J Ambient Intell Human Comput (2015) 6:481–494
DOI 10.1007/s12652-015-0259-x

ORIGINAL RESEARCH

Differential evolution with distributed direction information based mutation operators: an optimization technique for big data

Zewu Peng · Jingliang Liao · Yiqiao Cai

Received: 22 January 2015 / Accepted: 18 February 2015 / Published online: 1 March 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract With the rapid advance in networking, data storage, and data collection technique, big data is fast expanding in various scientific and engineering fields, such as physical, social and biological sciences. Thanks to solving difficult optimization problems without detailed prior knowledge, evolutionary algorithm (EA) has become a powerful optimization technique for dealing with complex problems in big data. This study focuses on differential evolution (DE), which is one of the most successful and popular EAs and distinguishes from other EAs with its mutation mechanism. However, for the mutation operators of most DE algorithms, the base and difference vectors are always randomly selected from the whole population, where the population information is not utilized effectively. In this study, a novel DE framework with distributed direction information based mutation operators (DE-DDI) is proposed. In DE-DDI, the distributed topology is employed to create a neighborhood for each individual in the population first and then the direction information derived from the neighbors is introduced into the mutation operator of DE. Therefore, the neighborhood and direction information are fully utilized to exploit the regions of better individuals and guide the search to the promising area. In order to test the performance of the proposed algorithm, DE-DDI is applied to several original DE algorithms, as well as the advanced DE variants. The results clearly indicate that DE-DDI is able to improve the performance of the DE algorithms studied.

Keywords Differential evolution · Distributed topology · Neighborhood information · Direction information · Mutation strategy · Big data

1 Introduction

During the past decade, large amounts of data have been generated in various scientific and engineering fields due to the development of high throughput technologies. With such large size of data, it becomes difficult to perform effective analysis by the existing traditional methods. This promotes the rise of big data, which has drawn huge attention from researchers in information sciences, policy and decision makers in governments and enterprises (Philip Chen and Zhang 2014). Big data is characterized by large-volume, complex, growing data with multiple, autonomous sources. Due to these characteristics, various challenges and issues related to big data are put forward, mainly existing in difficulties in data capture, data storage, and data analysis and data visualization. Recently, more and more fields involve big data, such as social network, bioinformatics, e-commerce, and so on. In order to capture the value from big data, a lot of techniques have been developed (Philip Chen and Zhang 2014), e.g., optimization methods, data mining, knowledge-based platform, social network analysis, etc. These big data techniques involve many disciplines and overlap with each other frequently (Philip Chen and Zhang 2014).

For the big data problems, they always have the characteristics of the large search space, sparse and incomplete data, potentially complex fitness landscapes and dynamic

Z. Peng
Information center of Guangdong Power Grid Company Limited, Guangzhou, Guangdong 510006, People's Republic of China

J. Liao · Y. Cai (✉)
College of Computer Science and Technology, Huaqiao University, Xiamen, Fujian 361000, People's Republic of China
e-mail: yiqiao00@163.com; caiyq@hqu.edu.cn

and uncertain (Thomas and Jin 2014). Inspired from natural evolution process, evolutionary algorithms (EAs) are suitable for dealing with these problems, as they require little or no prior knowledge of problems and can cope with multiple objectives and constraints based on data provenance or heterogeneous sources. Thus, it makes EAs a powerful tool for dealing with complex problems in big data. As an optimization technique for big data, EAs have the great potential in tacking complex problems, and have attracted great attention from both academia and industry in many fields. They also have been applied successfully to solve the quantitative problems in various fields, such as physics, biology, engineering, and so on.

Differential evolution (DE), proposed in (Storn and Price 1997), is a simple yet powerful EA for global numerical optimization. Recently, DE has become one of the most widely used for handling global optimization problems (Das and Suganthan 2011). Furthermore, DE has been successfully applied in many science and engineering fields, such as pattern recognition (Campomanes-Álvarez et al. 2014), signal processing (Das and Konar 2006), satellite communications (Wang and Cai 2015), vehicle routing problem (Zhou and Wang 2015), and so on.

During the last decade, there are many enhanced DE variants proposed in the literature (Das and Suganthan 2011). In these advanced DE variants, modifications mostly focus on devising the new mutation operators (Das et al. 2009; Zhang and Sanderson 2009; Wang et al. 2014a), employing the self-adaptive strategies for control parameters (Qin et al. 2009; Yang et al. 2014), proposing the ensemble strategies (Tang et al. 2014), developing the hybrid DE with other optimization methods (Sun et al. 2005; Cai et al. 2014a; Li et al. 2015) and population topology (Dorronsoro and Bouvry 2011), etc.

Generally, the mutant vector can be treated as the lead individual to search the decision space and is constructed by adding a scaled difference vector to a base vector. However, we have observed that these two vectors (i.e., the base and difference vectors) in most DE variants are always randomly selected. In this case, the population information could not be fully utilized to guide the search of DE.

In order to alleviate this drawback and improve the performance of DE, a new DE framework with distributed topology based mutation operator (DE-DDI) is proposed in this study. In DE-DDI, a distributed topology is first employed to define a neighborhood for each vector. Then, the neighbors of each vector are divided into better and worse groups according to their fitness compared to that of it. Finally, the direction information is introduced into mutation by selecting the vectors from the better and worse groups, respectively to construct the difference vector. In this way, DE-DDI not only utilizes the information of neighboring individuals to exploit the regions of minima but also incorporates the direction information of population to prevent individuals from entering an undesired region and move to a promising area. Hence, the population information composed by neighborhood information and direction information can be simultaneously and fully utilized in DE-DDI to guide the search of DE.

To evaluate the effectiveness of the proposed method, extensive experiments have been carried out on CEC 2005. With the analysis of the extensive experiments, we can clearly find that DE-DDI is able to improve the performance of the DE algorithms studied.

The main contributions of this paper include the following:

- Both neighborhood and direction information are fully and simultaneously utilized in the mutation strategy to generate the mutant.
- DE-DDI provides a simple yet powerful method for improving the explorative ability of DE. In addition, DE-DDI is simple and easily applied to other DE variants.
- The extensive experiments are carried out to show the effectiveness of DE-DDI. The results demonstrate that DDI 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-DDI 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. Without loss of generality, we consider the optimization problem to be minimized is $f(X)$, $X = [x^1, x^2, \ldots, x^D] \in R^D$ and D is the dimension of the decision variables. DE evolves a population of NP vectors representing the candidate solutions. Each vector 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 size of the population and G is the number of current generation.

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 constrained by the prescribed minimum and maximum bounds. That is, the jth parameter of the ith individual is initialized by

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

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

2.2 Mutation

Following initialization, DE employs the mutation strategy to generate a mutant vector $V_{i,G}$ with respect to each individual $X_{i,G}$ (called target vector) in the current population. Six most frequently used mutation strategies in the literature 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

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

- DE/current-to-best/

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

- DE/rand-to-best/1

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

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, and the mutation factor F is a positive control parameter for scaling the difference vector. More details can be found in (Das and Suganthan 2011; Storn and Price 1997).

2.3 Crossover

After the mutation phase, 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. The binomial crossover is widely used, which can be defined as follows:

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

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, we reinitialized it 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 $U_{i,G}$ for the next generation. The selection operator is given by

$$X_{i,G+1} = \begin{cases} U_{i,G} & \text{if } f(U_{i,G}) \le f(X_{i,G}); \\ X_{i,G} & \text{otherwise.} \end{cases} \tag{9}$$

3 Related work

In the section, we focus on the related work on how the population information, especially neighborhood and direction information, has been utilized in DE to improve its performance.

3.1 Neighborhood information

There are two main types of neighborhood information: one relies on the population topology and the other on the geographical locations on the fitness landscape. More details about the neighborhood concepts utilized in DE could be found in (Epitropakis et al. 2011).

In the first one, the neighbors of each individual do not necessary lie in the vicinity of its topological region in the search space. Different from the original DE algorithm, many DE variants utilize the neighborhood information with the structured population. In these DE variants, the individuals for the mutation strategies are selected according to a neighbor list constructed from the population topologies. Two main canonical kinds of structured population in DE could be found in literature, i.e., cellular DE (cDE) (Noman and Iba 2011) and distributed DE (dDE) (Weber et al. 2010, 2011; Neri et al. 2011). Recently, several population topologies, e.g., cellular, distributed, ring, small-world, were introduced in DE to improve its performance (Dorronsoro and Bouvry 2011).

The second kind of neighborhood information is derived from the current population during the evolutionary process. In this, we name some individuals be the neighbors of one 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 by using an affinity matrix based on the Euclidean distance to select the individuals for the mutation strategies. For improving the performance of DE, the learning-enhanced DE (LeDE) was proposed in (Cai et al. 2012). In LeDE, the neighborhood of each individual, which involved in the intra-cluster learning strategy, is defined based on the identified clusters. In (Wang et al. 2014b), by the cooperation of objective-wise learning process, all the objectives of the considered solution could be simultaneously guided to optimize in parallel.

3.2 Direction information

In DE, the difference vector of the mutation strategies is always be constructed in a random manner. In order to overcome this drawback of DE, several DE variants are proposed by using the direction information to construct the difference vector.

In (Wang and Xiang 2008), a new mutation strategy, which is identified as DE/rand/±mean, was proposed. In this strategy, the population is partitioned into two subpopulations according to the mean fitness value of all individuals. Then the different vector is constructed by randomly selecting two vectors from the better and worse subpopulation, respectively. Recently, a novel DE framework, DE with neighborhood and direction information (NDi-DE), was proposed by designing three types of direction information for mutation (Cai and Wang 2013). In NDi-DE, the direction information is derived from two sources, namely, the best and worst near-neighbor individuals. Then, three types of direction information based on the direction information with different sources are introduced to guide search. In the further work (Cai et al. 2014b; Cai and Du 2014), an adaptive operator selection (AOS) mechanism was introduced into NDi-DE for different mutation strategies. In this way, a good balance between exploration and exploitation of the novel DE framework (aNDi-DE) could be dynamically achieved. In (Bi and Xiao 2011), by using the direction information with the current best solution and the best previous solution of each individual, the authors proposed a classification-based self-adaptive DE.

4 DE-DDI

In this section, the proposed framework, i.e., DE-DDI, is described in detail. First, the motivation of this study is given first. Second, the two main components of DDI, i.e., distributed topology-based neighborhood and mutation with direction information, are presented. Third, the complete proposed framework is shown.

4.1 Motivations

As mentioned above, both neighborhood information and direction information can be utilized to improve the performance of DE, but they are not fully and simultaneously exploited in the evolutionary process for most DE algorithms. Furthermore, in most DE algorithms, the base and difference vectors are randomly selected for mutation, which cannot utilize the population information to guide the search effectively. Thus, based on these considerations, we utilize the population information composed by neighborhood and direction information to propose a new mutation operator to enhance the performance of DE.

4.2 Distributed topology-based neighborhood

In order to define a neighborhood for each individual in the population, a distributed topology with *NP* individuals is employed first. In this topology (with four islands illustrated in Fig. 1), the population is partitioned into four islands, which are evolved by the independent DE. We also refer to the populations in the islands as subpopulations of the algorithm, as it is commonly done in the literature. In this way, individuals in the same island are neighbors.

For the individuals in different islands to communicate with each other, when one island does not evolve a better individual for Q generations, we employ a process called recombination. In this process, each island will be reconstructed by a same number of individuals, which are randomly selected from the current whole population. This will allow the algorithm to better benefit from the diversity of solutions in the different islands.

4.3 Mutation with direction information

Based on the defined neighbors with distributed topology, the direction information is introduced into mutation operator by selecting several individuals to construct the difference vectors. The base vector, i.e., $X_{r1,G}$ in DE/rand/1, is randomly

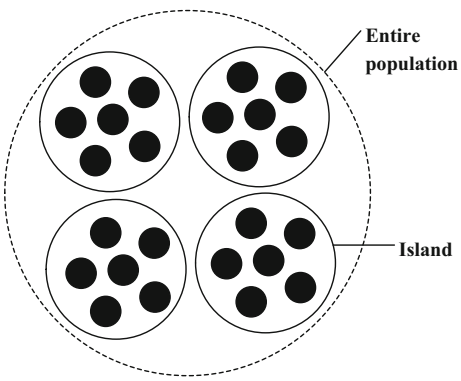

Fig. 1 I4

selected from the neighbors of X_i, first. Then, with respect to the fitness of the base vector, all of the neighbors of X_i are partitioned into the better and worse groups. Finally, the terminal point of the difference vector, i.e., X_{r2} in DE/rand/1, is randomly selected from the better group and the start point, i.e., X_{r3} in DE/rand/1, is randomly selected from the worse group. In this way, the difference vector with good direction information that directing at the better individual from the worse one can be obtained to guide the search in the solution space.

4.4 The framework of DE-DDI

As described above, it's clear that DE-DDI works through a simple cycle of stages, presented in Fig. 2. In this paper, we name the complete framework of DE-DDI with the DE/rand/1 strategy as DE-DDI/rand/1. And the corresponding pseudo-code of DE-DDI/rand/1 is shown in Algorithm 1 where the differences with respect to DE/rand/1 are highlighted with "*". It is clear that the proposed DE-DDI only affects the mutation stage, hence it could be directly and easily applied to most of the DE algorithms.

For the mutation operators which employ the best individual (e.g., DE/best/1), when applying DE-DDI to it, the best one in the neighborhood of the current individual will be selected as the best individual in DE-DDI. As for constructing the difference vector, when the base vector is the best or worst vector in the neighborhood, the vectors are randomly selected from the neighborhood and the difference vector will be constructed by directing at the better solution from the worse one.

Algorithm 1 DE-DDI/rand/1

1: Generate the initial population P^G and set $G = 1$;
2: Evaluate the fitness for each individual in P^G;
3: **While** the terminated condition is not satisfied **do**
4: **For** each individual $X_{i,G}$ **do**
5: *Randomly select the base vector X_{r1} from the neighborhood of $X_{i,G}$;
6: *Partition all the neighbors of $X_{i,G}$ into better and worse groups by comparing with the fitness of $X_{i,G}$;
7: *Randomly select X_{r2}, X_{r3} from the better and worse groups respectively;
8: Use Eq. (2) to generate a mutant vector
9: Use Eq. (8) to generate a trial vector;
10: Use Eq. (9) to determine the survived vector;
11: **End For**
12: Set $G = G + 1$
13: **End while**

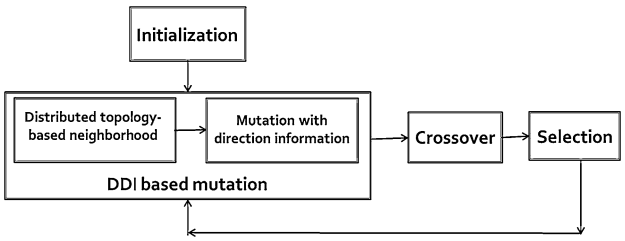

Fig. 2 Main stages of the DE-DDI

5 Experimental results and analysis

In order to evaluate the performance of DE-DDI, 25 classic 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. Second, the experimental setup is shown. Finally, the simulation results are analyzed and discussed.

5.1 Benchmark function

In this section, 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). According to the, they can be categorized into four groups: unimodal functions ($F1$–$F5$), basic multimodal functions ($F6$–$F12$), expanded multimodal functions ($F13$–$F14$) and hybrid composition functions ($F15$–$F25$).

5.2 Parameter settings

In order to maintain a fair and reliable comparison between DE-DDI and its corresponding competitors, the same random initial population is employed in this study. And the parameters are set as follows unless a change is mentioned.

- Dimension: $D = 30$;
- Population size: $NP = 100$;
- Mutation factor: $F = 0.5$;
- Crossover rate: $CR = 0.9$;
- Type of distributed topology: I4 ($n = 4$);
- Stagnation tolerance: $Q = 20$;
- Number of runs: $NumR = 25$;
- Maximum number of function evaluations: $MNFEs = 10{,}000 \times D$.

In the experiments, the comparisons between the four original DE algorithms (i.e., DE/rand/2, DE/best/2, DE/current-to-best/1 and DE/rand-to-best/1) and their corresponding DE-DDI algorithms are conducted first. Then, we compare the performance of two advanced DE

Table 1 Mean and standard deviation of the best error values obtained by DE-DDI and the corresponding DE algorithms on all the functions at 30*D*

Functions	DE/rand/2			DE-DDI/rand/2		DE/best/2			DE-DDI/best/2		DE/current-to-best/1			DE-DDI/current-to-best/1	
F1	8.37e−001	2.86e−001	+	**1.04e−020**	**3.18e−020**	**7.66e−028**	**2.71e−028**	−	1.59e−009	7.54e−009	3.58e+003	2.05e+003	+	**1.02e+003**	**2.56e+003**
F2	7.65e+003	1.80e+003	+	**5.56e+002**	**2.62e+002**	**1.06e−014**	**1.21e−014**	−	4.74e−011	1.07e−010	5.64e+003	2.23e+003	+	**3.34e+002**	**2.83e+002**
F3	4.85e+007	1.30e+007	+	**1.28e+007**	**4.63e+006**	**1.37e+005**	**7.33e+004**	=	1.98e+005	1.19e+005	6.51e+006	5.13e+006	+	**2.21e+005**	**1.20e+005**
F4	1.46e+004	2.59e+003	+	**3.84e+003**	**1.97e+003**	**9.37e−005**	**1.36e−004**	=	1.31e−004	2.90e−004	5.59e+002	4.98e+002	+	**6.67e−003**	**8.27e−003**
F5	8.07e+003	7.28e+002	+	**3.65e+003**	**5.02e+002**	**3.83e+001**	**5.26e+001**	=	6.86e+001	6.61e+001	7.36e+003	1.66e+003	+	**4.05e+003**	**9.12e+002**
F6	5.77e+003	4.08e+003	+	**2.22e+001**	**2.20e+001**	**6.38e−001**	**1.49e+000**	−	8.87e+001	1.06e+002	3.10e+008	2.70e+008	+	**2.23e+007**	**1.99e+007**
F7	7.20e+000	2.22e+000	+	**3.31e−002**	**1.21e−001**	1.55e−002	1.61e−002	=	**1.26e−002**	**1.13e−002**	3.98e+003	4.98e+002	+	**4.65e+002**	**1.52e+002**
F8	2.09e+001	6.06e−002	=	**2.09e+001**	**4.91−002**	2.10e+001	3.58e−002	=	**2.09e+001**	**5.18e−002**	2.10e+001	4.46e−002	=	2.10e+001	5.62e−002
F9	2.10e+002	1.28e+001	=	**2.07e+002**	**1.08e+001**	1.82e+002	1.50e+001	+	**1.74e+002**	**8.78e+000**	7.54e+001	1.53e+001	+	**5.03e+001**	**1.36e+001**
F10	2.42e+002	9.26e+000	+	**2.29e+002**	**9.79e+000**	2.01e+002	1.31e+001	+	**1.84e+002**	**1.30e+001**	1.12e+002	3.37e+001	+	**5.34e+001**	**2.16e+001**
F11	**3.93e+001**	**1.30e+000**	=	3.96e+ 001	8.75e−001	3.95e+001	8.57e−001	=	**3.92e+001**	**1.22e+000**	**1.40e+001**	**2.31e+000**	=	1.70e+001	8.64e+000
F12	5.22e+005	4.47e+004	=	**4.90e+005**	**5.94e+004**	1.98e+003	4.16e+003	=	**1.42e+003**	**1.62e+003**	3.58e+004	2.06e+004	+	**5.22e+003**	**4.37e+003**
F13	2.02e+001	9.01e−001	+	**1.91e+001**	**1.19e+000**	**1.57e+001**	**1.46e+000**	=	1.58e+001	1.06e+000	**4.86e+000**	**3.25e+000**	=	5.29e+000	3.21e+000
F14	1.34e+001	1.58e−001	=	**1.34e+001**	**1.48e−001**	1.34e+001	1.72e−001	=	**1.33e+001**	**2.08e−001**	**1.17e+001**	**3.72e−001**	−	1.21e+001	3.91e−001
F15	4.05e+002	6.90e+000	+	**4.00e+002**	**0.00e+000**	3.21e+002	1.12e+002	=	**3.09e+002**	**9.28e+001**	4.99e+002	1.29e+002	+	**3.47e+002**	**1.21e+002**
F16	2.69e+002	1.12e+001	+	**2.50e+002**	**1.24e+001**	2.64e+002	8.60e+001	=	**2.32e+002**	**4.63e+001**	2.82e+002	1.88e+002	+	**1.24e+002**	**8.16e+001**
F17	3.05e+002	1.34e+001	+	**2.88e+002**	**2.70e+001**	3.29e+002	1.09e+002	+	**2.35e+002**	**1.74e+001**	2.57e+002	1.64e+002	+	**9.81e+001**	**5.59e+001**
F18	9.40e+002	2.18e+000	+	**9.20e+002**	**2.44e+000**	8.86e+002	4.37e+001	=	8.86e+002	4.41e+001	9.75e+002	4.75e+001	=	**9.63e+002**	**3.75e+001**
F19	9.39e+002	3.20e+000	+	**9.19e+002**	**3.03e+000**	8.99e+002	2.98e+001	=	**8.96e+002**	**3.61e+001**	9.92e+002	3.11e+001	+	**9.61e+002**	**5.24e+001**
F20	9.39e+002	3.51e+000	+	**9.20e+002**	**2.94e+000**	8.99e+002	2.98e+001	=	**8.91e+002**	**4.06e+001**	9.89e+002	3.22e+001	=	**9.70e+002**	**5.86e+001**
F21	5.00e+002	1.23e−001	+	5.00e+002	1.53e−005	5.25e+002	8.49e+001	=	**5.12e+002**	**6.00e+001**	1.06e+003	1.50e+002	+	**8.53e+002**	**3.14e+002**
F22	1.02e+003	1.54e+001	+	**9.70e+002**	**1.07e+001**	9.27e+002	1.35e+001	+	**9.13e+002**	**1.36e+001**	1.02e+003	3.24e+ 001	+	**9.43e+002**	**2.16e+001**
F23	5.35e+002	1.24e+000	+	**5.34e+002**	**3.85e−004**	5.69e+002	1.12e+002	=	**5.68e+002**	**1.15e+002**	1.09e+003	1.43e+ 002	+	**8.77e+002**	**2.30e+002**
F24	2.00e+002	1.16e−001	+	2.00e+002	0.00e+000	2.00e+002	0.00e+000	=	2.00e+002	0.00e+000	9.37e+002	2.44e+002	+	**2.00e+002**	**6.47e−005**
F25	2.29e+002	3.54e+000	+	**2.12e+002**	**1.03e+000**	2.09e+002	1.03e−001	=	2.09e+002	9.31e−002	1.45e+003	5.75e+001	+	**2.54e+002**	**2.18e+002**
w/t/l	−			20/5/0		−			4/18/3		−			19/5/1	

Functions	DE/rand-to-best/1			DE-DDI/rand-to-best/1		MDE_pBX			MDE_pBX-DDI		O DE			O DE-DDI	
F1	1.82e+003	9.01e+002	+	**7.13e+001**	**5.10e+001**	3.07e−001	7.76e−001	+	**4.00e−012**	**1.98e-011**	**1.86e−028**	**1.42e−028**	=	3.76e−028	4.38e−028
F2	4.24e+003	1.54e+003	+	**4.53e+002**	**4.24e+002**	7.89e−007	3.46e−006	+	**5.76e−028**	**6.32e-028**	2.39e−004	2.84e−004	+	**5.53e−007**	**5.06e−007**
F3	6.91e+006	5.75e+006	+	**1.70e+006**	**5.04e+006**	9.69e+005	2.15e+005	+	**3.40e+005**	**1.36e+005**	6.66e+005	2.99e+005	+	**3.03e+005**	**1.64e+005**
F4	7.66e+002	4.75e+002	+	**5.03e+000**	**2.30e+001**	**3.00e−001**	**1.50e+000**	=	5.89e−001	2.69e+000	1.08e−001	8.62e−002	+	**2.05e−002**	**2.34e−002**
F5	6.97e+003	1.40e+003	=	**6.34e+003**	**2.15e+003**	3.51e+003	4.89e+002	+	**3.16e+003**	**4.83e+002**	**1.82e+002**	**8.74e+001**	−	4.10e+002	3.28e+002
F6	1.37e+008	1.23e+008	+	**1.16e+007**	**1.24e+007**	3.30e+005	7.84e+005	+	**7.83e+002**	**1.29e+003**	5.39e+001	2.46e+001	=	4.25e+001	2.74e+ 001
F7	3.70e+003	3.51e+002	+	**4.25e+002**	**1.23e+002**	9.66e+000	1.29e+001	+	**5.20e−002**	**1.04e−001**	**1.18e−003**	**3.27e−003**	−	2.98e−002	6.04e−002

Table 1 continued

Functions	DE/rand-to-best/1			DE-DDI/rand-to-best/1		MDE_pBX			MDE_pBX-DDI		O DE			O DE-DDI	
F8	**2.09e+001**	**3.82e−002**	=	2.10e+001	4.00e−002	2.09e+001	4.47e−002	=	2.09e+001	6.20e−002	2.10e+001	4.88e−002	=	2.10e+001	5.39e−002
F9	6.57e+001	1.54e+001	+	**5.00e+001**	**1.16e+001**	**1.60e+001**	**4.30e+000**	=	1.74e+001	4.46e+000	7.89e+001	2.99e+001	+	**2.35e+001**	**1.02e+001**
F10	**8.80e+001**	**1.80e+001**	=	1.05e+002	4.32e+001	2.78e+001	8.34e+000	=	**2.62e+001**	**5.95e+000**	**5.09e+001**	**4.66e+001**	=	5.24e+001	1.87e+001
F11	**1.30e+001**	**1.88e+000**	−	2.13e+001	4.64e+000	2.41e+001	6.58e+000	+	**1.63e+001**	**5.43e+000**	**9.22e+000**	**9.33e+000**	−	2.00e+001	6.16e+000
F12	**5.90e+004**	**2.17e+004**	−	1.01e+005	3.77e+004	1.05e+004	7.53e+003	+	**4.97e+003**	**6.32e+003**	**2.27e+003**	**2.24e+003**	=	2.30e+003	2.05e+003
F13	**3.25e+000**	**7.09e−001**	−	4.05e+000	1.33e+000	**2.72e+000**	**4.55e−001**	=	2.89e+000	5.54e−001	6.19e+000	2.51e+000	+	**3.70e+000**	**9.29e−001**
F14	1.15e+001	4.79e−001	=	**1.14e+001**	**3.41e−001**	1.15e+ 001	7.45e−001	+	**1.03e+001**	**5.90e−001**	1.32e+001	2.92e−001	=	**1.31e+001**	**3.11e−001**
F15	4.59e+002	8.21e+001	+	**3.87e+002**	**1.16e+002**	3.32e+002	6.28e+001	=	**3.27e+002**	**8.06e+001**	**4.12e+002**	**3.31e+001**	=	4.24e+002	4.35e+001
F16	2.60e+002	2.18e+002	=	**2.50e+002**	**6.75e+001**	**1.11e+002**	**1.53e+002**	=	1.26e+002	1.49e+002	9.02e+001	6.89e+001	=	**7.82e+001**	**4.32e+001**
F17	2.35e+002	1.46e+002	=	**1.93e+002**	**7.53e+001**	1.36e+002	1.51e+002	=	**1.35e+002**	**1.54e+002**	1.56e+002	7.50e+001	=	**1.16e+002**	**6.47e+001**
F18	**9.79e+002**	**2.61e+001**	−	1.00e+003	3.84e+001	**8.88e+002**	**6.10e+001**	=	9.04e+002	4.67e+001	**9.01e+002**	**2.10e+001**	−	9.01e+002	3.05e+001
F19	**9.76e+002**	**4.12e+001**	−	1.01e+003	5.01e+001	**8.75e+002**	**6.44e+001**	=	9.06+002	4.73e+001	9.04e+002	8.17e−001	=	**8.70e+002**	**5.39e+001**
F20	**9.73e+002**	**4.13e+001**	=	9.91e+002	6.01e+001	**8.73e+002**	**6.46e+001**	=	9.05e+002	4.67e+001	9.05e+002	1.94e+000	+	**9.01e+002**	**3.05e+001**
F21	1.02e+003	1.68e+002	+	**7.91e+002**	**3.16e+002**	**5.11e+002**	**1.80e+001**	−	5.26e+002	1.30e+002	5.00e+002	0.00e+000	=	5.00e+002	0.00e+000
F22	9.98e+002	2.49e+001	+	**9.61e+002**	**2.42e+001**	**9.31e+002**	**1.42e+001**	=	9.34e+002	1.67e+001	**9.09e+002**	**1.16e+001**	=	9.12e+002	8.79e+000
F23	1.07e+003	1.12e+002	=	**9.68e+002**	**1.90e+002**	5.74e+002	1.37e+002	=	**5.59e+002**	**1.22e+002**	**5.34e+002**	**3.50e−004**	=	5.50e+002	8.06e+001
F24	8.83e+002	2.77e+002	+	**3.00e+002**	**2.77e+002**	2.00e+002	2.20e−006	=	2.00e+002	0.00e+000	2.00e+002	0.00e+000	=	2.00e+002	0.00e+000
F25	1.51e+003	3.36e+001	+	**1.24e+003**	**3.22e+002**	**2.03e+002**	**4.84e+000**	=	2.04e+002	5.33e+000	2.09e+002	2.51e−001	=	2.09e+002	2.28e−001
w/t/l	−			12/8/5		−			9/15/1		−			6/15/4	

The better values in terms of mean solution error and standard deviation compared between DE and the corresponding DE-DDI variant are highlighted in bold

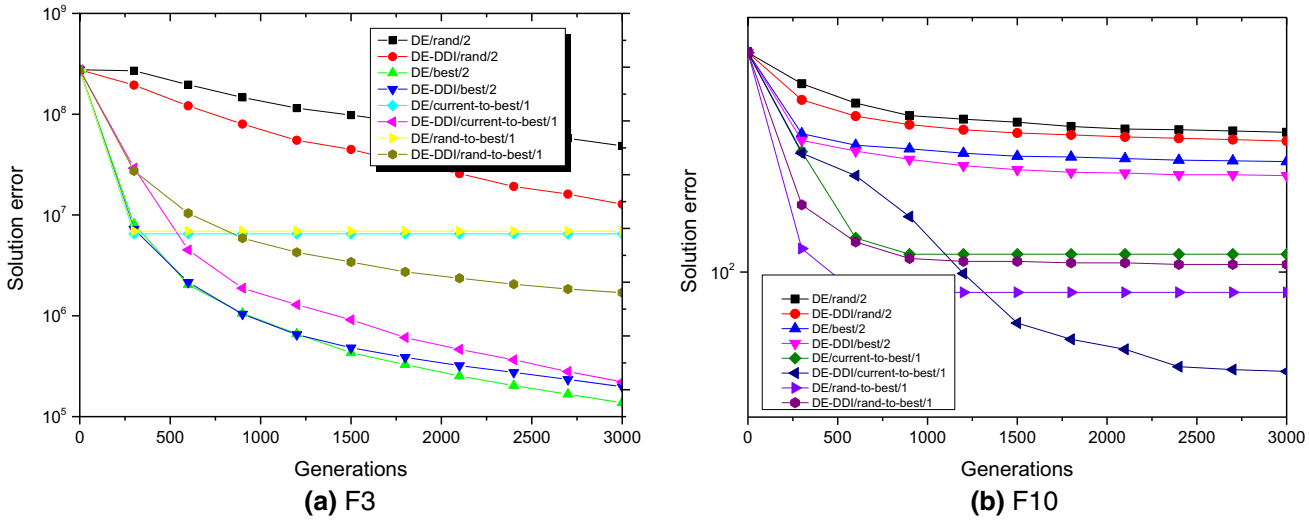

Fig. 3 Convergence graphs of the original DE and the corresponding DE-DDI for the selected functions at $D = 30$

Table 2 Results of the multi-problem Wilcoxon's test for DE-DDI versus the corresponding DE algorithm for all the functions at 30*D*

Algorithm	w/t/l	R+	R−	*p* value	$\alpha = 0.05$	$\alpha = 0.1$
DE-DDI/rand/2 vs DE/rand/2	20/5/0	315	10	3.20E−05	+	+
DE-DDI/best/2 vs DE/best/2	4/18/3	214	86	6.34E−02	=	+
E-DDI/current-to-best/1 vs DE/current-to-best/	19/5/1	294	6	3.70E−05	+	+
DE-DDI/rand-to-best/1 vs DE/rand-to-best/1	12/8/5	259	66	9.06E−03	+	+
MDE_pBX-DDI vs MDE_pBX	9/15/1	191	134	4.27E−01	=	=
ODE-DDI vs O DE	6/15/4	170.5	129.5	5.47E−01	=	=

variants with the corresponding DE-DDI variants, namely MDE_*p*BX (Islam et al. 2012) and ODE (Rahnamayan et al. 2008). All the parameters of these DE variants are set as their original papers. The simulations are carried out on an Intel Core i3 duo personal computer with 3.30-GHz central processing unit and 4-GB random access memory.

In order to show the significant differences among the algorithms, several nonparametric statistical tests (García et al. 2009; Derrac et al. 2011), are also carried out by the KEEL software (Alcalá-Fdez et al. 2009). The results of the single-problem Wilcoxon signed-rank test, at $\alpha = 0.05$ are summarized in the last row of the tables as "*w/t/l*", which means that DE-DDI wins, ties and loses on *w, t* and *l* functions, compared with its corresponding competitor.

5.3 Comparison with original DE algorithms

In this section, several classic DE mutation operators [see Eqs. (2)–(7)] are used in the experimental study. The results for all functions at $D = 30$ are shown in Table 1.

For all the functions at 30*D*, Table 1 shows that in most of the test functions DE-DDI provides significantly better results compared with their corresponding original DE methods. Specifically, for DE/rand/2, it exhibits substantial performance improvements in 20 out of 25 functions. For DE/best/2, DE-DDI is significantly better on four functions. For the exploitation strategy, DE-DDI can enhance the explorative ability of DE/current-to-best/1 to yield significantly performance improvement on 19 out of 25 functions. For DE/rand-to-best/1, DE-DDI is also significantly better on 12 functions.

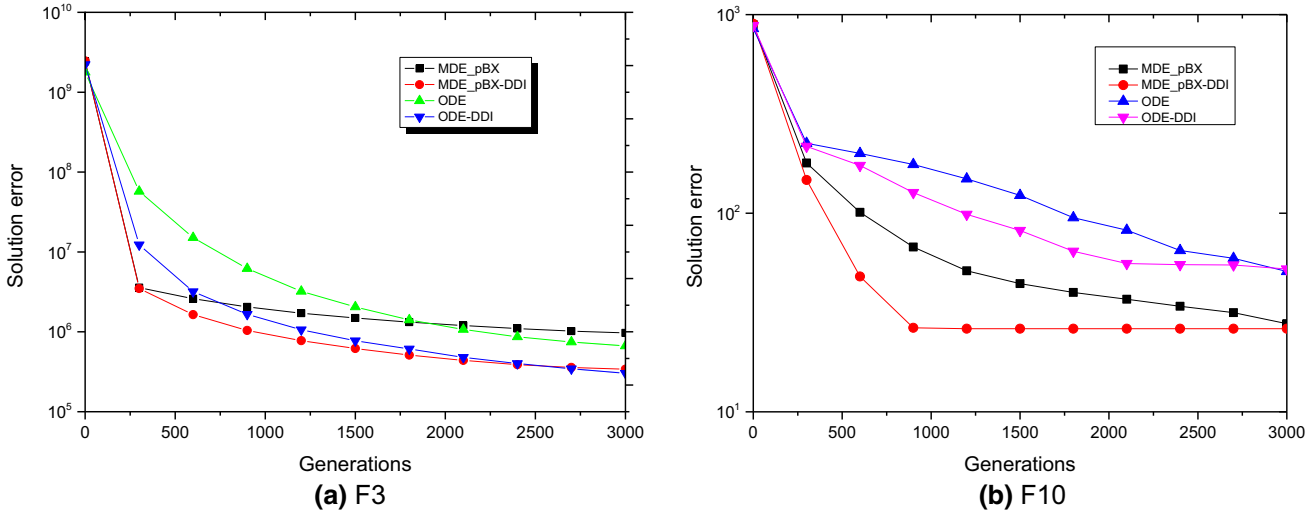

Fig. 4 Convergence graphs of the advanced DE and the corresponding DE-DDI for the selected functions at D = 30

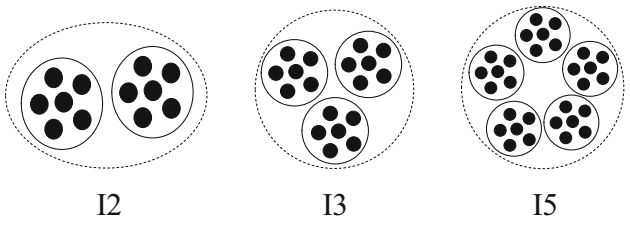

Fig. 5 Distributed topology types

Figure 3 shows that the convergence speed of DE-DDI is better than most of the corresponding original DE algorithms for most selected functions.

Furthermore, for clearly presenting the significant differences between DE-DDI and its corresponding DE method, the multi-problem Wilcoxon signed-rank test is also carried out on all the problems at 30*D*. The results are shown in Table 2. It is clear that DE-DDI can obtain the higher $R+$ values than $R-$ values in all the cases. In addition, the p value in most cases are less than 0.05, which means that DE-DDI is significantly better than most of the original DE algorithms.

5.4 Comparison with advanced DE algorithms

In order to evaluate the effectiveness of DE-DDI for the advanced DE variants, two recently proposed DE variants are employed, i.e., MDE_pBX and ODE.

In Table 1, DE-DDI can obtain significantly better results for MDE_pBX on nine functions. For ODE, DE-DDI is significantly better on six functions and is worse on four functions.

From Fig. 4, it is obvious that DE-DDI is better than the advanced DE variants in terms of the convergence speed for most of the selected functions at 30*D*.

Furthermore, the multi-problem Wilcoxon signed rank tests are employed and the results are shown in Table 2. It is obvious that DE-DDI can obtain the higher $R+$ values than $R-$ values in most cases. These results indicate that DE-DDI is better than most of its corresponding advanced DE variants.

5.5 Benefit of DE-DDI components

In this section, to identify the benefit of the components to DE-DDI, two DE variants is considered, i.e., DE-DIRT which only incorporates the neighborhood information of distributed topology into DE, and DE-DIR which only introduces the direction information into DE. In DE-DIRT, all the vectors for mutation are selected from the neighborhood of the current individual. In DE-DIR, based on the fitness of the randomly selected base vector, the whole population is partitioned into the better and worse groups, and the difference vector is constructed as that in DE-DDI. The experimental studies are carried out on the 25 functions at 30D, and two DE algorithms, i.e., DE/rand/2 and DE/best/2, are employed for comparison. Table 3 presents the results which including the results of the single-problem Wilcoxon signed-rank test and the average ranking values of the four DE variants by Friedman test.

Table 3 Mean and standard deviation of the best error values obtained by original DE, DE-DITR, DE-DIR and DE-DDI on all the functions at 30*D*

Func.	DE/rand/2			DE-DIRT/rand/2			DE-DIR/rand/2			DE-DDI/rand/2	
F1	8.37e−001	2.86e−001	+	**7.17e−009**	**8.06e−009**	+	**2.12e−004**	**1.42e−004**	+	**1.04e−020**	**3.18e−020**
F2	7.65e+003	1.80e+003	+	**4.24e+003**	**1.28e+003**	+	**1.10e+003**	**4.78e+002**	+	**5.56e+002**	**2.62e+002**
F3	4.85e+007	1.30e+007	+	**3.78e+007**	**1.28e+007**	+	**1.51e+007**	**6.79e+006**	+	**1.28e+007**	**4.63e+006**
F4	1.46e+004	2.59e+003	+	**1.17e+004**	**2.40e+003**	+	**5.07e+003**	**1.75e+003**	+	**3.84e+003**	**1.97e+003**
F5	8.07e+003	7.28e+002	+	**6.90e+003**	**8.44e+002**	+	**4.45e+003**	**4.87e+002**	+	**3.65e+003**	**5.02e+002**
F6	5.77e+003	4.08e+003	+	**2.46e+001**	**1.47e+000**	+	**5.03e+001**	**3.59e+001**	+	**2.22e+001**	**2.20e+001**
F7	7.20e+000	2.22e+000	+	**5.06e−001**	**1.89e−001**	+	**8.58e−001**	**1.10e−001**	+	**3.31e−002**	**1.21e−001**
F8	2.09e+001	6.06e−002	=	2.10e+001	5.12e−002	=	2.09e+001	6.51e−002	=	**2.09e+001**	**4.91e−002**
F9	2.10e+002	1.28e+001	=	2.12e+002	9.69e+000	=	**2.07e+002**	**1.06e+001**	=	**2.07e+002**	**1.08e+001**
F10	2.42e+002	9.26e+000	=	**2.34e+002**	**1.41e+001**	+	**2.32e+002**	**1.28e+001**	+	**2.29e+002**	**9.79e+000**
F11	3.93e+001	1.30e+000	=	3.95e+001	1.11e+000	=	3.96e+001	1.19e+000	=	3.96e+001	8.75e−001
F12	5.22e+005	4.47e+004	=	**5.12e+005**	**7.20e+004**	=	**4.95e+005**	**5.02e+004**	=	**4.90e+005**	**5.94e+004**
F13	2.02e+001	9.01e−001	+	**1.92e+001**	**1.10e+000**	+	**1.92e+001**	**1.59e+000**	+	**1.91e+001**	**1.19e+000**
F14	1.34e+001	1.58e−001	=	1.34e+001	1.45e−001	=	1.35e+001	9.64e−002	=	**1.34e+001**	**1.48e−001**
F15	4.05e+002	6.90e+000	+	**4.00e+002**	**0.00e+000**	=	4.05e+002	6.61e+001	+	**4.00e+002**	**0.00e+000**
F16	2.69e+002	1.12e+001	=	**2.63e+002**	**1.40e+001**	+	**2.58e+002**	**1.26e+001**	+	**2.50e+002**	**1.24e+001**
F17	3.05e+002	1.34e+001	=	**2.99e+002**	**1.32e+001**	+	**2.86e+002**	**1.77e+001**	+	**2.88e+002**	**2.70e+001**
F18	9.40e+002	2.18e+000	+	**9.30e+002**	**2.82e+000**	+	**9.28e+002**	**3.15e+000**	+	**9.20e+002**	**2.44e+000**
F19	9.39e+002	3.20e+000	+	**9.30e+002**	**2.74e+000**	+	**9.27e+002**	**3.02e+000**	+	**9.19e+002**	**3.03e+000**
F20	9.39e 002	3.51e+000	+	**9.29e+002**	**2.54e+000**	+	**9.26e+002**	**2.75e+000**	+	**9.20e+002**	**2.94e+000**
F21	5.00e+002	1.23e−001	+	5.00e+002	0.00e+000	+	5.00e+002	3.41e−005	+	5.00e+002	1.53e−005
F22	1.02e+003	1.54e+001	+	**1.01e+003**	**1.61e+001**	+	**9.76e+002**	**1.11e+001**	+	**9.70e+002**	**1.07e+001**
F23	5.35e+002	1.24e000	+	**5.34e+002**	**4.50e−004**	+	**5.34e+002**	**1.08e−003**	+	**5.34e+002**	**3.85e−004**
F24	2.00e+002	1.16e−001	+	2.00e+002	0.00e+000	+	2.00e+002	8.96e−005	+	2.00e+002	0.00e+000
F25	2.29e+002	3.54e+000	+	**2.18e+002**	**2.38e+000**	+	**2.13e+002**	**1.46e+000**	+	**2.12e+002**	**1.03e+000**
w/t/l	−			17/8/0			19/6/0			20/5/0	
ARV	3.54			2.72			2.32			1.42	
Rank	4			3			2			1	
Func.	DE/best/2			DE-DIRT/best/2			DE-DIR/best/2			DE-DDI/best/2	
F1	7.66e−028	2.71e−028	=	8.96e−028	3.50e−028	=	**6.59e−028**	**2.19e−028**	−	1.59e−009	7.54e−009
F2	1.06e−014	1.21e−014	+	**2.31e−021**	**4.18e−021**	−	3.36e−012	3.70e−012	−	4.74e−011	1.07e−010
F3	1.37e+005	7.33e+004	+	**9.18e+004**	**5.28e+004**	=	1.78e+005	9.67e+004	=	1.98e+005	1.19e+005
F4	9.37e−005	1.36e−004	+	**7.97e−006**	**1.37e−005**	−	2.81e−004	3.77e−004	=	1.31e−004	2.90e−004
F5	3.83e+001	5.26e+001	−	1.73e+002	2.27e+002	=	6.62e+001	1.20e+002	=	6.86e+001	6.61e+001
F6	6.38e−001	1.49e+000	=	9.57e−001	1.74e+000	=	**4.78e−001**	**1.32e+000**	−	8.87e+001	1.06e+002
F7	1.55e−002	1.61e−002	=	**1.51e−002**	**1.32e−002**	=	**1.21e−002**	**1.43e−002**	=	**1.26e−002**	**1.13e−002**
F8	2.10e+001	3.58e−002	=	**2.09e+001**	**6.94e−002**	+	**2.09e+001**	**5.85e−002**	=	**2.09e+001**	**5.18e−002**
F9	1.82e+002	1.50e+001	=	1.85e+002	1.18e+001	=	1.86e+002	1.37e+001	+	**1.74e+002**	**8.78e+000**
F10	2.01e+002	1.31e+001	+	**1.94e+002**	**1.50e+001**	=	2.02e+002	1.66e+001	+	**1.84e+002**	**1.30e+001**
F11	3.95e+001	8.57e−001	=	**3.93e+001**	**9.86e−001**	=	**3.92e+001**	**1.02e+000**	=	**3.92e+001**	**1.22e+000**
F12	1.98e+003	4.16e+003	=	2.67e+003	5.72e+003	=	**1.75e+003**	**2.50e+003**	=	**1.42e+003**	**1.62e+003**
F13	1.57e+001	1.46e+000	=	**1.55e+001**	**1.53e+000**	=	**1.53e+001**	**1.71e+000**	=	1.58e+001	1.06+000
F14	1.34e+001	1.72e−001	=	1.34e+001	1.41e−001	=	**1.33e+001**	**2.10e−001**	=	**1.33e+001**	**2.08e−001**
F15	3.21e+002	1.12e+002	=	**3.18e+002**	**1.09e+002**	−	3.71e+002	8.25e+001	=	**3.09e+002**	**9.28e+001**
F16	2.64e+002	8.60e+001	=	3.27e+002	1.14e+002	=	**2.30e+002**	**3.02e+001**	=	**2.32e+002**	**4.63e+001**
F17	3.29e+002	1.09e+002	=	3.45e+002	1.20e+002	=	**2.87e+002**	**7.75e+001**	+	**2.35e+002**	**1.74e+001**
F18	8.86e+002	4.37e+001	−	8.96e+002	4.31e+001	=	9.02e+002	2.14e+001	=	8.86e+02	4.41e+001

Table 3 continued

Func.	DE/best/2			DE-DIRT/best/2			DE-DIR/best/2			DE-DDI/best/2	
F19	8.99e+002	2.98e+001	=	**8.90e+002**	**4.63e+001**	=	9.06e+002	1.99e+000	=	**8.96e+002**	**3.61e+001**
F20	8.99e+002	2.98e+001	=	**8.78e+002**	**5.50e+001**	=	9.06e+002	2.40e+000	=	**8.91e+002**	**4.06e+001**
F21	5.25e+002	8.49e+001	=	5.88e+002	1.71e+002	=	**5.24e+002**	**8.31e+001**	=	**5.12e+002**	**6.00e+001**
F22	9.27e+002	1.35e+001	=	9.27e+002	1.41e+001	=	**9.18e+002**	**1.34e+001**	+	**9.13e+002**	**1.36e+001**
F23	5.69e+002	1.12e+002	=	6.00e+002	1.68e+002	=	6.02e+002	1.53e+002	=	**5.68e+002**	**1.15e+002**
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
F25	2.09e+002	1.03e−001	=	2.21e+002	5.82e+001	=	2.09e+002	1.57e−001	=	2.09e+002	9.31e−002
w/t/l	−			4/19/2			1/21/3			4/18/3	
ARV	2.68			2.70			2.50			2.12	
Rank	3			4			2			1	

The better values in terms of mean solution error and standard deviation compared between DE and the corresponding DE-DDI variant are highlighted in bold

Table 4 Results of the multi-problem Wilcoxon's test between original DE, DE-DIRT, DE-DIR and DE-DDI for all the functions at 30*D*

Algorithm	(1)	(2)	(3)	(4)	Algorithm	(1)	(2)	(3)	(4)
DE/rand/2 (1)	-	16.5 □	16.0 □	10.0□	**DE/best/2 (1)**	-	179.0	173.0	86.0□
DE-DIRT/rand/2 (2)	283.5•	-	67.0□	12.5□	**DE-DIRT/best/2 (2)**	121.0	-	139.5	93.5□
DE-DIR/rand/2 (3)	309.0•	258.0•	-	22.5□	**DE-DIR/best/2 (3)**	152.0	185.5	-	100.0
DE-DDI/rand/2 (4)	315.0•	287.5•	302.5•	-	**DE-DDI/best/2 (4)**	214.0	231.5	200.0	-

Upper diagonal of level significance $\alpha = 0.9$, lower diagonal level of significance $\alpha = 0.95$

• The method in the row improves the method of the column

□ The method in the column improves the method of the row

From Table 3, it is clear that DE-DIRT, DE-DIR and DE-DDI are significantly better than its corresponding original DE algorithm in these two cases. Specifically, for DE/rand/2, DE-DIRT and DE-DIR significantly outperform the original DE algorithm on 17 and 19 functions, respectively, while DE-DDI is significantly better than the original DE algorithm on 20 functions and only loses on no functions. For DE/best/2, DE-DIRT and DE-DIR wins on four and one function, respectively, while DE-DDI significantly obtains better results on four functions.

From the average ranking values obtained by Friedman test, DE-DDI achieves the best value for all the cases. Furthermore, it is interesting to find out that DE-DIRT and DE-DIR perform better than the original DE algorithms in most of functions.

The multi-problem Wilcoxon signed rank tests between the original DE, DE-DIRT, DE-DIR and DE-DDI are also carried out and the results are presented in Table 4. It is obvious that DE-DDI obtains the higher $R+$ values than $R-$ values in all the cases.

5.6 Parameter study

To test the influence of I*n* on the performance of DE-DDI, the used widely DE algorithm, i.e., DE/rand/1, is used for comparison and several types of distributed topology, i.e., I2, I3, I4 and I5, are considered. Figure 5 illustrates the distributed topology with different neighborhood sizes.

From Table 5, it is clear that DE-DDI with different I*n* is better than DE/rand/1 in most of the cases except I5. In the case of I2, I3, I4 and I5, DE-DDI is significantly better than DE/rand/1 on 9, 9, 9 and 8 functions, respectively. According to the results of the statistical tests in Table 6, with I2, I3, I4 and I5, DE-DDI can obtain the higher $R+$ values than $R-$ values. It indicates that DE-DDI with most of different distributed topology types significantly outperforms DE/rand/1 overall.

According to the results in Tables 5 and 6, we observe that DE-DDI is not sensitive to the distributed topology type. For the previous works (Weber et al. 2011, 2010; Neri

Table 5 Mean and standard deviation of the best error values obtained by DE/rand/1 and DE-DDI/rand/1 with different type of distributed topology on all the functions at 30*D*

Functions	DE/rand/1			DE-DDI/rand/1										
				I2			**I3**			**I4**			**I5**	
F1	1.62e−029	5.59e−029	−	1.69e−028	1.63e−028	−	2.01e−028	1.68e−028	−	1.31e−028	1.28e−028	−	3.53e−028	4.25e−028
F2	5.82e−005	4.46e−005	+	**2.13e−015**	**3.55e−015**	+	**2.12e−014**	**3.52e−014**	+	**2.44e−010**	**3.19e−010**	+	**1.92e−006**	**2.51e−006**
F3	4.74e+005	2.65e+005	+	**9.27e+004**	**5.46e+004**	+	**8.40e+004**	**4.35e+004**	+	**1.66e+005**	**8.28e+004**	+	**2.60e+005**	**1.36e+005**
F4	3.28e−002	5.23e−002	+	**1.05e−006**	**2.35e−006**	+	**6.78e−006**	**1.10e−005**	+	**2.10e−003**	**4.91e−003**	=	2.22e−001	5.97e−001
F5	4.77e+001	3.66e+001	−	1.33e+002	1.57e+002	−	7.13e+002	4.24e+002	−	1.22e+003	4.60e+002	−	1.87e+003	4.71e+002
F6	1.88e+000	1.40e+000	=	3.19e+000	3.80e+000	−	3.15e+001	2.50e+001	−	4.71e+001	3.59e+001	−	4.58e+001	3.41e+001
F7	4.93e−004	2.46e−003	−	1.20e−002	1.13e−002	−	2.89e−002	2.37e−002	−	2.82e−002	2.00e−002	−	4.89e−002	5.59e−002
F8	2.10e+001	4.46e−002	−	2.11e+001	6.08e−002	−	2.10e+001	6.65e−002	−	2.11e+001	5.28e−002	−	2.11e+001	7.12e−002
F9	1.33e+002	3.41e+001	+	**3.45e+001**	**1.19e+001**	+	**3.67e+001**	**1.07e+001**	+	**3.34e+001**	**7.05e+000**	+	**3.69e+001**	**1.14e+001**
F10	1.81e+002	9.31e+000	+	**7.09e+001**	**3.26e+001**	+	**8.01e+001**	**2.74e+001**	+	**8.69e+001**	**2.94e+001**	+	**7.44e+001**	**2.39e+001**
F11	3.97e+001	1.06e+000	+	**2.67e+001**	**8.84e+000**	+	**2.72e+001**	**4.96e+000**	+	**2.87e+001**	**5.96e+000**	+	**2.96e+001**	**4.79e+000**
F12	2.41e+003	2.26e+003	=	1.80e+003	2.26e+003	=	1.50e+003	1.43e+003	=	4.73e+003	4.82e+003	−	7.42e+003	5.92e+003
F13	1.50e+001	9.66e−001	+	**4.10e+000**	**9.97e−001**	+	**3.71e+000**	**9.08e−001**	+	**3.85e+000**	**1.36e+000**	+	**3.98e+000**	**1.03e+000**
F14	1.34e+001	1.09e−001	=	1.32e+001	5.29e−001	=	1.34e+001	3.19e−001	–	1.35e+001	4.13e−001	=	1.33e+001	2.94e−001
F15	4.04e+002	3.51e+001	=	4.08e+002	6.40e+001	=	3.72e+002	6.79e+001	=	3.97e+002	6.02e+001	=	3.93e+002	4.83e+001
F16	2.08e+002	7.66e+000	+	**1.07e+002**	**6.61e+001**	+	**1.32e+002**	**8.84e+001**	+	**1.10e+002**	**6.71e+001**	+	**1.00e+002**	**3.12e+001**
F17	2.33e+002	8.35e+000	+	**1.15e+002**	**3.81e+001**	+	**1.08e+002**	**7.39e+001**	+	**1.43e+002**	**9.39e+001**	+	**1.27e+002**	**4.33e+001**
F18	9.00e+002	2.09e+001	−	9.04e+002	2.19e+001	=	8.84e+002	4.81e+001	=	8.90e+002	4.62e+001	=	8.73e+002	6.06e+001
F19	9.00e+002	2.10e+001	−	9.00e+002	3.03e+001	=	8.93e+002	4.14e+001	=	8.68e+002	5.66e+001	=	8.71e+002	5.90e+001
F20	9.00e+002	2.10e+001	−	9.08e+002	2.40e+000	=	8.89e+002	4.53e+001	=	8.92e+002	4.70e+001	=	8.75e+002	5.77e+001
F21	5.00e+002	0.00e+000	=	5.00e+002	0.00e+000	=	5.00e+002	6.23e−006	=	5.00e+002	8.81e−006	=	5.00e+002	1.76e−005
F22	9.07e+002	9.41e+000	−	9.24e+002	1.82e+001	−	9.27e+002	1.87e+001	−	9.39e+002	1.93e+001	−	9.47e+002	2.30e+001
F23	5.34e+002	3.16e−004	−	5.50e+002	8.06e+001	=	5.67e+002	1.12e+002	−	5.67e+002	1.12e+002	−	5.34e+002	7.43e−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
F25	2.09e+002	2.20e−001	=	2.09e+002	2.06e−001	−	2.10e+002	3.18e−001	−	2.10e+002	3.85e−001	−	2.11e+002	3.70e−001
w/t/l	−			9/7/9			9/9/7			9/7/9			8/8/9	
ARV	3.30			2.56			2.56			3.24			3.34	
Ran k	4			2			2			3			5	

The better values in terms of mean solution error and standard deviation compared between DE and the corresponding DE-DDI variant are highlighted in bold

Table 6 Results of the multi-problem Wilcoxon's test for DE/rand/1 versus DE-DDI with different type of distributed topology on all the functions at 30*D*

Algorithm	(1)	(2)	(3)	(4)	(5)
DE_rand_1 (1)	–	130.0	98.0[b]	131.0	111.5
DE-DDI/rand/1 with I2 (2)	195.0	–	171.5	215.0[a]	190.5
DE-DDI/rand/1 with I3 (3)	227.0	153.5	–	254.5[a]	212.5
DE-DDI/rand/1 with I4 (4)	194.0	85.0	70.5[b]	–	155.5
DE-DDI/rand/1 with I5 (5)	188.5	109.5	112.5	144.5	–

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

et al. 2011), we choose I4 for DE-DDI for the benchmark problems in this study.

5.7 Application to three real-world problems

In order to test the effectiveness of DE-DDI on real-world problems, three problems are selected from (Das and Suganthan 2010; Eshelman et al. 1997) for testing in this section. Table 7 presents the results.

As Table 7 shows, it is clear that DE-DDI can obtain the better solutions than the corresponding DE variants in most of the cases. Specifically, for LEP and FMP, DE-DDI both are better than the original and advanced DE variants in three and two cases, respectively. And for SRP, DE-DDI is better than DE in five cases. In sum, the results of Table 7 indicate that DE-DDI is able to enhance the effectiveness of DE on the real-world problems studied.

6 Conclusions

In this paper, a simple and effective framework, DE with distributed direction information based mutation operators (DE-DDI), has been presented for global numerical optimization. First, a simple type of distributed topology is used to define a neighborhood for each individual. Then, the direction information is introduced into the mutation operator by constructing the difference vector with the neighbors. In this way, the distributed direction information composed by neighborhood and direction information can be simultaneously and effectively used to guide the search of DE. From the extensive experimental study, it is clear that DE-DDI is able to enhance the performance of most DE algorithms considered.

Table 7 Mean and standard deviation of the best error values obtained by the de algorithms and DE-DDI for the real-world application problems

Algorithm	LEP		FMP		SRP	
	DE	DE-DDI	DE	DE-DDI	DE	DE-DDI
DE/rand/2	3.36e−002	9.16e−003	**2.23e−010**	**1.68e−010**	1.38e+001	1.86e+000	**6.29e+000**	**7.85e+000**	2.46e+000	1.07e−001	**2.01e+000**	**3.75e−001**
DE/best/2	**0.00e+000**	**0.00e+000**	1.99e−015	8.59e−015	**3.61e+000**	**5.40e+000**	7.24e+000	6.16e+000	2.45e+000	1.16e−001	**1.92e+000**	**3.93e−001**
DE/rand-to-best/1	4.57e−001	1.72e+000	**6.69e−004**	**2.28e−003**	**9.60e+000**	**6.64e+000**	1.05e+001	6.03e+000	2.44e+000	1.32e−001	**1.46e+000**	**3.90e−001**
DE/current-to-best/1	**8.12e−001**	**1.24e+000**	3.17e+000	2.84e+000	1.41e+001	4.09e+000	**9.46e+000**	**7.19e+000**	1.76e+000	5.51e−001	**1.58e+000**	**2.21e−001**
CoDE	2.32e+002	4.08e+001	**8.88e−002**	**7.49e−002**	**5.53e+000**	**2.91e+000**	8.74e+000	5.79e+000	2.16e+000	8.30e−002	**1.57e+000**	**2.80e−001**
MDE_pBX	**2.52e−003**	**6.30e−003**	5.78e+002	2.82e+002	**3.87e+000**	**6.63e+000**	2.55e+001	1.78e+000	**2.00e+000**	**2.51e−001**	2.99e+000	3.06e−001

The better values in terms of mean solution error and standard deviation compared between DE and the corresponding DE-DDI variant are highlighted in bold

In the future, DE-DDI will be applied to other DE variants and more real-world problems.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China (61305085), 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, del Jesús MJ, Ventura S, Garrell J, Otero J, Romero C, Bacardit J, Rivas VM (2009) KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput 13(3):307–318

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

Cai Y, Du J (2014) Enhanced differential evolution with adaptive direction information. In: proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC 2014), Beijing, China, IEEE, pp 305–312

Cai Y, Wang J (2013) Differential evolution with neighborhood and direction information for numerical optimization. Cybern IEEE Trans 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, Du J, Chen W (2014a) Enhancing the search ability of differential evolution through competent leader. Int J High Perform Syst Archit 5(1):50–62

Cai Y, Wang J, Chen Y, Wang T, Tian H, Luo W (2014b) Adaptive direction information in differential evolution for numerical optimization. Soft Comput:1–30

Campomanes-Álvarez BR, Cordón Ó, Damas S, Ibáñez Ó (2014) Computer-based craniofacial superimposition in forensic identification using soft computing. J Ambient Intell Humaniz Comput 5(5):683–697

Das S, Konar A (2006) Two-dimensional IIR filter design with modern search heuristics: A comparative study. Int J Comput Intell Appl 6(03):329–355

Das S, Suganthan P (2010) Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Jadavpur University, Kolkata

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

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

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

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

Eshelman LJ, Mathias KE, Schaffer JD (1997) Convergence controlled variation. Found Genet Algorithms 4:203–224

García S, Fernández 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

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. Syst Man Cybern Part B Cybern IEEE Trans 42(2):482–500

Li Y, Zhan Z, Gong Y, Chen W, Zhang J, Li Y (2015) Differential evolution with an evolution path: a deep evolutionary algorithm. IEEE Trans Cybern. doi:10.1109/TCYB.2014.2360752

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. AI 2010: advances in artificial intelligence. Springer, Heidelberg, pp 293–302

Philip Chen C, Zhang C-Y (2014) Data-intensive applications, challenges, techniques and technologies: a survey on big data. Inf Sci 275:314–347

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

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

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

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

Sun J, Zhang Q, Tsang EP (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. doi:10.1109/TEVC.2014.2360890

Thomas SA, Jin Y (2014) Reconstructing biological gene regulatory networks: where optimization meets big data. Evol Intel 7(1):29–47

Wang J, Cai Y (2015) Multiobjective evolutionary algorithm for frequency assignment problem in satellite communications. Soft Comput

Wang Y-X, Xiang Q-L (2008) Exploring new learning strategies in differential evolution algorithm. In: Evolutionary computation, 2008. CEC 2008 (IEEE World Congress on computational intelligence), IEEE Congress on, 2008, IEEE, pp 204–209

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

Wang J, Zhong C, Zhou Y, Zhou Y (2014b) Multiobjective optimization algorithm with objective-wise learning for continuous multiobjective problems. J Ambient Intell Humaniz Comput. doi:10.1007/s12652-014-0218-y

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 45(2):302–315

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

Zhou Y, Wang J (2015) A local search-based multiobjective optimization algorithm for multiobjective vehicle routing problem with time windows. IEEE Syst J. doi:10.1109/JSYST.2014.2300201