $\boldsymbol{C}^{(g)}$ at generation g can be described as:

$$\boldsymbol{C}^{(g)} = \boldsymbol{B}^{(g)}\boldsymbol{D}^{(g)2}\boldsymbol{B}^{(g)\mathrm{T}} \tag{14}$$

where each column of the orthogonal matrix $\boldsymbol{B}^{(g)}$ is the corresponding Eigenvector of $\boldsymbol{C}^{(g)}$, and each diagonal element of the diagonal matrix $\boldsymbol{D}^{(g)}$ is the corresponding Eigenvalue of $\boldsymbol{C}^{(g)}$. By doing this, $\boldsymbol{B}^{(g)}$ includes an orthonormal basis of Eigenvectors and forms an Eigen coordinate system.

Note that $\boldsymbol{B}^{(g)\mathrm{T}}$ has the capability to rotate a vector into the Eigen coordinate system. After the rotation, the elements of the resulting vector are related to the projections onto the Eigenvectors [72]. Based on the above property, the target vector $\vec{x}_i^{(g)}$ and its mutant vector $\vec{v}_i^{(g)}$ are transformed into the Eigen coordinate system:

$$\vec{x}_i'^{(g)} = \boldsymbol{B}^{(g)\mathrm{T}}\vec{x}_i^{(g)} \tag{15}$$

$$\vec{v}_i'^{(g)} = \boldsymbol{B}^{(g)\mathrm{T}}\vec{v}_i^{(g)} \tag{16}$$

$g = 0$; // g is the generation number
Initialize $\vec{m}^{(0)}$ and $\boldsymbol{C}^{(0)}$;
Generate an initial population $\boldsymbol{P}^{(0)} = \{\vec{x}_1^{(0)}, \ldots, \vec{x}_{NP}^{(0)}\}$ by uniformly and randomly sampling NP individuals in the decision space;
Evaluate the objective function values of the initial population $f(\vec{x}_1^{(0)}), \ldots, f(\vec{x}_{NP}^{(0)})$;
$FEs = NP$; // FEs records the number of fitness evaluations
While $FEs < MaxFEs$ **do** // $MaxFEs$ represents the maximum number of fitness evaluations
    $\boldsymbol{O}^{(g)} = \emptyset$, $\boldsymbol{E}^{(g)} = \emptyset$, and $\boldsymbol{P}^{(g+1)} = \emptyset$;
    For $i = 1 : NP$ **do**
        Implement the mutation and crossover operators of DE in the original coordinate system to generate a trial vector $\vec{u}_{i\_1}^{(g)}$ for the target vector $\vec{x}_i^{(g)}$, and $\boldsymbol{O}^{(g)} = \boldsymbol{O}^{(g)} \cup \vec{u}_{i\_1}^{(g)}$;
        Implement the mutation and crossover operators of DE in the Eigen coordinate system to generate another trial vector $\vec{u}_{i\_2}^{(g)}$ for the target vector $\vec{x}_i^{(g)}$ according to (14)-(18), and $\boldsymbol{E}^{(g)} = \boldsymbol{E}^{(g)} \cup \vec{u}_{i\_2}^{(g)}$;
        Evaluate the objective function values of $\vec{u}_{i\_1}^{(g)}$ and $\vec{u}_{i\_2}^{(g)}$;
        Implement the selection operator of DE to select the best one from $\vec{x}_i^{(g)}$, $\vec{u}_{i\_1}^{(g)}$, and $\vec{u}_{i\_2}^{(g)}$, denoted as $\vec{x}_i^{(g+1)}$;
        $\boldsymbol{P}^{(g+1)} = \boldsymbol{P}^{(g+1)} \cup \vec{x}_i^{(g+1)}$;
    End For
    $FEs = FEs + 2 * NP$;
    Select the best NP individuals from $\boldsymbol{O}^{(g)} \cup \boldsymbol{E}^{(g)}$, denoted as $\vec{x}_{1:2*NP}^{(g+1)}, \ldots, \vec{x}_{NP:2*NP}^{(g+1)}$, and use these NP individuals and $\vec{m}^{(g)}$ to compute $\boldsymbol{C}^{(g+1)}$ according to (12) and (13). Subsequently, use these NP individuals to compute $\vec{m}^{(g+1)}$ according to (9);
    $g = g + 1$;
End While

Fig. 1. Pseudocode of CPI-DE.

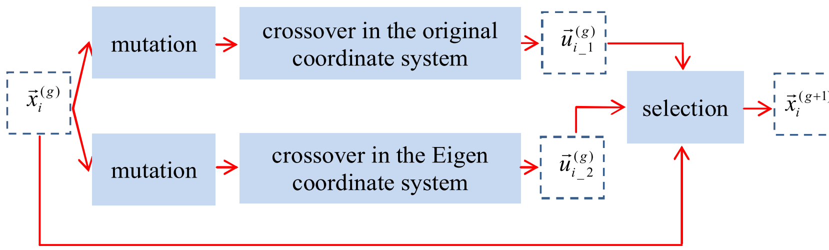

Fig. 2. The mutation, crossover, and selection of CPI-DE.

Afterward, the crossover operator is implemented in the Eigen coordinate system and a trial vector $\vec{u}_i'^{(g)} = \left(u_{i,1}'^{(g)}, \ldots, u_{i,D}'^{(g)}\right)$ is produced:

$$u_{i,j}'^{(g)} = \begin{cases} v_{i,j}'^{(g)}, & \text{if } rand(0,1) \le CR \text{ or } j = j_{rand} \\ x_{i,j}'^{(g)}, & \text{otherwise} \end{cases}, j = 1, \ldots, D \quad (17)$$

Since $\boldsymbol{B}^{(g)}$ is able to rotate the result back into the original coordinate system [72], the final trial vector $\vec{u}_i^{(g)}$ in the original coordinate system can be obtained by the following transformation:

$$\vec{u}_i^{(g)} = \boldsymbol{B}^{(g)} \vec{u}_i'^{(g)} \quad (18)$$

4.4. CPI-DE

The Pseudocode of CPI-DE is shown in Fig. 1. As introduced previously, in CPI-DE the cumulative distribution information of the population is used to update the covariance matrix by rank-*NP*-update, and then the Eigen coordinate system is established by the Eigen decomposition of the covariance matrix. At each generation, for each target vector, two trial vectors are generated by implementing the mutation and crossover operators in both the original coordinate system and the Eigen coordinate system. Thereafter, the best one among the target vector and two corresponding trial vectors will enter the next generation. The above mutation, crossover, and selection are shown in Fig. 2. Meanwhile, the covariance matrix and the mean vector of the search distribution are updated at the end of each generation accordingly.

In this paper, the advantages of implementing the crossover operator in the Eigen coordinate system are twofold:

- At the early stage of evolution, the population maintains high diversity. Under this condition, by utilizing the cumulative distribution information of the population, the covariance matrix has the capabilities to quickly provide reasonable search distribution and to continuously guide the population toward the promising areas.
- At the middle and later stages of evolution, the diversity of population may gradually decrease and the search may concentrate on a relatively small area. In this case, by utilizing the covariance

Table 1
Experimental results of DE/rand/1/bin, CPI-DE/rand/1/bin, DE/current-to-best/1/bin, and CPI-DE/current-to-best/1/bin over 51 independent runs on 28 test functions with 30D from IEEE CEC2013 using 300,000 FEs.

Test Functions with 30D from IEEE CEC2013		DE/rand/1/bin (Mean Error ± Std Dev)	CPI-DE/rand/1/bin (Mean Error ± Std Dev)	DE/current-to-best/1/bin (Mean Error ± Std Dev)	CPI-DE/current-to-best/1/bin (Mean Error ± Std Dev)
Unimodal Functions	$CEC2013_1$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
	$CEC2013_2$	1.28E+08 ± 2.54E+07–	3.70E−03 ± 1.21E−02	**3.50E+07 ± 1.78E+07**–	**0.00E+00 ± 0.00E+00**
	$CEC2013_3$	1.49E+09 ± 5.11E+08–	8.69E+04 ± 6.87E+04	1.49E+06 ± 4.08E+06–	1.28E+02 ± 5.91E+02
	$CEC2013_4$	**3.24E+04 ± 5.65E+03**–	**0.00E+00 ± 0.00E+00**	**1.60E+04 ± 3.50E+03**–	**0.00E+00 ± 0.00E+00**
	$CEC2013_5$	**1.98E−06 ± 8.94E−07**–	**0.00E+00 ± 0.00E+00**	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
Basic Multimodal Functions	$CEC2013_6$	2.62E+01 ± 1.28E+01–	1.56E+00 ± 6.27E+00	1.63E+01 ± 4.69E+00–	3.10E+00 ± 8.59E+00
	$CEC2013_7$	6.67E+01 ± 7.09E+00–	1.07E+01 ± 4.90E+00	5.93E+00 ± 4.61E+00–	3.74E+00 ± 4.34E+00
	$CEC2013_8$	2.09E+01 ± 4.77E−02≈	2.09E+01 ± 4.31E−02	2.09E+01 ± 5.56E−02≈	2.09E+01 ± 4.17E−02
	$CEC2013_9$	3.95E+01 ± 1.13E+00–	3.89E+01 ± 1.25E+00	3.71E+01 ± 3.04E+00–	2.06E+01 ± 1.36E+01
	$CEC2013_{10}$	2.11E+02 ± 5.34E+01–	3.86E−04 ± 1.93E−03	1.59E−02 ± 1.00E−02–	1.54E−03 ± 3.84E−03
	$CEC2013_{11}$	9.08E+01 ± 9.45E+00+	1.18E+02 ± 8.88E+00	7.33E+00 ± 2.97E+00–	6.75E+00 ± 2.82E+00
	$CEC2013_{12}$	2.32E+02 ± 1.22E+01–	1.91E+02 ± 1.05E+01	1.96E+02 ± 1.16E+01–	1.78E+02 ± 9.83E+00
	$CEC2013_{13}$	2.29E+02 ± 1.49E+01–	1.90E+02 ± 1.27E+01	1.93E+02 ± 1.31E+01–	1.80E+02 ± 1.19E+01
	$CEC2013_{14}$	4.26E+03 ± 1.77E+02+	4.67E+03 ± 2.18E+02	3.36E+03 ± 4.99E+02+	4.15E+03 ± 3.89E+02
	$CEC2013_{15}$	7.25E+03 ± 2.97E+02≈	7.29E+03 ± 2.17E+02	7.21E+03 ± 2.62E+02≈	7.19E+03 ± 2.92E+02
	$CEC2013_{16}$	2.49E+00 ± 2.91E−01≈	2.50E+00 ± 2.52E−01	2.46E+00 ± 2.97E−01≈	2.47E+00 ± 2.94E−01
	$CEC2013_{17}$	1.72E+02 ± 1.04E+01≈	1.78E+02 ± 1.16E+01	1.35E+02 ± 8.73E+00≈	1.39E+02 ± 1.30E+01
	$CEC2013_{18}$	2.69E+02 ± 1.20E+01–	2.23E+02 ± 1.21E+01	2.23E+02 ± 1.15E+01–	2.13E+02 ± 9.46E+00
	$CEC2013_{19}$	1.69E+01 ± 1.22E+00–	1.59E+01 ± 1.03E+00	1.13E+01 ± 1.21E+00+	1.25E+01 ± 1.17E+00
	$CEC2013_{20}$	1.28E+01 ± 2.13E−01≈	1.25E+01 ± 2.67E−01	1.25E+01 ± 3.11E−01≈	1.22E+01 ± 3.09E−01
Composition Functions	$CEC2013_{21}$	2.63E+02 ± 5.01E+01+	3.09E+02 ± 8.98E+01	2.70E+02 ± 7.02E+01+	3.01E+02 ± 8.67E+01
	$CEC2013_{22}$	4.71E+03 ± 3.91E+02+	5.09E+03 ± 3.53E+02	2.76E+03 ± 1.06E+03+	3.87E+03 ± 6.29E+02
	$CEC2013_{23}$	7.60E+03 ± 2.53E+02–	7.29E+03 ± 2.76E+02	7.31E+03 ± 2.89E+02–	7.15E+03 ± 3.42E+02
	$CEC2013_{24}$	2.84E+02 ± 6.56E+00–	2.10E+02 ± 3.67E+00	2.17E+02 ± 2.01E+01≈	2.16E+02 ± 1.29E+01
	$CEC2013_{25}$	3.09E+02 ± 4.13E+00–	2.87E+02 ± 2.51E+01	2.46E+02 ± 8.78E+00–	2.39E+02 ± 3.99E+00
	$CEC2013_{26}$	2.09E+02 ± 1.94E+00–	2.03E+02 ± 2.59E+01	2.51E+02 ± 7.12E+01–	2.41E+02 ± 5.63E+01
	$CEC2013_{27}$	1.28E+03 ± 2.89E+01–	9.41E+02 ± 2.91E+02	7.82E+02 ± 2.96E+02–	4.48E+02 ± 1.02E+02
	$CEC2013_{28}$	3.00E+02 ± 8.82E−05≈	3.00E+02 ± 1.49E−06	3.20E+02 ± 1.44E+02≈	3.19E+02 ± 1.36E+02
+		4		4	
–		17		15	
≈		7		9	

matrix, the modality of the fitness landscape can be identified and the crossover in the Eigen coordinate system is able to strengthen the exploitation ability of DE in the area surrounded by the population.

On the other hand, in order to keep the superiority and the search behavior of the original DE, the crossover operator is also executed in the original coordinate system.

Based on the above discussion, CPI-DE not only achieves a tradeoff between the exploration and exploitation in DE, but also provides a simple yet efficient framework to incorporate the cumulative distribution information of the population into DE. Moreover, CPI-DE does not introduce its own parameter.

Remark 1: In rank-μ-update of CMA-ES, μ parents in the population are used to generate λ offspring by a multivariate normal distribution. Subsequently, the best μ individuals are selected from these λ offspring and used to update the covariance matrix. In rank-*NP*-update of CPI-DE, *NP* individuals in the population are used to create $2*NP$ offspring by implementing the mutation and crossover operators in both the original coordinate system and the Eigen coordinate system. Afterward, the best *NP* individuals are chosen from these $2*NP$ offspring and used to update the covariance matrix. Hence, rank-*NP*-update in CPI-DE is a natural extension of rank-μ-update in CMA-ES. The relationship between rank-*NP*-update in CPI-DE and rank-μ-update in CMA-ES is shown in Fig. 3.

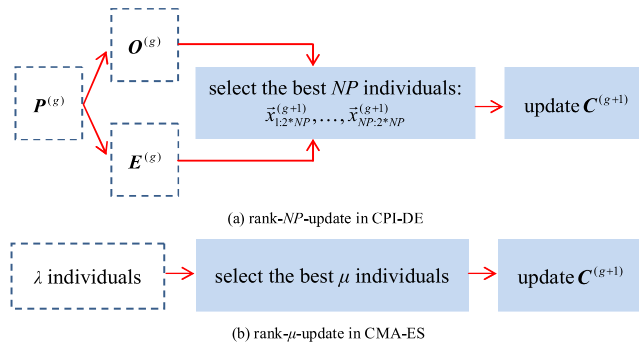

Fig. 3. The relationship between rank-*NP*-update in CPI-DE and rank-μ-update in CMA-ES.

5. Experimental study

In this paper, two sets of benchmark test functions are employed to demonstrate the effectiveness of CPI-DE, i.e., 28 test functions with 30 dimensions (30D) and 50 dimensions (50D) at IEEE CEC2013 [14], and 30 test functions with 30 dimensions (30D) and 50 dimensions (50D) at IEEE CEC2014 [15]. The 28 test functions in the first set are denoted as $CEC2013_1$-$CEC2013_{28}$, and the 30 test functions in the second set are denoted as $CEC2014_1$-$CEC2014_{30}$.

In our experiments, the function error value $(f(\vec{x}^{best})-f(\vec{x}^{*}))$ of each run is recorded, where $\vec{x}^{*}$ is the optimal solution and $\vec{x}^{best}$ is the best solution found at the end of a run. The average and standard deviation of the function error values in all runs (denoted as "Mean Error" and "Std Dev") are considered as two performance metrics to assess the performance of the algorithms. Moreover, Wilcoxon's rank sum test at a 0.05 significance level is used to test the statistical significance between pairwise algorithms. According to the suggestions in Refs. [14,15], the maximum number of fitness evalu-

Table 2
Experimental results of DE/rand/1/bin, CPI-DE/rand/1/bin, DE/current-to-best/1/bin, and CPI-DE/current-to-best/1/bin over 51 independent runs on 28 test functions with 50D from IEEE CEC2013 using 500,000 FEs.

Test Functions with 50D from IEEE CEC2013		DE/rand/1/bin (Mean Error ± Std Dev)	CPI-DE/rand/1/bin (Mean Error ± Std Dev)	DE/current-to-best/1/bin (Mean Error ± Std Dev)	CPI-DE/current-to-best/1/bin (Mean Error ± Std Dev)
Unimodal Functions	$CEC2013_1$	8.68E+01 ± 1.71E+01–	7.85E−01 ± 1.60E−01	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
	$CEC2013_2$	5.54E+08 ± 7.93E+07–	3.11E+05 ± 1.48E+05	2.75E+08 ± 6.31E+07–	3.14E+02 ± 1.29E+03
	$CEC2013_3$	5.19E+10 ± 4.67E+09–	8.64E+09 ± 2.01E+09	3.51E+08 ± 6.14E+08–	4.27E+06 ± 8.78E+06
	$CEC2013_4$	8.01E+04 ± 7.21E+03–	5.13E+01 ± 1.66E+01	**4.82E+04 ± 6.36E+03**–	**0.00E+00 ± 0.00E+00**
	$CEC2013_5$	2.94E+01 ± 3.08E+00–	1.21E+00 ± 2.12E−01	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
Basic Multimodal Functions	$CEC2013_6$	1.19E+02 ± 2.15E+01–	4.40E+01 ± 2.85E−01	4.40E+01 ± 7.63E−01–	4.34E+01 ± 4.43E−12
	$CEC2013_7$	1.55E+02 ± 8.54E+00–	9.28E+01 ± 9.54E+00	6.73E+01 ± 1.85E+01–	2.07E+01 ± 8.69E+00
	$CEC2013_8$	2.11E+01 ± 3.50E−02≈	2.11E+01 ± 3.01E−02	2.11E+01 ± 3.39E−02≈	2.11E+01 ± 3.41E−02
	$CEC2013_9$	7.26E+01 ± 1.64E+00≈	7.29E+01 ± 9.43E−01	7.24E+01 ± 1.32E+00–	7.16E+01 ± 1.88E+00
	$CEC2013_{10}$	2.57E+03 ± 2.50E+02–	1.16E+00 ± 3.09E−02	1.28E+00 ± 1.92E−01–	9.18E−04 ± 2.87E−03
	$CEC2013_{11}$	3.47E+02 ± 1.79E+01+	3.74E+02 ± 1.81E+01	1.20E+01 ± 2.66E+00+	9.34E+01 ± 2.15E+01
	$CEC2013_{12}$	5.43E+02 ± 1.92E+01–	3.97E+02 ± 1.48E+01	4.06E+02 ± 1.65E+01–	3.72E+02 ± 1.61E+01
	$CEC2013_{13}$	5.50E+02 ± 2.25E+01–	3.96E+02 ± 1.62E+01	4.08E+02 ± 1.51E+01–	3.74E+02 ± 1.45E+01
	$CEC2013_{14}$	1.00E+04 ± 2.29E+02≈	1.04E+04 ± 3.70E+02	9.76E+03 ± 3.39E+02+	1.02E+04 ± 3.41E+02
	$CEC2013_{15}$	1.40E+04 ± 3.79E+02≈	1.40E+04 ± 3.79E+02	1.40E+04 ± 3.59E+02≈	1.39E+04 ± 4.03E+02
	$CEC2013_{16}$	3.29E+00 ± 3.12E−01≈	3.33E+00 ± 3.64E−01	3.39E+00 ± 2.67E−01≈	3.37E+00 ± 2.66E−01
	$CEC2013_{17}$	5.15E+02 ± 1.86E+01–	4.84E+02 ± 1.85E+01	3.28E+02 ± 1.45E+01≈	3.32E+02 ± 1.42E+01
	$CEC2013_{18}$	6.57E+02 ± 1.98E+01–	4.73E+02 ± 1.55E+01	4.56E+02 ± 1.19E+01–	4.25E+02 ± 1.60E+01
	$CEC2013_{19}$	1.22E+02 ± 2.35E+01–	3.65E+01 ± 1.66E+00	2.74E+01 ± 1.48E+00+	2.95E+01 ± 1.12E+00
	$CEC2013_{20}$	2.28E+01 ± 2.09E−01≈	2.26E+01 ± 2.17E−01	2.25E+01 ± 2.56E−01≈	2.23E+01 ± 2.22E−01
Composition Functions	$CEC2013_{21}$	1.09E+03 ± 5.45E+02–	2.80E+02 ± 2.29E+02	7.08E+02 ± 4.38E+02–	6.68E+02 ± 4.24E+02
	$CEC2013_{22}$	1.06E+04 ± 2.80E+02≈	1.09E+04 ± 4.16E+02	9.28E+03 ± 6.39E+02+	9.95E+03 ± 3.97E+02
	$CEC2013_{23}$	1.44E+04 ± 3.28E+02≈	1.41E+04 ± 3.43E+02	1.41E+04 ± 3.83E+02≈	1.39E+04 ± 4.36E+02
	$CEC2013_{24}$	3.78E+02 ± 4.40E+00–	3.39E+02 ± 1.24E+01	3.15E+02 ± 3.74E+01–	2.35E+02 ± 1.58E+01
	$CEC2013_{25}$	4.13E+02 ± 4.42E+00≈	4.13E+02 ± 7.06E+00	3.41E+02 ± 3.61E+01–	2.84E+02 ± 2.03E+01
	$CEC2013_{26}$	3.64E+02 ± 8.47E+01≈	3.47E+02 ± 1.35E+02	4.19E+02 ± 1.04E+02–	3.79E+02 ± 9.32E+01
	$CEC2013_{27}$	2.15E+03 ± 3.59E+01≈	2.11E+03 ± 6.56E+01	1.99E+03 ± 2.16E+02–	9.83E+02 ± 3.95E+02
	$CEC2013_{28}$	4.84E+02 ± 8.86E+00–	4.05E+02 ± 7.82E−01	5.86E+02 ± 7.54E+02–	5.16E+02 ± 5.84E+02
+		1		4	
–		16		16	
≈		11		8	

ations (FEs) *MaxFEs* was set to $10,000 * D$ and the average function error value smaller than 10^{-8} was taken as zero.

Note that when our framework is applied to a specified DE algorithm, the name of this DE algorithm will be changed by adding four letters "CPI-". For example, DE/rand/1/bin under our framework is called CPI-DE/rand/1/bin.

5.1. CPI-DE for two classic DE versions

The proposed DE framework, i.e., CPI-DE, is firstly applied to two classic versions of DE introduced in Section 2, i.e., DE/rand/1/bin and DE/current-to-best/1/bin. DE/rand/1/bin selects individuals for mutation in a random manner and does not add any search bias; therefore it is an unbiased DE. In contrast, DE/current-to-best/1/bin is a relatively greedy DE, since the information of the best individual in the population is exploited to produce the trial vectors. The aim here is to investigate how CPI-DE influences the performance of unbiased and greedy DE.

The parameter settings of these two classic versions of DE were: $NP = D$, $F = 0.9$, and $CR = 0.5$. For each test function, 51 independent runs were implemented. The experimental results of CEC2013$_1$-CEC2013$_{28}$ with 30D and 50D have been reported in Tables 1 and 2, and the experimental results of CEC2014$_1$-CEC2014$_{30}$ with 30D and 50D have been reported in Tables 3 and 4, where "+", "−", and "≈" denote that the performance of a classic DE version is better than, worse than, and similar to that of its augmented algorithm, respectively. One of the first observations from Tables 1–4 is that our framework is able to enhance the performance of these two classic DE versions on the majority of test functions.

The detailed performance comparisons from Tables 1–4 are summarized as follows:

- In the case of CEC2013$_1$-CEC2013$_{28}$ with $D = 30$, CPI-DE/rand/1/bin and CPI-DE/current-to-best/1/bin exhibit better performance than their original algorithms on 17 and 15 test functions, respectively. When $D = 50$, they surpass their original algorithms on 16 test functions. In terms of CEC2014$_1$-CEC2014$_{30}$ with $D = 30$, CPI-DE/rand/1/bin and CPI-DE/current-to-best/1/bin perform better than their original algorithms on 21 and 15 test functions, respectively. With respect to $D = 50$, they have an edge over their original algorithms on 20 and 19 test functions, respectively. However, it can be seen from Tables 1–4 that the number of test functions that DE/rand/1/bin and DE/current-to-best/1/bin beat their augmented algorithms is less than five.
- For CEC2013$_1$-CEC2013$_{28}$, our framework fails to consistently provide the results of higher quality on three basic multimodal functions (i.e., CEC2013$_{11}$, CEC2013$_{14}$, and CEC2013$_{19}$) and two composition functions (i.e., CEC2013$_{21}$ and CEC2013$_{22}$). As far as CEC2014$_1$-CEC2014$_{30}$ are considered, the similar phenomenon can also be observed on three simple multimodal functions (i.e., CEC2014$_8$, CEC2014$_{10}$, and CEC2014$_{12}$) and one hybrid function (i.e., CEC2014$_{22}$). The failure could be because our framework generates two trial vectors for each target vector, which results in a smaller number of iterations.
- The performance of CPI-DE/rand/1/bin and CPI-DE/current-to-best/1/bin is better than or similar to that of their original

Table 3
Experimental results of DE/rand/1/bin, CPI-DE/rand/1/bin, DE/current-to-best/1/bin, and CPI-DE/current-to-best/1/bin over 51 independent runs on 30 test functions with 30D from IEEE CEC2014 using 300,000 FEs.

Test Functions with 30D from IEEE CEC2014		DE/rand/1/bin (Mean Error ± Std Dev)	CPI-DE/rand/1/bin (Mean Error ± Std Dev)	DE/current-to-best/1/bin (Mean Error ± Std Dev)	CPI-DE/current-to-best/1/bin (Mean Error ± Std Dev)
Unimodal Functions	CEC2014$_1$	9.73E+07 ± 1.74E+07−	1.97E−05 ± 1.73E−05	**1.87E+07 ± 9.06E+06**−	**0.00E+00 ± 0.00E+00**
	CEC2014$_2$	5.89E+02 ± 2.62E+02−	4.90E−06 ± 3.12E−06	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
	CEC2014$_3$	**1.16E+01 ± 5.01E+00**−	**0.00E+00 ± 0.00E+00**	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
Simple Multimodal Functions	CEC2014$_4$	1.30E+02 ± 1.32E+01−	1.24E+00 ± 8.87E+00	2.45E+01 ± 3.42E+01−	2.48E+00 ± 1.24E+01
	CEC2014$_5$	2.08E+01 ± 5.39E−02≈	2.09E+01 ± 5.71E−02	2.09E+01 ± 6.28E−02≈	2.09E+01 ± 5.15E−02
	CEC2014$_6$	3.13E+01 ± 1.16E+00−	2.53E+01 ± 7.11E+00	1.03E+00 ± 1.08E+00−	7.86E−01 ± 9.08E−01
	CEC2014$_7$	2.84E−01 ± 1.82E−01−	1.56E−06 ± 1.11E−05	5.80E−04 ± 2.00E−03≈	5.77E−04 ± 2.10E−03
	CEC2014$_8$	7.33E+01 ± 8.22E+00+	9.99E+01 ± 8.87E+00	6.82E+00 ± 2.69E+00−	6.51E+00 ± 2.73E+00
	CEC2014$_9$	2.07E+02 ± 1.28E+01−	1.94E+02 ± 1.06E+01	1.73E+02 ± 1.47E+01≈	1.70E+02 ± 1.26E+01
	CEC2014$_{10}$	3.27E+03 ± 2.65E+02+	3.76E+03 ± 3.12E+02	2.45E+02 ± 3.50E+02+	1.34E+03 ± 8.31E+02
	CEC2014$_{11}$	6.50E+03 ± 2.94E+02≈	6.55E+03 ± 2.35E+02	6.36E+03 ± 4.13E+02≈	6.52E+03 ± 2.88E+02
	CEC2014$_{12}$	2.01E+00 ± 2.75E−01+	2.13E+00 ± 2.71E−01	1.93E+00 ± 2.30E−01+	2.15E+00 ± 2.84E−01
	CEC2014$_{13}$	5.48E−01 ± 6.60E−02−	4.60E−01 ± 5.60E−02	3.91E−01 ± 5.33E−02−	3.41E−01 ± 5.06E−02
	CEC2014$_{14}$	3.49E−01 ± 6.13E−02−	2.61E−01 ± 3.40E−02	3.40E−01 ± 1.63E−01−	2.68E−01 ± 6.72E−02
	CEC2014$_{15}$	2.03E+01 ± 1.33E+00−	1.65E+01 ± 8.33E−01	1.63E+01 ± 1.03E+00−	1.51E+01 ± 8.25E−01
	CEC2014$_{16}$	1.24E+01 ± 2.21E−01≈	1.25E+01 ± 1.91E−01	1.22E+01 ± 2.44E−01≈	1.22E+01 ± 2.86E−01
Hybrid Functions	CEC2014$_{17}$	2.17E+06 ± 5.36E+05−	1.31E+03 ± 1.81E+02	4.53E+05 ± 2.99E+05−	5.79E+02 ± 3.23E+02
	CEC2014$_{18}$	1.23E+04 ± 6.31E+03−	4.74E+01 ± 6.65E+00	4.22E+02 ± 2.75E+02−	3.73E+01 ± 2.03E+01
	CEC2014$_{19}$	1.04E+01 ± 1.14E+00−	6.12E+00 ± 1.36E+00	6.15E+00 ± 9.95E−01−	4.89E+00 ± 8.02E−01
	CEC2014$_{20}$	2.02E+02 ± 3.37E+01−	3.62E+01 ± 4.23E+00	7.95E+01 ± 8.81E+00−	2.99E+01 ± 1.17E+01
	CEC2014$_{21}$	1.31E+05 ± 4.06E+04−	7.80E+02 ± 1.35E+02	9.68E+03 ± 7.01E+03−	5.16E+02 ± 2.40E+02
	CEC2014$_{22}$	1.76E+02 ± 5.76E+01−	1.70E+02 ± 6.86E+01	1.27E+02 ± 1.00E+02+	1.45E+02 ± 1.19E+02
Composition Functions	CEC2014$_{23}$	3.15E+02 ± 4.28E−05≈	3.15E+02 ± 4.36E−13	3.15E+02 ± 4.02E−13≈	3.15E+02 ± 4.01E−13
	CEC2014$_{24}$	2.08E+02 ± 3.60E+00−	2.00E+02 ± 1.59E−02	2.24E+02 ± 4.64E+00≈	2.21E+02 ± 5.87E+00
	CEC2014$_{25}$	2.24E+02 ± 2.95E+00−	2.02E+02 ± 2.73E−02	2.08E+02 ± 2.73E+00−	2.02E+02 ± 2.67E−01
	CEC2014$_{26}$	1.00E+02 ± 5.97E−02≈	1.00E+02 ± 4.79E−02	1.00E+02 ± 5.09E−02≈	1.00E+02 ± 5.31E−02
	CEC2014$_{27}$	6.04E+02 ± 1.29E+02−	3.25E+02 ± 4.17E+01	3.57E+02 ± 4.42E+01≈	3.55E+02 ± 4.44E+01
	CEC2014$_{28}$	1.00E+03 ± 2.41E+01≈	9.97E+02 ± 2.90E+01	7.88E+02 ± 5.24E+01≈	7.82E+02 ± 5.06E+01
	CEC2014$_{29}$	1.03E+04 ± 3.42E+03−	7.12E+02 ± 1.19E+02	2.25E+03 ± 8.89E+02−	6.80E+02 ± 1.16E+02
	CEC2014$_{30}$	4.94E+03 ± 9.06E+02−	1.01E+03 ± 1.70E+02	1.48E+03 ± 4.17E+02−	9.23E+02 ± 4.25E+02
+		3		3	
−		21		15	
≈		6		12	

Table 4
Experimental results of DE/rand/1/bin, CPI-DE/rand/1/bin, DE/current-to-best/1/bin, and CPI-DE/current-to-best/1/bin over 51 independent runs on 30 test functions with 50D from IEEE CEC2014 using 500,000 FEs.

Test Functions with 50D from IEEE CEC2014		DE/rand/1/bin (Mean Error ± Std Dev)	CPI-DE/rand/1/bin (Mean Error ± Std Dev)	DE/current-to-best/1/bin (Mean Error ± Std Dev)	CPI-DE/current-to-best/1/bin (Mean Error ± Std Dev)
Unimodal Functions	$CEC2014_1$	4.74E + 08 ± 6.65E + 07−	3.87E + 05 ± 1.43E + 05	1.75E + 08 ± 4.54E + 07−	1.01E + 03 ± 6.10E + 03
	$CEC2014_2$	1.47E + 09 ± 2.91E + 08−	1.36E + 06 ± 2.65E + 05	8.54E + 02 ± 1.10E + 03−	7.76E − 06 ± 7.24E − 06
	$CEC2014_3$	8.54E + 04 ± 7.65E + 03−	5.52E + 00 ± 1.53E + 00	**3.93E + 04 ± 4.72E + 03**−	**0.00E + 00 ± 0.00E + 00**
Simple Multimodal Functions	$CEC2014_4$	3.89E + 02 ± 4.35E + 01−	9.35E + 01 ± 2.42E + 00	9.60E + 01 ± 3.65E + 00−	3.91E + 01 ± 4.48E + 01
	$CEC2014_5$	2.11E + 01 ± 3.43E − 02≈	2.11E + 01 ± 4.52E − 02	2.11E + 01 ± 3.95E − 02≈	2.11E + 01 ± 3.43E − 02
	$CEC2014_6$	6.40E + 01 ± 1.50E + 00−	6.18E + 01 ± 2.27E + 00	4.94E + 00 ± 9.12E + 00−	1.49E + 00 ± 1.27E + 00
	$CEC2014_7$	3.60E + 00 ± 4.17E − 01−	8.43E − 01 ± 6.84E − 02	**1.93E − 04 ± 1.38E − 03**−	**0.00E + 00 ± 0.00E + 00**
	$CEC2014_8$	2.80E + 02 ± 1.25E + 01+	3.09E + 02 ± 1.21E + 01	1.30E + 01 ± 3.01E + 00+	9.93E + 01 ± 2.66E + 01
	$CEC2014_9$	4.61E + 02 ± 1.67E + 01−	4.10E + 02 ± 1.53E + 01	3.91E + 02 ± 1.38E + 01−	3.71E + 02 ± 1.61E + 01
	$CEC2014_{10}$	9.14E + 03 ± 3.25E + 02+	9.75E + 03 ± 3.19E + 02	5.24E + 03 ± 1.61E + 03+	7.83E + 03 ± 6.91E + 02
	$CEC2014_{11}$	1.30E + 04 ± 3.98E + 02≈	1.30E + 04 ± 3.31E + 02	1.30E + 04 ± 3.34E + 02≈	1.29E + 04 ± 3.83E + 02
	$CEC2014_{12}$	3.09E + 00 ± 2.94E − 01+	3.21E + 00 ± 2.60E − 01	3.10E + 00 ± 2.63E − 01≈	3.14E + 00 ± 2.48E − 01
	$CEC2014_{13}$	7.97E − 01 ± 7.75E − 02−	6.57E − 01 ± 6.78E − 02	5.66E − 01 ± 6.55E − 02−	4.93E − 01 ± 5.65E − 02
	$CEC2014_{14}$	5.59E − 01 ± 2.24E − 01−	3.00E − 01 ± 4.85E − 02	4.47E − 01 ± 2.53E − 01−	3.45E − 01 ± 1.61E − 01
	$CEC2014_{15}$	4.79E + 02 ± 1.69E + 02−	3.60E + 01 ± 1.55E + 00	3.45E + 01 ± 1.38E + 00−	3.16E + 01 ± 1.78E + 00
	$CEC2014_{16}$	2.24E + 01 ± 1.93E − 01≈	2.23E + 01 ± 1.96E − 01	2.22E + 01 ± 2.71E − 01≈	2.21E + 01 ± 2.58E − 01
Hybrid Functions	$CEC2014_{17}$	2.16E + 07 ± 5.69E + 06−	3.90E + 03 ± 2.90E + 02	1.00E + 07 ± 3.17E + 06−	3.21E + 03 ± 3.05E + 02
	$CEC2014_{18}$	1.06E + 05 ± 4.23E + 04−	1.45E + 02 ± 9.90E + 00	1.71E + 03 ± 1.26E + 03−	1.58E + 02 ± 2.66E + 01
	$CEC2014_{19}$	3.04E + 01 ± 1.60E + 00−	1.81E + 01 ± 1.91E + 00	1.35E + 01 ± 9.16E − 01−	1.21E + 01 ± 8.88E − 01
	$CEC2014_{20}$	2.89E + 04 ± 6.63E + 03−	1.10E + 02 ± 9.87E + 00	1.07E + 04 ± 3.09E + 03−	1.47E + 02 ± 3.94E + 01
	$CEC2014_{21}$	8.66E + 06 ± 2.18E + 06−	2.54E + 03 ± 1.89E + 02	3.62E + 06 ± 1.27E + 06−	2.13E + 03 ± 2.91E + 02
	$CEC2014_{22}$	1.29E + 03 ± 1.25E + 02≈	1.27E + 03 ± 1.49E + 02	1.11E + 03 ± 2.13E + 02+	1.21E + 03 ± 1.74E + 02
Composition Functions	$CEC2014_{23}$	3.44E + 02 ± 6.23E − 02≈	3.44E + 02 ± 3.03E − 04	3.44E + 02 ± 4.50E − 13≈	3.44E + 02 ± 4.59E − 13
	$CEC2014_{24}$	3.07E + 02 ± 2.49E + 00−	2.86E + 02 ± 2.62E + 00	2.67E + 02 ± 2.70E + 00≈	2.68E + 02 ± 2.84E + 00
	$CEC2014_{25}$	2.91E + 02 ± 9.85E + 00−	2.06E + 02 ± 4.36E − 01	2.37E + 02 ± 8.99E + 00−	2.05E + 02 ± 3.26E − 01
	$CEC2014_{26}$	1.00E + 02 ± 6.79E − 02≈	1.00E + 02 ± 5.80E − 02	1.24E + 02 ± 6.12E + 01−	1.00E + 02 ± 5.49E − 02
	$CEC2014_{27}$	1.77E + 03 ± 3.77E + 01−	1.19E + 03 ± 1.47E + 02	3.86E + 02 ± 5.46E + 01≈	3.83E + 02 ± 5.07E + 01
	$CEC2014_{28}$	1.63E + 03 ± 1.00E + 02≈	1.60E + 03 ± 4.28E + 01	1.06E + 03 ± 4.00E + 01≈	1.03E + 03 ± 5.15E + 01
	$CEC2014_{29}$	3.58E + 05 ± 1.36E + 05−	3.29E + 03 ± 1.47E + 03	3.24E + 04 ± 2.06E + 04−	8.05E + 02 ± 9.30E + 01
	$CEC2014_{30}$	7.06E + 04 ± 1.88E + 04−	8.99E + 03 ± 4.07E + 02	1.00E + 04 ± 1.87E + 03−	8.70E + 03 ± 6.36E + 02
+		3		3	
−		20		19	
≈		7		8	

algorithms on all the unimodal function, regardless of the test sets and the number of decision variables.

- Overall, the performance improvement provides by our framework is quite significant on four IEEE CEC2013 test functions (i.e., $CEC2013_2$, $CEC2013_3$, $CEC2013_4$, and $CEC2013_{10}$) and five IEEE CEC2014 test functions ($CEC2014_1$, $CEC2014_{17}$, $CEC2014_{18}$, $CEC2014_{21}$, and $CEC2014_{29}$).
- Compared with the original algorithms, CPI-DE/rand/1/bin and CPI-DE/current-to-best/1/bin have the capability to achieve 100% successful runs for nine cases, which have been highlighted in **boldface** in Tables 1–4.
- It seems that under our framework, the increase of the dimension (from 30 to 50) does not have a remarkable influence on the performance improvement. It is also interesting to note that the advantage of CPI-DE/current-to-best/1/bin over DE/current-to-best/1/bin increases as the number of dimension increases.

In summary, CPI-DE is an effective framework to improve the performance of two classic DE versions (DE/rand/1/bin and DE/current-to-best/1/bin) in the case of two sets of benchmark test functions with 30D and 50D from IEEE CEC2013 and IEEE CEC2014, which indicates that the cumulative population distribution information does play an important role in DE. The convergence graphs of the average function error values derived from two classic DE versions and their augmented algorithms have been given in Fig. 4 for two test functions, i.e., $CEC2013_2$ with 30D and $CEC2013_{10}$ with 30D.

5.2. CPI-DE for three state-of-the-art DE variants

In order to further assess the effectiveness of the proposed framework, CPI-DE is applied to three state-of-the-art DE variants, i.e., JADE [6], jDE [7], and SaDE [8]. In our experiments, the parameter settings of JADE, jDE, and SaDE were the same as in the original papers. Moreover, when applying our framework to these three DE variants, the parameter settings were kept unchanged.

The comparisons were carried out on $CEC2013_1$-$CEC2013_{28}$ with 30D and 50D and $CEC2014_1$-$CEC2014_{30}$ with 30D and 50D. Tables 5 and 6 present the experimental results of $CEC2013_1$-$CEC2013_{28}$, and Tables 7 and 8 summarize the experimental results of $CEC2014_1$-$CEC2014_{30}$, where "+", "−", and "≈" denote that the performance of a DE variant is better than, worse than, and similar to that of its augmented algorithm, respectively.

From Tables 5–8, we can give the following comments:

- It is clear that the proposed framework offers substantial improvement on the performance of JADE, jDE, and SaDE on a lot of test functions. More specifically, for $CEC2013_1$-$CEC2013_{28}$ with 30D, CPI-JADE, CPI-jDE, and CPI-SaDE perform better than their original algorithms on 14, 15, and 15 test functions, respectively. In the case of $D = 50$, CPI-JADE, CPI-jDE, and CPI-SaDE beat their original algorithms on 13, 18, and 16 test functions, respectively. In addition, regarding to $CEC2014_1$-$CEC2014_{30}$, CPI-JADE, CPI-jDE, and CPI-SaDE outperform their original algorithms on 14, 14, and 22 test functions, respectively when $D = 30$. In the case of $D = 50$, the performance of CPI-JADE, CPI-jDE, and CPI-SaDE is better than that of their original algorithms on 13,

Table 5
Experimental results of JADE, CPI-JADE, jDE, CPI-jDE, SaDE, and CPI-SaDE over 51 independent runs on 28 test functions with 30D from IEEE CEC2013 using 300,000 FEs.

Test Functions with 30D from IEEE CEC2013		JADE (Mean Error ± Std Dev)	CPI-JADE (Mean Error ± Std Dev)	jDE (Mean Error ± Std Dev)	CPI-jDE (Mean Error ± Std Dev)	SaDE (Mean Error ± Std Dev)	CPI-SaDE (Mean Error ± Std Dev)
Unimodal Functions	CEC2013$_1$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
	CEC2013$_2$	**7.02E+03 ± 4.59E+03**–	**0.00E+00 ± 0.00E+00**	1.34E+05 ± 7.99E+04–	2.12E–01 ± 1.49E+00	3.73E+05 ± 1.83E+05–	1.00E+04 ± 8.32E+03
	CEC2013$_3$	4.19E+05 ± 1.29E+06–	2.33E+02 ± 1.21E+03	1.55E+06 ± 2.19E+06–	2.96E–01 ± 1.36E+00	1.85E+07 ± 2.57E+07–	1.71E+02 ± 1.06+03
	CEC2013$_4$	**4.90E+03 ± 1.26E+04**–	**0.00E+00 ± 0.00E+00**	**2.39E+01 ± 2.83E+01**–	**0.00E+00 ± 0.00E+00**	**3.17E+03 ± 1.37E+03**–	**0.00E+00 ± 0.00E+00**
	CEC2013$_5$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
Basic Multimodal Functions	CEC2013$_6$	1.63E+00 ± 6.28E+00≈	1.03E+00 ± 5.17E+00	1.37E+01 ± 4.75E+00–	1.01E+01 ± 5.05E+00	3.47E+01 ± 2.99E+01–	5.37E+00 ± 6.70E+00
	CEC2013$_7$	3.60E+00 ± 4.40E+00–	2.34E+00 ± 2.22E+00	2.80E+00 ± 2.42E+00–	1.52E+00 ± 2.23E+00	2.56E+01 ± 1.32E+01≈	2.60E+01 ± 1.46E+01
	CEC2013$_8$	2.09E+01 ± 9.20E–02≈	2.09E+01 ± 4.92E–02	2.09E+01 ± 4.74E–02≈	2.09E+01 ± 5.25E–02	2.09E+01 ± 4.25E–02≈	2.09E+01 ± 5.66E–02
	CEC2013$_9$	2.69E+01 ± 1.54E+00–	2.60E+01 ± 1.44E+00	2.58E+01 ± 3.83E+00–	2.00E+01 ± 6.29E+00	1.85E+01 ± 2.82E+00+	2.28E+01 ± 4.94E+00
	CEC2013$_{10}$	3.50E–02 ± 2.08E–02–	3.02E–02 ± 1.65E–02	3.54E–02 ± 2.14E–02–	6.08E–03 ± 7.17E–03	2.61E–01 ± 1.44E–01–	6.40E–02 ± 4.11E–02
	CEC2013$_{11}$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	**2.34E–01 ± 6.77E–01**–	**0.00E+00 ± 0.00E+00**
	CEC2013$_{12}$	2.40E+01 ± 5.10E+00–	2.36E+01 ± 3.16E+00	6.06E+01 ± 9.13E+00–	5.04E+01 ± 7.82E+00	4.62E+01 ± 1.11E+01–	4.44E+01 ± 1.01E+01
	CEC2013$_{13}$	4.55E+01 ± 1.35E+01–	4.28E+01 ± 1.18E+01	9.19E+01 ± 1.61E+01–	7.86E+01 ± 1.36E+01	9.89E+01 ± 1.88E+01–	7.96E+01 ± 1.82E+01
	CEC2013$_{14}$	3.38E–02 ± 2.35E–02+	2.95E+00 ± 8.42E–01	2.44E–03 ± 6.77E–03+	4.32E+01 ± 1.17E+01	1.12E+00 ± 1.40E+00+	3.48E+00 ± 1.90E+00
	CEC2013$_{15}$	3.30E+03 ± 3.13E+02–	3.20E+03 ± 3.27E+02	5.25E+03 ± 3.92E+02–	5.02E+03 ± 4.51E+02	4.67E+03 ± 1.03E+03–	4.57E+03 ± 3.99E+02
	CEC2013$_{16}$	1.90E+00 ± 6.49E–01+	2.21E+00 ± 4.99E–01	2.40E+00 ± 3.03E–01≈	2.38E+00 ± 2.02E–01	2.19E+00 ± 3.10E–01≈	2.15E+00 ± 3.29E–01
	CEC2013$_{17}$	3.04E+01 ± 2.65E–14≈	3.04E+01 ± 1.44E–03	3.04E+01 ± 9.42E–07+	3.24E+01 ± 5.22E–01	3.04E+01 ± 4.25E–02≈	3.04E+01 ± 5.58E–03
	CEC2013$_{18}$	7.66E+01 ± 7.25E+00≈	7.62E+01 ± 6.25E+00	1.56E+02 ± 1.53E+01–	1.44E+02 ± 1.34E+01	1.28E+02 ± 4.44E+01≈	1.30E+02 ± 1.08E+01
	CEC2013$_{19}$	1.45E+00 ± 1.01E–01+	1.65E+00 ± 1.30E–01	1.66E+00 ± 1.30E–01+	2.20E+00 ± 2.06E–01	4.10E+00 ± 7.04E–01–	2.63E+00 ± 2.85E–01
	CEC2013$_{20}$	1.05E+01 ± 5.07E–01≈	1.03E+01 ± 4.77E–01	1.17E+01 ± 3.41E–01≈	1.15E+01 ± 3.34E–01	1.07E+01 ± 6.75E–01≈	1.12E+01 ± 5.22E–01
Composition Functions	CEC2013$_{21}$	3.09E+02 ± 7.22E+01≈	3.08E+02 ± 8.64E+01	2.76E+02 ± 7.28E+01+	3.08E+02 ± 9.43E+01	3.20E+02 ± 7.70E+01–	3.06E+02 ± 8.87E+01
	CEC2013$_{22}$	9.80E+01 ± 2.50E+01+	1.19E+02 ± 2.09E+01	1.14E+02 ± 1.92E+01+	2.30E+02 ± 6.64E+01	1.21E+02 ± 2.88E+01+	2.09E+02 ± 8.14E+01
	CEC2013$_{23}$	3.46E+03 ± 4.74E+02–	3.20E+03 ± 4.23E+02	5.26E+03 ± 4.45E+02–	4.91E+03 ± 3.62E+02	4.91E+03 ± 1.11E+03–	4.81E+03 ± 4.69E+02
	CEC2013$_{24}$	2.16E+02 ± 1.68E+01–	2.09E+02 ± 1.18E+01	2.10E+02 ± 8.38E+00–	2.04E+02 ± 4.35E+00	2.28E+02 ± 6.08E+00–	2.16E+02 ± 5.58E+00
	CEC2013$_{25}$	2.75E+02 ± 9.97E+00–	2.60E+02 ± 1.37E+01	2.53E+02 ± 9.63E+00–	2.48E+02 ± 9.52E+00	2.65E+02 ± 1.06E+01–	2.57E+02 ± 1.25E+01
	CEC2013$_{26}$	2.27E+02 ± 5.48E+01–	2.15E+02 ± 3.99E–01	2.00E+00 ± 4.73E–03≈	2.00E+00 ± 1.44E–05	2.07E+02 ± 2.97E+01≈	2.06E+02 ± 2.76E+01
	CEC2013$_{27}$	6.98E+02 ± 2.30E+02–	4.85E+02 ± 2.16E+02	6.80E+02 ± 2.25E+02–	4.10E+02 ± 1.19E+02	5.88E+02 ± 6.93E+01–	5.20E+02 ± 1.14E+02
	CEC2013$_{28}$	3.00E+02 ± 0.00E+00≈	3.00E+02 ± 0.00E+00	3.00E+02 ± 0.00E+00≈	3.00E+02 ± 0.00E+00	3.00E+02 ± 0.00E+00≈	3.00E+02 ± 0.00E+00
+		4		5		3	
–		14		15		15	
≈		10		8		10	

Table 6
Experimental results of JADE, CPI-JADE, jDE, CPI-jDE, SaDE, and CPI-SaDE over 51 independent runs on 28 test functions with 50D from IEEE CEC2013 using 500,000 FEs.

Test Functions with 50D from IEEE CEC2013		JADE (Mean Error ± Std Dev)	CPI-JADE (Mean Error ± Std Dev)	jDE (Mean Error ± Std Dev)	CPI-jDE (Mean Error ± Std Dev)	SaDE (Mean Error ± Std Dev)	CPI-SaDE (Mean Error ± Std Dev)
Unimodal Functions	$CEC2013_1$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
	$CEC2013_2$	2.65E+04 ± 1.54E+04–	4.28E+03 ± 8.26E+03	5.10E+05 ± 1.65E+05–	2.77E+04 ± 3.04E+04	7.33E+05 ± 2.24E+05–	6.90E+04 ± 3.56E+04
	$CEC2013_3$	2.76E+06 ± 5.23E+06–	8.47E+05 ± 2.68E+06	3.98E+06 ± 4.52E+06–	2.74E+04 ± 9.27E+04	6.47E+07 ± 6.36E+07–	2.22E+05 ± 5.01E+05
	$CEC2013_4$	**8.00E+03 ± 1.90E+04–**	**0.00E+00 ± 0.00E+00**	**1.77E+02 ± 1.82E+02–**	**0.00E+00 ± 0.00E+00**	**4.66E+03 ± 1.66E+03–**	**0.00E+00 ± 0.00E+00**
	$CEC2013_5$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
Basic Multimodal Functions	$CEC2013_6$	4.36E+01 ± 1.11E+00–	4.34E+01 ± 2.83E−14	4.39E+01 ± 8.65E−01–	4.34E+01 ± 1.10E−02	5.18E+01 ± 2.08E+01–	4.33E+01 ± 8.35E+00
	$CEC2013_7$	2.31E+01 ± 9.99E+00–	1.80E+01 ± 9.51E+00	1.62E+01 ± 6.23E+00–	1.01E+01 ± 4.48E+00	4.86E+01 ± 1.01E+01+	6.14E+01 ± 1.25E+01
	$CEC2013_8$	2.11E+01 ± 7.14E−02≈	2.11E+01 ± 3.48E−02	2.11E+01 ± 4.56E−02≈	2.11E+01 ± 2.67E−02	2.11E+01 ± 4.54E−02≈	2.11E+01 ± 4.41E−02
	$CEC2013_9$	5.43E+01 ± 2.78E+00–	5.38E+01 ± 2.55E+00	5.35E+01 ± 4.75E+00–	4.32E+01 ± 1.20E+01	3.90E+01 ± 4.21E+00≈	3.97E+01 ± 8.69E+00
	$CEC2013_{10}$	3.55E−02 ± 1.75E−02≈	3.60E−02 ± 2.38E−02	4.66E−02 ± 4.20E−02≈	4.73E−02 ± 2.80E−02	2.70E−01 ± 1.29E−01–	1.02E−01 ± 6.09E−02
	$CEC2013_{11}$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	**1.91E+00 ± 1.46E+00–**	**0.00E+00 ± 0.00E+00**
	$CEC2013_{12}$	5.65E+01 ± 1.11E+01+	5.77E+01 ± 9.17E+00	1.08E+02 ± 1.70E+01–	9.42E+01 ± 1.29E+01	1.22E+02 ± 2.19E+01–	1.14E+02 ± 2.07E+01
	$CEC2013_{13}$	1.33E+02 ± 2.44E+01≈	1.35E+02 ± 2.57E+01	1.82E+02 ± 2.79E+01–	1.56E+02 ± 2.28E+01	2.50E+02 ± 3.90E+01–	2.31E+02 ± 3.44E+01
	$CEC2013_{14}$	4.38E−02 ± 2.60E−02+	1.44E+01 ± 3.04E+00	5.94E−03 ± 1.34E−02+	9.29E+01 ± 1.68E+01	6.03E+00 ± 3.76E+00+	5.92E+01 ± 1.36E+01
	$CEC2013_{15}$	7.00E+03 ± 3.77E+02–	6.89E+03 ± 4.04E+02	9.92E+03 ± 5.30E+02–	9.36E+03 ± 5.36E+02	8.53E+03 ± 2.25E+03+	8.98E+03 ± 6.85E+02
	$CEC2013_{16}$	2.15E+00 ± 8.14E−01+	2.48E+00 ± 6.91E−01	3.18E+00 ± 3.64E−01–	3.07E+00 ± 3.00E−01	3.00E+00 ± 2.77E−01–	2.58E+00 ± 5.03E−01
	$CEC2013_{17}$	5.07E+01 ± 4.58E−14≈	5.08E+01 ± 1.31E−02	5.07E+01 ± 7.48E−14+	5.45E+01 ± 8.76E−01	5.14E+01 ± 5.18E−01≈	5.21E+01 ± 3.52E−01
	$CEC2013_{18}$	1.40E+02 ± 1.09E+01≈	1.43E+02 ± 9.54E+00	2.84E+02 ± 2.36E+01–	2.36E+02 ± 1.97E+01	1.57E+02 ± 5.80E+01+	2.56E+02 ± 2.21E+01
	$CEC2013_{19}$	2.75E+00 ± 1.69E−01+	3.18E+00 ± 2.24E−01	2.89E+00 ± 2.32E−01+	3.90E+00 ± 2.96E−01	1.13E+01 ± 2.25E+00–	5.82E+00 ± 7.45E−01
	$CEC2013_{20}$	1.96E+01 ± 5.53E−01≈	1.94E+01 ± 6.54E−01	2.14E+01 ± 4.60E−01–	2.10E+01 ± 4.87E−01	1.97E+01 ± 1.13E+00≈	2.03E+01 ± 5.63E−01
Composition Functions	$CEC2013_{21}$	8.19E+02 ± 3.84E+02–	7.40E+02 ± 5.13E+02	5.74E+02 ± 4.53E+02–	4.98E+02 ± 4.01E+02	8.58E+02 ± 3.34E+02–	7.47E+02 ± 4.06E+02
	$CEC2013_{22}$	1.30E+01 ± 7.12E+00+	5.67E+01 ± 5.13E+01	2.79E+01 ± 2.23E+01+	1.77E+02 ± 9.41E+01	2.68E+01 ± 3.42E+01+	2.69E+02 ± 1.93E+02
	$CEC2013_{23}$	7.31E+03 ± 8.24E+02–	6.87E+03 ± 4.82E+02	9.69E+03 ± 8.21E+02–	9.27E+03 ± 6.82E+02	8.24E+03 ± 1.90E+03+	9.05E+03 ± 8.21E+02
	$CEC2013_{24}$	2.49E+02 ± 2.10E+01≈	2.49E+02 ± 1.67E+01	2.57E+02 ± 1.39E+01–	2.35E+02 ± 1.20E+01	2.77E+02 ± 1.01E+01–	2.58E+02 ± 9.37E+00
	$CEC2013_{25}$	3.55E+02 ± 1.71E+01–	3.31E+02 ± 3.19E+01	3.05E+02 ± 2.02E+01–	2.98E+02 ± 8.91E+00	3.45E+02 ± 1.17E+01–	3.29E+02 ± 1.79E+01
	$CEC2013_{26}$	3.65E+02 ± 9.76E+01–	3.08E+02 ± 1.03E+02	2.89E+02 ± 9.89E+01–	2.57E+02 ± 8.30E+01	2.86E+02 ± 9.35E+01–	2.75E+02 ± 8.76E+01
	$CEC2013_{27}$	1.39E+03 ± 3.33E+02–	1.24E+03 ± 3.24E+02	1.17E+03 ± 2.97E+02–	9.06E+02 ± 2.17E+02	1.19E+03 ± 1.01E+02–	1.09E+03 ± 1.49E+02
	$CEC2013_{28}$	5.73E+02 ± 6.99E+02–	4.57E+02 ± 4.11E+02	4.00E+02 ± 5.27E−14≈	4.00E+02 ± 1.60E−14	5.96E+02 ± 7.93E+02–	4.00E+02 ± 4.68E−14
+		5		4		6	
–		13		18		16	
≈		10		6		6	

Table 7
Experimental results of JADE, CPI-JADE, jDE, CPI-jDE, SaDE, and CPI-SaDE over 51 independent runs on 30 test functions with 30D from IEEE CEC2014 using 300,000 FEs.

Test Functions with 30D from IEEE CEC2014		JADE (Mean Error ± Std Dev)	CPI-JADE (Mean Error ± Std Dev)	jDE (Mean Error ± Std Dev)	CPI-jDE (Mean Error ± Std Dev)	SaDE (Mean Error ± Std Dev)	CPI-SaDE (Mean Error ± Std Dev)
Unimodal Functions	CEC2014$_{1}$	**6.09E+02 ± 1.18E+03**–	**0.00E+00 ± 0.00E+00**	7.35E+04 ± 6.12E+04–	1.10E–08 ± 7.67E–08	3.60E+05 ± 2.74E+05–	9.15E+01 ± 3.10E+02
	CEC2014$_{2}$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
	CEC2014$_{3}$	**9.86E–04 ± 5.95E–03**–	**0.00E+00 ± 0.00E+00**	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	**1.92E+01 ± 5.60E+01**–	**0.00E+00 ± 0.00E+00**
Simple Multimodal Functions	CEC2014$_{4}$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	5.09E+00 ± 1.48E+01–	6.84E–02 ± 1.30E–01	4.13E+01 ± 3.65E+01–	8.94E–05 ± 4.41E–04
	CEC2014$_{5}$	2.03E+01 ± 3.23E–02≈	2.03E+01 ± 3.68E–02	2.03E+01 ± 3.80E–02≈	2.04E+01 ± 3.74E–02	2.05E+01 ± 4.94E–02≈	2.04E+01 ± 3.93E–02
	CEC2014$_{6}$	9.15E+00 ± 2.21E+00–	3.44E+00 ± 3.57E+00	3.39E+00 ± 3.97E+00–	1.06E+00 ± 1.64E+00	4.86E+00 ± 2.15E+00–	1.29E+00 ± 1.78E+00
	CEC2014$_{7}$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	**1.12E–02 ± 1.50E–02**–	**0.00E+00 ± 0.00E+00**
	CEC2014$_{8}$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	**5.85E–02 ± 2.36E–01**–	**0.00E+00 ± 0.00E+00**
	CEC2014$_{9}$	2.62E+01 ± 4.96E+00–	2.24E+01 ± 5.33E+00	4.40E+01 ± 5.33E+00–	4.16E+01 ± 5.88E+00	3.72E+01 ± 8.61E+00+	4.16E+01 ± 7.19E+00
	CEC2014$_{10}$	8.16E–03 ± 1.18E–02+	3.83E–01 ± 7.47E–02	1.22E–03 ± 4.94E–03+	3.36E+00 ± 1.61E+00	2.82E–01 ± 4.35E–01–	1.17E–01 ± 9.29E–02
	CEC2014$_{11}$	1.67E+03 ± 2.13E+02≈	1.77E+03 ± 2.55E+02	2.41E+03 ± 3.11E+02≈	2.69E+03 ± 3.23E+02	3.25E+03 ± 5.37E+02–	2.63E+03 ± 2.79E+02
	CEC2014$_{12}$	2.67E–01 ± 3.57E–02+	3.95E–01 ± 8.64E–02	4.56E–01 ± 6.46E–02+	5.48E–01 ± 9.15E–02	7.95E–01 ± 9.96E–02–	5.59E–01 ± 9.48E–02
	CEC2014$_{13}$	2.20E–01 ± 3.25E–02–	2.04E–01 ± 3.38E–02	3.04E–01 ± 3.54E–02–	2.85E–01 ± 3.66E–02	2.66E–01 ± 4.05E–02+	2.93E–01 ± 4.07E–02
	CEC2014$_{14}$	2.41E–01 ± 3.18E–02–	2.32E–01 ± 3.35E–02	2.83E–01 ± 2.95E–02–	2.58E–01 ± 2.98E–02	2.35E–01 ± 3.70E–02≈	2.36E–01 ± 2.93E–02
	CEC2014$_{15}$	3.20E+00 ± 4.55E–01≈	3.26E+00 ± 3.78E–01	5.89E+00 ± 7.23E–01≈	5.91E+00 ± 8.20E–01	4.10E+00 ± 1.40E+00+	6.29E+00 ± 1.01E+00
	CEC2014$_{16}$	9.30E+00 ± 4.61E–01+	9.70E+00 ± 2.79E–01	9.85E+00 ± 3.81E–01+	1.03E+01 ± 3.22E–01	1.10E+01 ± 2.64E–01–	1.03E+01 ± 4.09E–01
Hybrid Functions	CEC2014$_{17}$	1.91E+04 ± 1.08E+05–	1.16E+03 ± 3.81E+02	1.13E+03 ± 9.03E+02–	2.56E+02 ± 1.74E+02	1.40E+04 ± 1.36E+04–	8.08E+02 ± 3.74E+02
	CEC2014$_{18}$	1.14E+02 ± 1.97E+02–	9.47E+01 ± 3.42E+01	1.66E+01 ± 6.53E+00–	1.18E+01 ± 4.24E+00	3.52E+02 ± 4.95E+02–	6.91E+01 ± 2.73E+01
	CEC2014$_{19}$	4.48E+00 ± 7.56E–01+	4.89E+00 ± 7.64E–01	4.36E+00 ± 5.94E–01–	4.15E+00 ± 6.88E–01	6.31E+00 ± 1.15E+01–	4.78E+00 ± 8.97E–01
	CEC2014$_{20}$	3.11E+03 ± 3.01E+03–	1.12E+01 ± 5.24E+00	1.16E+01 ± 3.51E+00–	8.53E+00 ± 1.66E+00	1.39E+02 ± 2.02E+02–	2.45E+01 ± 1.52E+01
	CEC2014$_{21}$	1.33E+04 ± 4.12E+04–	3.33E+02 ± 1.54E+02	2.74E+02 ± 1.71E+02–	7.09E+01 ± 7.18E+01	4.46E+03 ± 7.23E+03–	2.32E+02 ± 1.19E+02
	CEC2014$_{22}$	1.44E+02 ± 7.74E+01–	9.99E+01 ± 6.09E+01	1.08E+02 ± 7.15E+01–	6.08E+01 ± 4.59E+01	1.54E+02 ± 5.78E+01–	1.00E+02 ± 6.49E+01
Composition Functions	CEC2014$_{23}$	3.15E+02 ± 4.01E–13≈	3.15E+02 ± 4.01E–13	3.15E+02 ± 4.01E–13≈	3.15E+02 ± 4.15E–13	3.15E+02 ± 2.24E–13≈	3.15E+02 ± 2.29E–13
	CEC2014$_{24}$	2.25E+02 ± 3.60E+00≈	2.24E+02 ± 2.93E+00	2.25E+02 ± 2.56E+00≈	2.23E+02 ± 8.32E–01	2.26E+02 ± 2.79E+00≈	2.24E+02 ± 9.76E–01
	CEC2014$_{25}$	2.03E+02 ± 1.13E+00≈	2.03E+02 ± 5.77E–01	2.03E+02 ± 5.31E–01≈	2.02E+02 ± 3.22E–01	2.08E+02 ± 2.54E+00–	2.03E+02 ± 1.07E+00
	CEC2014$_{26}$	1.02E+02 ± 1.39E+01–	1.00E+02 ± 2.92E–02	1.00E+02 ± 4.02E–02≈	1.00E+02 ± 3.39E–02	1.11E+02 ± 3.24E+01–	1.00E+02 ± 4.09E–02
	CEC2014$_{27}$	3.35E+02 ± 4.68E+01+	3.53E+02 ± 5.03E+01	3.62E+02 ± 4.69E+01+	3.84E+02 ± 3.54E+01	4.20E+02 ± 4.42E+01–	3.70E+02 ± 4.18E+01
	CEC2014$_{28}$	7.96E+02 ± 4.63E+01≈	8.02E+02 ± 4.34E+01	7.99E+02 ± 2.68E+01≈	8.02E+02 ± 2.78E+01	8.93E+02 ± 3.46E+01–	8.23E+02 ± 3.26E+01
	CEC2014$_{29}$	8.28E+02 ± 3.27E+02–	8.13E+02 ± 7.12E+01	8.13E+02 ± 7.12E+01–	5.51E+02 ± 2.48E+02	1.10E+03 ± 2.16E+02–	6.41E+02 ± 1.62E+02
	CEC2014$_{30}$	1.66E+03 ± 7.61E+02–	1.40E+03 ± 7.24E+02	1.40E+03 ± 5.06E+02–	7.23E+02 ± 3.11E+02	1.48E+03 ± 5.40E+02–	7.60E+02 ± 3.65E+02
+		5		4		3	
–		14		14		22	
≈		11		12		5	

Table 8
Experimental results of JADE, CPI-JADE, jDE, CPI-jDE, SaDE, and CPI-SaDE over 51 independent runs on 30 test functions with 50D from IEEE CEC2014 using 500,000 FEs.

Test Functions with 50D from IEEE CEC2014		JADE (Mean Error ± Std Dev)	CPI-JADE (Mean Error ± Std Dev)	jDE (Mean Error ± Std Dev)	CPI-jDE (Mean Error ± Std Dev)	SaDE (Mean Error ± Std Dev)	CPI-SaDE (Mean Error ± Std Dev)
Unimodal Functions	$CEC2014_{1}$	1.45E+04 ± 9.43E+03−	4.73E+00 ± 3.33E+01	4.58E+05 ± 2.03E+05−	1.97E+04 ± 2.08E+04	9.38E+05 ± 2.96E+05−	5.53E+04 ± 3.31E+04
	$CEC2014_{2}$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	9.19E−09 ± 1.83E−08+	3.52E+02 ± 1.24E+03	3.65E+03 ± 4.02E+03−	2.27E−02 ± 8.85E−02
	$CEC2014_{3}$	**3.96E+03 ± 2.41E+03**−	**0.00E+00 ± 0.00E+00**	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	**3.04E+03 ± 1.64E+03**−	**0.00E+00 ± 0.00E+00**
Simple Multimodal Functions	$CEC2014_{4}$	2.35E+01 ± 4.12E+01−	9.70E+00 ± 2.94E+01	8.70E+01 ± 1.92E+01−	5.97E+01 ± 3.77E+01	9.34E+01 ± 3.89E+01−	1.68E+01 ± 2.41E+01
	$CEC2014_{5}$	2.03E+01 ± 2.81E−02≈	2.04E+01 ± 4.04E−02	2.04E+01 ± 3.30E−02≈	2.05E+01 ± 3.99E−02	2.07E+01 ± 4.62E−02≈	2.05E+01 ± 3.82E−02
	$CEC2014_{6}$	1.56E+01 ± 6.56E+00−	4.62E+00 ± 2.95E+00	8.88E+00 ± 7.14E+00−	7.65E+00 ± 7.59E+00	1.77E+01 ± 3.55E+00−	9.27E+00 ± 2.27E+00
	$CEC2014_{7}$	2.84E−03 ± 6.74E−03−	7.24E−04 ± 2.57E−03	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	1.43E−02 ± 1.36E−02−	4.63E−03 ± 6.52E−03
	$CEC2014_{8}$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00	**1.46E+00 ± 1.78E+00**−	**0.00E+00 ± 0.00E+00**
	$CEC2014_{9}$	5.15E+01 ± 7.85E+00−	4.97E+01 ± 7.66E+00	9.15E+01 ± 9.51E+00−	9.10E+01 ± 1.03E+01	8.78E+01 ± 1.45E+01−	8.62E+01 ± 1.23E+01
	$CEC2014_{10}$	1.17E−02 ± 1.33E−02+	1.93E+00 ± 5.04E−01	7.34E−04 ± 2.96E−03+	7.58E+00 ± 2.70E+00	1.57E+00 ± 1.11E+00+	6.05E+00 ± 2.53E+00
	$CEC2014_{11}$	3.84E+03 ± 3.04E+02+	4.06E+03 ± 2.77E+02	5.22E+03 ± 3.62E+02+	5.47E+03 ± 3.86E+02	6.49E+03 ± 1.70E+03−	5.81E+03 ± 5.12E+02
	$CEC2014_{12}$	2.50E−01 ± 3.42E−02+	3.96E−01 ± 6.93E−02	4.93E−01 ± 5.40E−02+	5.98E−01 ± 6.22E−02	1.10E+00 ± 1.10E−01−	7.05E−01 ± 7.51E−02
	$CEC2014_{13}$	3.29E−01 ± 4.25E−02−	3.07E−01 ± 5.63E−02	3.84E−01 ± 4.45E−02−	3.63E−01 ± 4.17E−02	4.26E−01 ± 5.82E−02−	4.16E−01 ± 6.58E−02
	$CEC2014_{14}$	3.04E−01 ± 8.56E−02≈	3.06E−01 ± 6.30E−02	3.26E−01 ± 5.50E−02−	2.90E−01 ± 2.09E−02	3.09E−01 ± 3.72E−02−	2.98E−01 ± 2.97E−02
	$CEC2014_{15}$	7.17E+00 ± 8.18E−01+	7.64E+00 ± 8.11E−01	1.20E+01 ± 1.28E+00≈	1.19E+01 ± 1.53E+00	1.46E+01 ± 4.25E+00+	1.66E+01 ± 2.30E+00
	$CEC2014_{16}$	1.77E+01 ± 3.71E−01≈	1.78E+01 ± 4.28E−01	1.82E+01 ± 3.72E−01≈	1.85E+01 ± 3.55E−01	2.01E+01 ± 3.19E−01−	1.93E+01 ± 3.65E−01
Hybrid Functions	$CEC2014_{17}$	2.40E+03 ± 6.31E+02−	2.26E+03 ± 4.13E+02	2.16E+04 ± 1.32E+04−	1.95E+03 ± 5.08E+02	5.72E+04 ± 3.41E+04−	2.05E+03 ± 5.70E+02
	$CEC2014_{18}$	1.74E+02 ± 5.03E+01≈	1.74E+02 ± 5.18E+01	5.91E+02 ± 7.38E+02−	1.35E+02 ± 4.59E+01	5.82E+02 ± 5.63E+02−	1.45E+02 ± 3.04E+01
	$CEC2014_{19}$	1.30E+01 ± 6.01E+00−	1.09E+01 ± 5.16E+00	1.30E+01 ± 4.48E+00≈	1.28E+01 ± 5.21E+00	1.39E+01 ± 6.45E+00+	2.28E+01 ± 1.24E+01
	$CEC2014_{20}$	8.27E+03 ± 6.67E+03−	1.86E+02 ± 4.63E+01	4.86E+01 ± 1.64E+01−	4.54E+01 ± 1.89E+01	8.79E+02 ± 6.50E+02−	2.18E+02 ± 8.91E+01
	$CEC2014_{21}$	1.25E+03 ± 3.07E+02+	1.49E+03 ± 3.61E+02	8.53E+03 ± 7.94E+03−	9.13E+02 ± 3.41E+02	6.25E+04 ± 3.33E+04−	1.23E+03 ± 3.70E+02
	$CEC2014_{22}$	4.81E+02 ± 1.58E+02−	3.59E+02 ± 1.31E+02	5.40E+02 ± 1.12E+02−	3.64E+02 ± 1.12E+02	4.80E+02 ± 1.43E+02≈	4.78E+02 ± 1.47E+02
Composition Functions	$CEC2014_{23}$	3.44E+02 ± 4.26E−13≈	3.44E+02 ± 4.30E−13	3.44E+02 ± 4.17E−13≈	3.44E+02 ± 3.19E−13	3.44E+02 ± 2.85E−13≈	3.44E+02 ± 2.87E−13
	$CEC2014_{24}$	2.74E+02 ± 1.66E+00≈	2.74E+02 ± 2.14E+00	2.68E+02 ± 2.17E+00≈	2.67E+02 ± 2.05E+00	2.75E+02 ± 3.44E+00≈	2.72E+02 ± 3.48E+00
	$CEC2014_{25}$	2.16E+02 ± 7.03E+00−	2.08E+02 ± 3.78E+00	2.07E+02 ± 1.47E+00≈	2.07E+02 ± 1.34E+00	2.17E+02 ± 8.71E+00−	2.11E+02 ± 1.01E+01
	$CEC2014_{26}$	1.00E+02 ± 1.33E−01≈	1.08E+02 ± 2.73E+01	1.00E+02 ± 3.69E−02≈	1.00E+02 ± 3.95E−02	1.94E+02 ± 2.36E+01−	1.35E+02 ± 4.80E+01
	$CEC2014_{27}$	4.48E+02 ± 5.03E+01+	4.75E+02 ± 6.08E+01	4.48E+02 ± 7.13E+01−	4.04E+02 ± 6.97E+01	7.67E+02 ± 6.81E+01−	6.08E+02 ± 6.72E+01
	$CEC2014_{28}$	1.18E+03 ± 5.71E+01≈	1.21E+03 ± 2.21E+02	1.09E+03 ± 3.35E+01≈	1.13E+03 ± 5.08E+01	1.41E+03 ± 1.25E+02−	1.23E+03 ± 1.04E+02
	$CEC2014_{29}$	9.00E+02 ± 6.73E+01−	8.86E+02 ± 5.73E+01	1.04E+03 ± 1.95E+02−	7.80E+02 ± 7.33E+01	1.43E+03 ± 3.82E+02−	8.70E+02 ± 6.22E+01
	$CEC2014_{30}$	9.71E+03 ± 7.21E+02+	1.07E+04 ± 8.37E+02	8.70E+03 ± 4.77E+02+	9.20E+03 ± 7.44E+02	1.19E+04 ± 1.78E+03−	1.02E+04 ± 1.13E+03
+		7		5		3	
−		13		13		23	
≈		10		12		4	

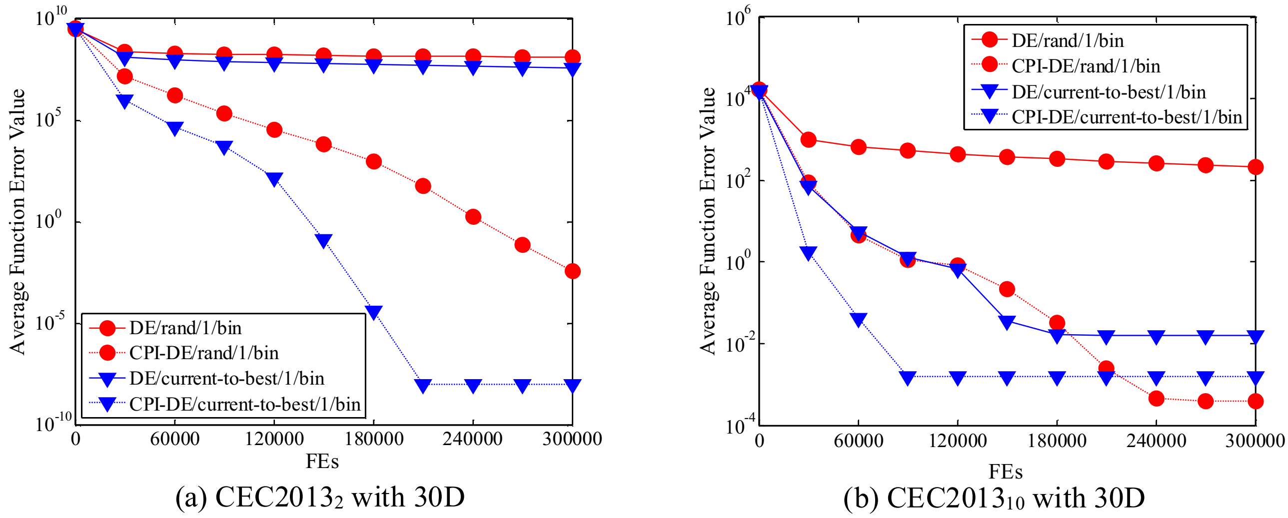

Fig. 4. Evolution of the average function error values derived from two classic DE versions (DE/rand/1/bin and DE/current-to-best/1/bin) and their augmented algorithms versus the number of FE_S on $CEC2013_2$ with 30D and $CEC2013_{10}$ with 30D.

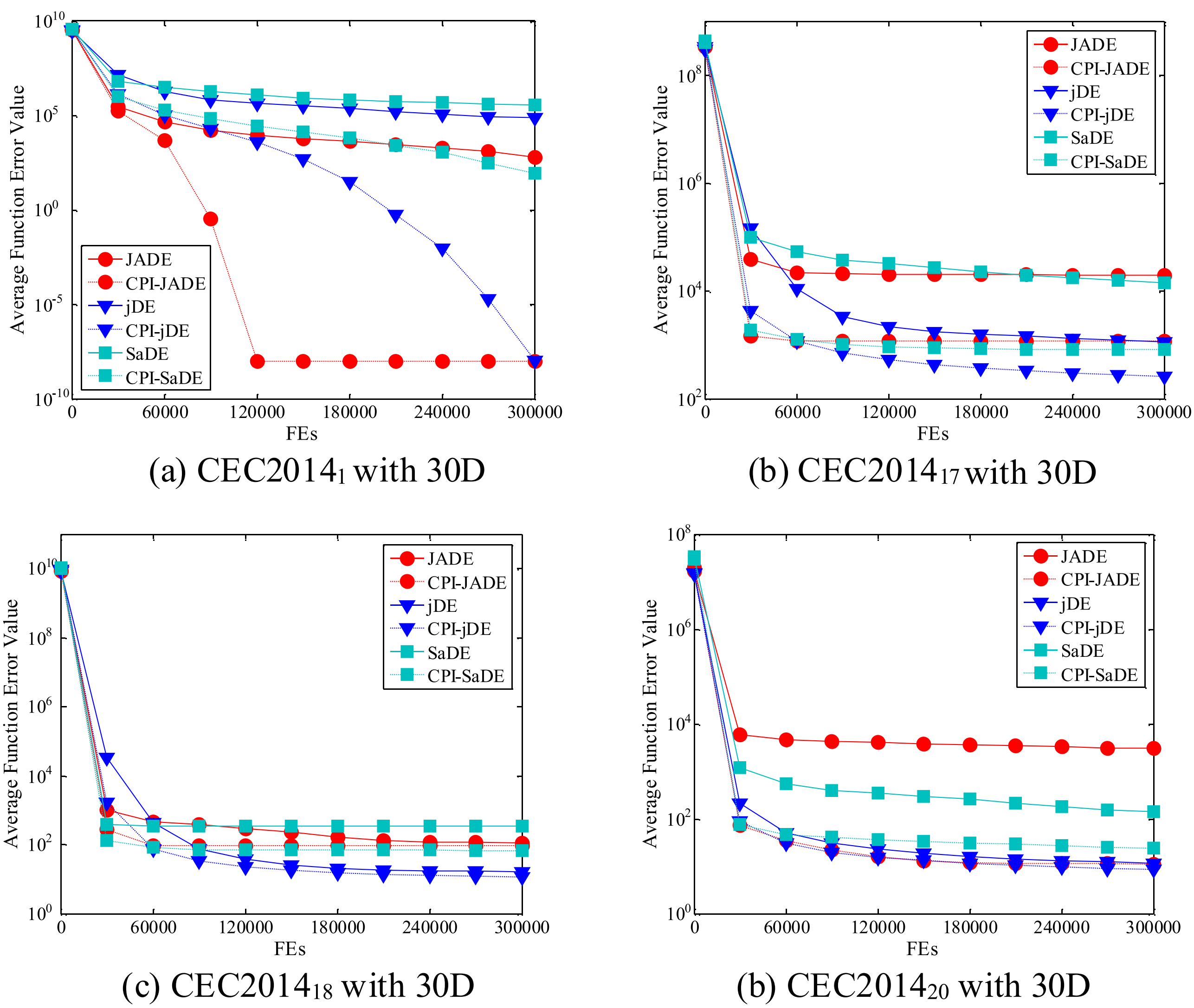

Fig. 5. Evolution of the average function error values derived from three state-of-the-art DE variants (JADE, jDE, and SaDE) and their augmented algorithms versus the number of FE_S on $CEC2014_1$ with 30D, $CEC2014_{17}$ with 30D, $CEC2014_{18}$ with 30D, and $CEC2014_{20}$ with 30D.

Table 9
Experimental results of SPI-JADE, JADE/eig, CoJADE, and CPI-JADE over 51 independent runs on 28 test functions with 30D from IEEE CEC2013 using 300,000 FEs.

Test Functions with 30D from IEEE CEC2013		SPI-JADE (Mean Error ± Std Dev)	JADE/eig (Mean Error ± Std Dev)	CoJADE (Mean Error ± Std Dev)	CPI-JADE (Mean Error ± Std Dev)
Unimodal Functions	$CEC2013_1$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
	$CEC2013_2$	1.20E+03 ± 1.89E+03–	9.42E+03 ± 7.67E+03–	3.37E+03 ± 3.39E+03–	0.00E+00 ± 0.00E+00
	$CEC2013_3$	8.19E+02 ± 3.82E+03–	7.15E+02 ± 2.91E+03–	1.55E+03 ± 8.06E+03–	2.33E+02 ± 1.21E+03
	$CEC2013_4$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
	$CEC2013_5$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
Basic Multimodal Functions	$CEC2013_6$	1.55E+00 ± 6.27E+00≈	3.79E+00 ± 5.24E+00–	2.66E+00 ± 7.92E+00–	1.03E+00 ± 5.17E+00
	$CEC2013_7$	2.33E+00 ± 2.44E+00≈	2.33E+00 ± 2.44E+00≈	2.34E+00 ± 3.44E+00≈	2.34E+00 ± 2.22E+00
	$CEC2013_8$	2.09E+01 ± 5.43E–02≈	2.09E+01 ± 4.24E–02≈	2.09E+01 ± 8.41E–02≈	2.09E+01 ± 4.92E–02
	$CEC2013_9$	2.59E+01 ± 1.57E+00≈	2.62E+01 ± 1.50E+00≈	2.58E+01 ± 1.86E+00≈	2.60E+01 ± 1.44E+00
	$CEC2013_{10}$	2.97E–02 ± 1.80E–02≈	2.33E–02 ± 1.25E–02+	3.05E–02 ± 1.71E–02≈	3.02E–02 ± 1.65E–02
	$CEC2013_{11}$	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00≈	0.00E+00 ± 0.00E+00
	$CEC2013_{12}$	2.35E+01 ± 3.76E+00≈	2.60E+01 ± 4.09E+00–	2.44E+01 ± 4.83E+00–	2.36E+01 ± 3.16E+00
	$CEC2013_{13}$	4.52E+01 ± 1.18E+01–	5.06E+01 ± 1.10E+01–	4.88E+01 ± 1.14E+01–	4.28E+01 ± 1.18E+01
	$CEC2013_{14}$	2.92E+00 ± 9.08E–01≈	5.74E+00 ± 1.67E+00–	5.25E+00 ± 1.40E+00–	2.95E+00 ± 8.42E–01
	$CEC2013_{15}$	3.23E+03 ± 3.12E+02≈	3.24E+03 ± 3.39E+02≈	3.19E+03 ± 3.79E+02≈	3.20E+03 ± 3.27E+02
	$CEC2013_{16}$	2.19E+00 ± 4.92E–01≈	1.71E+00 ± 7.23E–01+	1.87E+00 ± 5.87E–01+	2.21E+00 ± 4.99E–01
	$CEC2013_{17}$	3.04E+01 ± 1.53E–03≈	3.04E+01 ± 3.53E–03≈	3.04E+01 ± 4.97E–03≈	3.04E+01 ± 1.44E–03
	$CEC2013_{18}$	7.75E+01 ± 6.65E+00–	7.78E+01 ± 7.26E+00–	7.68E+01 ± 7.35E+00–	7.62E+01 ± 6.25E+00
	$CEC2013_{19}$	1.67E+00 ± 1.25E–01≈	1.75E+00 ± 1.35E–01–	1.77E+00 ± 1.42E–01–	1.65E+00 ± 1.30E–01
	$CEC2013_{20}$	1.03E+01 ± 4.92E–01≈	1.05E+01 ± 5.95E–01≈	1.04E+01 ± 4.90E–01≈	1.03E+01 ± 4.77E–01
Composition Functions	$CEC2013_{21}$	3.11E+02 ± 8.00E+01≈	3.21E+02 ± 8.41E+01–	2.93E+02 ± 7.72E+01+	3.08E+02 ± 8.64E+01
	$CEC2013_{22}$	1.17E+02 ± 1.30E+01≈	1.39E+02 ± 3.98E+01–	1.37E+02 ± 3.33E+01–	1.19E+02 ± 2.09E+01
	$CEC2013_{23}$	3.23E+03 ± 3.92E+02≈	3.30E+03 ± 3.92E+02–	3.34E+03 ± 3.90E+02–	3.20E+03 ± 4.23E+02
	$CEC2013_{24}$	2.08E+02 ± 6.83E+00≈	2.15E+02 ± 1.49E+01–	2.10E+02 ± 1.19E+01≈	2.09E+02 ± 1.18E+01
	$CEC2013_{25}$	2.59E+02 ± 1.19E+01≈	2.67E+02 ± 1.31E+01–	2.68E+02 ± 1.26E+01–	2.60E+02 ± 1.37E+01
	$CEC2013_{26}$	2.16E+02 ± 4.21E+01≈	2.19E+02 ± 4.67E+01–	2.26E+02 ± 5.45E+01–	2.15E+02 ± 3.99E+01
	$CEC2013_{27}$	4.82E+02 ± 2.02E+02≈	6.68E+02 ± 2.24E+02–	6.19E+02 ± 2.11E+02–	4.85E+02 ± 2.16E+02
	$CEC2013_{28}$	3.00E+02 ± 0.00E+00≈	3.00E+02 ± 0.00E+00≈	3.00E+02 ± 0.00E+00≈	3.00E+02 ± 0.00E+00
+		0	2	2	
–		4	15	13	
≈		24	11	13	

13, and 23 test functions, respectively. However, JADE, jDE, and SaDE surpass their augmented algorithms on at most seven test functions.

- CPI-JADE, CPI-jDE, and CPI-SaDE can produce better or similar results compared with their original algorithms on all the unimodal functions except that the performance of CPI-jDE is worse than that of jDE on $CEC2014_2$ with 50D as shown in Table 8. Moreover, CPI-JADE, CPI-jDE, and CPI-SaDE can achieve significant improvements on three unimodal functions from IEEE CEC2013 (i.e., $CEC2013_2$-$CEC2013_4$) and one unimodal function from IEEE CEC2014 (i.e., $CEC2014_1$).
- Compared with the original DE variants, the augmented algorithms can consistently provide the optimal solutions for 17 cases, which have been shown in **boldface** in Tables 5–8.
- CPI-JADE, CPI-jDE, and CPI-SaDE show similar superiority against their original algorithms when the dimension increases from 30 to 50. Moreover, the advantages of CPI-jDE and CPI-SaDE increase as the number of dimension increases for $CEC2013_1$-$CEC2013_{28}$ and $CEC2014_1$-$CEC2014_{30}$, respectively.

The above comparison suggests that the performance of JADE, jDE, and SaDE can be significantly refined under our framework, which again verifies the importance of utilizing the cumulative population distribution information in DE. Some convergence graphs for the performance comparison between the three state-of-the-art DE variants and their augmented algorithms have been given in Fig. 5.

5.3. Cumulative population distribution information vs single population distribution information in DE

The aim of this subsection is to investigate the performance difference between cumulative population distribution information and single population distribution information in DE. To this end, two groups of experiments have been implemented.

- In the first group of experiments, the population distribution information of the current generation has been utilized in our framework.
- In the second group of experiments, CPI-DE is compared with the methods in Refs. [11,12], which also exploit the distribution information of the current population in DE. In Ref. [11], all the individuals in the current population are used to compute the covariance matrix and the crossover is implemented in the Eigen coordinate system with a probability 0.5. In contrast, the method in Ref. [12] selects the best *ps***NP* individuals from the population to compute the covariance matrix and the crossover is executed in the Eigen coordinate system with a probability *pb*.

JADE [6] is considered as the instance algorithm and $CEC2013_1$-$CEC2013_{28}$ with 30D are adopted in the experimental study. The compared method in the first group of experiments is denoted as SPI-JADE, and the compared methods in the second group of experiments are denoted as JADE/eig and CoJADE. The source codes of JADE/eig and CoJADE were obtained from the authors of Refs. [11,12], respectively. It is necessary to emphasize that SPI-JADE, JADE/eig, and CoJADE only exploit the distribution information from a single population of one generation to update the covariance matrix, compared with CPI-JADE.

Table 9 reports the experimental results, where "+", "–", and "≈" denote that the performance of the corresponding algorithm is better than, worse than, and similar to that of CPI-JADE, respectively. As shown in this table, CPI-JADE demonstrates either similar or better performance against SPI-JADE on all the test functions and SPI-JADE cannot beat CPI-JADE even on one test function. In addition, CPI-JADE performs better than JADE/eig and CoJADE on 15 and 13 test functions, respectively. Whereas, JADE/eig and CoJADE surpass CPI-JADE on only two test functions. The above comparison suggests that cumulative population distribution information is able to provide a more reasonable estimation of the covariance matrix compared with single population distribution information.

On the other hand, it can be seen from Table 9 that the overall performance of CPI-JADE and SPI-JADE is better than that of JADE/eig and CoJADE. In CPI-JADE and SPI-JADE, the crossover is implemented in a deterministic manner, i.e., two trial vectors are generated for each target vector by the crossover in both the original coordinate system and the Eigen coordinate system. However, JADE/eig and CoJADE implement the crossover via a random manner, i.e., one trial vector is created for each target vector by the crossover in either the original coordinate system or the Eigen coordinate system. Clearly, the above comparison supports the deterministic manner, by which not only could the superiority and the search behavior of the original DE be kept, but also the modality of the fitness landscape could be identified by utilizing the cumulative population distribution information in DE. Moreover, a problem-dependent parameter in JADE/eig and CoJADE, i.e., the rate at which the crossover is implemented in the Eigen coordinate system, has been eliminated in CPI-JADE and SPI-JADE through the deterministic manner.

5.4. Crossover in the Eigen coordinate system vs crossover in the original coordinate system

We have introduced two kinds of crossover in DE, i.e., the crossover in the Eigen coordinate system and the crossover in the original coordinate system. One may be interested in the roles of these two kinds of crossover. At each generation, for each target vector if the trial vector generated by the crossover in the Eigen coordinate system is better than the trial vector generated by the crossover in the original coordinate system, then it is called a successful crossover in the Eigen coordinate system. Otherwise, it is called a successful crossover in the original coordinate system. With the termination of the gth generation, there are $g * NP$ trial vectors generated by the crossover in the Eigen coordinate system and by the crossover in the original coordinate system, respectively. Suppose that the number of successful crossover in the Eigen coordinate system is NS. Thus, the number of successful crossover in the original coordinate system is $(g * NP - NS)$. Then, the success rates of the crossover in the Eigen coordinate system and in the original coordinate system are computed as $NS/(g * NP)$ and $1 - NS/(g * NP)$, respectively. In this subsection, the change of such success rates is monitored during the evolution.

In the experiments, JADE is still chosen as the instance algorithm. In addition, three test functions with 30D from IEEE CEC2014 are used to produce the experimental results: the unimodal function CEC2014_3, the simple multimodal function CEC2014_6, and the hybrid function CEC2014_{17}. Since CEC2014_3, CEC2014_6, and CEC2014_{17} represent different kinds of test functions, it is expected to carry out a multi-facet investigation on the roles of the crossover in the Eigen coordinate system and the crossover in the original coordinate system. Fig. 6 depicts the evolution of the success rates of these two kinds of crossover. Note that the success rates are initialized to 0.5 for both of them.

From Fig. 6, two interesting phenomena can be observed:

- The relationship between the success rates of these two kinds of crossover can be divided into three types. In the first type, the success rate of the crossover in the Eigen coordinate system is consistently higher than that of the crossover in the original coordinate system throughout the whole evolutionary process (as shown in Fig. 6(b)). Nevertheless, the second type is contrary to the first type (as shown in Fig. 6(c)). For the third type, the success rate of the crossover in the Eigen coordinate system is lower at the early stage of the evolution yet higher at the later stage of the evolution, compared with that of the crossover in the original coordinate system (as shown in Fig. 6(a)). The occurrence of the above three types of relationship is dependent on the characteristics of test functions, such as the number and size of the basin of attraction. For example, for the unimodal function (such as CEC2014_3), there is only one basin of attraction. In the later stage of evolution, the search will be implemented around this basin of attraction. As a result, the modality of the fitness landscape can be well identified by utilizing the cumulative population distribution information, which leads to a higher success rate of the crossover in the Eigen coordinate system.
- During the evolution, the success rates of these two kinds of crossover exhibit opposite tendencies. That is, the success rate of the crossover in the Eigen coordinate system gradually increases while the success rate of the crossover in the original coordinate system gradually decreases. It is not difficult to understand since more and more information of the population will be exploited to update the covariance matrix over the course of evolution, and thus, the capability of the covariance matrix associated with cumulative population distribution information identifying the modality of the fitness landscape will be strengthened gradually. Consequently, the crossover in the Eigen coordinate system

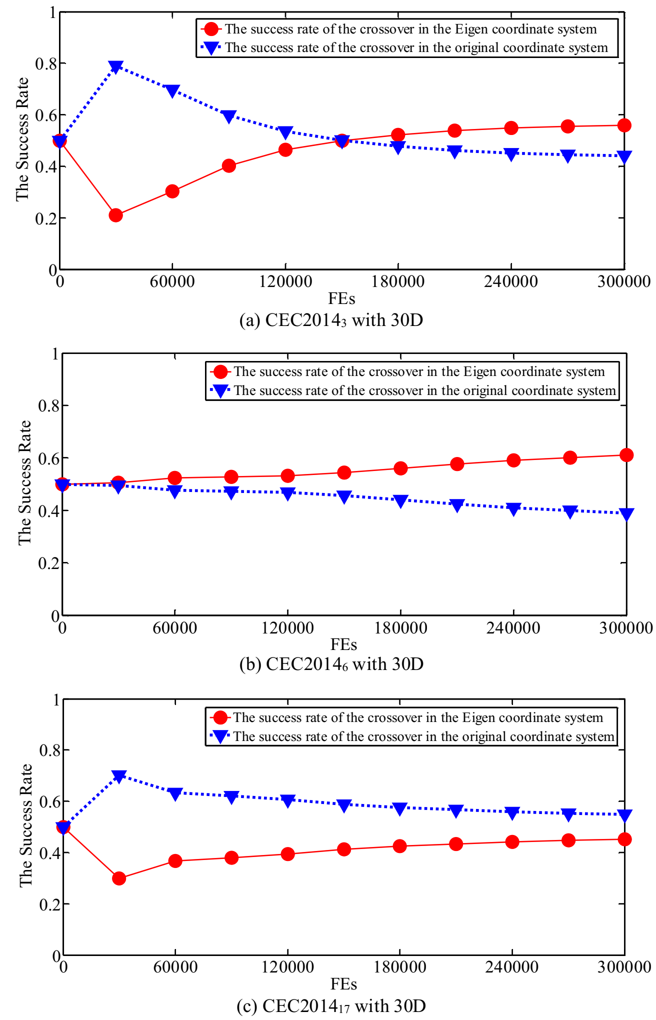

Fig. 6. The success rates of the crossover in the Eigen coordinate system and the crossover in the original coordinate system during the evolution.

becomes more and more powerful to guide the population toward the promising areas.

Based on the above discussion, we can give the following comments:

- Since neither of these two kinds of crossover can consistently provide better success rate on different kinds of test functions, both of them are very important for our framework.
- It seems that the crossover in the Eigen coordinate system plays a more and more important role along with the evolution.

6. Conclusion

A simple yet efficient DE framework, which is referred as CPI-DE, has been presented in this paper. In CPI-DE, the cumulative population distribution information is utilized to establish an Eigen coordinate system for DE's crossover. Moreover, CPI-DE performs the crossover in both the original coordinate system and the Eigen coordinate system in a deterministic manner. As a result, two trial vectors are generated for each target vector and the best one among the target vector and two trial vectors will survive into the next generation. CPI-DE does not add any extra parameters into DE.

CPI-DE has been applied to two classic DE versions and three state-of-the-art DE variants. Experiments across two benchmark test sets from IEEE CEC2013 and IEEE CEC2014 show that CPI-DE has the capability to significantly enhance the performance of DE. Moreover, the effectiveness of utilizing cumulative population distribution information in DE has been demonstrated by experiments, and the roles of the crossover in the Eigen coordinate system and the crossover in the original coordinate system have been analyzed. In the future, we will propose some advanced DE variants based on our framework.

The Matlab source code of CPI-DE can be downloaded from Y. Wang's homepage: http://ist.csu.edu.cn/YongWang.htm.

Acknowledgments

The authors would like to thank the anonymous reviewers for their very constructive and helpful suggestions. This work was supported in part by the National Basic Research Program 973 of China (Grant No. 2011CB013104), in part by the Innovation-driven Plan in Central South University (No. 2015CXS012 and No. 2015CX007), in part by the National Natural Science Foundation of China under Grant 61273314, in part by the EU Horizon 2020 Marie Sklodowska-Curie Individual Fellowships (Project ID: 661327), in part by the Hunan Provincial Natural Science Fund for Distinguished Young Scholars (Grant No. 2016JJ1018), in part by the Program for New Century Excellent Talents in University under Grant NCET-13-0596, and in part by State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology.

References

[1] R. Storn, K. Price, Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces, Tech. Rep. TR-95-012, Berkeley, CA, 1995.
[2] R. Storn, K.V. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[3] Y. Wang, B.-C. Wang, H.-X. Li, G.G. Yen, Incorporating objective function information into the feasibility rule for constrained evolutionary optimization, IEEE Trans. Cybern. (in press). http://dx.doi.org/10.1109/tcyb.2015.2493239.
[4] Y. Wang, H.-X. Li, G.G. Yen, W. Song, MOMMOP: multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems, IEEE Trans. Cybern. 45 (4) (2015) 830–843.
[5] G. Jia, Y. Wang, Z. Cai, Y. Jin, An improved (μ+λ)-constrained differential evolution for constrained optimization, Inf. Sci. 222 (2013) 302–322.
[6] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.
[7] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657.
[8] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.
[9] R. Mallipeddi, P.N. Suganthan, Q.K. Pan, M.F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2) (2011) 1679–1696.
[10] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (1) (2011) 55–66.
[11] S. Guo, C. Yang, Enhancing differential evolution utilizing Eigenvector-based crossover operator, IEEE Trans. Evol. Comput. 19 (1) (2015) 31–49.
[12] Y. Wang, H.-X. Li, T. Huang, L. Li, Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, Appl. Soft Comput. 18 (2014) 232–247.
[13] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2) (2001) 159–195.
[14] J.J. Liang, B.Y. Qu, P.N. Suganthan, A.G. Hernandez-Diaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-parameter Optimization, Zhengzhou Univ., China, and Nanyang Technol. Univ., Singapore, 2013, Tech. Rep. 201212 (January).
[15] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-parameter Numerical Optimization, Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, 2013 (December).
[16] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.
[17] Y. Zhou, X. Li, L. Gao, A differential evolution algorithm with intersect mutation operator, Appl. Soft Comput. 13 (1) (2013) 390–401.
[18] X. Zhang, S.Y. Yuen, A directional mutation operator for differential evolution algorithms, Appl. Soft Comput. 30 (2015) 529–548.
[19] Z. Hu, S. Xiong, X. Wang, Q. Su, M. Liu, Z. Chen, Subspace clustering mutation operator for developing convergent differential evolution algorithm, Math. Probl. Eng. 2014 (2014) 18 (Article ID 154626).
[20] W. Gong, Z. Cai, Differential evolution with ranking-based mutation operators, IEEE Trans. Cybern. 43 (6) (2013) 2066–2081.
[21] Y. Cai, J. Wang, Differential evolution with neighborhood and direction information for numerical optimization, IEEE Trans. Cybern. 43 (6) (2013) 2202–2215.
[22] J. Wang, J. Liao, Y. Zhou, Y. Cai, Differential evolution enhanced with multiobjective sorting-based mutation operators, IEEE Trans. Cybern. 44 (12) (2014) 2792–2805.
[23] S. Guo, C.C. Yang, P.H. Hsu, J.S. Tsai, Improving differential evolution with successful-parent-selecting framework, IEEE Trans. Evol. Comput. 19 (5) (2015) 717–730.
[24] W.J. Yu, J.J. Li, J. Zhang, M. Wan, Differential evolution using mutation strategy with adaptive greediness degree control, Proc. Genet. Evol. Comput. Conf. (GECCO) (2014) 73–80.
[25] H. Wang, S. Rahnamayan, H. Sun, M.G. Omran, Gaussian bare-bones differential evolution, IEEE Trans. Cybern. 43 (2) (2013) 634–647.
[26] S.Z. Zhao, P.N. Suganthan, Empirical investigations into the exponential crossover of differential evolutions, Swarm Evol. Comput. 9 (2013) 27–36.
[27] Y. Wang, Z. Cai, Q. Zhang, Enhancing the search ability of differential evolution through orthogonal crossover, Inf. Sci. 185 (1) (2012) 153–177.
[28] W. Gong, Z. Cai, Y. Wang, Repairing the crossover rate in adaptive differential evolution, Appl. Soft Comput. 15 (2014) 149–168.
[29] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, Proc. Congr. Evol. Comput. (CEC) (2013) 71–78.
[30] R. Tanabe, A.S. Fukunaga, Improving the search performance of shade using linear population size reduction, Proc. Congr. Evol. Comput. (CEC) (2014) 1658–1665.
[31] T. Takahama, S. Sakai, An adaptive differential evolution considering correlation of two algorithm parameters, Proceedings of the 7th International Conference on Soft Computing and Intelligent Systems (SCIS), and 15th International Symposium on Advanced Intelligent Systems (ISIS) (2014) 618–623.
[32] J. Aalto, J. Lampinen, A mutation and crossover adaptation mechanism for differential evolution algorithm, Proc. Congr. Evol. Comput. (CEC) (2014) 451–458.
[33] R. Sarker, S. Elsayed, T. Ray, Differential evolution with dynamic parameters selection for optimization problems, IEEE Trans. Evol. Comput. 18 (5) (2014) 689–707.
[34] R.J. He, Z.Y. Yang, Differential evolution with adaptive mutation and parameter control using lévy probability distribution, J. Comput. Sci. Technol. 27 (5) (2012) 1035–1055.
[35] W.J. Yu, J. Zhang, Adaptive differential evolution with optimization state estimation, Proc. Genet. Evol. Comput. Conf. (GECCO) (2012) 1285–1292.
[36] W. Zhu, Y. Tang, J. Fang, W. Zhang, Adaptive population tuning scheme for differential evolution, Inf. Sci. 223 (2013) 164–191.

[37] A. Zamuda, J. Brest, E. Mezura-Montes, Structured population size reduction differential evolution with multiple mutation strategies on CEC 2013 real parameter optimization, Proc. Congr. Evol. Comput. (CEC) (2013) 1925–1931.
[38] C. Segura, C.A. Coello Coello, E. Segredo, C. León, An analysis of the automatic adaptation of the crossover rate in differential evolution, Proc. Congr. Evol. Comput. (CEC) (2014) 459–466.
[39] C. Segura, C.A. Coello Coello, E. Segredo, C. León, On the adaptation of the mutation scale factor in differential evolution, Optim. Lett. 9 (1) (2015) 189–198.
[40] L. Tang, Y. Dong, J. Liu, Differential evolution with an individual-dependent mechanism, IEEE Trans. Evol. Comput. 19 (4) (2015) 560–574.
[41] W. Yi, L. Gao, X. Li, Y. Zhou, A new differential evolution algorithm with a hybrid mutation operator and self-adapting control parameters for global optimization problems, Appl. Intell. 42 (4) (2015) 642–660.
[42] Q. Fan, X. Yan, Self-adaptive differential evolution algorithm with discrete mutation control parameters, Expert Syst. Appl. 42 (3) (2015) 1551–1572.
[43] Q. Fan, X. Yan, Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies, IEEE Trans. Cybern. 46 (1) (2016) 219–232.
[44] X. Li, M. Yin, Modified differential evolution with self-adaptive parameters method, J. Comb. Optim. 31 (2) (2016) 546–576.
[45] T. Takahama, S. Sakai, Differential evolution with dynamic strategy and parameter selection by detecting landscape modality, Proc. Congr. Evol. Comput. (CEC) (2012) 1–8.
[46] X. Zhou, Z. Wu, H. Wang, S. Rahnamayan, Enhancing differential evolution with role assignment scheme, Soft Comput. 18 (11) (2014) 2209–2225.
[47] P. Bujok, J. Tvrdík, Parallel migration model employing various adaptive variants of differential evolution, Lect. Notes Comput. Sci. 7269 (2012) 39–47.
[48] C.L. Huo, Y.S. Lien, Y.H. Yu, T.Y. Sun, Effectively multi-swarm sharing management for differential evolution, Proc. Congr. Evol. Comput. (CEC) (2012) 1–8.
[49] J.I. Kushida, A. Hara, T. Takahama, A. Kido, Island-based differential evolution with varying subpopulation size, Proceedings of the Sixth International Workshop on Computational Intelligence & Applications (IWCIA) (2013) 119–124.
[50] Y. Zhou, X. Li, L. Gao, A novel two-layer hierarchical differential evolution algorithm for global optimization, Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC) (2013) 2916–2921.
[51] S.S. Thangavelu, C.S. Velayutham, Combining different differential evolution variants in an island based distributed framework-an investigation, Adv. Intell. Syst. Comput. 320 (2015) 593–605.
[52] J. Cheng, G. Zhang, F. Neri, Enhancing distributed differential evolution with multicultural migration for global numerical optimization, Inf. Sci. 247 (2013) 72–93.
[53] H. Peng, Z. Wu, Heterozygous differential evolution with Taguchi local search, Soft Comput. 19 (11) (2015) 3273–3291.
[54] J. Li, A hybrid differential evolution algorithm with opposition-based learning, Proceedings of the 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC) vol. 1 (2012) 85–89.
[55] H. Liu, Z. Wu, H. Wang, S. Rahnamayan, C. Deng, Improved differential evolution with adaptive opposition strategy, Proc. Congr. Evol. Comput. (CEC) (2014) 1776–1783.
[56] R. Polakova, J. Tvrdik, P. Bujok, Controlled restart in differential evolution applied to CEC 2014 benchmark functions, Proc. Congr. Evol. Comput. (CEC) (2014) 2230–2236.
[57] D. Meng, An improved differential evolution with adaptive disturbance for numerical optimization, Adv. Inf. Sci. Serv. Sci. 5 (1) (2013), http://dx.doi.org/10.4156/AISS.vol5.issue1.3.
[58] M. Yang, C. Li, Z. Cai, J. Guan, Differential evolution with auto-enhanced population diversity, IEEE Trans. Cybern. 45 (2) (2015) 302–315.
[59] Y.L. Li, Z.H. Zhan, Y.J. Gong, W.N. Chen, J. Zhang, Y. Li, Differential evolution with an evolution path: a deep evolutionary algorithm, IEEE Trans. Cybern. 45 (9) (2015) 1798–1810.
[60] X. Lu, K. Tang, X. Yao, Classification- and regression-assisted differential evolution for computationally expensive problems, J. Comput. Sci. Technol. 27 (5) (2012) 1024–1034.
[61] X. Lu, K. Tang, B. Sendhoff, X. Yao, A new self-adaptation scheme for differential evolution, Neurocomputing 146 (2014) 2–16.
[62] R. Mallipeddi, M. Lee, Surrogate model assisted ensemble differential evolution algorithm, Proc. Congr. Evol. Comput. (CEC) (2012) 1–8.
[63] W. Xiang, S. Ma, M. An, hABCDE: a hybrid evolutionary algorithm based on artificial bee colony algorithm and differential evolution, Appl. Math. Comput. 238 (2014) 370–386.
[64] D. Kovačević, N. Mladenović, B. Petrović, P. Milošević, DE-VNS: self-adaptive differential evolution with crossover neighborhood search for continuous global optimization, Comput. Oper. Res. 52 (2014) 157–169.
[65] H. Guo, Y. Li, J. Li, H. Sun, D. Wang, X. Chen, Differential evolution improved with self-adaptive control parameters based on simulated annealing, Swarm Evol. Comput. 19 (2014) 52–67.
[66] W. Shao, L. Shang, L. Ma, Z. Shao, X. Ying, Hybrid differential evolution/estimation of distribution algorithm based on adaptive incremental learning, J. Comput. Inf. Syst. 10 (12) (2014) 5355–5364.
[67] P. Roy, M.J. Islam, M.M. Islam, Self-adaptive genetically programmed differential evolution, Proceedings of the 7th International Conference on Electrical & Computer Engineering (ICECE) (2012) 639–642.
[68] P. Musigawan, S. Chiewchanwattana, K. Sunat, Improved differential evolution via cuckoo search operator, Lect. Notes Comput. Sci. 7663 (2012) 465–472.
[69] A.K. Qin, K. Tang, H. Pan, S. Xia, Self-adaptive differential evolution with local search chains for real-parameter single-objective optimization, Proc. Congr. Evol. Comput. (CEC) (2014) 467–474.
[70] A.P. Piotrowski, Adaptive memetic differential evolution with global and local neighborhood-based mutation operator, Inf. Sci. 241 (2013) 164–194.
[71] P. Larranaga, J.A. Lozano, E. Bengoetxea, Estimation of Distribution Algorithms Based on Multivariate Normal and Gaussian Networks, Technical Report, Dept. of Computer Science and Artificial Intelligence University of the Basque Country, 2001.
[72] N. Hansen, The CMA Evolution Strategy: A Tutorial, 2011 https://www.lri.fr/~hansen/.