Soft Comput
DOI 10.1007/s00500-015-1682-9

METHODOLOGIES AND APPLICATION

Cellular direction information based differential evolution for numerical optimization: an empirical study

Jingliang Liao[1] · Yiqiao Cai[1] · Tian Wang[1] · Hui Tian[1] · Yonghong Chen[1]

Abstract Differential evolution (DE) is a well-known evolutionary algorithm which has been successfully applied in many scientific and engineering fields. In most DE algorithms, the base and difference vectors for mutation are randomly selected from the current population. That is, the useful population information cannot be fully exploited to guide the search of DE through mutation. Furthermore, the selection of parents in mutation has been verified to be critical for the DE performance. Therefore, to alleviate this drawback and improve the performance of DE, a novel DE algorithm with a directional mutation based on cellular topology is proposed in this study. This proposed algorithm is named as Cellular Direction Information based DE (DE-CDI). In DE-CDI, the cellular topology is employed first to define a neighborhood for each individual of population and then the direction information based on the neighborhood is incorporated into the mutation operator. In this way, DE-CDI not only utilizes the neighborhood information to exploit the regions of better individuals and accelerate convergence, but also introduces the direction information to guide the search to the promising area. To evaluate the performance of the proposed method, DE-CDI is applied to the original DE algorithms, as well as several advanced DE variants. Experimental results demonstrate the high performance of DE-CDI for most DE algorithms studied.

Communicated by V. Loia.

✉ Yiqiao Cai
yiqiao00@163.com; caiyq@hqu.edu.cn

1 College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China

Keywords Differential evolution · Cellular topology · Neighborhood information · Direction information · Mutation operator · Numerical optimization

1 Introduction

Differential evolution (DE), proposed by Storn and Price (Storn and Price 1997; Storn et al. 2005), is a simple yet efficient evolutionary algorithm for global numerical optimization. It has many attractive characteristics, such as ease to use, compact structure, robustness and speediness. Recently, DE has been extended to handle large-scale, multi-objective, constrained, dynamic and uncertain optimization problems (Das and Suganthan 2011). Furthermore, DE has been successfully applied to diverse domains of science and engineering, such as pattern recognition (Das et al. 2008; Omran et al. 2005), chemical engineering (Wang and Jang 2000; Lampinen 1999), engineering design (Joshi and Sanderson 1999; Rogalsky et al. 1999), signal processing (Das and Konar 2006), satellite communications (Wang and Cai 2015) and so on.

In DE, two main factors significantly influence the performance of DE: the control parameters (i.e., population size *NP*, scaling factor *F* and crossover rate *Cr*) and the evolutionary operators (i.e., mutation, crossover and selection). During the past decade, there are many enhanced DE variants proposed in the literature. According to Neri and Tirronen (2010), these advanced DE variants can be divided into two categories: algorithms with additional components and algorithms with a modified DE structure. In these DE variants, modifications mostly focus on devising the new mutation operators (Zhang and Sanderson 2009; Das et al. 2009; Wang et al. 2014), employing the self-adaptive strategies for control parameters (Qin et al. 2009; Brest et al. 2006; Yang et al.

Published online: 19 April 2015

2015), proposing the ensemble strategies (Qin et al. 2009; Wang et al. 2011; Tang et al. 2015), developing the hybrid DE with other optimization methods (Sun et al. 2005), etc.

In DE, the mutant vector, generated by the mutation operator, can be treated as a lead individual to explore the search space and is constructed by adding a difference vector to a base vector. However, these two vectors (i.e., the base and difference vectors) in most DE variants are always randomly selected from the current population. That is, the useful population information cannot be fully utilized to guide the search through mutation.

In order to alleviate this drawback and enhance the performance of DE, a new DE algorithm with a directional mutation based on cellular topology is proposed in this study, which is named as Cellular Direction Information based DE (DE-CDI). In DE-CDI, a cellular topology is employed first to define a neighborhood for each individual of population. Then, the neighbors of each individual are divided into the superior and inferior groups according to their fitness. Finally, the direction information is incorporated into mutation by selecting two vectors from the superior and inferior groups, respectively. Here, all the parents for mutation are selected from the neighbors of the current vector. In this way, DE-CDI not only utilizes the information of neighboring individuals to exploit the regions of minima and accelerate convergence, but also incorporates the direction information of population to guide the search to the promising area. Therefore, the population information composed by neighborhood and direction information can be simultaneously and fully utilized in DE-CDI to guide the search.

To evaluate the effectiveness of the proposed method, DE-CDI is applied to six original DE algorithms, as well as several advanced DE variants. With the analysis of the extensive experiments on a set of benchmark functions, we can clearly find that DE-CDI is able to improve the performance of most DE algorithms studied.

The main contributions of this study include the following:

- Both neighborhood and direction information are fully and simultaneously utilized in selecting parents for mutation to guide the search of DE.
- DE-CDI provides a simple yet powerful approach for improving the search ability of DE. In addition, DE-CDI can be easily applied to other DE variants.
- The extensive experiments are carried out to show the effectiveness of DE-CDI. The results demonstrate that CDI is able to enhance the performance of most DE algorithms studied.

The rest of this paper is organized as follows: In Sect. 2, the original DE is introduced. Section 3 briefly reviews some related work. The proposed DE-CDI is presented in detail in Sect. 4. In Sect. 5, experimental results are reported. Finally, the conclusions are drawn in Sect. 6.

2 DE

In this study, DE is for solving the numerical optimization problem (Storn and Price 1997). Without loss of generality, the optimization problem which is considered to be minimized is $f(X)$, $X \in R^D$, and D is the dimension of the decision variables. DE evolves a population of vectors representing the candidate solutions. Every individual is denoted as $X_{i,G} = (x^1_{i,G}, x^2_{i,G}, \ldots, x^D_{i,G})$, where $i = 1, 2, \ldots, NP$, NP is the population size and G is the current generation.

The DE algorithmic schemes can be classified using the notation "DE/x/y/z", where DE means differential evolution algorithm, x means the method of selecting the parent that constitutes the base vector, y means the number of difference vectors that are used to perturb x and z stands for the crossover type employed.

2.1 Initialization

In DE, the initial population should cover the entire search space as much as possible by uniformly randomizing individuals within the search space. Here, the jth parameter in the ith individual is initialized as follows:

$$x^j_{i,G} = L_j + \text{rand}(0, 1) \times (U_j - L_j), \tag{1}$$

where rand$(0, 1)$ represents a uniformly distributed random number within the range [0, 1], and L_j and U_j represent the lower and upper bounds of the jth variable, respectively.

2.2 Mutation

DE employs the mutation strategy to generate a mutant vector $V_{i,G}$ with respect to each individual $X_{i,G}$. Here, several widely used mutation strategies are listed as follows:

- DE/rand/1

$$V_{i,G} = X_{r1,G} + F \times (X_{r2,G} - X_{r3,G}) \tag{2}$$

- DE/rand/2

$$V_{i,G} = X_{r1,G} + F \times (X_{r2,G} - X_{r3,G}) + F \times (X_{r4,G} - X_{r5,G}) \tag{3}$$

- DE/best/1

$$V_{i,G} = X_{\text{best},G} + F \times (X_{r1,G} - X_{r2,G}) \tag{4}$$

- DE/best/2

$$V_{i,G} = X_{\text{best},G} + F \times (X_{r1,G} - X_{r2,G}) + F \times (X_{r3,G} - X_{r4,G}) \quad (5)$$

- DE/current-to-best/1

$$V_{i,G} = X_{i,G} + F \times (X_{\text{best},G} - X_{i,G}) + F \times (X_{r1,G} - X_{r2,G}) \quad (6)$$

- DE/rand-to-best/1

$$V_{i,G} = X_{r1,G} + F \times (X_{\text{best},G} - X_{r1,G}) + F \times (X_{r2,G} - X_{r3,G}), \quad (7)$$

where F is called the mutation scaling factor, and the indices $r1$, $r2$, $r3$, $r4$ and $r5$ are mutually exclusive integers randomly generated within the range $[1, NP]$, which are also different from the index i. $X_{\text{best},G}$ is the best individual vector at generation G. More details of them can be found in Storn and Price (1997), Storn et al. (2005) and Das and Suganthan (2011).

2.3 Crossover

The crossover operator is applied to each pair of $X_{i,G}$ and $V_{i,G}$ to generate a trial vector $U_{i,G}$. There are two kinds of crossover scheme: binomial and exponential (Storn and Price 1997; Storn et al. 2005). The binomial crossover, as it is widely used, is outlined as follows:

$$u_{i,G}^{j} = \begin{cases} v_{i,G}^{j} & \text{if } \text{rand}(0,1) \leq Cr \text{ or } j = j_{\text{rand}} \\ x_{i,G}^{j} & \text{otherwise,} \end{cases} \quad (8)$$

where $Cr \in [0, 1]$ is called the crossover rate. j_{rand} is a randomly chosen integer in the range $[1, D]$. If $u_{i,G}^{j}$ is out of the boundary, it will be reinitialized within the range $[L_j, U_j]$.

2.4 Selection

The selection operator selects the better one from each pair of $X_{i,G}$ and $U_{i,G}$ according to their fitness values for the next generation. The selection operator is given as follows:

$$X_{i,G} = \begin{cases} U_{i,G} & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \quad (9)$$

3 Related work

This section briefly reviews some recently improved DE variants that utilize the population information, especially neighborhood and direction information, in the mutation operators.

The neighborhood concepts are usually used to enhance the performance of DE by the following ways: offspring generation operator, structured population and local search (Epitropakis et al. 2011). There are two main types of neighborhood information: index-based and distance-based. For the index-based neighborhood, the neighbors of each individual do not necessary lie in the vicinity of its topological region in the search space. Many DE variants utilize the index-based neighborhood information with the structured population. In these DE variants, the individuals for mutation are selected according to a neighbor list constructed from the population topologies. Two main canonical kinds of structured population in DE is employed in literature, i.e., cellular DE (cDE) (Noman and Iba 2011; Noroozi et al. 2011; Dorronsoro and Bouvry 2010) and distributed DE (dDE) (Weber et al. 2011, 2010; Neri et al. 2011). In De Falco et al. (2014), the impact of several network topologies on the performance of distributed differential evolution was investigated. By using the ring topology, a neighborhood-based mutation operator was proposed in Das et al. (2009). In Omran et al. (2006), the self-adaptive DE was enhanced using a ring neighborhood topology. In Omran et al. (2009), by employing the concept of index neighborhoods, the bare bones DE algorithm was proposed. To balance the exploration and exploitation, an adaptive memetic DE was proposed (Piotrowski 2013), where the ring topology was used to construct the neighborhood of individuals for local model. In Hu et al. (2014), the ring neighborhood topology was employed to improve a memetic DE algorithm. Recently, several population topologies, e.g. distributed, cellular, ring, small-world, have been introduced in DE to improve its performance (Dorronsoro and Bouvry 2011). For the distance-based neighborhood, the individuals are regarded as the neighbors of an individual when they locate in the vicinity of its topological region in the search space. In Epitropakis et al. (2011), a proximity-based DE framework (ProDE) was proposed using an affinity matrix based on the Euclidean distance to select the individuals for mutation. For improving the performance of DE, the learning-enhanced DE (LeDE) was proposed in Cai et al. (2012). In LeDE, the neighbors of each individual were defined based on the identified clusters. In Liang et al. (2014), a novel DE variant that employed a modified Fitness Euclidean-distance Ratio technique was proposed. In Sarkar et al. (2015), an effective new grouping strategy namely adaptive clustering and reclustering was proposed based on Fuzzy c-means clustering technique and was integrated with a hybrid of crowding niching technique and DE. To enhance the performance on multi-modal problems, an ensemble crowding DE with neighborhood mutation was proposed (Hui and Suganthan 2013).

Due to that difference vectors in mutation of DE are always constructed in a random manner, the direction concept is incorporated into many DE variants to enhance the exploration ability. In Wang and Xiang (2008), a new mutation strategy, named as DE/rand/±mean, was proposed. In this strategy, the population is partitioned into two groups and the difference vector is constructed by randomly selecting the vectors from the two groups respectively. In Fan and Lampinen (2003), a trigonometric mutation DE (TDE) was proposed with a probabilistic triangle mutation strategy that incorporates the direction information into DE. Recently, a novel DE framework, DE with neighborhood and direction information (NDi-DE), was proposed by designing three types of direction information for mutation to guide the search (Cai and Wang 2013). After that, adaptive operator selection mechanism was introduced into NDi-DE for different mutation strategies to dynamically balance the exploration and exploitation (Cai et al. 2015). In Bi and Xiao (2011), using the direction information with the current best solution and the best previous solution of each individual, a classification-based self-adaptive DE was proposed. In Iorio and Li (2006) and Liu et al. (2009), the direction information was also employed to enhance the performance of DE for the multi-objective optimization. In Zhang and Yuen (2015), with a difference vector pool containing good direction information, a directional mutation DE (DMDE) was proposed to speed up the convergence of DE. Based a distributed topology, a directional strategy was employed to enhance convergence speed by using the direction information in Gou et al. (2015).

4 DE-CDI

In this section, the proposed algorithm, i.e., DE-CDI, is described in detail. First, the motivations of this study are given. Second, two main components of CDI based mutation, i.e., cellular topology-based neighborhood and neighborhood-based directional mutation, are presented. Third, the complete proposed framework of DE-CDI is shown. Finally, the complexity of DE-CDI is discussed.

4.1 Motivations

As mentioned in Sect. 3, many attempts with neighborhood or/and direction information of population are effective to guide the search of DE. Furthermore, the selection of parents in mutation has been verified to be critical for the DE performance (Noman and Iba 2011; Noroozi et al. 2011; Dorronsoro and Bouvry 2010; Weber et al. 2011, 2010; Neri et al. 2011; Das et al. 2009; Omran et al. 2006, 2009; Cai et al. 2012; Wang and Xiang 2008; Fan and Lampinen 2003; Cai and Wang 2013; Bi and Xiao 2011). However, in most DE algorithms, on the one hand, the base and difference vectors are always selected randomly or locally. That it, both neighborhood and direction information are not simultaneously and fully utilized during the evolutionary process of DE. On the other hand, the individuals are often guided only with the best individual of population, even if the best individual is far away from the global optimum. By this way, the individuals may be trapped in local optimum more frequently, especially for the multimodal problems.

Based on these considerations, a directional mutation based on cellular topology is introduced into the mutation operator of DE to exploit the neighborhood and direction information of population simultaneously. Thus, a simple and effective framework, DE-CDI, is proposed. The primary idea of DE-CDI is to incorporate cellular topology and direction information into mutation to enhance the performance of DE. In CDI-based mutation, there are two main components: cellular topology-based neighborhood and neighborhood-based directional mutation.

4.2 Cellular topology-based neighborhood

In order to define a neighborhood for each individual, the cellular topology is introduced into the population. In the neighborhood-based on cellular topology, the population is spatially structured in a two-dimensional toroidal grid. Each grid-point contains exactly one individual, and the neighborhood of an individual is defined by the surrounding grid-points (Noman and Iba 2011). Here, the cellular topology-based neighborhood is called Cn, where n indicates the number of total individuals in the neighborhood (including the current individual itself and its closest neighbors). Figure 1 illustrates the concept with C9.

When employing Cn in DE-CDI, the population is structured in a regular grid of two dimensions and each individual can only interact with its surrounding neighboring solutions during the evolutionary process. As shown in Fig. 1, the individual at the center position in the grid (i.e., X_i) can generate an offspring only with its eight closest neighbors (i.e., the solid circles). That is, the interactions among individuals are

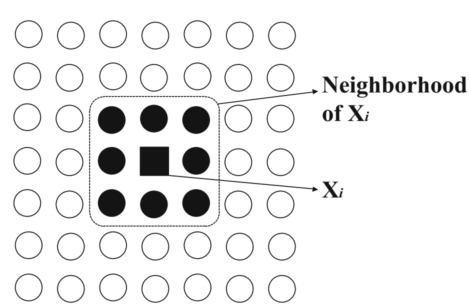

Fig. 1 C9

restricted to the closest ones, which will lead to slow diffusion of information throughout the population.

In this way, the neighborhood information based on cellular topology will be exploited fully to guide the search of DE-CDI through selecting parents from the neighbors. Although various neighborhood topologies, e.g., distributed, ring or small-world topology, are proposed for DE (Dorronsoro and Bouvry 2011), the cellular topology can obtain the more attractive and robust results in our preliminary tests. In the future work, the detailed comparisons of the DE-CDI variants with other neighborhood topologies will be made.

4.3 Neighborhood based directional mutation

Based on the defined neighborhood with cellular topology, direction information is introduced into mutation of DE by selecting neighbors to construct difference vectors. The new proposed mutation operator is named as neighborhood-based directional mutation (NDM). In NDM, all the parents in mutation are selected from the neighbors and the difference vectors are constructed by directing at a better neighbor from a worse one. Concretely, according to the fitness of the base vector, all of the neighbors are partitioned into the superior and inferior groups, and the difference vectors are constructed by randomly selecting two neighbors from the superior and inferior groups respectively.

In order to illustrate how NDM works, a simple example for DE/rand/1 with direction information based on C9 is shown in Fig. 2. As illustrated in Fig. 2, the base vector, $X_{r1,G}$, is randomly selected from the neighbors of $X_{i,G}$ first. Then, based on the fitness of $X_{r1,G}$, all the eight neighbors of $X_{i,G}$ are partitioned into the superior and inferior groups. Finally, the terminal point of the difference vector, $X_{r2,G}$, is randomly selected from the superior group, and the start point, $X_{r3,G}$, is randomly selected from the inferior group. In this way, the direction information is incorporated into mutation for guiding the search of DE to the promising area.

4.4 The framework of DE-CDI

The framework of DE-CDI is illustrated in Fig. 3. The pseudo-code of DE-CDI with DE/rand/1 (named as DE-CDI/rand/1) is also shown in Algorithm 1 where the differences with respect to DE/rand/1 are highlighted with "$*$". It is clear that DE-CDI only affects the mutation operator and it can be directly and easily applied to most DE algorithms.

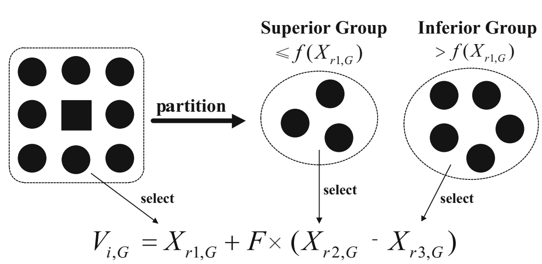

Fig. 2 DE/rand/1 with direction information based on C9

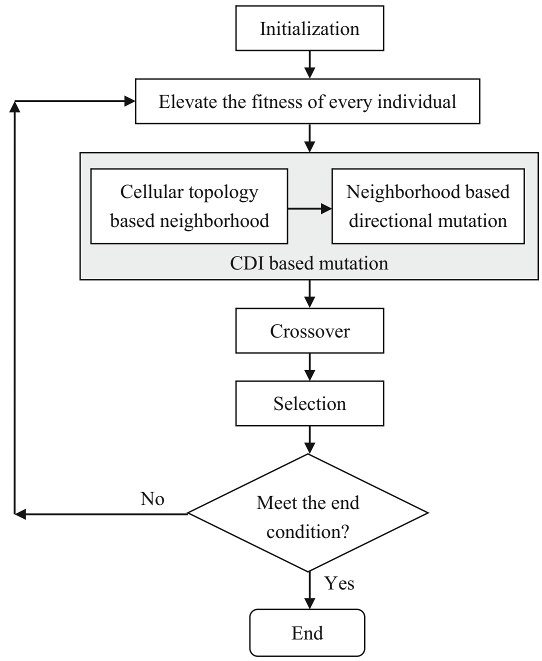

Fig. 3 Flowchart of DE-CDI

When applying DE-CDI to the mutation operator that employs the best individual (e.g., DE/best/1, DE/ current-to-best/1 and DE/rand-to-best/1), the best neighbor of the current individual will be served as the best individual for mutation. That is, the best individual is refined in these mutation operators. As for constructing the difference vector, when the base vector is the best or worst vector in the neighborhood, two vectors will be randomly selected from the neighborhood of the base vector and the difference vector will be constructed by directing at the better vector from the worse one.

4.5 The complexity of DE-CDI

Compared with the original DE algorithm, the additional computation of DE-CDI depends on the selection of parents for mutation, i.e., steps 5–7 in Algorithm 1. During one generation, selecting individuals to construct difference vector based on Cn will take $n-1$ times of comparison with the base vector. Therefore, the additional computation of DE-CDI is $O(n)$. Since the complexity of the original DE algorithm is $O(G_{\max} \times NP \times D)$ where $G_{\max}$ is the maximal number of generation, the total complexity of DE-CDI is $O(G_{\max} \times NP \times \max(D, n))$.

Algorithm 1 DE-CDI/rand/1

```
Generate the initial population P^G and set G = 1;
Evaluate the fitness for each individual in P^G;
While the terminated condition is not satisfied do
  For each individual X_{i,t} do
    * Randomly select the base vector X_{r1} from the neighborhood of X_{i,G};
    * Partition all the neighbors of X_{i,G} into the superior and inferior groups by comparing their fitness with X_{i,G};
    * Randomly select X_{r2} and X_{r3} from the superior and inferior groups respectively;
    Use Eq. (2) to generate a mutant vector;
    Use Eq. (8) to generate a trial vector;
    Use Eq. (9) to determine the survived vector;
  End For
  Set G = G + 1;
End while
```

5 Experimental result and analysis

In order to evaluate the performance of DE-CDI, 25 benchmark functions from the CEC2005 special session on real-parameter optimization (Suganthan et al. 2005) and three real-world problems (Eshelman et al. 1997; Das and Suganthan 2010) are used. In this section, the benchmark functions are presented first. Then, the experimental setup is given. Finally, the simulation results are analyzed and discussed.

5.1 Benchmark functions

25 benchmark functions are used, denoted as $F1$–$F25$, which are from the special session on real-parameter optimization of the 2005 IEEE Congress on Evolutionary Computation (CEC 2005) (Suganthan et al. 2005). According to Suganthan et al. (2005), these functions can be categorized into four groups: unimodal functions ($F1$–$F5$), basic multimodal functions ($F6$–$F12$), expanded multimodal functions ($F13$–$F14$) and hybrid composition functions ($F15$–$F25$). More details of them can be found in Rogalsky et al. (1999).

5.2 Parameter settings

For a fair comparison, the same random initial population is used to evaluate the performance of different algorithms. The parameters are set as follows unless a change is mentioned:

- Dimension of each function (D): 30 and 50.
- Population size (NP): 100.
- Mutation factor (F): 0.5.
- Crossover factor (Cr): 0.9.
- Number of neighbors (n): 13 (i.e., C13).
- Maximum number of function evaluations (MNFEs): $10^4 \times D$.
- Number of runs (NumR): 25.

In the experimental study, comparisons between six original DE algorithms (i.e., DE/rand/1, DE/rand/2, DE/best/1, DE/best/2, DE/current-to-best/1 and DE/ rand-to-best/1) and their corresponding DE-CDI algorithms are made first. Then, we compare the performance of DE-CDI with advanced DE variants, i.e., CoDE (Wang et al. 2011), jDE (Brest et al. 2006), ODE (Rahnamayan et al. 2008) and SaDE (Qin et al. 2009). All the parameters of these DE variants are set as their original papers except NP in CoDE and SaDE. For comparing the algorithms with the same number of neighbors, e.g., $n = 25$ (C25) or 49 (C49), NP in CoDE and SaDE is set to 100. Simulations are carried out on an Intel Core i3 duo PC with 3.30-GHz CPU and 4 GB RAM.

To show the significant differences among the competitors, several nonparametric statistical tests (García et al. 2009; Derrac et al. 2011) are also carried out by the KEEL software (Alcalá-Fdez et al. 2015). The results of single-problem Wilcoxon signed-rank test at $\alpha = 0.05$ (García et al. 2009; Derrac et al. 2011) are shown in the tables as "$+/=/-$", which means that DE-CDI wins, ties and loses on the specific function when compared with its competitor.

5.3 Comparison with original DE algorithms

Six original DE mutation operators (see Eqs. 2–7) are used here. The results for all the functions at $30D$ and $50D$ are shown in Tables 1 and 2, respectively. The better values in terms of mean solution error and standard deviation compared between DE and its corresponding DE-CDI variants are highlighted in boldface.

For the functions at $30D$, the results of Table 1 show that DE-CDI can provide significantly better results than the corresponding original DE algorithms on most functions. Specifically, for DE/best/1, DE-CDI exhibits substantial performance improvements on 24 out of 25 functions. For DE/current-to-best/1 and DE/rand-to-best/1, DE-CDI is significantly better on 21 and 21 functions, respectively. For DE/rand/2, DE-CDI can effectively enhance its exploitative ability and yield significant improvements on 22 functions. For DE/rand/1 and DE/best/2, DE-CDI is significantly better on 11 and 9 functions, respectively.

For the functions at $50D$, the results of Table 2 also show that DE-CDI is consistently superior to most of the corresponding DE algorithms. DE-CDI/best/1 is significantly better than DE/best/1 on 22 functions, while DE-CDI/current-to-best/1 and DE-CDI/rand-to-best/1 are significantly better than their corresponding original DE algorithms on 21 and 21 functions, respectively. For DE/rand/1 and DE/rand/2, DE-CDI significantly outperforms them on 10 and 22 functions, respectively. For DE/best/2, DE-CDI is also significantly better on 12 functions.

Table 1 Mean and standard deviation of the best error values obtained by DE-CDI and the original DE algorithm on all the functions at 30D

Function	DE/rand/1		DE-CDI/rand/1	DE/rand/2		DE-CDI/rand/2	DE/best/1		DE-CDI/best/1
F1	**1.62e−029 5.59e−029**	−	1.94e−028 1.49e−028	8.37e−001 2.86e−001	+	**1.86e−012 8.53e−013**	3.70e+003 1.69e+003	+	**1.72e+001 4.95e+001**
F2	5.82e−005 4.46e−005	+	**9.65e−013 1.53e−012**	7.65e+003 1.80e+003	+	**4.38e+000 2.37e+000**	6.37e+003 3.78e+003	+	**1.56e−002 7.27e−002**
F3	4.74e+005 2.65e+005	+	**1.39e+005 6.23e+004**	4.85e+007 1.30e+007	+	**1.05e+006 4.41e+005**	1.52e+007 1.33e+007	+	**3.53e+005 1.71e+005**
F4	3.28e−002 5.23e−002	+	**5.21e−006 1.17e−005**	1.46e+004 2.59e+003	+	**8.57e+001 3.85e+001**	2.46e+002 3.97e+002	+	**5.10e−001 6.59e−001**
F5	**4.77e+001 3.66e+001**	−	2.89e+002 1.68e+002	8.07e+003 7.28e+002	+	**1.21e+003 2.38e+002**	9.04e+003 2.15e+003	+	**2.79e+003 5.33e+002**
F6	**1.88e+000 1.40e+000**	−	3.17e+001 2.25e+001	5.77e+003 4.08e+003	+	**1.61e+001 1.63e+001**	3.40e+008 2.51e+008	+	**1.41e+007 3.80e+007**
F7	**4.93e−004 2.46e−003**	−	2.43e−002 1.74e−002	7.20e+000 2.22e+000	+	**3.69e−005 3.94e−005**	3.65e+003 5.48e+002	+	**3.56e+000 6.33e+000**
F8	2.10e+001 4.46e−002	=	2.09e+001 5.31e−002	2.09e+001 6.06e−002	=	2.09e+001 6.41e−002	2.09e+001 5.73e−002	=	2.10e+001 5.59e−002
F9	1.33e+002 3.41e+001	+	**2.84e+001 7.30e+000**	2.10e+002 1.28e+001	+	**1.94e+002 1.08e+001**	1.06e+002 2.60e+001	+	**6.75e+001 1.88e+001**
F10	1.81e+002 9.31e+000	+	**4.34e+001 2.12e+001**	2.42e+002 9.26e+000	+	**2.05e+002 9.29e+000**	1.60e+002 3.86e+001	+	**9.42e+001 2.10e+001**
F11	3.97e+001 1.06e+000	+	**3.10e+001 6.60e+000**	3.93e+001 1.30e+000	=	3.97e+001 1.14e+000	**2.22e+001 2.41e+000**	−	2.91e+001 5.51e+000
F12	2.41e+003 2.26e+003	=	4.12e+003 8.03e+003	5.22e+005 4.47e+004	+	**2.70e+005 8.67e+004**	1.11e+005 4.94e+004	+	**5.13e+003 4.19e+003**
F13	1.50e+001 9.66e−001	+	**2.86e+000 6.36e−001**	2.02e+001 9.01e−001	+	**1.68e+001 8.94e−001**	7.74e+000 1.95e+000	+	**3.79e+000 9.85e−001**
F14	1.34e+001 1.09e−001	+	**1.32e+001 1.83e−001**	1.34e+001 1.58e−001	=	1.34e+001 1.94e−001	1.18e+001 4.34e−001	=	1.21e+001 6.59e−001
F15	4.04e+002 3.51e+001	=	4.04e+002 6.76e+001	4.05e+002 6.90e+000	+	**3.96e+002 4.52e+001**	5.65e+002 5.82e+001	+	**4.12e+002 5.37e+001**
F16	2.08e+002 7.66e+000	+	**5.53e+001 1.25e+001**	2.69e+002 1.12e+001	+	**2.29e+002 1.02e+001**	2.91e+002 1.24e+002	+	**1.39e+002 4.69e+001**
F17	2.33e+002 8.35e+000	+	**5.88e+001 2.61e+001**	3.05e+002 1.34e+001	+	**2.55e+002 1.15e+001**	3.22e+002 1.31e+002	+	**1.75e+002 1.26e+002**
F18	9.00e+002 2.09e+001	=	8.90e+002 4.60e+001	9.40e+002 2.18e+000	+	**9.09e+002 1.28e+000**	9.83e+002 3.18e+001	+	**9.03e+002 6.05e+001**
F19	9.00e+002 2.10e+001	+	**8.98e+002 3.72e+001**	9.39e+002 3.20e+000	+	**9.09e+002 1.10e+000**	9.88e+002 3.31e+001	+	**9.05e+002 6.12e+001**
F20	9.00e+002 2.10e+001	=	8.80e+002 5.09e+001	9.39e+002 3.51e+000	+	**9.09e+002 1.40e+000**	9.91e+002 4.10e+001	+	**8.96e+002 6.12e+001**
F21	5.00e+002 0.00e+000	=	5.12e+002 6.00e+001	5.00e+002 1.23e−001	+	**5.00e+002 0.00e+000**	1.06e+003 1.71e+002	+	**7.53e+002 3.23e+002**
F22	9.07e+002 9.41e+000	=	8.99e+002 1.29e+001	1.02e+003 1.54e+001	+	**9.35e+002 6.42e+000**	1.06e+003 6.74e+001	+	**9.36e+002 2.51e+001**
F23	5.34e+002 3.16e−004	=	5.34e+002 1.18e−002	5.35e+002 1.24e+000	+	**5.34e+002 1.70e−004**	1.11e+003 1.42e+002	+	**7.05e+002 2.82e+002**
F24	2.00e+002 0.00e+000	=	2.00e+002 0.00e+000	2.00e+002 1.16e−001	+	**2.00e+002 0.00e+000**	1.08e+003 2.51e+002	+	**2.00e+002 0.00e+000**
F25	2.09e+002 2.20e−001	=	2.09e+002 1.91e−001	2.29e+002 3.54e+000	+	**2.09e+002 2.27e−001**	1.56e+003 5.37e+001	+	**2.52e+002 2.13e+002**
SUM (+/ = /−)	–	11/10/4		–	22/3/0		–	22/2/1	

Function	DE/best/2		DE-CDI/best/2	DE/current-to-best/1		DE-CDI/current-to-best/1	DE/rand-to-best/1		DE-CDI/rand-to-best/1
F1	**7.66e−028 2.71e−028**	−	1.04e−021 2.25e−021	3.58e+003 2.05e+003	+	**5.61e−002 2.79e−001**	1.82e+003 9.01e+002	+	**9.67e+000 1.94e+001**
F2	**1.06e−014 1.21e−014**	−	1.14e−005 4.00e−005	5.64e+003 2.23e+003	+	**5.22e−020 1.67e−019**	4.24e+003 1.54e+003	+	**6.64e+001 1.41e+002**
F3	1.37e+005 7.33e+004	=	1.45e+005 8.51e+004	6.51e+006 5.13e+006	+	**2.79e+004 1.68e+004**	6.91e+006 5.75e+006	+	**3.78e+005 4.21e+005**
F4	**9.37e−005 1.36e−004**	−	4.07e−003 7.30e−003	5.59e+002 4.98e+002	+	**2.01e−007 8.80e−007**	7.66e+002 4.75e+002	+	**3.15e−003 8.97e−003**
F5	**3.83e+001 5.26e+001**	−	1.96e+002 2.48e+002	7.36e+003 1.66e+003	+	**2.37e+002 2.23e+002**	6.97e+003 1.40e+003	+	**2.13e+003 4.36e+002**

Table 1 continued

Function	DE/best/2		DE-CDI/best/2	DE/current-to-best/1		DE-CDI/current-to-best/1	DE/rand-to-best/1		DE-CDI/rand-to-best/1
F6	6.38e−001 1.49e+000	+	**3.19e−001 1.10e+000**	3.10e+008 2.70e+008	+	**2.29e+003 4.81e+003**	1.37e+008 1.23e+008	+	**1.04e+006 1.38e+006**
F7	1.55e−002 1.61e−002	+	**4.43e−003 6.51e−003**	3.98e+003 4.98e+002	+	**5.18e−002 3.49e−002**	3.70e+003 3.51e+002	+	**1.26e+001 2.00e+001**
F8	2.10e+001 3.58e−002	=	2.09e+001 7.63e−002	2.10e+001 4.46e−002	=	2.09e+001 8.84e−002	2.09e+001 3.82e−002	=	2.09e+001 5.10e−002
F9	1.82e+002 1.50e+001	+	**1.73e+002 1.15e+001**	7.54e+001 1.53e+001	+	**4.21e+001 1.15e+001**	6.57e+001 1.54e+001	+	**4.85e+001 9.61e+000**
F10	2.01e+002 1.31e+001	+	**1.80e+002 6.86e+000**	1.12e+002 3.37e+001	+	**5.32e+001 1.63e+001**	8.80e+001 1.80e+001	+	**5.40e+001 1.08e+001**
F11	3.95e+001 8.57e−001	=	3.93e+001 9.70e−001	**1.40e+001 2.31e+000**	−	2.13e+001 4.66e+000	**1.30e+001 1.88e+000**	−	2.14e+001 5.03e+000
F12	1.98e+003 4.16e+003	=	1.49e+003 1.51e+003	3.58e+004 2.06e+004	+	**2.50e+003 3.25e+003**	5.90e+004 2.17e+004	+	**1.10e+004 1.09e+004**
F13	1.57e+001 1.46e+000	=	1.51e+001 9.27e−001	4.86e+000 3.25e+000	=	3.54e+000 7.55e−001	3.25e+000 7.09e−001	=	3.54e+000 8.85e−001
F14	1.34e+001 1.72e−001	=	1.33e+001 1.85e−001	**1.17e+001 3.72e−001**	−	1.22e+001 3.07e−001	1.15e+001 4.79e−001	=	1.14e+001 5.30e−001
F15	**3.21e+002 1.12+002**	−	3.92e+002 4.00e+001	4.99e+002 1.29e+002	+	**4.11e+002 7.49e+001**	4.59e+002 8.21e+001	+	**3.91e+002 6.89e+001**
F16	2.64e+002 8.60e+001	+	**2.05e+002 2.05e+001**	2.82e+002 1.88e+002	+	**1.01e+002 7.16e+001**	2.60e+002 2.18e+002	+	**8.34e+001 3.29e+001**
F17	3.29e+002 1.09e+002	+	**2.24e+002 1.27e+001**	2.57e+002 1.64e+002	+	**9.59e+001 3.64e+001**	2.35e+002 1.46e+002	+	**1.17e+002 1.09e+002**
F18	8.86e+002 4.37e+001	=	9.04e+002 2.17e+001	9.75e+002 4.75e+001	+	**8.88e+002 5.60e+001**	9.79e+002 2.61e+001	+	**8.51e+002 6.82e+001**
F19	8.99e+002 2.98e+001	=	9.00e+002 3.01e+001	9.92e+002 3.11e+001	+	**8.78e+002 5.97e+001**	9.76e+002 4.12e+001	+	**8.78e+002 6.92e+001**
F20	8.99e+002 2.98e+001	=	9.04e+002 2.19e+001	9.89e+002 3.22e+001	+	**8.54e+002 6.23e+001**	9.73e+002 4.13e+001	+	**8.82e+002 6.65e+001**
F21	5.25e+002 8.49e+001	=	5.12e+002 6.00e+001	1.06e+003 1.50e+002	+	**5.53e+002 1.82e+002**	1.02e+003 1.68e+002	+	**6.25e+002 2.46e+002**
F22	9.27e+002 1.35e+001	+	**9.13e+002 8.81e+000**	1.02e+003 3.24e+001	+	**9.21e+002 1.77e+001**	9.98e+002 2.49e+001	+	**9.27e+002 1.79e+001**
F23	5.69e+002 1.12e+002	+	**5.34e+002 3.62e−004**	1.09e+003 1.43e+002	+	**6.64e+002 2.09e+002**	1.07e+003 1.12e+002	+	**6.47e+002 1.62e+002**
F24	2.00e+002 0.00e+000	=	2.00e+002 0.00e+000	9.37e+002 2.44e+002	+	**2.00e+002 0.00e+000**	8.83e+002 2.77e+002	+	**2.00e+002 2.76e−003**
F25	2.09e+002 1.03e−001	+	**2.09e+002 5.81e−002**	1.45e+003 5.75e+001	+	**2.21e+002 5.83e+001**	1.51e+003 3.36e+001	+	**2.09e+002 4.10e+000**
SUM (+/ = /−)	–		9/11/5	–		21/2/2	–		21/3/1

Table 2 Mean and standard deviation of the best error values obtained by DE-CDI and the original DE algorithm on all the functions at 50D

Function	DE/rand/1		DE-CDI/rand/1	DE/rand/2		DE-CDI/rand/2	DE/best/1		DE-CDI/best/1
F1	**1.94e−028 1.65e−028**	−	7.43e−028 3.84e−028	4.03e+002 1.39e+002	+	**3.77e−012 3.32e−012**	2.66e+004 4.72e+003	+	**1.77e+002 4.95e+002**
F2	4.54e+000 2.33e+000	+	**1.26e−006 1.42e−006**	7.46e+004 8.70e+003	+	**2.36e+003 8.17e+002**	2.86e+004 8.04e+003	+	**1.72e−002 5.78e−002**
F3	2.57e+006 8.74e+005	+	**2.39e+005 9.09e+004**	4.37e+008 6.12e+007	+	**1.48e+007 4.45e+006**	1.45e+008 6.33e+007	+	**1.04e+006 2.96e+005**
F4	4.45e+002 3.25e+002	+	**1.79e+001 2.21e+001**	9.65e+004 9.33e+003	+	**1.35e+004 2.89e+003**	6.19e+003 5.29e+003	+	**1.64e+003 1.55e+003**
F5	**1.80e+003 3.49e+002**	−	3.20e+003 4.02e+002	2.13e+004 1.13e+003	+	**4.66e+003 5.61e+002**	1.89e+004 3.04e+003	+	**7.04e+003 1.04e+003**
F6	**3.73e+001 2.47e+001**	−	1.55e+002 2.52e+002	5.90e+006 2.76e+006	+	**4.03e+001 1.65e+001**	4.33e+009 2.97e+009	+	**3.96e+007 1.12e+008**
F7	**2.17e−003 4.57e−003**	−	1.28e−002 1.53e−002	2.02e+002 7.11e+001	+	**4.43e−004 1.98e−003**	8.06e+003 8.66e+002	+	**9.87e+000 1.62e+001**
F8	2.11e+001 3.79e−002	=	2.11e+001 3.90e−002	2.11e+001 3.65e−002	=	2.11e+001 3.12e−002	2.11e+001 3.96e−002	=	2.11e+001 4.61e−002
F9	2.19e+002 3.43e+001	+	**8.17e+001 1.18e+001**	4.71e+002 1.57e+001	+	**4.02e+002 1.40e+001**	2.76e+002 3.24e+001	+	**1.67e+002 2.49e+001**
F10	3.59e+002 1.45e+001	+	**7.41e+001 1.47e+001**	5.20e+002 2.69e+001	+	**4.12e+002 1.55e+001**	4.67e+002 8.04e+001	+	**2.30e+002 4.79e+001**
F11	7.31e+001 1.42e+000	+	**6.80e+001 6.58e+000**	7.26e+001 1.61e+000	=	7.25e+001 1.62e+000	**4.71e+001 4.41e+000**	−	5.98e+001 4.93e+000
F12	8.43e+003 7.54e+003	=	9.75e+003 7.55e+003	2.57e+006 2.00e+005	+	**8.64e+005 3.48e+005**	8.20e+005 1.83e+005	+	**2.50e+004 1.76e+004**
F13	3.00e+001 1.39e+000	+	**5.91e+000 1.52e+000**	4.29e+001 2.24e+000	+	**3.32e+001 1.29e+000**	2.90e+001 6.46e+000	+	**9.79e+000 2.77e+000**
F14	2.31e+001 2.19e−001	+	**2.29e+001 2.34e−001**	2.31e+001 1.42e−001	=	2.31e+001 1.59e−001	2.13e+001 7.44e−001	+	**2.04e+001 6.91e−001**
F15	3.92e+002 4.00e+001	=	3.77e+002 6.56e+001	4.52e+002 2.89e+000	+	**3.92e+002 3.99e+001**	5.81e+002 5.97e+001	+	**3.89e+002 3.94e+001**
F16	2.55e+002 6.67e+000	+	**5.98e+001 1.15e+001**	3.60e+002 1.23e+001	+	**2.85e+002 9.52e+000**	3.73e+002 1.11e+002	+	**1.50e+002 3.98e+001**
F17	2.81e+002 1.01e+001	+	**5.90e+001 8.75e+000**	4.15e+002 1.64e+001	+	**3.22e+002 1.09e+001**	3.14e+002 8.61e+001	+	**1.83e+002 7.90e+001**
F18	**8.95e+002 4.87e+001**	−	9.20e+002 4.61e+001	1.03e+003 1.13e+001	+	**9.05e+002 4.68e+001**	1.09e+003 3.16e+001	+	**1.02e+003 1.82e+001**
F19	**8.71e+002 1.28e+002**	−	9.37e+002 2.98e+001	1.04e+003 9.59e+000	+	**9.20e+002 2.51e+001**	1.09e+003 2.73e+001	+	**1.03e+003 1.59e+001**
F20	**9.05e+002 3.95e+001**	−	9.44e+002 1.22e+001	1.04e+003 1.01e+001	+	**9.00e+002 5.12e+001**	1.09e+003 2.70e+001	+	**1.02e+003 2.19e+001**
F21	5.00e+002 1.65e−005	=	5.00e+002 1.25e−005	6.46e+002 5.92e+001	+	**5.00e+002 6.23e−006**	1.22e+003 5.31e+001	+	**8.40e+002 3.58e+002**
F22	**9.55e+002 1.27e+001**	−	9.62e+002 7.33e+000	1.14e+003 1.88e+001	+	**9.94e+002 9.22e+000**	1.13e+003 6.55e+001	+	**1.01e+003 2.11e+001**
F23	**5.39e+002 1.54e−002**	−	5.65e+002 1.28e+002	7.16e+002 5.63e+001	+	**5.39e+002 2.00e−001**	1.24e+003 5.50e+001	+	**9.69e+002 3.19e+002**
F24	2.00e+002 3.11e−006	=	2.00e+002 3.11e−006	6.42e+002 1.11e+002	+	**2.00e+002 0.00e+000**	1.30e+003 2.54e+001	+	**1.20e+003 2.08e+002**
F25	**2.14e+002 2.22e−001**	−	2.15e+002 5.42e−001	3.52e+002 2.51e+001	+	**2.16e+002 1.09e+000**	1.71e+003 3.50e+001	+	**8.83e+002 5.09e+002**
SUM (+/ = /−)	–		10/5/10	–		22/3/0	–		23/1/1

Function	DE/best/2		DE-CDI/best/2	DE/current-to-best/1		DE-CDI/current-to-best/1	DE/rand-to-best/1		DE-CDI/rand-to-best/1
F1	**2.04e−027 4.86e−028**	−	1.38e−024 5.90e−024	2.01e+004 5.90e+003	+	**1.87e+001 6.02e+001**	1.43e+004 3.79e+003	+	**2.93e+002 3.62e+002**
F2	**5.11e−005 3.63e−005**	−	3.53e−001 1.07e+000	2.41e+004 5.52e+003	+	**1.10e−013 2.98e−013**	2.07e+004 5.86e+003	+	**6.71e+001 1.42e+002**
F3	**5.32e+005 1.91e+005**	−	1.76e+006 6.94e+005	1.21e+008 7.44e+007	+	**2.04e+005 2.30e+005**	9.31e+007 3.10e+007	+	**3.51e+006 1.21e+006**
F4	2.69e+002 1.93e+002	+	**1.24e+002 1.36e+002**	7.73e+003 2.25e+003	+	**1.47e+001 3.67e+001**	9.40e+003 3.02e+003	+	**3.33e+002 3.07e+002**
F5	2.74e+003 6.07e+002	+	**2.12e+003 5.00e+002**	1.51e+004 1.74e+003	+	**3.04e+003 5.63e+002**	1.36e+004 1.63e+003	+	**6.13e+003 6.63e+002**
F6	1.78e+000 2.05e+000	=	8.49e−001 1.52e+000	3.02e+009 1.82e+009	+	**1.07e+006 2.75e+006**	1.71e+009 8.36e+008	+	**3.41e+007 4.63e+007**

Table 2 continued

Function	DE/best/2		DE-CDI/best/2	DE/current-to-best/1		DE-CDI/current-to-best/1	DE/rand-to-best/1		DE-CDI/rand-to-best/1
F7	9.54e−003 1.31e−002	+	**3.24e−003 9.69e−003**	7.78e+003 7.52e+002	+	**8.36e+000 4.06e+001**	7.31e+003 4.93e+002	+	**5.15e+001 5.32e+001**
F8	2.11e+001 2.93e−002	=	2.11e+001 3.86e−002	2.11e+001 3.96e−002	=	2.11e+001 2.99e−002	2.11e+001 3.45e−002	=	2.11e+001 4.19e−002
F9	3.90e+002 3.17e+001	+	**3.49e+002 1.63e+001**	1.93e+002 3.38e+001	+	**1.27e+002 2.22e+001**	1.75e+002 3.16e+001	+	**1.29e+002 1.65e+001**
F10	4.22e+002 3.66e+001	+	**3.64e+002 1.05e+001**	2.87e+002 5.07e+001	+	**1.69e+002 2.94e+001**	2.85e+002 4.26e+001	+	**1.51e+002 2.25e+001**
F11	7.27e+001 1.12e+000	=	7.31e+001 1.12e+000	**3.49e+001 3.14e+000**	−	5.12e+001 4.36e+000	**3.16e+001 2.58e+000**	−	4.85e+001 6.96e+000
F12	7.03e+003 6.07e+003	=	9.58e+003 1.26e+004	3.35e+005 1.48e+005	+	**7.89e+003 8.34e+003**	4.80e+005 1.64e+005	+	**5.59e+004 3.22e+004**
F13	3.27e+001 2.06e+000	+	**3.05e+001 1.09e+000**	8.57e+000 1.88e+000	+	**6.66e+000 1.40e+000**	8.88e+000 1.91e+000	+	**6.59e+000 1.55e+000**
F14	2.30e+001 1.60e−001	=	2.31e+001 1.60e−001	**2.13e+001 3.69e−001**	−	2.21e+001 3.42e−001	2.09e+001 4.53e−001	=	**2.10e+001 4.82e−001**
F15	**2.95e+002 9.72e+001**	−	3.76e+002 6.64e+001	5.15e+002 1.15e+002	+	**3.98e+002 2.37e+001**	5.03e+002 6.31e+001	+	**4.00e+002 4.03e+001**
F16	3.16e+002 5.60e+001	+	**2.57e+002 8.31e+000**	2.40e+002 1.25e+002	+	**1.32e+002 8.38e+001**	2.21e+002 9.79e+001	+	**1.13e+002 6.58e+001**
F17	3.50e+002 5.43e+001	+	**2.80e+002 1.15e+001**	2.64e+002 9.91e+001	+	**1.16e+002 6.88e+001**	2.12e+002 9.56e+001	+	**1.15e+002 7.07e+001**
F18	9.01e+002 1.30e+002	=	8.87e+002 1.29e+002	1.06e+003 2.44e+001	+	**9.77e+002 1.81e+001**	1.06e+003 2.09e+001	+	**9.95e+002 1.23e+001**
F19	9.21e+002 3.69e+001	+	**8.71e+002 1.74e+002**	1.05e+003 2.33e+001	+	**9.81e+002 1.58e+001**	1.05e+003 1.81e+001	+	**9.93e+002 1.38e+001**
F20	9.22e+002 3.74e+001	=	8.98e+002 1.28e+002	1.05e+003 2.32e+001	+	**9.70e+002 1.50e+001**	1.05e+003 1.79e+001	+	**9.99e+002 2.07e+001**
F21	5.12e+002 6.00e+001	=	5.00e+002 1.25e−005	1.21e+003 3.89e+001	+	**7.91e+002 3.44e+002**	1.18e+003 1.08e+002	+	**8.20e+002 3.41e+002**
F22	9.92e+002 1.46e+001	+	**9.70e+002 8.55e+000**	1.12e+003 4.71e+001	+	**9.87e+002 1.71e+001**	1.09e+003 4.53e+001	+	**9.90e+002 2.36e+001**
F23	5.90e+002 1.45e+002	+	**5.54e+002 6.98e+001**	1.23e+003 4.72e+001	+	**8.05e+002 2.23e+002**	1.20e+003 7.26e+001	+	**1.00e+003 2.53e+002**
F24	2.82e+002 2.83e+002	=	2.00e+002 3.11e−006	1.25e+003 1.30e+001	+	**5.13e+002 4.66e+002**	1.23e+003 1.85e+001	+	**1.08e+003 3.31e+002**
F25	2.14e+002 5.53e−001	+	**2.13e+002 2.15e−001**	1.65e+003 4.11e+001	+	**2.56e+002 1.98e+002**	1.67e+003 3.13e+001	+	**1.01e+003 4.52e+002**
SUM (+/ = /−)	–	12/9/4		–	22/1/2		–	22/2/1	

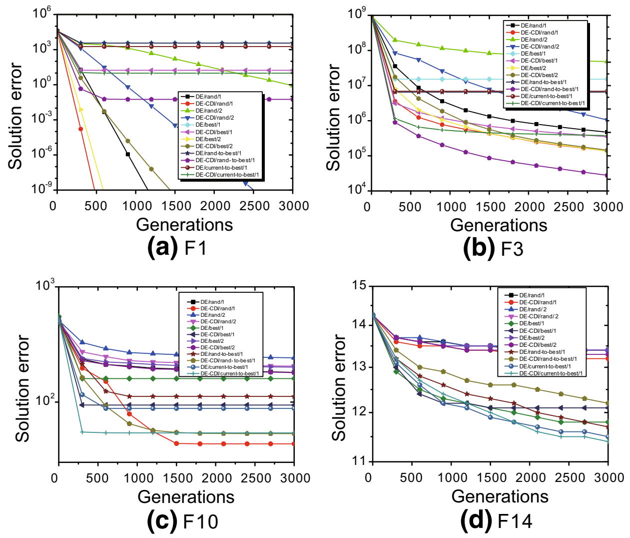

Fig. 4 Convergence graphs of the original DE algorithm and the corresponding DE-CDI for the selected functions at 30D

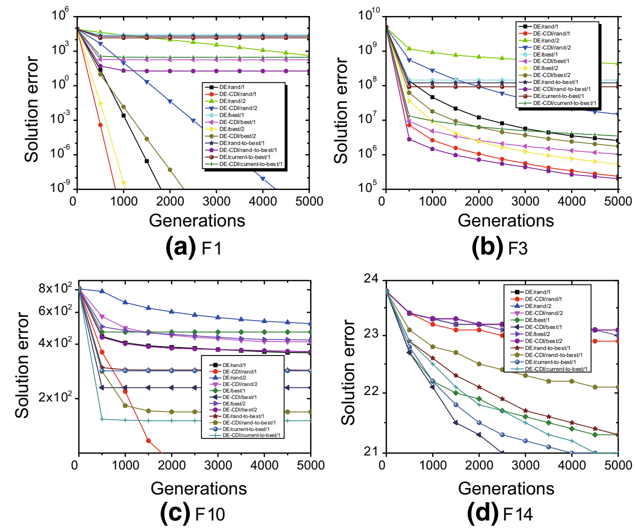

Fig. 5 Convergence graphs of the original DE algorithm and the corresponding DE-CDI for the selected functions at 50D

Figures 4 and 5 also show that the convergence speed of DE-CDI is better than the corresponding original DE algorithms for most selected functions at 30D and 50D.

Furthermore, to show the significant differences between DE-CDI and its corresponding DE algorithm, the multi-problem Wilcoxon signed-rank test (García et al. 2009; Derrac et al. 2011) is also carried out on all the functions at 30D and 50D. The results are shown in Tables 3 and 4, respectively. For the functions at 30D, it is clear that DE-CDI can obtain the higher $R+$ values than $R-$ values in all the cases. In addition, the p values in most cases are less than 0.05, which means that DE-CDI is significantly better than most original DE algorithms overall. For the functions at 50D, the similar results can also be obtained from Table 4.

5.4 Comparison with advanced DE algorithms

In order to evaluate the effectiveness of DE-CDI for the advanced DE variants, four DE variants are used, i.e., CoDE (Wang et al. 2011), jDE (Brest et al. 2006), ODE (Rahnamayan et al. 2008) and SaDE (Qin et al. 2009). The results

Table 3 Results of the multi-problem Wilcoxon's test for DE-CDI versus the original DE algorithm for all the functions at 30D

Algorithm	$+/=/-$	$R+$	$R-$	p value	$\alpha = 0.05$	$\alpha = 0.1$
DE-CDI/rand/1 vs DE/rand/1	11/10/4	228	97	7.58E−02	=	+
DE-CDI/rand/2 vs DE/rand/2	22/3/0	315	10	3.60E−05	+	+
DE-CDI/best/1 vs DE/best/1	22/2/1	318	7	2.70E−05	+	+
DE-CDI/best/2 vs DE/best/2	9/11/5	200.5	124.5	3.00E−01	=	=
DE-CDI/current-to-best/1 vs DE/current-to-best/1	21/2/2	319	6	2.40E−05	+	+
DE-CDI/rand-to-best/1 vs DE/rand-to-best/1	21/3/1	295	5	3.20E−05	+	+

Table 4 Results of the multi-problem Wilcoxon's test for DE-CDI versus the original DE algorithm for all the functions at 50D

Algorithm	$+/=/-$	$R+$	$R-$	p value	$\alpha = 0.05$	$\alpha = 0.1$
DE-CDI/rand/1 vs DE/rand/1	10/5/10	161.5	138.5	7.32E−01	=	=
DE-CDI/rand/2 vs DE/rand/2	22/3/0	323.5	1.5	1.30E−05	+	+
DE-CDI/best/1 vs DE/best/1	23/1/1	298	2	2.10E−05	+	+
DE-CDI/best/2 vs DE/best/2	12/9/4	221	79	4.11E−02	+	+
DE-CDI/current-to-best/1 vs DE/current-to-best/1	22/1/2	296	4	2.80E−05	+	+
DE-CDI/rand-to-best/1 vs DE/rand-to-best/1	22/2/1	296	4	2.80E−05	+	+

Table 5 Mean and standard deviation of the best error values obtained by DE-CDI and the advanced DE algorithm on all the functions at 30D

Function	CoDE	CoDE-CDI	jDE	jDE-CDI	ODE	ODE-CDI	SaDE	SaDE-CDI
F1	3.00e−003 6.30e−004	+ **1.05e−008 4.65e−009**	0.00e+000 0.00e+000	= 0.00e+000 0.00e+000	1.86e−028 1.42e−028	= 1.99e−028 1.89e−028	0.00e+000 0.00e+000	= 0.00e+000 0.00e+000
F2	1.17e+004 2.03e+003	+ **6.85e+002 1.49e+002**	1.11e−005 1.19e−005	+ **3.32e−006 4.84e−006**	2.39e−004 2.84e−004	+ **1.07e−005 7.43e−006**	1.91e+000 9.22e−001	+ **3.79e−001 1.64e−001**
F3	5.24e+007 1.43e+007	+ **1.42e+007 3.33e+006**	**1.81e+005 1.02e+005**	− 3.10e+005 1.83e+005	6.66e+005 2.99e+005	+ **4.41e+005 1.79e+005**	9.77e+006 2.67e+006	+ **1.96e+006 4.46e+005**
F4	1.87e+004 2.99e+003	+ **2.60e+003 5.65e+002**	4.80e−001 1.37e+000	= 2.60e−001 3.00e−001	1.08e−001 8.62e−002	+ **2.32e−002 2.61e−002**	1.32e+002 5.55e+001	= 1.19e+002 5.41e+001
F5	5.69e+003 5.06e+002	+ **1.22e+003 3.20e+002**	**8.94e+002 4.12e+002**	− 1.31e+003 4.51e+002	1.82e+002 8.74e+001	+ **3.59e+001 2.94e+001**	2.55e+003 4.46e+002	+ **2.09e+003 3.89e+002**
F6	4.40e+002 6.67e+001	+ **2.76e+001 5.24e+000**	1.97e+001 2.17e+001	+ **1.08e+001 1.98e+001**	5.39e+001 2.46e+001	+ **1.81e+000 2.35e+000**	5.40e+001 3.30e+001	+ **2.03e+001 3.77e+000**
F7	2.72e+001 5.21e+000	+ **2.51e−001 9.95e−002**	1.28e−002 8.51e−003	+ **6.60e−003 6.02e−003**	1.18e−003 3.27e−003	= 3.94e−003 7.61e−003	1.26e−002 7.98e−003	+ **7.33e−003 4.61e−003**
F8	2.09e+001 5.49e−002	= 2.09e+001 6.40e−002	2.10e+001 4.42e−002	= 2.10e+001 4.60e−002	2.10e+001 4.88e−002	= 2.10e+001 4.45e−002	2.09e+001 4.70e−002	= 2.09e+001 4.27e−002
F9	1.76e+001 1.54e+000	= 1.71e+001 1.91e+000	0.00e+000 0.00e+000	= 0.00e+000 0.00e+000	7.89e+001 2.99e+001	+ **2.49e+001 1.28e+001**	0.00e+000 0.00e+000	= 0.00e+000 0.00e+000
F10	2.07e+002 1.51e+001	+ **1.78e+002 1.13e+001**	5.80e+001 8.63e+000	+ **5.00e+001 8.03e+000**	5.09e+001 4.66e+001	= 4.14e+001 2.69e+001	**7.44e+001 7.09e+000**	− 8.21e+001 7.59e+000
F11	3.51e+001 9.66e−001	= 3.44e+001 1.36e+000	2.76e+001 1.82e+000	= 2.78e+001 1.24e+000	**9.22e+000 9.33e+000**	− 1.45e+001 9.96e+000	2.77e+001 1.23e+000	= 2.81e+001 1.28e+000
F12	1.59e+005 2.44e+004	+ **1.16e+005 1.89e+004**	1.51e+004 8.70e+003	+ **6.61e+003 5.57e+003**	2.27e+003 2.24e+003	= 1.64e+003 1.95e+003	2.06e+004 4.51e+003	= 1.95e+004 4.91e+003
F13	9.84e+000 7.31e−001	+ **8.71e+000 6.58e−001**	1.67e+000 1.67e−001	= 1.73e+000 1.63e−001	6.19e+000 2.51e+000	+ **3.03e+000 7.50e−001**	**2.40e+000 1.54e−001**	− 2.48e+000 1.82e−001
F14	1.33e+001 1.48e−001	+ **1.32e+001 1.31e−001**	1.30e+001 1.79e−001	= 1.29e+001 2.74e−001	1.32e+001 2.92e−001	+ **1.30e+001 3.46e−001**	1.27e+001 2.03e−001	= 1.28e+001 1.58e−001
F15	4.00e+002 1.67e−002	+ **3.92e+002 3.99e+001**	3.13e+002 1.25e+002	= 3.12e+002 1.32e+002	4.12e+002 3.31e+001	= 4.12e+002 3.34e+001	2.67e+002 1.16e+002	= 2.01e+002 1.59e+002
F16	2.29e+002 1.04e+001	+ **1.98e+002 7.80e+000**	7.77e+001 1.09e+001	+ **6.81e+001 8.76e+000**	9.02e+001 6.89e+001	= 8.52e+001 5.97e+001	1.06e+002 1.78e+001	= 1.15e+002 1.16e+001
F17	2.64e+002 1.53e+001	+ **2.31e+002 1.37e+001**	1.31e+002 2.04e+001	+ **1.12e+002 2.42e+001**	1.56e+002 7.50e+001	= 1.35e+002 7.60e+001	**1.58e+002 1.58e+001**	− 1.70e+002 1.55e+001
F18	9.08e+002 2.62e−001	+ **9.06e+002 2.03e−001**	9.07e+002 1.61e+000	+ **9.06e+002 1.21e+000**	9.01e+002 2.10e+001	= 8.97e+002 2.92e+001	**8.23e+002 4.72e+001**	− 8.70e+002 5.38e+001
F19	9.08e+002 4.31e−001	+ **9.06e+002 1.74e−001**	9.06e+002 1.57e+000	= 9.07e+002 1.74e+000	**9.04e+002 8.17e−001**	− 9.05e+002 1.51e+000	8.64e+002 5.77e+001	= 8.88e+002 4.47e+001
F20	9.08e+002 4.20e−001	+ **9.06e+002 1.77e−001**	9.02e+002 2.14e+001	= 9.07e+002 1.21e+000	9.05e+002 1.94e+000	= 8.93e+002 3.49e+001	8.59e+002 5.80e+001	= 8.65e+002 5.43e+001
F21	5.00e+002 2.14e−004	+ **5.00e+002 0.00e+000**	5.00e+002 0.00e+000	= 5.00e+002 0.00e+000	5.00e+002 0.00e+000	= 5.00e+002 0.00e+000	5.00e+002 0.00e+000	= 5.00e+002 1.53e−005
F22	9.54e+002 1.03e+001	+ **9.02e+002 9.69e+000**	9.03e+002 8.50e+000	= 9.05e+002 1.14e+001	9.09e+002 1.16e+001	= 9.05e+002 9.22e+000	9.35e+002 6.96e+000	= 9.34e+002 8.98e+000
F23	5.34e+002 4.52e−004	+ **5.34e+002 3.70e−004**	5.34e+002 1.58e−004	= 5.34e+002 1.58e−004	5.34e+002 3.50e−004	= 5.34e+002 3.15e−004	**5.34e+002 6.80e−005**	− 5.34e+002 1.08e−004
F24	2.00e+002 3.51e−004	+ **2.00e+002 4.92e−006**	2.00e+002 0.00e+000	= 2.00e+002 0.00e+000	2.00e+002 0.00e+000	= 2.00e+002 0.00e+000	2.00e+002 0.00e+000	= 2.00e+002 0.00e+000
F25	2.56e+002 9.67e+000	+ **2.16e+002 1.43e+000**	2.10e+002 2.29e−001	= 2.09e+002 3.13e−001	2.09e+002 2.51e−001	+ **2.09e+002 1.31e−001**	2.11e+002 4.57e−001	= 2.11e+002 5.41e−001
SUM (+/ = /−)	–	22/3/0	–	8/15/2	–	9/14/2	–	5/15/5

Table 6 Mean and standard deviation of the best error values obtained by DE-CDI and the advanced DE algorithm on all the functions at $50D$

Function	CoDE	CoDE-CDI	jDE	jDE-CDI	ODE	ODE-CDI	SaDE	SaDE-CDI
F1	9.71e−001 1.07e−001	+ **9.91e−008 3.85e−008**	0.00e+000 0.00e+000	= 0.00e+000 0.00e+000	1.25e−027 1.19e−027	= 1.40e−027 1.46e−027	0.00e+000 0.00e+000	= 0.00e+000 0.00e+000
F2	5.73e+004 5.62e+003	+ **8.46e+003 1.95e+003**	8.94e−002 1.57e−001	= 2.43e−001 3.38e−001	5.72e+000 3.00e+000	+ **1.39e+000 1.16e+000**	3.22e+002 1.18e+002	+ **1.06e+002 4.69e+001**
F3	2.30e+008 3.29e+007	+ **4.81e+007 1.10e+007**	**3.92e+005 1.24e+005**	− 1.64e+006 8.48e+005	4.28e+006 1.40e+006	+ **2.38e+006 5.76e+005**	2.72e+007 4.76e+006	+ **1.18e+007 3.16e+006**
F4	7.38e+004 8.33e+003	+ **2.14e+004 3.80e+003**	**5.88e+002 4.21e+002**	− 9.31e+002 7.26e+002	5.84e+002 3.19e+002	= 4.92e+002 3.75e+002	6.95e+003 2.38e+003	= 6.45e+003 1.62e+003
F5	1.54e+004 5.90e+002	+ **5.13e+003 4.51e+002**	3.31e+003 5.98e+002	= 3.25e+003 5.12e+002	2.30e+003 3.63e+002	= 2.14e+003 3.71e+002	5.40e+003 7.51e+002	= 5.37e+003 3.59e+002
F6	1.32e+004 2.24e+003	+ **5.33e+001 2.82e+001**	3.57e+001 2.40e+001	= 3.40e+001 3.40e+001	1.19e+005 4.20e+005	+ **4.38e+001 2.18e+001**	9.94e+001 4.08e+001	+ **3.70e+001 3.64e+000**
F7	3.78e+002 5.29e+001	+ **1.39e−001 6.84e−002**	2.66e−003 6.91e−003	= 5.81e−003 9.41e−003	1.28e−003 4.44e−003	= 2.17e−003 5.14e−003	3.62e−003 7.24e−003	= 2.59e−003 5.91e−003
F8	2.11e+001 3.51e−002	= 2.11e+001 5.28e−002	2.11e+001 3.48e−002	= 2.11e+001 4.54e−002	2.11e+001 3.36e−002	= 2.11e+001 3.68e−002	2.11e+001 3.13e−002	= 2.11e+001 3.34e−002
F9	7.69e+001 4.44e+000	+ **7.32e+001 4.47e+000**	0.00e+000 0.00e+000	= 0.00e+000 0.00e+000	1.98e+002 6.87e+001	+ **4.40e+001 1.22e+001**	0.00e+000 0.00e+000	= 0.00e+000 0.00e+000
F10	4.46e+002 1.34e+001	+ **3.73e+002 1.13e+001**	9.87e+001 8.32e+000	+ **8.70e+001 1.75e+001**	1.25e+002 1.06e+002	= 7.80e+001 3.92e+001	**1.94e+002 1.45e+001**	− 2.06e+002 1.44e+001
F11	6.81e+001 1.42e+000	= 6.76e+001 1.54e+000	5.42e+001 2.30e+000	= 5.38e+001 3.09e+000	**1.61e+001 5.37e+000**	− 3.02e+001 1.32e+001	5.57e+001 2.68e+000	= 5.49e+001 2.02e+000
F12	8.19e+005 8.74e+004	+ **6.30e+005 6.11e+004**	4.86e+004 2.93e+004	+ **1.82e+004 1.26e+004**	1.33e+004 1.13e+004	= 1.16e+004 9.48e+003	9.23e+004 1.67e+004	= 9.19e+004 1.69e+004
F13	2.72e+001 1.56e+000	+ **2.23e+001 1.40e+000**	2.99e+000 1.60e−001	= 3.11e+000 1.76e−001	1.40e+001 4.89e+000	+ **5.97e+000 2.11e+000**	**5.47e+000 3.10e−001**	− 5.65e+000 2.85e−001
F14	2.31e+001 1.60e−001	= 2.30e+001 1.19e−001	2.25e+001 1.79e−001	= 2.26e+001 2.49e−001	2.29e+001 2.26e−001	= 2.29e+001 3.38e−001	2.22e+001 2.61e−001	= 2.22e+001 2.49e−001
F15	4.01e+002 1.87e−001	+ **3.76e+002 6.61e+001**	3.56e+002 8.21e+001	= 3.16e+002 9.78e+001	4.00e+002 5.72e−001	= 3.96e+002 2.00e+001	2.83e+002 9.46e+001	= 2.64e+002 1.11e+002
F16	3.13e+002 9.77e+000	+ **2.64e+002 1.20e+001**	8.26e+001 1.02e+001	+ **7.07e+001 1.05e+001**	**6.44e+001 5.41e+001**	− 8.02e+001 6.39e+001	**1.31e+002 1.12e+001**	− 1.44e+002 1.19e+001
F17	3.55e+002 1.15e+001	+ **3.01e+002 1.09e+001**	1.75e+002 1.65e+001	+ **1.58e+002 1.84e+001**	1.39e+002 9.37e+001	= 1.46e+002 9.52e+001	**2.10e+002 1.70e+001**	− 2.25e+002 1.28e+001
F18	9.38e+002 2.67e+000	+ **9.17e+002 8.47e−001**	**9.25e+002 3.61e+000**	− 9.27e+002 2.82e+000	**8.96e+002 4.93e+001**	− 9.17e+002 2.46e+001	9.45e+002 4.68e+001	+ **9.21e+002 3.67e+001**
F19	9.38e+002 2.52e+000	+ **9.17e+002 9.46e−001**	**9.24e+002 3.05e+000**	− 9.26e+002 3.12e+000	**8.63e+002 6.20e+001**	− 9.06e+002 4.12e+001	9.45e+002 3.41e+001	+ **8.91e+002 6.37e+001**
F20	9.38e+002 1.92e+000	+ **9.17e+002 8.89e−001**	9.25e+002 2.57e+000	= 9.26e+002 3.65e+000	**9.02e+002 4.53e+001**	− 9.14e+002 3.43e+001	9.52e+002 1.40e+001	+ **9.10e+002 4.90e+001**
F21	9.90e+002 1.26e+002	+ **7.85e+002 2.58e+002**	5.00e+002 1.53e−005	= 5.00e+002 1.39e−005	5.00e+002 1.39e−005	= 5.00e+002 1.39e−005	5.00e+002 1.25e−005	= 5.00e+002 1.25e−005
F22	9.70e+002 9.39e+000	+ **9.29e+002 3.32e+000**	9.41e+002 9.75e+000	= 9.40e+002 1.40e+001	9.58e+002 1.06e+001	= 9.60e+002 1.23e+001	9.83e+002 7.09e+000	= 9.88e+002 8.62e+000
F23	1.01e+003 4.56e+001	+ **8.62e+002 2.26e+002**	5.39e+002 3.47e−005	= 5.39e+002 5.57e−005	5.39e+002 1.26e−002	= 5.39e+002 1.98e−002	5.39e+002 6.68e−005	+ **5.39e+002 4.83e−005**
F24	2.00e+002 8.07e−002	+ **2.00e+002 2.20e−006**	2.00e+002 0.00e+000	= 2.00e+002 2.20e−006	2.00e+002 3.11e−006	= 2.00e+002 3.11e−006	2.41e+002 2.04e+002	= 2.00e+002 0.00e+000
F25	4.18e+002 2.87e+001	+ **2.30e+002 2.48e+000**	2.14e+002 7.65e−001	+ **2.14e+002 4.91e−001**	2.14e+002 3.07e−001	+ **2.13e+002 2.70e−001**	**2.16e+002 7.16e−001**	− 2.17e+002 8.70e−001
SUM (+/=/−)	–	22/3/0	–	5/16/4	–	6/14/5	–	7/13/5

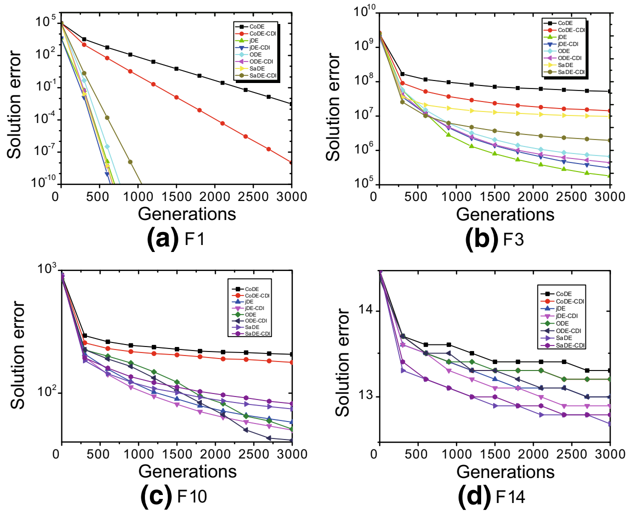

Fig. 6 Convergence graphs of the advanced DE variants and the corresponding DE-CDI for the selected functions at 30D

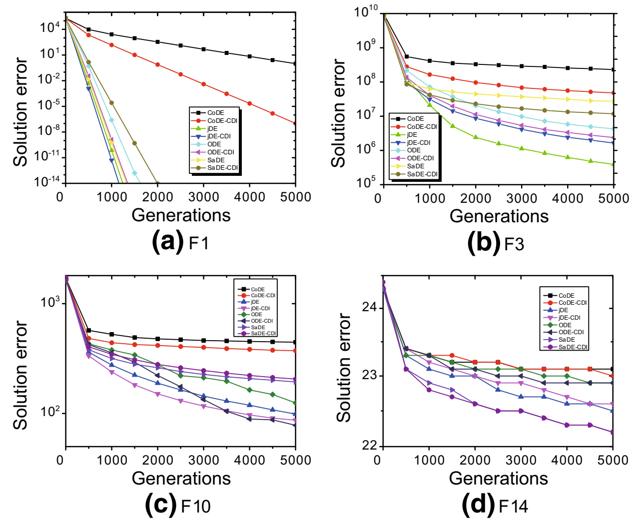

Fig. 7 Convergence graphs of the advanced DE variants and the corresponding DE-CDI for the selected functions at 50D

for the functions at 30D and 50D are presented in Tables 5 and 6, respectively.

In Table 5, it is obvious that DE-CDI exhibits significant improvements for most advanced DE variants. Specifically, for the DE variants with ensemble strategies, CoDE-CDI is significantly better than CoDE on 22 functions, while SaDE-CDI significantly outperforms SaDE on five functions. For jDE, DE-CDI significantly outperforms it on eight functions and is outperformed by it on two functions. For ODE, DE-CDI is significantly better on nine functions and is worse on two functions.

For the functions at 50D in Table 6, DE-CDI also significantly improves the performance of most DE variants. For CoDE and SaDE, DE-CDI is significantly better on 22 and 7 functions, respectively. For jDE, DE-CDI is significantly better on five functions. For ODE, DE-CDI can obtain the significant better results on six functions.

From Figs. 6 and 7, it is clear that DE-CDI is better than the advanced DE variants in terms of the convergence speed for most selected functions at 30D and 50D.

Furthermore, the multi-problem Wilcoxon signed rank tests at $\alpha = 0.05$ and $\alpha = 0.1$ are employed to show the significant differences between DE-CDI and its corresponding DE variant, and the results are shown in Tables 7 and 8. It is obvious that DE-CDI can obtain the higher $R+$ values than $R-$ values in most cases. These results indicate that DE-CDI is able to improve the performance of most advanced DE variants overall.

Summarily, the results of Tables 5, 6, 7 and 8 indicate that DE-CDI can also bring the beneficial to most advanced DE variants studied.

Table 7 Results of the multi-problem Wilcoxon's test for DE-CDI versus the advanced DE algorithm for all the functions at 30D

Algorithm	$+/=/-$	$R+$	$R-$	p value	$\alpha = 0.05$	$\alpha = 0.1$
CoDE-CDI vs CoDE	22/3/0	320	5	1.60E−05	+	+
jDE-CDI vs jDE	8/15/2	197	128	3.30E−01	=	=
ODE-CDI vs ODE	9/14/2	269.5	55.5	3.71E−03	+	+
SaDE-CDI vs SaDE	5/15/5	166.5	133.5	6.27E−01	=	=

Table 8 Results of the multi-problem Wilcoxon's test for DE-CDI versus the advanced DE algorithm for all the functions at 50D

Algorithm	$+/=/-$	$R+$	$R-$	p value	$\alpha = 0.05$	$\alpha = 0.1$
CoDE-CDI vs CoDE	22/3/0	323.5	1.5	1.10E−05	+	+
jDE-CDI vs jDE	5/16/4	166	134	6.35E−01	=	=
ODE-CDI vs ODE	6/14/5	190	110	2.47E−01	=	=
SaDE-CDI vs SaDE	7/13/5	246.5	78.5	2.30E−02	+	+

Table 9 Mean and standard deviation of the best error values obtained by DE-CDI and NDi-DE on all the functions at 30D

Function	NDi-DE/rand/1	DE-CDI/rand/1	NDi-DE/best/1	DE-CDI/best/1	NDi-CoDE	CoDE-CDI	NDi-ODE	ODE-CDI
F1	1.79e−028 1.54e−028	= 1.94e−028 1.49e−028	1.97e+000 3.44e+000	= 1.72e+001 4.95e+001	2.32e−004 1.97e−004	+ **1.05e−008 4.65e−009**	2.45e−028 2.30e−028	= 1.99e−028 1.89e−028
F2	**1.59e−020 2.07e−020**	− 9.65e−013 1.53e−012	1.19e−012 7.27e−013	+ **1.56e−002 7.27e−002**	7.30e+003 2.56e+003	+ **6.85e+002 1.49e+002**	**1.01e−012 1.54e−012**	− 1.07e−005 7.43e−006
F3	**1.02e+005 4.96e+004**	− 1.39e+005 6.23e+004	3.96e+004 3.34e+004	+ **3.53e+005 1.71e+005**	4.27e+007 9.49e+006	+ **1.42e+007 3.33e+006**	**2.03e+005 9.69e+004**	− 4.41e+005 1.79e+005
F4	**7.48e−011 1.19e−010**	− 5.21e−006 1.17e−005	2.24e−002 1.12e−001	+ **5.10e−001 6.59e−001**	1.43e+004 3.35e+003	+ **2.60e+003 5.65e+002**	**4.00e−006 1.09e−005**	− 2.32e−002 2.61e−002
F5	**4.32e+001 7.15e+001**	− 2.89e+002 1.68e+002	2.08e+003 6.08e+002	+ **2.79e+003 5.33e+002**	3.99e+003 5.12e+002	+ **1.22e+003 3.20e+002**	3.04e+001 4.04e+001	= 3.59e+001 2.94e+001
F6	**6.43e−001 1.49e+000**	− 3.17e+001 2.25e+001	5.08e+004 1.51e+005	+ **1.41e+007 3.80e+007**	1.28e+002 6.55e+001	+ **2.76e+001 5.24e+000**	3.60e+001 2.42e+001	+ **1.81e+000 2.35e+000**
F7	**6.89e−003 1.06e−002**	− 2.43e−002 1.74e−002	5.47e+003 1.92e+002	+ **3.56e+000 6.33e+000**	2.98e+000 1.83e+000	+ **2.51e−001 9.95e−002**	7.88e−003 1.06e−002	= 3.94e−003 7.61e−003
F8	2.10e+001 6.23e−002	+ **2.09e+001 5.31e−002**	2.09e+001 7.29e−002	= 2.10e+001 5.59e−002	2.10e+001 4.40e−002	= 2.09e+001 6.40e−002	2.09e+001 7.30e−002	= 2.10e+001 4.45e−002
F9	3.02e+001 8.73e+000	= 2.84e+001 7.30e+000	8.23e+001 1.94e+001	+ **6.75e+001 1.88e+001**	**7.20e+000 4.08e+000**	− 1.71e+001 1.91e+000	2.47e+001 5.98e+000	= 2.49e+001 1.28e+001
F10	4.41e+001 1.34e+001	= 4.34e+001 2.12e+001	1.05e+002 4.35e+001	= 9.42e+001 2.10e+001	1.99e+002 1.95e+001	+ **1.78e+002 1.13e+001**	3.77e+001 1.11e+001	= 4.14e+001 2.69e+001
F11	**1.53e+001 5.07e+000**	− 3.10e+001 6.60e+000	**2.33e+001 5.52e+000**	− 2.91e+001 5.51e+000	3.52e+001 1.44e+000	+ **3.44e+001 1.36e+000**	**8.81e+000 3.52e+000**	− 1.45e+001 9.96e+000
F12	1.64e+003 3.10e+003	= 4.12e+003 8.03e+003	4.03e+004 2.25e+004	+ **5.13e+003 4.19e+003**	1.30e+005 2.89e+004	= 1.16e+005 1.89e+004	1.67e+003 2.35e+003	= 1.64e+003 1.95e+003
F13	3.41e+000 8.21e−001	+ **2.86e+000 6.36e−001**	6.19e+000 1.90e+000	+ **3.79e+000 9.85e−001**	8.73e+000 1.04e+000	= 8.71e+000 6.58e−001	3.19e+000 6.25e−001	= 3.03e+000 7.50e−001
F14	**1.27e+001 5.66e−001**	− 1.32e+001 1.83e−001	1.19e+001 5.66e−001	= 1.21e+001 6.59e−001	1.33e+001 1.38e−001	+ **1.32e+001 1.31e−001**	**1.26e+001 4.04e−001**	− 1.30e+001 3.46e−001
F15	4.06e+002 5.05e+001	= 4.04e+002 6.76e+001	4.06e+002 7.98e+001	= 4.12e+002 5.37e+001	4.00e+002 9.53e−004	+ **3.92e+002 3.99e+001**	4.04e+002 6.11e+001	= 4.12e+002 3.34e+001
F16	6.84e+001 3.01e+001	= 5.53e+001 1.25e+001	2.13e+002 1.33e+002	+ **1.39e+002 4.69e+001**	2.18e+002 1.40e+001	+ **1.98e+002 7.80e+000**	**5.73e+001 1.69e+001**	− 8.52e+001 5.97e+001
F17	6.25e+001 1.29e+001	= 5.88e+001 2.61e+001	2.35e+002 1.38e+002	+ **1.75e+002 1.26e+002**	2.51e+002 1.36e+001	+ **2.31e+002 1.37e+001**	**6.17e+001 1.93e+001**	− 1.35e+002 7.60e+001
F18	8.99e+002 2.99e+001	= 8.90e+002 4.60e+001	9.20e+002 3.37e+001	= 9.03e+002 6.05e+001	9.07e+002 3.17e−001	+ **9.06e+002 2.03e−001**	8.98e+002 2.96e+001	= 8.97e+002 2.92e+001
F19	**8.90e+002 4.01e+001**	− 8.98e+002 3.72e+001	9.20e+002 2.43e+001	= 9.05e+002 6.12e+001	9.07e+002 4.46e−001	+ **9.06e+002 1.74e−001**	8.82e+002 4.68e+001	= 9.05e+002 1.51e+000
F20	8.82e+002 4.71e+001	= 8.80e+002 5.09e+001	9.21e+002 2.43e+001	= 8.96e+002 6.12e+001	9.07e+002 4.09e−001	+ **9.06e+002 1.77e−001**	8.94e+002 3.55e+001	= 8.93e+002 3.49e+001
F21	5.00e+002 0.00e+000	= 5.12e+002 6.00e+001	6.41e+002 2.51e+002	= 7.53e+002 3.23e+002	5.00e+002 1.03e−004	+ **5.00e+002 0.00e+000**	5.12e+002 6.00e+001	= 5.00e+002 0.00e+000
F22	8.95e+002 1.22e+001	= 8.99e+002 1.29e+001	9.91e+002 4.37e+001	+ **9.36e+002 2.51e+001**	9.42e+002 1.23e+001	+ **9.02e+002 9.69e+000**	**8.97e+002 1.32e+001**	− 9.05e+002 9.22e+000
F23	5.34e+002 8.52e−003	= 5.34e+002 1.18e−002	7.94e+002 2.50e+002	= 7.05e+002 2.82e+002	5.34e+002 4.51e−004	+ **5.34e+002 3.70e−004**	5.34e+002 1.66e−003	= 5.34e+002 3.15e−004
F24	2.00e+002 0.00e+000	= 2.00e+002 0.00e+000	2.00e+002 9.25e−001	= 2.00e+002 0.00e+000	2.00e+002 7.93e−005	+ **2.00e+002 4.92e−006**	2.00e+002 0.00e+000	= 2.00e+002 0.00e+000
F25	2.21e+002 5.82e+001	+ **2.09e+002 1.91e−001**	1.68e+003 1.61e+001	+ **2.52e+002 2.13e+002**	2.37e+002 7.88e+000	+ **2.16e+002 1.43e+000**	2.09e+002 1.17e−001	= 2.09e+002 1.31e−001
SUM (+/ = /−)	–	3/13/9	–	13/11/1	–	21/3/1	–	1/16/8

Table 10 Mean and standard deviation of the best error values obtained by DE-CDI and NDi-DE on all the functions at 50D

Function	NDi-DE/rand/1	DE-CDI/rand/1	NDi-DE/best/1	DE-CDI/best/1	NDi-CoDE	CoDE-CDI	NDi-ODE	ODE-CDI
F1	9.26e−028 4.53e−028	= 7.43e−028 3.84e−028	7.73e+003 2.83e+003	+ **1.77e+002 4.95e+002**	7.23e−003 8.86e−003	+ **9.91e−008 3.85e−008**	7.50e−012 3.75e−011	= 1.40e−027 1.46e−027
F2	**4.89e−010 8.21e−010**	− 1.26e−006 1.42e−006	1.50e+004 5.63e+003	+ **1.72e−002 5.78e−002**	4.50e+004 6.78e+003	+ **8.46e+003 1.95e+003**	**3.03e−005 2.97e−005**	− 1.39e+000 1.16e+000
F3	**1.39e+005 5.39e+004**	− 2.39e+005 9.09e+004	7.88e+007 5.34e+007	+ **1.04e+006 2.96e+005**	1.75e+008 2.97e+007	+ **4.81e+007 1.10e+007**	**3.30e+005 1.26e+005**	− 2.38e+006 5.76e+005
F4	**7.11e−001 8.99e−001**	− 1.79e+001 2.21e+001	2.26e+003 1.99e+003	= 1.64e+003 1.55e+003	6.23e+004 7.83e+003	+ **2.14e+004 3.80e+003**	**1.63e+001 2.66e+001**	− 4.92e+002 3.75e+002
F5	**2.36e+003 3.36e+002**	− 3.20e+003 4.02e+002	1.39e+004 2.29e+003	+ **7.04e+003 1.04e+003**	1.07e+004 1.43e+003	+ **5.13e+003 4.51e+002**	2.44e+003 3.73e+002	+ **2.14e+003 3.71e+002**
F6	**2.04e+001 1.66e+001**	− 1.55e+002 2.52e+002	1.50e+009 8.23e+008	+ **3.96e+007 1.12e+008**	4.25e+002 2.03e+002	+ **5.33e+001 2.82e+001**	1.26e+003 4.90e+003	+ **4.38e+001 2.18e+001**
F7	**7.18e−003 1.16e−002**	− 1.28e−002 1.53e−002	7.39e+003 3.30e+002	+ **9.87e+000 1.62e+001**	5.35e+000 3.67e+000	+ **1.39e−001 6.84e−002**	**2.07e−003 4.59e−003**	− 2.17e−003 5.14e−003
F8	2.11e+001 4.00e−002	= 2.11e+001 3.90e−002	2.11e+001 3.45e−002	= 2.11e+001 4.61e−002	2.11e+001 3.08e−002	= 2.11e+001 5.28e−002	2.11e+001 3.39e−002	= 2.11e+001 3.68e−002
F9	7.66e+001 2.07e+001	= 8.17e+001 1.18e+001	2.82e+002 4.18e+001	+ **1.67e+002 2.49e+001**	**2.93e+001 1.09e+001**	− 7.32e+001 4.47e+000	6.63e+001 1.91e+001	+ **4.40e+001 1.22e+001**
F10	7.83e+001 2.04e+001	= 7.41e+001 1.47e+001	4.11e+002 5.47e+001	+ **2.30e+002 4.79e+001**	4.07e+002 1.76e+001	+ **3.73e+002 1.13e+001**	9.01e+001 2.56e+001	+ **7.80e+001 3.92e+001**
F11	**3.94e+001 1.06e+001**	− 6.80e+001 6.58e+000	**5.13e+001 5.28e+000**	− 5.98e+001 4.93e+000	6.85e+001 1.56e+000	+ **6.76e+001 1.54e+000**	2.67e+001 7.87e+000	= 3.02e+001 1.32e+001
F12	7.51e+003 6.33e+003	= 9.75e+003 7.55e+003	5.14e+005 1.56e+005	+ **2.50e+004 1.76e+004**	6.09e+005 1.33e+005	= 6.30e+005 6.11e+004	1.18e+004 1.06e+004	= 1.16e+004 9.48e+003
F13	6.63e+000 1.60e+000	= 5.91e+000 1.52e+000	2.64e+001 7.55e+000	+ **9.79e+000 2.77e+000**	**2.00e+001 3.07e+000**	− 2.23e+001 1.40e+000	5.97e+000 1.24e+000	= 5.97e+000 2.11e+000
F14	2.26e+001 5.71e−001	= 2.29e+001 2.34e−001	2.16e+001 6.39e−001	+ **2.04e+001 6.91e−001**	2.31e+001 1.36e−001	+ **2.30e+001 1.19e−001**	**2.26e+001 4.79e−001**	− 2.29e+001 3.38e−001
F15	3.54e+002 8.48e+001	= 3.77e+002 6.56e+001	5.69e+002 7.86e+001	+ **3.89e+002 3.94e+001**	**3.69e+002 7.27e+001**	− 3.76e+002 6.61e+001	3.76e+002 6.63e+001	= 3.96e+002 2.00e+001
F16	5.96e+001 1.38e+001	= 5.98e+001 1.15e+001	3.54e+002 9.76e+001	+ **1.50e+002 3.98e+001**	2.89e+002 1.57e+001	+ **2.64e+002 1.20e+001**	5.85e+001 1.44e+001	= 8.02e+001 6.39e+001
F17	6.41e+001 1.89e+001	= 5.90e+001 8.75e+000	2.96e+002 8.18e+001	+ **1.83e+002 7.90e+001**	3.28e+002 1.81e+001	+ **3.01e+002 1.09e+001**	**7.12e+001 3.12e+001**	− 1.46e+002 9.52e+001
F18	8.93e+002 1.29e+002	= 9.20e+002 4.61e+001	1.04e+003 3.98e+001	+ **1.02e+003 1.82e+001**	9.27e+002 3.54e+000	+ **9.17e+002 8.47e−001**	9.10e+002 4.35e+001	= 9.17e+002 2.46e+001
F19	**9.18e+002 3.69e+001**	− 9.37e+002 2.98e+001	1.05e+003 2.81e+001	+ **1.03e+003 1.59e+001**	9.27e+002 2.81e+000	+ **9.17e+002 9.46e−001**	9.17e+002 3.54e+001	+ **9.06e+002 4.12e+001**
F20	**9.19e+002 3.68e+001**	− 9.44e+002 1.22e+001	1.04e+003 3.12e+001	= 1.02e+003 2.19e+001	9.26e+002 2.60e+000	+ **9.17e+002 8.89e−001**	**9.09e+002 4.85e+001**	− 9.14e+002 3.43e+001
F21	5.24e+002 8.31e+001	= 5.00e+002 1.25e−005	1.13e+003 1.37e+002	+ **8.40e+002 3.58e+002**	9.40e+002 1.86e+002	+ **7.85e+002 2.58e+002**	5.00e+002 1.08e−005	= 5.00e+002 1.39e−005
F22	**9.51e+002 1.29e+001**	− 9.62e+002 7.33e+000	1.11e+003 5.63e+001	+ **1.01e+003 2.11e+001**	9.57e+002 1.01e+001	+ **9.29e+002 3.32e+000**	9.55e+002 1.29e+001	= 9.60e+002 1.23e+001
F23	5.67e+002 9.73e+001	= 5.65e+002 1.28e+002	1.17e+003 1.07e+002	+ **9.69e+002 3.19e+002**	9.62e+002 1.53e+002	+ **8.62e+002 2.26e+002**	5.53e+002 7.00e+001	+ **5.39e+002 1.98e−002**
F24	2.00e+002 3.11e−006	= 2.00e+002 3.11e−006	1.26e+003 2.41e+001	+ **1.20e+003 2.08e+002**	2.00e+002 2.71e−003	+ **2.00e+002 2.20e−006**	2.00e+002 3.11e−006	= 2.00e+002 3.11e−006
F25	**2.14e+002 3.47e−001**	− 2.15e+002 5.42e−001	1.80e+003 1.83e+001	+ **8.83e+002 5.09e+002**	3.16e+002 2.70e+001	+ **2.30e+002 2.48e+000**	2.13e+002 3.05e−001	= 2.13e+002 2.70e−001
SUM (+/ = /−)	–	0/14/11	–	21/3/1	–	20/2/3	–	6/12/7

Table 11 Results of the multi-problem Wilcoxon's test for DE-CDI versus NDi-DE for all the functions at 30D

Algorithm	+/ = /−	$R+$	$R-$	p value	$\alpha = 0.05$	$\alpha = 0.1$
DE-CDI/rand/1 vs NDi-DE/rand/1	3/13/9	133	192	1.00e+00	=	=
DE-CDI/best/1 vs NDi-DE/best/1	13/11/1	182	118	3.53E−01	=	=
CoDE-CDI vs NDi-CoDE	21/3/1	285.5	14.5	6.70E−05	+	+
ODE-CDI vs NDi-ODE	1/16/8	101.5	198.5	1.00e+00	=	=

Table 12 Results of the multi-problem Wilcoxon's test for DE-CDI versus NDi-DE for all the functions at 50D

Algorithm	+/ = /−	$R+$	$R-$	p value	$\alpha = 0.05$	$\alpha = 0.1$
DE-CDI/rand/1 vs NDi-DE/rand/1	0/14/11	63.5	261.5	1.00e+00	−	−
DE-CDI/best/1 vs NDi-DE/best/1	21/3/1	298	2	1.70E−05	+	+
CoDE-CDI vs NDi-CoDE	20/2/3	271.5	53.5	3.11E−03	+	+
ODE-CDI vs NDi-ODE	6/12/7	134	166	1.00e+00	=	=

Table 13 Average computational time (in seconds) used by all the DE algorithms and their corresponding NDi-DE and DE-CDI variants

Function	30D					50D				
	DE	NDi-DE	DE-CDI	Ratio1	Ratio2	DE	NDi-DE	DE-CDI	Ratio1	Ratio2
F1	3.588	33.821	3.995	1.113	8.466	16.257	80.813	16.685	1.026	4.843
F2	3.915	19.148	4.625	1.181	4.140	13.785	48.880	13.941	1.011	3.506
F3	3.988	15.287	4.525	1.135	3.378	13.042	45.801	13.341	1.023	3.433
F4	4.026	15.816	4.660	1.157	3.394	13.724	44.463	13.947	1.016	3.188
F5	2.578	15.026	2.925	1.135	5.137	7.408	39.327	7.565	1.021	5.199
F6	6.252	26.721	6.604	1.056	4.046	17.615	76.757	17.908	1.017	4.286
F7	4.422	19.706	4.893	1.107	4.027	13.868	62.276	14.243	1.027	4.372
F8	4.454	21.863	4.868	1.093	4.491	13.553	42.533	13.945	1.029	3.050
F9	4.356	18.362	4.761	1.093	3.857	13.594	57.956	13.757	1.012	4.213
F10	4.339	16.962	4.782	1.102	3.547	13.532	54.141	13.758	1.017	3.935
F11	102.666	115.593	103.142	1.005	1.121	218.216	322.655	218.815	1.003	1.475
F12	95.387	114.472	97.527	1.022	1.174	331.804	489.458	332.436	1.002	1.472
F13	7.472	21.570	7.514	1.006	2.871	25.926	71.932	26.241	1.012	2.741
F14	13.579	28.485	13.592	1.001	2.096	42.778	69.282	42.982	1.005	1.612
F15	230.323	250.545	232.197	1.008	1.079	689.966	742.986	694.158	1.006	1.070
F16	230.438	244.753	232.187	1.008	1.054	692.254	728.524	694.030	1.003	1.050
F17	230.900	242.120	232.484	1.007	1.041	691.545	716.976	694.295	1.004	1.033
F18	230.669	244.085	232.834	1.009	1.048	695.526	727.576	696.387	1.001	1.045
F19	230.852	244.428	232.819	1.009	1.050	693.948	731.694	696.132	1.003	1.051
F20	230.554	244.043	232.650	1.009	1.049	692.988	729.835	693.007	1.000	1.053
F21	254.672	270.637	256.856	1.009	1.054	759.837	802.121	760.931	1.001	1.054
F22	257.338	270.045	260.303	1.012	1.037	769.463	795.093	770.067	1.001	1.032
F23	256.583	273.930	258.809	1.009	1.058	763.027	803.024	769.837	1.009	1.043
F24	160.112	179.249	161.538	1.009	1.110	496.938	547.394	497.288	1.001	1.101
F25	161.033	172.423	161.877	1.005	1.065	497.525	532.460	499.864	1.005	1.065
Avg.	109.380	124.764	110.519	1.010	1.129	327.925	374.558	329.022	1.003	1.138

Ratio1 means the value that the cost of DE-CDI is divided by that of the original DE algorithm. Ratio2 means defined as the value that the cost of NDi-DE is divided by that of the DE-CDI

Table 14 Mean and standard deviation of the best error values obtained by DMDE, cDMDE and DE-CDI on all the functions at 30D

Function	DE-CDI/rand/2		DMDE/rand/2		cDMDE/rand/2	DE-CDI/best/1		DMDE/best/1		cDMDE/best/1
F1	**1.86e−012 8.53e−013**	−	9.53e−001 4.29e−001	−	1.87e−009 1.05e−009	1.72e+001 4.95e+001	−	2.78e+003 2.26e+003	+	**2.46e−027 4.49e−027**
F2	**4.38e+000 2.37e+000**	−	7.72e+003 1.73e+003	−	2.32e+001 6.75e+000	1.56e−002 7.27e−002	−	1.12e+004 4.60e+003	+	**1.26e−025 2.76e−025**
F3	**1.05e+006 4.41e+005**	−	4.77e+007 1.29e+007	−	2.03e+006 5.80e+005	3.53e+005 1.71e+005	−	3.43e+007 2.01e+007	+	**2.41e+004 1.78e+004**
F4	**8.57e+001 3.85e+001**	−	1.51e+004 2.25e+003	−	3.16e+002 9.31e+001	5.10e−001 6.59e−001	−	6.97e+002 7.86e+002	+	**3.51e−005 1.46e−004**
F5	**1.21e+003 2.38e+002**	−	8.18e+003 7.53e+002	−	1.94e+003 3.19e+002	2.79e+003 5.33e+002	−	1.01e+004 2.18e+003	+	**1.91e+003 6.76e+002**
F6	**1.61e+001 1.63e+001**	−	5.66e+003 3.46e+003	−	1.88e+001 7.22e−001	1.41e+007 3.80e+007	−	1.84e+008 2.50e+008	+	**9.57e−001 1.74e+000**
F7	**3.69e−005 3.94e−005**	−	5.38e+000 1.38e+000	−	1.45e−003 8.63e−004	3.56e+000 6.33e+000	−	1.10e+003 4.63e+002	+	**2.96e−002 3.17e−002**
F8	2.09e+001 6.41e−002	=	2.09e+001 6.34e−002	=	2.09e+001 4.93e−002	2.10e+001 5.59e−002	=	2.09e+001 4.87e−002	=	2.10e+001 5.54e−002
F9	1.94e+002 1.08e+001	−	2.11e+002 1.05e+001	=	1.99e+002 1.02e+001	6.75e+001 1.88e+001	−	1.30e+002 3.43e+001	+	**5.30e+001 1.19e+001**
F10	2.05e+002 9.29e+000	−	2.43e+002 1.71e+001	=	2.10e+002 1.05e+001	9.42e+001 2.10e+001	−	2.08e+002 6.05e+001	+	**7.74e+001 1.98e+001**
F11	3.97e+001 1.14e+000	=	3.95e+001 1.25e+000	=	3.95e+001 1.27e+000	2.91e+001 5.51e+000	+	**2.55e+001 2.74e+000**	=	2.66e+001 3.28e+000
F12	**2.70e+005 8.67e+004**	−	5.06e+005 6.77e+004	−	3.40e+005 8.02e+004	5.13e+003 4.19e+003	−	1.63e+005 4.63e+004	=	3.53e+003 5.45e+003
F13	**1.68e+001 8.94e−001**	−	2.07e+001 1.15e+000	−	1.74e+001 8.47e−001	3.79e+000 9.85e−001	−	1.37e+001 4.19e+000	=	4.03e+000 1.59e+000
F14	1.34e+001 1.94e−001	=	1.34e+001 1.49e−001	=	1.34e+001 1.39e−001	1.21e+001 6.59e−001	=	1.20e+001 4.80e−001	+	**1.17e+001 6.30e−001**
F15	3.96e+002 4.52e+001	−	4.07e+002 8.90e+000	=	4.00e+002 0.00e+000	4.12e+002 5.37e+001	−	6.31e+002 1.25e+002	=	4.32e+002 5.91e+001
F16	2.29e+002 1.02e+001	−	2.65e+002 1.43e+001	=	2.32e+002 1.40e+001	1.39e+002 4.69e+001	−	3.60e+002 1.30e+002	=	1.90e+002 1.37e+002
F17	2.55e+002 1.15e+001	−	3.04e+002 1.38e+001	=	2.61e+002 1.59e+001	1.75e+002 1.26e+002	−	3.17e+002 1.28e+002	=	1.46e+002 1.04e+002
F18	9.09e+002 1.28e+000	−	9.40e+002 3.29e+000	+	**9.06e+002 2.21e+001**	9.03e+002 6.05e+001	−	1.03e+003 5.10e+001	=	9.01e+002 5.23e+001
F19	**9.09e+002 1.10e+000**	−	9.39e+002 4.80e+000	−	9.10e+002 1.38e+000	9.05e+002 6.12e+001	−	1.02e+003 3.35e+001	=	9.19e+002 3.69e+001
F20	**9.09e+002 1.40e+000**	−	9.40e+002 3.85e+000	−	9.10e+002 1.09e+000	8.96e+002 6.12e+001	−	1.03e+003 4.38e+001	=	9.04e+002 5.38e+001
F21	5.00e+002 0.00e+000	−	5.00e+002 9.40e−002	=	5.00e+002 0.00e+000	7.53e+002 3.23e+002	−	1.13e+003 1.47e+002	+	**6.97e+002 3.01e+002**
F22	**9.35e+002 6.42e+000**	−	1.02e+003 8.65e+000	−	9.45e+002 7.46e+000	9.36e+002 2.51e+001	−	1.05e+003 6.66e+001	=	9.38e+002 2.71e+001
F23	**5.34e+002 1.70e−004**	−	5.35e+002 1.15e+000	−	5.34e+002 2.91e−004	7.05e+002 2.82e+002	−	1.19e+003 6.21e+001	=	8.24e+002 2.30e+002
F24	2.00e+002 0.00e+000	−	2.00e+002 1.68e−001	=	2.00e+002 0.00e+000	2.00e+002 0.00e+000	−	1.17e+003 2.28e+002	=	2.00e+002 0.00e+000
F25	**2.09e+002 2.27e−001**	−	2.25e+002 3.80e+000	−	2.10e+002 2.45e−001	2.52e+002 2.13e+002	−	1.40e+003 6.07e+001	+	**2.09e+002 1.99e+000**
SUM[a] (+/ = /−)	–		0/3/22		1/10/14	–		1/2/22		12/13/0
ARV	1.30		2.78		1.92	1.80		2.80		1.40
Rank	**1**		3		2	2		3		**1**
Function	CoDE-CDI		DMCoDE		cDMCoDE	ODE-CDI		DMODE		cDMODE
F1	**1.05e−008 4.65e−009**	−	1.07e−005 2.37e−005	−	1.59e−006 1.95e−006	1.99e−028 1.89e−028	=	2.24e−028 1.74e−028	=	2.53e−028 1.75e−028
F2	**6.85e+002 1.49e+002**	−	1.14e+004 1.62e+003	−	8.91e+003 1.30e+003	**1.07e−005 7.43e−006**	−	2.21e−004 2.84e−004	−	4.52e−004 3.33e−004
F3	**1.42e+007 3.33e+006**	−	4.83e+007 9.29e+006	−	4.22e+007 7.55e+006	**4.41e+005 1.79e+005**	−	5.68e+005 2.37e+005	−	7.30e+005 4.03e+005
F4	**2.60e+003 5.65e+002**	−	1.77e+004 2.71e+003	−	1.59e+004 2.57e+003	**2.32e−002 2.61e−002**	−	6.93e−002 6.74e−002	−	2.66e−001 2.96e−001
F5	**1.22e+003 3.20e+002**	−	4.99e+003 4.95e+002	−	4.99e+003 5.24e+002	**3.59e+001 2.94e+001**	−	1.26e+002 8.79e+001	−	1.79e+002 9.06e+001

Table 14 continued

Function	CoDE-CDI		DMCoDE		cDMCoDE	ODE-CDI		DMODE		cDMODE
F6	**2.76e+001 5.24e+000**	–	2.46e+002 7.29e+001	–	2.01e+002 7.06e+001	**1.81e+000 2.35e+000**	–	3.95e+001 2.36e+001	–	6.90e+003 3.42e+004
F7	**2.51e−001 9.95e−002**	–	5.50e+000 4.07e+000	–	2.62e+000 1.61e+000	3.94e−003 7.61e−003	=	2.96e−004 1.48e−003	=	1.58e−003 3.69e−003
F8	2.09e+001 6.40e−002	=	2.09e+001 3.78e−002	=	2.09e+001 5.63e−002	2.10e+001 4.45e−002	=	2.10e+001 5.93e−002	=	2.09e+001 7.70e−002
F9	1.71e+001 1.91e+000	+	**1.40e+001 2.43e+000**	+	**1.36e+001 2.31e+000**	**2.49e+001 1.28e+001**	–	7.57e+001 2.72e+001	–	4.04e+001 1.96e+001
F10	**1.78e+002 1.13e+001**	–	2.07e+002 7.77e+000	–	2.02e+002 1.27e+001	4.14e+001 2.69e+001	=	4.92e+001 4.13e+001	=	4.25e+001 2.30e+001
F11	3.44e+001 1.36e+000	=	3.50e+001 1.15e+000	=	3.51e+001 1.42e+000	1.45e+001 9.96e+000	+	**8.33e+000 1.04e+001**	=	1.00e+001 7.20e+000
F12	**1.16e+005 1.89e+004**	–	1.61e+005 1.91e+004	–	1.46e+005 1.99e+004	1.64e+003 1.95e+003	=	2.34e+003 1.70e+003	=	2.43e+003 2.74e+003
F13	**8.71e+000 6.58e−001**	–	9.30e+000 7.27e−001	–	9.21e+000 7.39e−001	**3.03e+000 7.50e−001**	–	7.66e+000 2.27e+000	–	3.99e+000 1.93e+000
F14	1.32e+001 1.31e−001	–	1.33e+001 1.26e−001	=	1.33e+001 1.58e−001	1.30e+001 3.46e−001	=	1.30e+001 3.14e−001	=	1.28e+001 3.34e−001
F15	3.92e+002 3.99e+001	=	3.88e+002 5.95e+001	=	3.88e+002 4.38e+001	4.12e+002 3.34e+001	=	4.32e+002 4.76e+001	=	4.04e+002 2.00e+001
F16	**1.98e+002 7.80e+000**	–	2.44e+002 2.63e+001	–	2.27e+002 2.05e+001	8.52e+001 5.97e+001	=	1.15e+002 7.85e+001	=	6.17e+001 3.32e+001
F17	**2.31e+002 1.37e+001**	–	2.67e+002 2.09e+001	–	2.53e+002 1.58e+001	1.35e+002 7.60e+001	=	1.30e+002 8.71e+001	=	1.22e+002 7.82e+001
F18	**9.06e+002 2.03e−001**	–	9.07e+002 4.22e−001	–	9.08e+002 3.90e−001	8.97e+002 2.92e+001	=	8.92e+002 3.48e+001	=	8.76e+002 4.84e+001
F19	**9.06e+002 1.74e−001**	–	9.08e+002 4.28e−001	–	9.08e+002 4.55e−001	9.05e+002 1.51e+000	=	8.96e+002 2.90e+001	=	8.93e+002 3.50e+001
F20	**9.06e+002 1.77e−001**	–	9.08e+002 4.55e−001	–	9.08e+002 3.86e−001	8.93e+002 3.49e+001	=	8.97e+002 2.91e+001	=	8.89e+002 3.97e+001
F21	5.00e+002 0.00e+000	=	5.00e+002 6.23e−006	=	5.00e+002 0.00e+000	5.00e+002 0.00e+000	=	5.00e+002 0.00e+000	=	5.12e+002 6.00e+001
F22	**9.02e+002 9.69e+000**	–	9.54e+002 1.04e+001	–	9.43e+002 9.74e+000	9.05e+002 9.22e+000	=	9.09e+002 1.08e+001	–	9.13e+002 1.03e+001
F23	5.34e+002 3.70e−004	+	**5.34e+002 3.65e−004**	+	**5.34e+002 3.43e−004**	5.34e+002 3.15e−004	=	5.34e+002 2.96e−004	=	5.50e+002 8.06e+001
F24	2.00e+002 4.92e−006	=	2.00e+002 8.81e−006	=	2.00e+002 0.00e+000	2.00e+002 0.00e+000	=	2.00e+002 0.00e+000	=	2.00e+002 0.00e+000
F25	**2.16e+002 1.43e+000**	–	2.44e+002 8.28e+000	–	2.32e+002 6.13e+000	**2.09e+002 1.31e−001**	–	2.09e+002 2.13e−001	–	2.10e+002 3.01e−001
SUM[a] (+/ = /−)	–		2/5/18		2/6/17	–		1/16/8		0/16/9
ARV	1.32		2.58		2.10	1.74		2.14		2.12
Rank	**1**		3		2	**1**		3		2

The average ranking values (ARV) by Friedman test are also shown in the table

[a] The number of functions that the competitor wins, ties and loses when compared with the corresponding DE-CDI based on Wilcoxon signed-rank test at $\alpha = 0.05$

Table 15 Results of the multi-problem Wilcoxon's test between DMDE, cDMDE and DE-CDI for all the functions at 30D

Algorithm	(1)	(2)	(3)	Algorithm	(1)	(2)	(3)
DE-CDI/rand/2 (1)	–	315.0[a]	274.5[a]	DE-CDI/best/1 (1)	–	319.0[a]	92.0
DMDE/rand/2 (2)	10.0[b]	–	5.0[b]	DMDE/best/1 (2)	6.0[b]	–	4.0[b]
cDMDE/rand/2 (3)	25.5[b]	295.0[a]	–	cDMDE/best/1(3)	233.0[a]	321.0[a]	-
CoDE-CDI (1)	–	295.0[a]	293.0[a]	ODE-CDI (1)	–	242.5[a]	185.0
DMCoDE (2)	30.0[b]	–	42.0[b]	DMODE (2)	82.5[b]	–	157.5
cDMCoDE (3)	32.0[b]	258.0[a]	–	cDMODE (3)	115.0	142.5	-

Upper diagonal of level significance $\alpha = 0.9$, lower diagonal level of significance $\alpha = 0.95$
[a] The method in the row improves the method of the column
[b] The method in the column improves the method of the row

5.5 Comparison with NDi-DE

For investigating the effectiveness of DE-CDI with both neighborhood and direction information, a DE framework, i.e., NDi-DE (Cai and Wang 2013), is considered. Two original DE algorithms, i.e., DE/rand/1 and DE/best/1, and two advanced DE variants, i.e., CoDE and ODE, are used here for comparison. The results for all the functions at 30D and 50D are shown in Tables 9 and 10, respectively.

In Table 9, we can find that DE-CDI can obtain the comparable results to NDi-DE for the functions at 30D. For DE/rand/1 and DE/best/1, DE-CDI is significantly better than NDi-DE on 3 and 13 functions, respectively. For CoDE and ODE, DE-CDI can obtain the significant better results on 21 and 1 functions, respectively.

For the functions at 50D, the results of Table 10 show that DE-CDI consistently perform as well as NDi-DE. Specifically, for DE/rand/1 and DE/best/1, DE-CDI is significantly better than NDi-DE on 0 and 21 functions, respectively. For CoDE and ODE, DE-CDI significantly outperforms NDi-DE on 20 and 6 functions, respectively.

Furthermore, the multi-problem Wilcoxon signed rank tests at $\alpha = 0.05$ and $\alpha = 0.1$ are carried out between DE-CDI and NDi-DE. Tables 11 and 12 present the results for functions at 30D and 50D, respectively. For the functions at 30D, it is clear that DE-CDI can obtain the higher $R+$ values than $R-$ values in the cases of DE/best/1 and CoDE, while the lower $R+$ values than $R-$ values in DE/rand/1 and ODE. In addition, the p values in the case of CoDE are less than 0.05, which means that DE-CDI is significantly better than NDi-DE overall. For the functions at 50D, the results of the multi-problem Wilcoxon signed rank tests show that DE-CDI is significantly better than NDi-DE in the cases of DE/best/1 and CoDE and is outperformed by it in the case of DE/rand/1.

As discussed in Sect. 4.5, the complexity of DE-CDI is $O(G_{\max} \times NP \times \max(D, n))$. In Cai and Wang (2013), the complexity of NDi-DE is $O(G_{\max} \times NP \times NP \times D)$. To show the efficiency of DE-CDI, DE-CDI is compared with NDi-DE in terms of the average runtime on all the functions at 30D and 50D. The results are shown in Table 13. In Table 13, the runtime value for each function means the average overhead of all the DE algorithms considered in this study on the corresponding function. In Table 13, the value of Ratio1 means the cost of DE-CDI is divided by that of the original DE algorithm, and the value of Ratio2 means the cost of NDi-DE is divided by that of DE-CDI. The last column "Avg." means the average value for all the functions.

From Table 13, it is clear that the Ratio1 values are close to 1 for all the functions at 30D and 50D. It indicates that the additional computational cost of DE-CDI is trivial in all the cases when compared with the original DE algorithms. When compared with NDi-DE, the Ratio2 values are higher than 2 for most of $F1$–$F14$, which means that DE-CDI is more efficient than NDi-DE for these functions. For the functions $F15$–$F25$, most Ratio2 values are lower than 1.1. Different from $F1$–$F14$, the evaluations of $F15$–$F25$ are costly. Thus, the overhead of additional computational cost of NDi-DE becomes trivial, which has been verified in Cai and Wang (2013).

Overall, compared with NDi-DE, on the one hand, DE-CDI does not need to select the type of direction information and the scaling factor for different mutation operators. On the other hand, the results in Tables 9, 10, 11, 12 and 13 clearly show that DE-CDI can obtain the better or comparable results to NDi-DE with low complexity.

5.6 Comparison with DMDE

In order to further show the high performance of DE-CDI, a recently proposed DE variant with a directional mutation operator (DMDE) (Zhang and Yuen 2015), is considered for comparison. In DMDE, a pool of difference vectors between the child and the parent individuals is calculated when the fitness is improved at a generation, and the difference vectors are randomly selected and incorporated into the mutation operator (Zhang and Yuen 2015). In addition, to deeply study the effectiveness of DE-CDI with different implementations

Table 16 Mean and standard deviation of the best error values obtained by original DE, DE-CELL, DE-DIR and DE-CDI on all the functions at 30*D*

Function	DE/best/1	DE-CELL/best/1	DE-DIR/best/1	DE-CDI/best/1	DE/rand-to-best/1	DE-CELL/rand-to-best/1	DE-DIR/rand-to-best/1	DE-CDI/rand-to-best/1
F1	3.70e+003 1.69e+003	= 4.35e+003 1.79e+003	− 9.76e+003 4.42e+003	+ **1.72e+001 4.95e+001**	1.82e+003 9.01e+002	− 2.55e+003 1.09e+003	+ **1.53e+002 1.25e+002**	+ **9.67e+000 1.94e+001**
F2	6.37e+003 3.78e+003	= 6.18e+003 2.98e+003	− 1.76e+004 6.43e+003	+ **1.56e−002 7.27e−002**	4.24e+003 1.54e+003	− 6.38e+003 3.08e+003	+ **9.75e+002 8.55e+002**	+ **6.64e+001 1.41e+002**
F3	1.52e+007 1.33e+007	= 1.58e+007 1.14e+007	− 4.77e+007 2.54e+007	+ **3.53e+005 1.71e+005**	6.91e+006 5.75e+006	− 1.33e+007 6.86e+006	+ **6.72e+005 7.25e+005**	+ **3.78e+005 4.21e+005**
F4	2.46e+002 3.97e+002	= 4.06e+002 5.45e+002	− 1.64e+003 2.43e+003	+ **5.10e−001 6.59e−001**	7.66e+002 4.75e+002	− 1.32e+003 9.73e+002	+ **2.50e+001 5.70e+001**	+ **3.15e−003 8.97e−003**
F5	9.04e+003 2.15e+003	= 9.46e+003 2.75e+003	− 1.30e+004 2.48e+003	+ **2.79e+003 5.33e+002**	6.97e+003 1.40e+003	− 8.12e+003 1.72e+003	+ **4.03e+003 8.51e+002**	+ **2.13e+003 4.36e+002**
F6	3.40e+008 2.51e+008	= 4.10e+008 3.37e+008	− 1.64e+009 1.06e+009	+ **1.41e+007 3.80e+007**	1.37e+008 1.23e+008	− 2.76e+008 2.18e+008	+ **3.83e+006 6.86e+006**	+ **1.04e+006 1.38e+006**
F7	3.65e+003 5.48e+002	= 3.84e+003 5.16e+002	− 5.85e+003 4.76e+002	+ **3.56e+000 6.33e+000**	3.70e+003 3.51e+002	− 3.99e+003 4.50e+002	+ **2.89e+003 3.66e+002**	+ **1.26e+001 2.00e+001**
F8	2.09e+001 5.73e−002	= 2.10e+001 5.35e−002	= 2.09e+001 4.92e−002	= 2.10e+001 5.59e−002	2.09e+001 3.82e−002	= 2.09e+001 4.52e−002	= 2.09e+001 5.58e−002	= 2.09e+001 5.10e−002
F9	1.06e+002 2.60e+001	− 1.21e+002 2.55e+001	− 1.29e+002 2.47e+001	+ **6.75e+001 1.88e+001**	6.57e+001 1.54e+001	= 6.77e+001 1.56e+001	+ **5.46e+001 1.35e+001**	+ **4.85e+001 9.61e+000**
F10	1.60e+002 3.86e+001	= 1.62e+002 4.11e+001	− 2.02e+002 3.60e+001	+ **9.42e+001 2.10e+001**	8.80e+001 1.80e+001	− 1.04e+002 3.55e+001	+ **6.45e+001 2.13e+001**	+ **5.40e+001 1.08e+001**
F11	2.22e+001 2.41e+000	= 2.33e+001 3.38e+000	− 2.46e+001 2.72e+000	− 2.91e+001 5.51e+000	1.30e+001 1.88e+000	− 1.55e+001 2.39e+000	= 1.22e+001 2.70e+000	− 2.14e+001 5.03e+000
F12	1.11e+005 4.94e+004	= 1.06e+005 4.71e+004	− 1.57e+005 5.35e+004	+ **5.13e+003 4.19e+003**	5.90e+004 2.17e+004	= 6.87e+004 2.43e+004	= 4.62e+004 2.00e+004	+ **1.10e+004 1.09e+004**
F13	7.74e+000 1.95e+000	= 9.14e+000 2.73e+000	− 1.08e+001 3.01e+000	+ **3.79e+000 9.85e−001**	3.25e+000 7.09e−001	− 3.96e+000 9.52e−001	= 3.51e+000 7.44e−001	= 3.54e+000 8.85e−001
F14	1.18e+001 4.34e−001	= 1.20e+001 4.35e−001	− 1.22e+001 4.88e−001	= 1.21e+001 6.59e−001	1.15e+001 4.79e−001	= 1.15e+001 4.46e−001	= 1.12e+001 4.11e−001	= 1.14e+001 5.30e−001
F15	5.65e+002 5.82e+001	+ **5.07e+002 9.89e+001**	= 5.82e+002 5.93e+001	+ **4.12e+002 5.37e+001**	4.59e+002 8.21e+001	= 4.69e+002 8.62e+001	= 4.11e+002 7.38e+001	+ **3.91e+002 6.89e+001**
F16	2.91e+002 1.24e+002	= 2.56e+002 1.16e+002	= 3.45e+002 1.61e+002	+ **1.39e+002 4.69e+001**	2.60e+002 2.18e+002	= 2.90e+002 1.86e+002	= 2.14e+002 1.71e+002	+ **8.34e+001 3.29e+001**
F17	3.22e+002 1.31e+002	= 2.52e+002 1.34e+002	= 3.62e+002 1.47e+002	+ **1.75e+002 1.26e+002**	2.35e+002 1.46e+002	= 2.16e+002 1.60e+002	= 2.27e+002 1.72e+002	+ **1.17e+002 1.09e+002**
F18	9.83e+002 3.18e+001	= 9.92e+002 2.63e+001	− 1.03e+003 2.39e+001	+ **9.03e+002 6.05e+001**	9.79e+002 2.61e+001	− 9.91e+002 3.65e+001	+ **9.35e+002 1.60e+001**	+ **8.51e+002 6.82e+001**
F19	9.88e+002 3.31e+001	= 9.81e+002 2.10e+001	− 1.02e+003 3.05e+001	+ **9.05e+002 6.12e+001**	9.76e+002 4.12e+001	− 1.00e+003 2.79e+001	+ **9.26e+002 3.19e+001**	+ **8.78e+002 6.92e+001**
F20	9.91e+002 4.10e+001	= 9.91e+002 3.20e+001	− 1.02e+003 3.24e+001	+ **8.96e+002 6.12e+001**	9.73e+002 4.13e+001	− 9.90e+002 2.08e+001	+ **9.34e+002 3.36e+001**	+ **8.82e+002 6.65e+001**
F21	1.06e+003 1.71e+002	= 1.14e+003 1.25e+002	− 1.17e+003 6.17e+001	+ **7.53e+002 3.23e+002**	1.02e+003 1.68e+002	= 1.10e+003 1.10e+002	+ **7.61e+002 2.59e+002**	+ **6.25e+002 2.46e+002**
F22	1.06e+003 6.74e+001	= 1.05e+003 5.28e+001	= 1.04e+003 5.80e+001	+ **9.36e+002 2.51e+001**	9.98e+002 2.49e+001	− 1.02e+003 3.79e+001	+ **9.79e+002 3.29e+001**	+ **9.27e+002 1.79e+001**
F23	1.11e+003 1.42e+002	− 1.16e+003 7.64e+001	− 1.20e+003 6.26e+001	+ **7.05e+002 2.82e+002**	1.07e+003 1.12e+002	= 1.12e+003 1.28e+002	+ **8.83e+002 2.11e+002**	+ **6.47e+002 1.62e+002**
F24	1.08e+003 2.51e+002	= 1.04e+003 2.58e+002	− 1.20e+003 8.13e+001	+ **2.00e+002 0.00e+000**	8.83e+002 2.77e+002	= 9.94e+002 2.25e+002	+ **3.74e+002 2.31e+002**	+ **2.00e+002 2.76e−003**
F25	1.56e+003 5.37e+001	= 1.57e+003 3.88e+001	− 1.64e+003 4.60e+001	+ **2.52e+002 2.13e+002**	1.51e+003 3.36e+001	= 1.53e+003 3.88e+001	+ **1.45e+003 3.63e+001**	+ **2.09e+002 4.10e+000**
SUM (+/ = /−)	–	1/22/2	0/5/20	22/2/1	–	0/11/14	17/8/0	21/3/1
ARV	2.28	2.64	3.78	1.30	2.92	3.80	1.98	1.30
Rank	2	3	4	1	3	4	2	1
Function	CoDE	CoDE-CELL	CoDE-DIR	CoDE-CDI	ODE	ODE-CELL	ODE-DIR	ODE-CDI
F1	3.00e−003 6.30e−004	+ **2.04e−003 4.04e−004**	+ **6.21e−004 9.91e−005**	+ **1.05e−008 4.65e−009**	1.86e−028 1.42e−028	= 1.76e−028 2.08e−028	− 3.65e−028 2.63e−028	= 1.99e−028 1.89e−028
F2	1.17e+004 2.03e+003	+ **1.00e+004 1.61e+003**	+ **7.22e+003 1.27e+003**	+ **6.85e+002 1.49e+002**	2.39e−004 2.84e−004	= 5.64e−004 7.38e−004	+ **5.58e−008 7.34e−008**	+ **1.07e−005 7.43e−006**
F3	5.24e+007 1.43e+007	= 4.37e+007 8.16e+006	+ **4.07e+007 8.29e+006**	+ **1.42e+007 3.33e+006**	6.66e+005 2.99e+005	= 6.06e+005 3.30e+005	+ **3.07e+005 1.79e+005**	+ **4.41e+005 1.79e+005**
F4	1.87e+004 2.99e+003	+ **1.62e+004 2.55e+003**	+ **1.18e+004 1.77e+003**	+ **2.60e+003 5.65e+002**	1.08e−001 8.62e−002	− 2.57e−001 2.99e−001	+ **1.72e−003 4.14e−003**	+ **2.32e−002 2.61e−002**
F5	5.69e+003 5.06e+002	+ **5.05e+003 4.35e+002**	+ **4.20e+003 4.06e+002**	+ **1.22e+003 3.20e+002**	1.82e+002 8.74e+001	= 2.24e+002 1.04e+002	+ **1.67e+001 1.56e+001**	+ **3.59e+001 2.94e+001**

Table 16 continued

Function	CoDE	CoDE-CELL	CoDE-DIR	CoDE-CDI	ODE	ODE-CELL	ODE-DIR	ODE-CDI
F6	4.40e+002 6.67e+001	+ **3.94e+002 6.99e+001**	+ **2.73e+002 7.31e+001**	+ **2.76e+001 5.24e+000**	5.39e+001 2.46e+001	= 5.17e+001 2.66e+001	+ **5.34e−005 1.68e−004**	+ **1.81e+000 2.35e+000**
F7	2.72e+001 5.21e+000	+ **1.27e+001 2.79e+000**	+ **6.95e+000 1.29e+000**	+ **2.51e−001 9.95e−002**	1.18e−003 3.27e−003	= 1.77e−003 3.66e−003	− 1.48e−003 4.27e−003	= 3.94e−003 7.61e−003
F8	2.09e+001 5.49e−002	= 2.10e+001 4.38e−002	= 2.09e+001 4.22e−002	= 2.09e+001 6.40e−002	2.10e+001 4.88e−002	= 2.09e+001 5.50e−002	+ **2.09e+001 6.29e−002**	= 2.10e+001 4.45e−002
F9	1.76e+001 1.54e+000	= 1.76e+001 1.79e+000	+ **1.62e+001 1.67e+000**	= 1.71e+001 1.91e+000	7.89e+001 2.99e+001	+ **2.75e+001 2.26e+001**	+ **3.35e+001 2.30e+001**	+ **2.49e+001 1.28e+001**
F10	2.07e+002 1.51e+001	= 2.07e+002 1.34e+001	= 1.99e+002 1.22e+001	+ **1.78e+002 1.13e+001**	5.09e+001 4.66e+001	= 3.86e+001 1.49e+001	= 5.53e+001 4.25e+001	= 4.14e+001 2.69e+001
F11	3.51e+001 9.66e−001	= 3.49e+001 9.83e−001	= 3.54e+001 1.04e+000	= 3.44e+001 1.36e+000	9.22e+000 9.33e+000	= 1.15e+001 9.21e+000	= 1.19e+001 1.10e+001	− 1.45e+001 9.96e+000
F12	1.59e+005 2.44e+004	= 1.52e+005 1.95e+004	= 1.54e+005 2.14e+004	+ **1.16e+005 1.89e+004**	2.27e+003 2.24e+003	= 2.93e+003 3.76e+003	+ **1.30e+003 1.81e+003**	= 1.64e+003 1.95e+003
F13	9.84e+000 7.31e−001	= 9.43e+000 9.98e−001	= 9.48e+000 7.52e−001	+ **8.71e+000 6.58e−001**	6.19e+000 2.51e+000	+ **3.92e+000 1.89e+000**	= 5.52e+000 2.50e+000	+ **3.03e+000 7.50e−001**
F14	1.33e+001 1.48e−001	= 1.33e+001 1.45e−001	= 1.33e+001 1.49e−001	+ **1.32e+001 1.31e−001**	1.32e+001 2.92e−001	+ **1.28e+001 4.10e−001**	= 1.33e+001 2.50e−001	+ **1.30e+001 3.46e−001**
F15	4.00e+002 1.67e−002	= 4.00e+002 1.50e−002	+ **4.00e+002 3.00e−003**	+ **3.92e+002 3.99e+001**	4.12e+002 3.31e+001	= 4.28e+002 4.58e+001	= 3.92e+002 8.13e+001	= 4.12e+002 3.34e+001
F16	2.29e+002 1.04e+001	= 2.26e+002 1.15e+001	+ **2.14e+002 1.50e+001**	+ **1.98e+002 7.80e+000**	9.02e+001 6.89e+001	= 6.24e+001 2.68e+001	= 1.07e+002 6.88e+001	= 8.52e+001 5.97e+001
F17	2.64e+002 1.53e+001	= 2.59e+002 1.19e+001	+ **2.50e+002 1.66e+001**	+ **2.31e+002 1.37e+001**	1.56e+002 7.50e+001	= 1.41e+002 7.15e+001	= 1.47e+002 9.18e+001	= **1.35e+002 7.60e+001**
F18	9.08e+002 2.62e−001	= 9.08e+002 3.30e−001	+ **9.07e+002 3.16e−001**	+ **9.06e+002 2.03e−001**	9.01e+002 2.10e+001	= 8.89e+002 3.95e+001	= 9.05e+002 1.41e+000	= 8.97e+002 2.92e+001
F19	9.08e+002 4.31e−001	= 9.08e+002 3.27e−001	+ **9.07e+002 2.31e−001**	+ **9.06e+002 1.74e−001**	9.04e+002 8.17e−001	= 8.93e+002 3.49e+001	− 9.05e+002 1.59e+000	− 9.05e+002 1.51e+000
F20	9.08e+002 4.20e−001	= 9.08e+002 3.01e−001	+ **9.07e+002 2.29e−001**	+ **9.06e+002 1.77e−001**	9.05e+002 1.94e+000	= 8.76e+002 4.83e+001	= 8.97e+002 2.91e+001	= 8.93e+002 3.49e+001
F21	5.00e+002 2.14e−004	+ **5.00e+002 1.10e−004**	+ **5.00e+002 4.32e−005**	+ **5.00e+002 0.00e+000**	5.00e+002 0.00e+000	= 5.12e+002 6.00e+001	= 5.00e+002 0.00e+000	= 5.00e+002 0.00e+000
F22	9.54e+002 1.03e+001	+ **9.46e+002 9.03e+000**	+ **9.24e+002 8.16e+000**	+ **9.02e+002 9.69e+000**	9.09e+002 1.16e+001	= 9.13e+002 8.00e+000	= 9.04e+002 1.19e+001	= 9.05e+002 9.22e+000
F23	5.34e+002 4.52e−004	= 5.34e+002 3.30e−004	+ **5.34e+002 3.70e−004**	+ **5.34e+002 3.70e−004**	5.34e+002 3.50e−004	= 5.34e+002 2.91e−004	= 5.50e+002 8.06e+001	= 5.34e+002 3.15e−004
F24	2.00e+002 3.51e−004	+ **2.00e+002 2.57e−004**	+ **2.00e+002 5.86e−005**	+ **2.00e+002 4.92e−006**	2.00e+002 0.00e+000	= 2.00e+002 0.00e+000	= 2.00e+002 0.00e+000	= 2.00e+002 0.00e+000
F25	2.56e+002 9.67e+000	+ **2.43e+002 6.85e+000**	+ **2.44e+002 6.71e+000**	+ **2.16e+002 1.43e+000**	2.09e+002 2.51e−001	= 2.09e+002 1.84e−001	− 2.47e+002 1.90e+002	+ **2.09e+002 1.31e−001**
SUM (+/ = /−)	–	10/15/0	19/6/0	**22/3/0**	–	3/21/1	8/13/4	**9/14/2**
ARV	3.52	2.92	2.30	**1.26**	2.86	2.40	2.50	**2.24**
Rank	4	3	2	**1**	4	2	3	**1**

The average ranking values (ARV) by Friedman test are also shown in the table

Table 17 Results of the multi-problem Wilcoxon's test between original DE, DE-CELL, DE-DIR and DE-CDI for all the functions at $30D$

Algorithm	(1)	(2)	(3)	(4)	Algorithm	(1)	(2)	(3)	(4)
DE/best/1 (1)	–	193.0	295.0[a]	7.0[b]	DE/rand-to-best/1 (1)	–	313.5[a]	1.0[b]	5.0[b]
DE-CELL/best/1 (2)	107.0	–	318.0[a]	4.0[b]	DE-CELL/rand-to-best/1 (2)	11.5[b]	–	4.0[b]	3.0[b]
DE-DIR/best/1 (3)	5.0[b]	7.0[b]	–	5.0[b]	DE-DIR/rand-to-best/1 (3)	299.0[a]	296.0[a]	–	7.0[b]
DE-CDI/best/1 (4)	318.0[a]	296.0[a]	320.0[a]	–	DE-CDI/rand-to-best/1 (4)	295.0[a]	297.0[a]	293.0[a]	–
CoDE (1)	–	39.5[b]	18.5[b]	5.0[b]	ODE (1)	–	117.0	117.0	55.5[b]
CoDE-CELL (2)	285.5[a]	–	50.5[b]	1.5[b]	ODE-CELL (2)	183.0	–	170.0	115.5
CoDE-DIR (3)	306.5[a]	249.5[a]	–	13.0[b]	ODE-DIR (3)	208.0	155.0	–	140.5
CoDE-CDI (4)	320.0[a]	298.5[a]	312.0[a]	–	ODE-CDI (4)	269.5[a]	184.5	159.5	–

Upper diagonal of level significance $\alpha = 0.9$, lower diagonal level of significance $\alpha = 0.95$
[a] The method in the row improves the method of the column
[b] The method in the column improves the method of the row

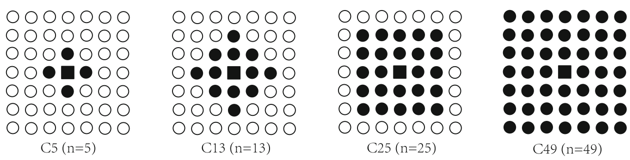

Fig. 8 Cellular topology-based neighborhood with different number of neighbors

of directional mutation, an improved DMDE with cellular topology, named cDMDE, is also presented here. In cDMDE, the cellular topology is incorporated into the population of DMDE. In this section, four DE algorithms, i.e., DE/rand/2 and DE/best/1, CoDE and ODE, are used. The results for all the functions at $30D$ are shown in Table 14. Furthermore, the results of the multi-problem Wilcoxon signed rank tests are also presented in Table 15.

In Table 14, it is clear that DE-CDI can obtain the better results than DMDE in all the cases. For DE/rand/2, DE-CDI is significantly better than DMDE on 22 functions. For DE/best/1, DE-CDI is significantly better than DMDE on 22 functions, while is outperformed by it on one function. For CoDE and ODE, DE-CDI can obtain the significant better results than DMDE on 18 and 8 functions, respectively. From the results of Friedman test, DE-CDI obtains the better average ranking values than DMDE in all the cases. The results of the multi-problem Wilcoxon signed rank tests in Table 15 also show that DE-CDI obtains the higher $R+$ values than $R-$ values in all the cases, when compared with DMDE. These results clearly indicate that DE-CDI is more effective than DMDE to enhance the performance of DE.

When compared with cDMDE, on the one hand, DE-CDI is consistently superior to it in most cases, except for DE/best/1. On the other hand, it is interesting to find that cDMDE can obtain the better results than DMDE in all the cases. Therefore, it is clear that the neighborhood information benefits DMDE. Furthermore, the effectiveness of DE-CDI is also demonstrated when it is implemented with different directional mutation operators.

5.7 Benefit of CDI components

In DE-CDI, the CDI-based mutation operator is composed by cellular topology-based neighborhood and neighborhood-based directional mutation. The experimental results presented above have shown that DE-CDI is effective for improving the performance of DE. In this section, to identify the benefit of these two components, two DE-CDI variants are considered: DE-CELL that only incorporates the cellular topology-based neighborhood into DE and DE-DIR that only introduces the neighborhood-based directional mutation into DE. In DE-CELL, all the individuals for mutation are selected from the neighborhood of the current individual. In DE-DIR, based on the randomly selected base vector, the whole population is partitioned into the superior and inferior groups and the difference vector is constructed as that in DE-CDI. Four DE algorithms, DE/best/1, DE/rand-to-best/1, CoDE and ODE, are used for comparison. The results for the functions at $30D$ are shown in Table 16.

From Table 16, both DE-CELL and DE-DIR perform worse than the corresponding original DE algorithm in two cases, while DE-CDI can obtain significantly better results than DE in all the cases. Specifically, for DE/best/1, DE-CELL and DE-DIR are significantly outperformed by the original DE algorithm on 2 and 20 functions, respectively, while DE-CDI is significantly better than the original DE algorithm on 22 functions. For DE/rand-to-best/1, DE-CELL and DE-DIR are significantly better on zero and 17 functions, respectively, while DE-CDI is significantly better on 21 functions. For CoDE and ODE, all the DE-CDI variants are significantly better on most functions. From the results of Friedman test (García et al. 2009; Derrac et al. 2011), DE-CDI obtains the best average ranking values in all the cases. In addition, both DE-CELL and DE-DIR perform better than the advanced DE algorithms overall.

To further show the advantage of combining the neighborhood and direction information in DE-CDI, the multi-

Table 18 Mean and standard deviation of the best error values obtained by DE/rand/1 and DE-CDI/rand/1 with different type of cellular topology on all the functions at 30*D*

Function	DE/rand/1	DE-CDI/rand/1									
			C5		C9		C13		C25		C49
F1	1.62e−029 5.59e−029	−	1.45e−028 1.13e−028	=	3.65e−029 7.22e−029	−	6.51e−029 9.91e−029	=	5.05e−029 8.50e−029	=	2.83e−029 6.69e−029
F2	5.82e−005 4.46e−005	+	**6.46e−008 2.61e−007**	+	**2.70e−014 3.70e−014**	+	**7.78e−014 8.80e−014**	+	**5.26e−011 5.46e−011**	+	**9.45e−009 1.87e−008**
F3	4.74e+005 2.65e+005	=	7.36e+005 6.39e+005	+	**1.34e+005 8.70e+004**	+	**1.45e+005 8.96e+004**	+	**2.10e+005 1.66e+005**	+	**1.99e+005 1.03e+005**
F4	3.28e−002 5.23e−002	−	1.35e+001 5.14e+001	+	**2.55e−006 6.51e−006**	+	**8.98e−007 1.72e−006**	+	**2.09e−005 4.00e−005**	+	**8.32e−005 7.23e−005**
F5	4.77e+001 3.66e+001	−	1.53e+003 6.31e+002	+	**9.39e+000 2.85e+001**	+	**3.29e+000 3.42e+000**	+	**1.42e+000 1.49e+000**	+	**4.04e+000 1.22e+001**
F6	1.88e+000 1.40e+000	−	5.54e+001 7.01e+001	=	1.75e+000 2.02e+000	+	**4.78e−001 1.32e+000**	+	**6.38e−001 1.49e+000**	+	**3.19e−001 1.10e+000**
F7	4.93e−004 2.46e−003	−	3.29e−002 7.89e−002	−	1.18e−002 1.12e−002	−	1.01e−002 9.48e−003	−	4.73e−003 6.55e−003	=	3.25e−003 5.02e−003
F8	2.10e+001 4.46e−002	=	2.09e+001 5.79e−002	=	2.09e+001 7.05e−002	=	2.09e+001 4.67e−002	=	2.10e+001 3.93e−002	=	2.10e+001 5.16e−002
F9	1.33e+002 3.41e+001	−	2.16e+002 1.56e+001	+	**2.63e+001 7.82e+000**	+	**2.37e+001 7.24e+000**	+	**1.84e+001 5.29e+000**	+	**7.30e+001 4.25e+001**
F10	1.81e+002 9.31e+000	−	1.93e+002 1.06e+001	+	**8.10e+001 6.26e+001**	+	**4.89e+001 4.24e+001**	+	**1.56e+002 4.57e+001**	=	1.82e+002 7.21e+000
F11	3.97e+001 1.06e+000	=	3.93e+001 8.01e−001	+	**3.74e+001 4.27e+000**	+	**3.32e+001 7.51e+000**	=	3.90e+001 2.57e+000	=	3.99e+001 7.85e−001
F12	2.41e+003 2.26e+003	−	1.38e+005 7.67e+004	=	1.72e+003 2.17e+003	+	**9.33e+002 1.49e+003**	+	**7.35e+002 1.59e+003**	+	**6.37e+002 9.68e+002**
F13	1.50e+001 9.66e−001	−	1.79e+001 1.02e+000	+	**3.54e+000 2.00e+000**	+	**2.74e+000 6.23e−001**	+	**5.49e+000 3.71e+000**	+	**1.40e+001 1.43e+000**
F14	1.34e+001 1.09e−001	+	**1.33e+001 1.31e−001**	+	**1.32e+001 1.76e−001**	+	**1.32e+001 2.04e−001**	+	**1.33e+001 1.42e−001**	=	1.33e+001 1.37e−001
F15	4.04e+002 3.51e+001	−	4.13e+002 5.34e+001	=	4.12e+002 4.39e+001	=	4.04e+002 3.51e+001	=	3.88e+002 5.26e+001	=	3.88e+002 6.66e+001
F16	2.08e+002 7.66e+000	−	2.22e+002 2.39e+001	+	**4.87e+001 9.82e+000**	+	**5.59e+001 1.14e+001**	+	**1.36e+002 7.23e+001**	=	2.03e+002 1.57e+001
F17	2.33e+002 8.35e+000	=	2.37e+002 8.70e+000	+	**1.69e+002 5.86e+001**	+	**1.18e+002 7.56e+001**	+	**2.03e+002 2.36e+001**	+	**2.25e+002 1.03e+001**
F18	9.00e+002 2.09e+001	=	8.85e+002 4.77e+001	−	9.02e+002 2.14e+001	+	**8.94e+002 3.56e+001**	=	9.01e+002 2.10e+001	=	8.96e+002 2.90e+001
F19	9.00e+002 2.10e+001	=	8.94e+002 4.33e+001	=	8.94e+002 3.53e+001	=	8.94e+002 3.54e+001	−	9.01e+002 2.12e+001	=	8.92e+002 3.48e+001
F20	9.00e+002 2.10e+001	=	**8.99e+002 4.24e+001**	+	**8.94e+002 3.55e+001**	=	8.93e+002 3.52e+001	−	9.05e+002 1.49e+000	=	9.05e+002 1.55e+000
F21	5.00e+002 0.00e+000	=	5.00e+002 0.00e+000	=	5.12e+002 6.00e+001	=	5.00e+002 0.00e+000	=	5.24e+002 8.31e+001	=	5.00e+002 0.00e+000
F22	9.07e+002 9.41e+000	−	9.42e+002 9.65e+000	=	9.05e+002 8.91e+000	=	9.04e+002 1.12e+001	=	9.03e+002 1.25e+001	+	**8.99e+002 1.11e+001**
F23	5.34e+002 3.16e−004	−	5.35e+002 2.02e+000	=	5.34e+002 1.66e−003	=	5.34e+002 3.34e−003	=	5.34e+002 2.56e−004	=	5.50e+002 8.06e+001
F24	2.00e+002 0.00e+000	=	2.00e+002 0.00e+000	=	2.00e+002 0.00e+000	=	2.00e+002 0.00e+000	=	2.00e+002 0.00e+000	=	2.00e+002 0.00e+000
F25	2.09e+002 2.20e−001	=	2.09e+002 2.81e−001	+	**2.09e+002 8.95e−002**	+	**2.09e+002 8.39e−002**	+	**2.09e+002 9.05e−002**	+	**2.09e+002 9.18e−002**
SUM (+/ = /−)	–	2/10/13		13/10/2		**15/8/2**		13/9/3		11/14/0	
ARV	4.26	4.82		2.96		**2.28**		3.34		3.34	
Rank	5	6		2		**1**		4		4	

Table 19 Results of the multi-problem Wilcoxon's test for DE/rand/1 versus DE-CDI with different type of cellular topology on all the functions at 30D

Algorithm	+/=/−	$R+$	$R-$	p value	$\alpha = 0.05$	$\alpha = 0.1$
DE-CDI with C5 vs DE/rand/1	2/10/13	67	233	1.00e+00	−	−
DE-CDI with C9 vs DE/rand/1	13/10/2	248	52	4.59E−03	+	+
DE-CDI with C13 vs DE/rand/1	15/8/2	283	17	1.29E−04	+	+
DE-CDI with C25 vs DE/rand/1	13/9/3	252	73	1.51E−02	+	+
DE-CDI with C49 vs DE/rand/1	11/14/0	250.5	74.5	1.47E−02	+	+

problem Wilcoxon signed rank tests are also carried out and the results are presented in Table 17. It can be found that DE-CDI obtains the higher $R+$ values than $R-$ values in all the cases. According to the multi-problem Wilcoxon signed rank test at $\alpha = 0.05$, DE-CDI is significantly better than DE-CELL and DE-DIR overall in most cases. These results clearly show that DE-CDI is better than the corresponding DE-CELL and DE-DIR overall.

From the results in Tables 16 and 17, we can obtain some interesting observations. First, both the neighborhood information and direction information are beneficial to enhancing the performance of most DE algorithms studied. Second, compared with the DE algorithm only with single information (i.e., neighborhood or direction information), DE-CDI can utilize the population information more effectively to guide the search.

5.8 Parameter study

In DE-CDI, there is a control parameter (i.e., n in Cn) that decides the neighborhood size of each individual. To test the influence of the neighborhood size on the performance of DE-CDI, the experiment studies with different n values are carried out. DE/rand/1 is employed for comparison here. Different number of neighbors, i.e., $n = 5$, 9, 13, 25 and 49, are considered. Figure 8 illustrates the cellular topology based neighborhood with different number of neighbors. The results are shown in Tables 18 and 19.

From Table 18, it is clear that DE-CDI with different n values is better than DE/rand/1 in all the cases except $n = 5$. In the cases of $n = 9$, 13, 25 and 49, DE-CDI is significantly better than DE/rand/1 on 13, 15, 13 and 11 functions, respectively, and is worse on 2, 2, 3 and zero functions, respectively. In addition, DE-CDI/rand/1 with different n values can obtain better average ranking value than DE/rand/1. According to the results of Table 19, in the cases of $n = 9$, 13, 25 and 49, DE-CDI obtains the higher $R+$ values than $R-$ values. It indicates that DE-CDI with different n values significantly outperforms DE/rand/1 overall. Furthermore, DE-CDI with $n = 13$ and $n = 9$ obtains the best and second best results. In the case of $n = 5$, DE-CDI obtains the worst results. The reason may lie in that the neighborhood size (e.g., $n = 5$) for each individual is too small to be partitioned into the superior and inferior groups, and the direction information cannot be utilized effectively to construct the difference vector.

According to the results in Tables 18 and 19, DE-CDI with $n = 13$ is a good choice for the benchmark functions in this study. In order to choose the appropriate neighborhood size for DE-CDI, adaptive or self-adaptive parameter control techniques (e.g., Zhang and Sanderson 2009; Qin et al. 2009; Islam et al. 2012) will be studied in the future work.

5.9 Application to real-world problems

In order to test the effectiveness of DE-CDI on real-world problems, three problems are selected from Eshelman et al. (1997) and Das and Suganthan (2010). The first two problems are from the CEC 2011 competition on testing EA on real-world numerical optimization problems (Das and Suganthan 2010). One is parameter estimation for frequency modulated sound waves (FMP) and the other is spread spectrum radar poly phase code design (SRP). FMP is a highly complex multi-modal problem with strong epistasis and SRP is with numerous local optima and has proven to be an *NP*-hard problem (Das and Suganthan 2010). The last problem is systems of linear equations problem (LEP) which has proven to be quite difficult for the optimizers (Suganthan et al. 2005). For each problem, MNFEs is set to 150,000 (Das and Suganthan 2010). Table 20 presents the results.

As Table 20 shows, it is clear that DE-CDI can obtain the better results than the corresponding DE algorithm in most cases. Specifically, for LEP and SRP, DE-CDI is better than the corresponding DE algorithms in 10 out of 12 cases. For FMP, DE-CDI is better than the corresponding DE algorithms in seven cases.

In sum, the results of Table 20 indicate that DE-CDI is able to enhance the performance of DE on the real-world problems considered.

6 Conclusions

In this study, a simple and effective framework, DE with cellular direction information (DE-CDI), has been presented for numerical optimization. In DE-CDI, the cellular topology is employed to define a neighborhood for each individual

Table 20 Mean and standard deviation of the best error values obtained by the DE algorithm and DE-CDI for the real-world application problems

Algorithm	LEP		FMP		SRP	
	DE	DE-CDI	DE	DE-CDI	DE	DE-CDI
DE/rand/1	1.06e−009 5.85e−010	**1.73e−012 8.13e−012**	**4.92e−001 2.46e+000**	1.77e+000 4.97e+000	2.44e+000 1.16e−001	**2.22e+000 4.05e−001**
DE/rand/2	3.36e−002 9.16e−003	**9.47e−005 3.99e−005**	1.38e+001 1.86e+000	**7.09e+000 4.84e+000**	2.46e+000 1.07e−001	**2.42e+000 9.51e−002**
DE/best/1	8.11e+000 6.39e+000	**9.70e−001 2.37e+000**	1.34e+001 5.39e+000	**1.10e+001 6.03e+000**	1.51e+000 1.98e−001	**1.43e+000 1.32e−001**
DE/best/2	**0.00e+000 0.00e+000**	6.01e−006 1.02e−005	**3.61e+000 5.40e+000**	9.51e−001 3.10e+000	2.45e+000 1.16e−001	**2.42e+000 1.13e−001**
DE/rand-to-best/1	4.57e−001 1.72e+000	**1.70e−011 5.90e−011**	9.60e+000 6.64e+000	**2.12e+000 4.38e+000**	2.44e+000 1.32e−001	**2.12e+000 1.76e−001**
DE/current-to-best/1	8.12e−001 1.24e+000	**8.42e−008 2.91e−007**	1.41e+001 4.09e+000	**6.07e+000 6.33e+000**	1.76e+000 5.51e−001	**1.50e+000 2.68e−001**
CoDE	2.32e+002 4.08e+001	**7.09e+001 1.45e+001**	5.53e+000 2.91e+000	**1.34e+000 1.65e+000**	2.16e+000 8.30e−002	**2.14e+000 1.12e−001**
JADE	0.00e+000 0.00e+000	0.00e+000 0.00e+000	**4.60e−001 2.30e+000**	1.84e+000 4.30e+000	1.49e+000 1.95e−001	**1.39e+000 1.60e−001**
jDE	4.67e−007 1.56e−006	**1.23e−011 2.11e−011**	1.90e−001 2.94e−001	**9.24e−003 3.09e−002**	2.00e+000 8.83e−002	**1.99e+000 9.41e−002**
MDE_pBX	**2.52e−003 6.30e−003**	1.72e+000 1.84e+000	**3.87e+000 6.63e+000**	7.44e+000 7.24e+000	2.00e+000 2.51e−001	**1.87e+000 2.43e−001**
ODE	6.52e−008 5.19e−008	**2.93e−008 2.29e−008**	**2.45e+000 4.96e+000**	9.61e−001 3.33e+000	**1.19e+000 1.89e−001**	1.24e+000 1.58e−001
SaDE	3.51e+001 1.11e+001	**1.23e+001 3.12e+000**	1.87e+000 1.75e+000	**1.23e+000 2.24e+000**	**1.90e+000 8.01e−002**	1.93e+000 8.29e−002

and the direction information is introduced into the mutation operator by constructing difference vector with the neighbors. In this way, the neighborhood and direction information can be simultaneously and effectively utilized to guide the search of DE. According to the extensive experimental study, it is obvious that DE-CDI is able to enhance the performance of most DE algorithms considered.

In the future, adaptive or self-adaptive methods for choosing number of neighbors in the cellular topology-based neighborhood will be investigated first. Then, the application of DE-CDI to other real-world problems will be studied. Finally, DE-CDI will also be extended to the multi-objective and large-scale optimization problems.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China (61305085, 61202468), the Natural Science Foundation of Fujian Province of China (2014J05074, 2014J01240), the Support Program for Innovative Team and Leading Talents of Huaqiao University (2014KJTD13) and the Fundamental Research Funds for the Central Universities (12BS216).

References

Alcalá-Fdez J, Sánchez L, García S (2015) KEEL: a software tool to assess evolutionary algorithms for data mining problems (online). http://www.keel.es/

Bi XJ, Xiao J (2011) Classification-based self-adaptive differential evolution with fast and reliable convergence performance. Soft Comput 15(8):1581–1599

Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10(6):646–657

Cai Y, Wang J (2013) Differential evolution with neighborhood and direction information for numerical optimization. IEEE Trans Cybern 43(6):2202–2215

Cai Y, Wang J, Yin J (2012) Learning-enhanced differential evolution for numerical optimization. Soft Comput 16(2):303–330

Cai Y, Wang J, Chen Y et al (2015) Adaptive direction information in differential evolution for numerical optimization. Soft Comput (in press)

Das S, Konar A (2006) Design of two dimensional IIR filters with modern search heuristics: a comparative study. Int J Comput Intell Appl 6(3):329–355

Das S, Suganthan PN (2010) Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. In: Technical report, Jadavpur University, West Bengal. Nanyang Technological University, Singapore

Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31

Das S, Abraham A, Konar A (2008) Adaptive clustering using improved differential evolution algorithm. IEEE Trans Syst Man Cybern A 38(1):218–237

Das S, Abraham A, Chakraborty UK, Konar A (2009) Differential evolution using a neighborhood-based mutation operator. IEEE Trans Evolut Comput 13(3):526–553

De Falco I, Della Cioppa A, Maisto D, Scafuri U, Taranino E (2014) Impact of the topology on the performance of distributed differential evolution. Appl Evol Comput 8602:75–85

Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18

Dorronsoro B, Bouvry P (2010) Differential evolution algorithms with cellular populations. In: Proceedings of the 11th PPSN, pp 320–330

Dorronsoro B, Bouvry P (2011) Improving classical and decentralized differential evolution with new mutation operator and population topologies. IEEE Trans Evol Comput 15(1):67–98

Epitropakis MG, Tasoulis DK, Pavlidis NG, Plagianakos VP, Vrahatis MN (2011) Enhancing differential evolution utilizing proximity based mutation operators. IEEE Trans Evol Comput 15(1):99–119

Eshelman LJ, Mathias KE, Schaffer JD (1997) Convergence controlled variation. In: Belew R, Vose M (eds) Foundations of genetic algorithms 4. Morgan Kaufmann, SanMateo, pp 203–224

Fan H, Lampinen J (2003) A trigonometric mutation operation to differential evolution. J Global Optim 27(1):105–129

García S, Fernandez A, Luengo J, Herrera F (2009) A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Comput 13(10):959–977

Gou J, Guo W, Hou F, Wang C, Cai Y (2015) Adaptive differential evolution with directional strategy and cloud model. Appl Intell 42(2):369–388

Hu Z, Cai X, Fan Z (2014) An improved memetic algorithm using ring neighborhood topology for constrained optimization. Soft Comput 18:2023–2041

Hui S, Suganthan PN (2013) Ensemble crowding differential evolution with neighborhood mutation for multimodal optimization. In: Proceedings of the IEEE symposium on differential evolution (SDE), pp 135–142. IEEE, New York

Iorio A, Li X (2006) Incorporating directional information within a differential evolution algorithm for multi-objective optimization. In: Proceedings of the 8th annual conference on genetic evolutionary computational, pp 691–698

Islam SM, Das S, Ghosh S, Roy S, Suganthan PN (2012) An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. IEEE Trans Syst Man Cybern B Cybern 42(2):482–500

Joshi R, Sanderson AC (1999) Minimal representation multi-sensor fusion using differential evolution. IEEE Trans Syst Man Cybern Part A 29(1):63–76

Lampinen J (1999) A bibliography of differential evolution algorithm. In: Technical report, Laboratory of Information Processing, Department of Information Technology, Lappeenranta University of Technology. http://www.lut.fi/jlampine/debiblio.htm (online)

Liang JJ, Qu B-Y, Mao X, Niu B, Wang D (2014) Differential evolution based on fitness Euclidean-distance ratio for multimodal optimization. Neurocomputing 137(5):252–260

Liu J, Fan Z, Goodman E (2009) SRDE: an improved differential evolution based on stochastic ranking. In: Proceedings of the 1st ACM/SIGEVO, pp 345–352

Neri F, Tirronen V (2010) Recent advances in differential evolution: a survey and experimental analysis. Artif Intell Rev 33(1/2):61–106

Neri F, Iacca G, Mininno E (2011) Disturbed exploitation compact differential evolution for limited memory optimization problems. Inf Sci 181(12):2469–2487

Noman N, Iba H (2011) Cellular differential evolution algorithm. In: Proceedings of the AII advanced artificial intelligence, pp 293–302

Noroozi V, Hashemi A, Meybodi M (2011) CellularDE: a cellular based differential evolution for dynamic optimization problems. In: Proceedings of the adapting natural computational algorithms, pp 340–349

Omran M, Engelbrecht AP, Salman A (2005) Differential evolution methods for unsupervised image classification. In: Proceedings of the 7th congress on evolutionary computation (CEC-2005), vol 2, pp 966–973. IEEE Press, Piscataway

Omran M, Engelbrecht A, Salman A (2006) Using the ring neighborhood topology with self-adaptive differential evolution. In: Jiao L, Wang L, Gao X-B, Liu J, Wu F (eds) Advances in natural computation. Springer, Berlin, pp 976–979

Omran M, Engelbrecht A, Salman A (2009) Bare bones differential evolution. Eur J Oper Res 196(1):128–139

Piotrowski AP (2013) Adaptive memetic differential evolution with global and local neighborhood-based mutation operators. Inf Sci 241(20):164–194

Qin A, Huang V, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13(2):398–417

Rahnamayan S, Tizhoosh HR, Salama MMA (2008) Opposition based differential evolution. IEEE Trans Evol Comput 12(1):64–79

Rogalsky T, Derksen RW, Kocabiyik S (1999) Differential evolution in aerodynamic optimization. In: Proceedings of the 46th annual conference on Canadian Aeronautics and Space Institute, pp 29–36

Sarkar S, Mukherjee R, Biswas S, Kundu R, Das S (2015) An adaptive clustering and re-clustering based crowding differential evolution for continuous multi-modal optimization. In: Proceedings of the 18th Asia Pacific symposium on intelligent and evolutionary systems, vol 1, pp 373–388. Springer, New York

Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359

Storn R, Price KV, Lampinen J (2005) Differential evolution—a practical approach to global optimization. Springer, Berlin

Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. In: KanGAL Report No. 2005005, Nanyang Technological University, Singapore. IIT Kanpur, India

Sun J, Zhang Q, Tsang EPK (2005) DE/EDA: a new evolutionary algorithm for global optimization. Inf Sci 169(3):249–262

Tang L, Dong Y, Liu J (2015) Differential evolution with an individual-dependent mechanism. IEEE Trans Evol Comput (in press)

Wang F-S, Jang H-J (2000) Parameter estimation of a bio-reaction model by hybrid differential evolution. In: Proceedings of the IEEE congress on evolutionary computation, vol 1, pp 410–417. IEEE Press, Piscataway

Wang YX, Xiang QL (2008) Exploring new learning strategies in differential evolution algorithm. In: Proceedings of the IEEE congress on evolutionary computational, pp 204–209

Wang J, Cai Y (2015) Multiobjective evolutionary algorithm for frequency assignment problem in satellite communications. Soft Comput 19(5):1229–1253

Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans Evol Comput 15(1):55–66

Wang J, Liao J, Zhou Y, Cai Y (2014) Differential evolution enhanced with multiobjective sorting based mutation operators. IEEE Trans Cybern 46(12):2792–2805

Weber M, Tirronen V, Neri F (2010) Scale factor inheritance mechanism in distributed differential evolution. Soft Comput 14(11):1187–1207

Weber M, Neri F, Tirronen V (2011) A study on scale factor in distributed differential evolution. Inf Sci 181(12):2488–2511

Yang M, Li C, Cai Z, Guan J (2015) Differential evolution with auto-enhanced population diversity. IEEE Trans Cybern (in press)

Zhang J, Sanderson A (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 13(5):945–958

Zhang X, Yuen SY (2015) A directional mutation operator for differential evolution algorithms. Appl Soft Comput 30:529–548