

Solving large-scale global optimization problems using enhanced adaptive differential evolution algorithm

Ali Wagdy Mohamed¹

Received: 24 October 2016 / Accepted: 20 March 2017 / Published online: 6 April 2017
© The Author(s) 2017. This article is an open access publication

Abstract This paper presents enhanced adaptive differential evolution (EADE) algorithm for solving high-dimensional optimization problems over continuous space. To utilize the information of good and bad vectors in the DE population, the proposed algorithm introduces a new mutation rule. It uses two random chosen vectors of the top and bottom $100p\%$ individuals in the current population of size NP , while the third vector is selected randomly from the middle $[NP-2(100p\%)]$ individuals. The mutation rule is combined with the basic mutation strategy DE/rand/1/bin, where the only one of the two mutation rules is applied with the probability of 0.5. This new mutation scheme helps to maintain effectively the balance between the global exploration and local exploitation abilities for searching process of the DE. Furthermore, we propose a novel self-adaptive scheme for gradual change of the values of the crossover rate that can excellently benefit from the past experience of the individuals in the search space during evolution process which, in turn, can considerably balance the common trade-off between the population diversity and convergence speed. The proposed algorithm has been evaluated on the 7 and 20 standard high-dimensional benchmark numerical optimization problems for both the IEEE CEC-2008 and the IEEE CEC-2010 Special Session and Competition on Large-Scale Global Optimization. The comparison results between EADE and its version and the other state-of-art algorithms that were all tested on these test suites indicate that the proposed algorithm and its version are highly competitive algorithms for solving large-scale global optimization problems.

Keywords Evolutionary computation · Global optimization · Differential evolution · Novel mutation · Self-adaptive crossover

Introduction

In general, global numerical optimization problem can be expressed as follows (without loss of generality, minimization problem is considered here):

$$\begin{aligned} \min f(x), \vec{x} = [x_1, x_2, \dots, x_D] \in \mathbb{R}^D; [x_j^L, x_j^U], \\ \forall j = 1, 2, \dots, D \end{aligned} \quad (1)$$

where f is the objective function, \vec{x} is the decision vector $\in \mathbb{R}^D$ space consisting of D variables, D is the problem dimension, i.e., the number of variables to be optimized, and x_j^L and x_j^U are the lower and upper bounds for each decision variable, respectively. The optimization of the large-scale problems of this kind (i.e. $D = 1000$) is considered as a challenging task, since the solution space of a problem often increases exponentially with the problem dimension and the characteristics of a problem may change with the scale [1]. Generally speaking, there are different types of real-world large-scale global optimization (LSGO) problems in engineering, manufacturing, economy applications, such as (bio-computing, data or web mining, scheduling, vehicle routing, etc.). To draw more attention to this challenge of optimization, the first competition on (LSGO) was held in CEC 2008 [2]. Consequently, in the recent few years, (LSGO) has gained considerable attention and has attracted much interest from Operations Research and Computer Science professionals, researchers, and practitioners as well as mathematicians and engineers. Therefore, the challenges mentioned above

✉ Ali Wagdy Mohamed
aliwagdy@gmail.com

¹ Operations Research Department, Institute of Statistical Studies and Research, Cairo University, Giza 12613, Egypt

have motivated the researchers to design and improve many kinds of efficient, effective, and robust various kinds of meta-heuristic algorithms that can solve (LSGO) problems with high-quality solution and high convergence performance with low computational cost. Evolutionary algorithms (EAs) have been proposed to meet the global optimization challenges. The structure of (EAs) has been inspired from the mechanisms of natural evolution. Due to their adaptability and robustness, EAs are especially capable in solving difficult optimization problems, such as highly nonlinear, non-convex, non-differentiable, and multi-modal optimization problems. In general, the process of (EAs) is based on the exploration and the exploitation of the search space through selection and reproduction operators [3]. Similar to other evolutionary algorithms (EAs), differential evolution (DE) is a stochastic population-based search method, proposed by Storn and Price [4]. The advantages are its simple of implementation, ease of use, speed, and robustness. Due to these advantages, it has successfully been applied for solving many real-world applications, such as admission capacity planning in higher education [5,6], financial markets dynamic modeling [7], solar energy [8], and many others. In addition, many recent studies prove that the performance of DE is highly competitive with and in many cases superior to other EAs in solving unconstrained optimization problems, constrained optimization problems, multi-objective optimization problems, and other complex optimization problems [9]. However, DE has many weaknesses as all other evolutionary search techniques. In general, DE has a good global exploration ability that can reach the region of global optimum, but it is slow at exploitation of the solution [10]. In addition, the parameters of DE are problem-dependent and it is difficult to adjust them for different problems. Moreover, DE performance decreases as search space dimensionality increases [11]. Finally, the performance of DE deteriorates significantly when the problems of premature convergence and/or stagnation occur [11,12]. The performance of DE basically depends on the mutation strategy, the crossover operator. Besides, the intrinsic control parameters (population size NP , scaling factor F , and the crossover rate CR) play a vital role in balancing the diversity of population and convergence speed of the algorithm. For the original DE, these parameters are user-defined and kept fixed during the run. However, many recent studies indicate that the performance of DE is highly affected by the parameter setting and the choice of the optimal values of parameters is always problem-dependent. Moreover, prior to an actual optimization process, the traditional time-consuming trial-and-error method is used for fine-tuning the control parameters for each problem. Alternatively, to achieve acceptable results even for the same problem, different parameter settings along with different mutation schemes at different stages of evolution are needed. Therefore, some techniques have been designed to

adjust control parameters in adaptive or self-adaptive manner instead of trial-and-error procedure plus new mutation rules have been developed to improve the search capability of DE [13–22]. Based on the above considerations, in this paper, we present a novel DE, referred as EADE, including two novel modifications: novel mutation rule and self-adaptive scheme for gradual change of CR values. In EADE, a novel mutation rule can balance the global exploration ability and the local exploitation tendency and enhance the convergence rate of the algorithm. Furthermore, a novel adaptation schemes for CR is developed that can benefit from the past experience through generations of evolutionary. Scaling factors are produced according to a uniform distribution to balance the global exploration and local exploitation during the evolution process. EADE has been tested on 20 benchmark test functions developed for the 2010 IEEE Congress on Evolutionary Computation (IEEE CEC 2010) [1]. Furthermore, EADE has been also tested on 7 benchmark test functions developed for the 2008 IEEE Congress on Evolutionary Computation (IEEE CEC 2008) [2]. The experimental results indicate that the proposed algorithm and its two versions are highly competitive algorithms for solving large-scale global optimization problems. The remainder of this paper is organized as follows. The next section briefly introduces DE and its operators followed by which the related work is reviewed. In the subsequent section, EADE algorithm is presented. The experimental results are given before the concluding section. Finally, the conclusions and future works are presented.

Differential evolution (DE)

This section provides a brief summary of the basic Differential Evolution (DE) algorithm. In simple DE, generally known as DE/rand/1/bin [23,24], an initial random population consists of NP vectors \vec{X} , $\forall i = 1, 2, \dots, NP$, is randomly generated according to a uniform distribution within the lower and upper boundaries (x_j^L, x_j^U) . After initialization, these individuals are evolved by DE operators (mutation and crossover) to generate a trial vector. A comparison between the parent and its trial vector is then done to select the vector which should survive to the next generation [9]. DE steps are discussed below:

Initialization

To establish a starting point for the optimization process, an initial population P^0 must be created. Typically, each j th component ($j = 1, 2, \dots, D$) of the i th individuals ($i = 1, 2, \dots, NP$) in the P^0 is obtained as follows:

$$x_{j,i}^0 = x_{j,L} + \text{rand}(0, 1) \cdot (x_{j,U} - x_{j,L}) \quad (2)$$

where rand (0,1) returns a uniformly distributed random number in [0, 1].

Mutation

At generation G, for each target vector x_i^G , a mutant vector v_i^G is generated according to the following:

$$v_i^G = x_{r_1}^G + F \cdot (x_{r_2}^G - x_{r_3}^G), r_1 \neq r_2 \neq r_3 \neq i \tag{3}$$

With randomly chosen indices $r_1, r_2, r_3 \in \{1, 2, \dots, NP\}$, F is a real number to control the amplification of the difference vector $(x_{r_2}^G - x_{r_3}^G)$. According to Storn and Price [4], the range of F is in [0, 2]. In this work, if a component of a mutant vector violates search space, then the value of this component is generated a new using (2). The other most frequently used mutations strategies are

$$\text{“DE/best/1”[17] : } v_i^G = x_{\text{best}}^G + F \cdot (x_{r_1}^G - x_{r_2}^G) \tag{4}$$

$$\text{“DE/best/2”[17] : } v_i^G = x_{\text{best}}^G + F \cdot (x_{r_1}^G - x_{r_2}^G) + F \cdot (x_{r_3}^G - x_{r_4}^G) \tag{5}$$

$$\text{“DE/rand/2”[17] : } v_i^G = x_{r_1}^G + F \cdot (x_{r_2}^G - x_{r_3}^G) + F \cdot (x_{r_4}^G - x_{r_5}^G) \tag{6}$$

$$\text{“DE/current-to-best/1”[17] : } v_i^G = x_i^G + F \cdot (x_{\text{best}}^G - x_i^G) + F \cdot (x_{r_1}^G - x_{r_2}^G) \tag{7}$$

$$\text{“DE/current-to-rand/1”[17] : } v_i^G = x_i^G + F \cdot (x_{r_1}^G - x_i^G) + F \cdot (x_{r_2}^G - x_{r_3}^G). \tag{8}$$

The indices r_1, r_2, r_3, r_4, r_5 are mutually integers randomly generated within the range [1, NP], which are also different from the index i . These indices are randomly generated once for each mutant vector. The scale factor F is a positive control parameter for scaling the difference vector. x_{best}^G is the best individual vector with the best fitness value in the population at generation G.

Crossover

There are two main crossover types, binomial and exponential. We here elaborate the binomial crossover. In the binomial crossover, the target vector is mixed with the mutated vector, using the following scheme, to yield the trial vector u_i^G .

$$u_{j,i}^G = \begin{cases} v_{j,i}^G, & \text{if } (\text{rand}_{j,i} \leq CR \text{ or } j = j_{\text{rand}}) \\ x_{j,i}^G, & \text{otherwise} \end{cases} \tag{9}$$

where $\text{rand}_{j,i}$ ($i \in [1, NP]$ and $j \in [1, D]$) is a uniformly distributed random number in [0,1], $CR \in [0, 1]$ called the crossover rate that controls how many components are inherited from the mutant vector, j_{rand} is a uniformly distributed

random integer in [1, D] that makes sure at least one component of trial vector is inherited from the mutant vector.

Selection

DE adapts a greedy selection strategy. If and only if the trial vector u_i^G yields as good as or a better fitness function value than x_i^G , then u_i^G is set to x_i^{G+1} . Otherwise, the old vector x_i^G is retained. The selection scheme is as follows (for a minimization problem):

$$x_i^{G+1} = \begin{cases} u_i^G, & f(u_i^G) \leq f(x_i^G) \\ x_i^G, & \text{otherwise} \end{cases} \tag{10}$$

A detailed description of standard DE algorithm is given in Fig. 1.

Related work

As previously mentioned, during the past few years, LSGO has attracted much attention by the researches due to its significance as many real-world problems and applications are high-dimensional problems in nature. Basically, the current EA-based LSGO research can be classified into two categories:

- Cooperative Co-evolution (CC) framework algorithms or divide-and-conquer methods.
- Non Cooperative Co-evolution (CC) framework algorithms or no divide-and-conquer methods.

Cooperative Co-evolution (CC) has become a popular and effective technique in Evolutionary Algorithms (EAs) for large-scale global optimization since its initiation in the publication of Potter and De Jong [26]. The main idea of CC is to partition the LSGO problem into a number of sub-problems, i.e., the decision variables of the problem are divided into smaller subcomponents, each of which is optimized using a separate EA. Using this divide-and-conquer method, the classical EAs are able to effectively solve many separable problems [26]. CC shows better performance on separable problems, but deteriorated on non-separable problems, because the interacting variables could not be grouped in one subcomponent. Recently, different versions of CC-based EAs have been developed and shown excellent performance. Yang et al. [27] proposed a new decomposition strategy called random grouping as a simple way of increasing the probability of grouping interacting variables in one subcomponent. According to this strategy, without any prior knowledge of the non-separability of a problem, subdivide a n-dimensional decision vector into m s-dimensional subcomponents. Later, Omidvar et al. [28] proposed DECC-DML algorithm which

```

01. Begin
02. G=0
03. Create a random initial population  $\bar{x}_i^G \forall i, i = 1, \dots, NP$ 
04. Evaluate  $f(\bar{x}_i^G) \forall i, i = 1, \dots, NP$ 
05. For G=1 to Gmax Do
06.   For i=1 to NP Do
07.     Select randomly  $r1 \neq r2 \neq r3 \neq i \in [1, NP]$ 
08.      $j_{rand} = \text{randint}(1, D)$ 
09.     For j=1 to D Do
10.       If  $(\text{rand}_{i,j}[0,1] < CR \text{ or } j = j_{rand})$  Then
11.          $u_{i,j}^G = x_{r1,j}^G + F \cdot (x_{r2,j}^G - x_{r3,j}^G)$ 
12.       Else
13.          $u_{i,j}^G = x_{i,j}^G$ 
14.       End If
15.     End For
16.     If  $(f(\bar{u}_i^G) \leq f(\bar{x}_i^G))$  Then
17.        $\bar{x}_i^{G+1} = \bar{u}_i^G$ 
18.     Else
19.        $\bar{x}_i^{G+1} = \bar{x}_i^G$ 
20.     End If
21.   End For
22.   G=G+1
23. End For
24. End

```

Fig. 1 Description of standard DE algorithm. $\text{rand}[0,1)$ is a function that returns a real number between 0 and 1. $\text{randint}(\text{min}, \text{max})$ is a function that returns an integer number between min and max. NP , G_{max} , CR , and F are user-defined parameters. D is the dimensionality of the problem

is a differential evolution algorithm adopting CC frame. They suggested a new decomposition strategy called delta grouping. The central idea of this technique was the improvement interval of interacting variables that would be limited if they were in different subcomponents. Delta method measures the averaged difference in a certain variable across the entire population and uses it for identifying interacting variables. The experimental results show that this new method is more effective than the existing random grouping method. However, DECC-DML is less efficient on non-separable functions with more than one group of rotated variables. Many CC-based algorithms have been developed during the past decade such as FEPCC [29], DECC-I, DECC-II [30], MLCC [31], SEE [32], and AM-CCPSO [33]. On the other hand, there are many other approaches that optimize LSGO problems as a whole, that is, no divide-and-conquer methods were used. Actually, it is considered as a challenging task as it needs to develop novel evolutionary operators that can promote and strengthen the capability of the algorithms to improve the overall optimization process in high-dimensional search space. Takahama and Sakai [34] proposed DE with landscape

modality detection and a diversity archive (LMDEa). In this method, the landscape modality is observed at every fixed generation. Based on the modality detection, F is controlled dynamically. LMDEa showed excellent performance for the large-scale optimization problems. Brest et al. [35] presented self-adaptive Differential Evolution algorithm (jDElsgo). In this approach, self-adaptive F and Cr control parameters and “rand/1/bin” strategy along with population size reduction mechanism are used. Similarly, Wang et al. [36] introduced a sequential Differential Evolution (DE) enhanced by neighborhood search (SDENS), where hybrid crossover strategy “rand/1/bin” and “rand/1/exp” are used. To search the neighbors of each individual, two trial individuals by local and global neighborhood search strategies are created. Then, the fittest one among the current individual and the two created trial individuals is selected as a new current individual. Molina et al. [37] put forward a memetic algorithm based on local search chains, named MA-SW-Chains, which assigned local search intensity to each individual depending on its features by changing different local search applications. Kabán et al. [38] proposed some fundamental roots

of the problem and make a start at developing a new and generic framework to yield effective and efficient estimations of distribution algorithms (EDAs)-type algorithms for large-scale continuous global optimization problems. Cheng and Jin [39] introduced a novel competitive swarm optimizer (CSO). The algorithm is fundamentally inspired by the particle swarm optimization but is conceptually very different. In the proposed CSO, neither the personal best position of each particle nor the global best position (or neighborhood best positions) is involved in updating the particles. Instead, a pair wise competition mechanism is introduced, where the particle that loses the competition will update its position by learning from the winner. Recently, different versions of non-CC-based EAs have been developed and shown excellent performance such as SL-PSO [40], GODE [41], and EDA-MCC [42]. In general, the proposed EADE algorithm belongs to this category.

EADE algorithm

In this section, we outline a novel DE algorithm, EADE, and explain the steps of the algorithm in details.

Novel mutation scheme

DE/rand/1 is the fundamental mutation strategy developed by Storn and Price [23,25], and it is reported to be the most successful and widely used scheme in the literature [9]. Obviously, in this strategy, the three vectors are chosen from the population at random for mutation and the base vector is then selected at random among the three. The other two vectors form the difference vector that is added to the base vector. Consequently, it is able to maintain population diversity and global search capability with no bias to any specific search direction, but it slows down the convergence speed of DE algorithms [15]. DE/rand/2 strategy, like the former scheme with extra two vectors that form another difference vector, which might lead to better perturbation than one-difference-vector-based strategies [15]. Furthermore, it can provide more various differential trial vectors than the DE/rand/1/bin strategy which increase its exploration ability of the search space. On the other hand, greedy strategies like DE/best/1, DE/best/2, and DE/current-to-best/1 incorporate the information of the best solution found so far in the evolutionary process to increase the local search tendency that leads to fast convergence speed of the algorithm. However, the diversity of the population and exploration capability of the algorithm may deteriorate or may be completely lost through a very small number of generations, i.e., at the beginning of the optimization process, that cause problems such stagnation and/or premature convergence. Consequently, to overcome the shortcomings of both types of mutation strategies, most of

the recent successful algorithms utilize the strategy candidate pool that combines different trail vector generation strategies that have diverse characteristics and distinct optimization capabilities, with different control parameter settings to be able to deal with a variety of problems with different features at different stages of evolution [15,17,41]. Contrarily, taking into consideration the weakness of existing greedy strategies, [16] introduced a new differential evolution (DE) algorithm, named JADE, to improve optimization performance by implementing a new mutation strategy “DE/current-to-pbest” with optional external archive and updating control parameters in an adaptive manner. Consequently, proposing new mutations strategies that can considerably improve the search capability of DE algorithms and increase the possibility of achieving promising and successful results in complex and large-scale optimization problems is still an open challenges for the evolutionary computation research. Therefore, this research uses a new mutation rule with a view of balancing the global exploration ability and the local exploitation tendency and enhancing the convergence rate of the algorithm. The proposed mutation strategy uses two random chosen vectors of the top and bottom $100p\%$ individuals in the current population of size NP , while the third vector is selected randomly from the middle ($NP-2(100p\%)$) individuals. The proposed mutation vector is generated in the following manner:

$$v_i^{G+1} = x_r^G + F1 \cdot (x_{p_best}^G - x_r^G) + F2 \cdot (x_r^G - x_{p_worst}^G) \quad (11)$$

where x_r^G is a random chosen vector from the middle ($NP-2(100p\%)$) individuals, and $x_{p_best}^G$ and $x_{p_worst}^G$ are randomly chosen as one of the top and bottom $100p\%$ individuals in the current population, respectively, with $p \in (0, 1]$, $F1$ and $F2$ are the mutation factors that are independently generated according to uniform distribution in $(0,1)$. Really, the main idea of the proposed novel mutation is based on that each vector learns from the position of the top best and the bottom worst individuals among the entire population of a particular generation. Obviously, from mutation Eq. (11), it can be observed that the incorporation of the objective function value in the mutation scheme has two benefits. First, the target vectors are not always attracted toward the same best position found so far by the entire population. Thus, the premature convergence at local optima can be almost avoided by following the same direction of the top best vectors which preserves the exploration capability. Secondly, avoiding the direction of the bottom worst vectors can enhances the exploitation tendency by guiding the search process to the promising regions of the search space, i.e., it concentrates the exploitation of some sub-regions of the search space. Therefore, the directed perturbations in the proposed mutation resemble the concept

of gradient as the difference vectors are directed from the worst vectors to the best vectors [43]. Thus, it is considerably used to explore the landscape of the objective function being optimized in different sub-regions around the best vectors within search space through optimization process. Thus, by utilizing and sharing the best and worst information of the DE population, the proposed directed mutation balances both global exploration capability and local exploitation tendency. The new mutation strategy is embedded into the DE algorithm and combined with the basic mutation strategy DE/rand/1/bin, where only one of the two mutation rules is applied with the probability of 0.5.

Parameter adaptation schemes in EADE

The successful performance of DE algorithm is significantly dependent upon the choice of its three control parameters: The scaling factor F , crossover rate CR, and population size NP [23,25]. In fact, they have a vital role, because they greatly influence the effectiveness, efficiency, and robustness of the algorithm. Furthermore, it is difficult to determine the optimal values of the control parameters for a variety of problems with different characteristics at different stages of evolution. In the proposed HDE algorithm, NP is kept as a user-specified parameter, since it highly depends on the problem complexity. Generally speaking, F is an important parameter that controls the evolving rate of the population, i.e., it is closely related to the convergence speed [15]. A small F value encourages the exploitation tendency of the algorithm that makes the search focus on neighborhood of the current solutions; hence, it can enhance the convergence speed. However, it may also lead to premature convergence [43]. On the other hand, a large F value improves the exploration capability of the algorithm that can makes the mutant vectors distribute widely in the search space and can increase the diversity of the population [43]. However, it may slow down the search [43] with respect to the scaling factors in the proposed algorithm, at each generation G , the scale factors $F1$ and $F2$ of each individual target vector are independently generated according to uniform distribution in $(0,1)$ to enrich the search behavior. The constant crossover (CR) reflects the probability with which the trial individual inherits the actual individual's genes, i.e., which and how many components are mutated in each element of the current population [17,43]. The constant crossover CR practically controls the diversity of the population [44]. As a matter of fact, if CR is high, this will increase the population diversity. Nevertheless, the stability of the algorithm may reduce. On the other hand, small values of CR increase the possibility of stagnation that may weak the exploration ability of the algorithm to open up new search space. In addition, CR is usually more sensitive to problems with different characteristics such as unimodality and multi-modality, and separable and non-separable prob-

lems. For separable problems, CR from the range $(0, 0.2)$ is the best, while for multi-modal, parameter dependent problems, CR in the range $(0.9,1)$ is suitable [45]. On the other hand, there are wide varieties of approaches for adapting or self-adapting control parameters values through optimization process. Most of these methods based on generating random values from uniform, normal, or Cauchy distributions or by generating different values from pre-defined parameter candidate pool besides use the previous experience (of generating better solutions) to guide the adaptation of these parameters [11, 15–17, 19, 45–49]. The presented work proposed a novel self-adaptation scheme for CR. The core idea of the proposed self-adaptation scheme for the crossover rate CR is based on the following fundamental principle. In the initial stage of the search process, the difference among individual vectors is large, because the vectors in the population are completely dispersed or the population diversity is large due to the random distribution of the individuals in the search space that requires a relatively smaller crossover value. Then, as the population evolves through generations, the diversity of the population decreases as the vectors in the population are clustered, because each individual gets closer to the best vector found so far. Consequently, to maintain the population diversity and improve the convergence speed, crossover should be gradually utilized with larger values along with the generations of evolution increased to preserve well genes in so far as possible and promote the convergence performance. Therefore, the population diversity can be greatly enhanced through generations. However, there is no an appropriate CR value that balances both the diversity and convergence speed when solve a given problem during overall optimization process. Consequently, to address this problem and following the SaDE algorithm [15], in this paper, a novel adaptation scheme for CR is developed that can benefit from the past experience through generations of evolutionary.

Crossover rate Adaptation At each generation G , the crossover probability CR_i of each individual target vector is independently generated randomly from pool A according to uniform distribution and the following procedure exists through generations. Where A is the pool of values of crossover rate CR that changes during and after the learning period LP, we set $LP = 10\%$ of GEN, G is the current generation number, and GEN is the maximum number of generations. The lower and upper limits of the ranges for (G) are experimentally determined, $CR_Flag_List [i]$ is the list that contains one of two binary values $(0,1)$ for each individual i through generation G , where 0 represents failure, no improvement, when the target vector is better than the trial vector during and after the learning period and 1 represent success, improvement, when the trial vector is better than the target vector during and after the learning period, the $failure_counter_list [i]$ is the list that monitors the work-

ing of individuals in terms of fitness function value during generations after completion the learning period, if there is no improvement in fitness, then the failure counter of this target vector is increased by unity. This process is repeated until it achieves pre-specified value of $Max_failure_counter$ which assigned a value 20 that is experimentally determined, $CR_Ratio_List [k]$ is the list that records the relative change improvement ratios between the trial and target objective function values with respect to each value k of the pool of values A of CR through generation G . It can be clearly seen from procedure 1 that, at $G = 1$, $CR = 0.05$ for each target vector and then, at each generation G , if the generated trial vector produced is better than the target vector, the relative change improvement ratio (RCIR) associated with this CR value is computed and the correspondence ratio is updated. On the other hand, during the learning period, if the target vector is better than the trial vector, then the CR value is chosen randomly from the associated pool A of CR values, that is gradually added more values, according to generation number and hence, for this CR value, there is no improvement and its ratio remains unchanged. However, after termination of the learning period, if the target vector is better than the trial vector, i.e., if there is no improvement in fitness, then the $failure_counter$ is increased by one in each generation till this value achieves a pre-specified value of $Max_failure_counter$ which assigned a value 20, then this CR value should change to a new value that is randomly selected from the pool A of CR values that is taken in range 0.1–0.9 in steps of 0.1 and 0.05 and 0.95 which are also included as lower and upper values, respectively. Note that the (RCIR) is only updated if there is an improvement; Otherwise, it remains constant. Thus, the CR value with maximum ratio is continuously changing according to the evolution process at each subsequent generation. In fact, although all test problems included in this study have optimum of zero, the absolute value is used in calculating (RCIR) as a general rule to deal with positive, negative, or mixed values of objective function. Concretely, Fig. 2 shows that, during the first half of the learning period, the construction of pool A of CR values ensures the diversity of the population, such that the crossover probability for i th individual target increases gradually in staircase along with the generations of evolution process increased. Taking into consideration that the probability of chosen small CR values is greater than the probability of chosen the larger CR values as the diversity of the population is still large. In addition, in the second half of the learning period, there is a larger values of 0.9 and 0.95 which are added to the pool as it favors non-separable functions. However, all the values have an equally likely chance of occurrence to keep on the diversity with different values of CR. Consequently, the successful CR values with high relative change improvement ratio in this period will be survive to be used in the next generations of the optimization process until it fails to achieve improvement for a

specific value of 20, then it must be changed randomly by a new value. Thus, the value of CR is adaptively changed as the diversity of the population changes through generations. Distinctly, it varies from one individual to another during generations, and also, it is different from one function to another one being optimized. In general, adaptive control parameters with different values during the optimization process in successive generations enrich the algorithm with controlled-randomness which enhances the global optimization performance of the algorithm in terms of exploration and exploitation capabilities. Therefore, it can be concluded that the proposed novel adaptation scheme for gradual change of the values of the crossover rate can excellently benefit from the past experience of the individuals in the search space during evolution process which, in turn, can considerably balance the common trade-off between the population diversity and convergence speed. The pseudocode of EADE is presented in Fig. 3.

Experimental study

Benchmark functions

The performance of the proposed EADE algorithm has been tasted on 20 scalable optimization functions for the CEC 2010 special session and competition on large-Scale Global Optimization. A detailed description of these test functions can be found in [1]. These 20 test functions can be divided into four classes:

1. Separable functions F_1 – F_3 ;
2. Partially separable functions, in which a small number of variables are dependent, while all the remaining ones are independent ($m = 50$) F_4 – F_8 ;
3. Partially separable functions that consist of multiple independent subcomponents, each of which is m -non-separable ($m = 50$) F_9 – F_{18} ;
4. Fully non-separable functions F_{19} – F_{20} ;

where the sphere function, the rotated elliptic function, Schwefels Problem 1.2, Rosenbrock function, the rotated Rastrigins function, and the rotated Ackleys function are used as the basic functions. The control parameter used to define the degree of separability of a given function in the given test suite is set as $m = 50$. The dimensions (D) of functions are 1000. In addition, the performance of the proposed EADE algorithm has also been tasted on 7 scalable optimization functions for the CEC 2008 special session and competition on Large-Scale Global Optimization. A detailed description of these test functions can be found in [2]. These 7 test functions are Shifted Sphere Function (F_1), Shifted Schwefel's Problem 2.21 (F_2), Shifted Rosenbrock's Function (F_3),

If ((CR_Flag_List [i] = 0) and ($G \leq LP$)), If the target vector is better than the trial vector during the learning period, then :

$$Cr = \begin{cases} \text{Randomly select one value from } A, A = [0.05] & , 0 \leq G < (1/6) * (LP) \\ \text{Randomly select one value from } A, A = [0.05, 0.1, 0.2] & , (1/6) * (LP) \leq G < (1/4) * (LP) \\ \text{Randomly select one value from } A, A = [0.05, 0.1, 0.2, 0.3, 0.4] & , (1/4) * (LP) \leq G < (1/3) * (LP) \\ \text{Randomly select one value from } A, A = [0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6] & , (1/3) * (LP) \leq G < (5/12) * (LP) \\ \text{Randomly select one value from } A, A = [0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8] & , (5/12) * (LP) \leq G < (1/2) * (LP) \\ \text{Randomly select one value from } A, A = [0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95] & , (1/2) * (LP) \leq G < (1) * (LP) \end{cases}$$

Else If ((CR_Flag_List [i] = 0) and ($G > LP$)), If the target vector is better than the trial vector after the learning period, then:

If the failure_counter_list [i] = Max_failure_counter ,

Randomly select one value from list A , $A = [0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95]$.

else

$failure_counter_list[i] = failure_counter_list[i] + 1;$

End If

Else ((CR_Flag_List [i] = 1)) , If the trial vector is better than the target vector through generations, then:

Select the CR value from A with maximum relative change improvement ratio (RCIR) from CR_Ratio_List [k].

End IF

Fig. 2 Pseudocode of crossover rate CR

Shifted Rastrigin's Function (F_4), Shifted Griewank's Function (F_5), Shifted Ackley's Function (F_6), and FastFractal "DoubleDip" Function (F_7). The dimensions (D) of functions are 100, 500, and 1000. These functions can be divided into two classes:

1. Separable functions: F_1 , F_4 , F_5 and F_6 ;
2. non-separable functions F_2 , F_3 and F_7 .

Note that F_5 is grouped as a non-separable function, because the product component becomes less significant with the increase of dimension [39].

Parameter settings and involved algorithms

To evaluate the performance of algorithm, experiments were conducted on these two test suites. We adopt the solution

error measure ($f(x) - f(x^*)$), where $f(x)$ is the best solution obtained by algorithms in one run and $f(x^*)$ is well-known global optimum of each benchmark function, which is recorded after $1.2e+05$, $6.0e+05$, and $3.0e+06$ function evaluations (FEs) for CEC'2010 and $5.0e+03 * D$ function evaluations (FEs) for CEC'2008, respectively. All experiments for each function run 25 times independently and statistical results are provided including the best, median, mean, worst results, and the standard deviation. The population size in EADE was set to 50 for CEC'2010 and 50 for $D = 100$ and 100 for $D = 500$ and 1000 for CEC'2008, respectively. The p parameter is set to 0.1, i.e., the top 10% high-quality and bottom 10% low-quality solutions in the mutation are considered. The learning period (LP) and the maximum failure counter (MFC) are set to 10% of total generations and 20 generations, respectively. It is a comparatively good parameter combination that has


```

01. Begin
02. G=0
03. Create a random initial population  $\vec{x}_i^G \forall i, i = 1, \dots, NP$ ,
04. set the Learning Period (LP) = 10% GEN, set the Max_failure_counter = 20,  $p=0.1$ 
05. For each  $\vec{x}_i^G$ , set the failure_counter_list[ $i$ ] = 0, set the CR_Flag_List [ $i$ ] = 0,
06. For each CR values in the list, set the CR_Ratio_List [ $k$ ] = 0  $\forall k, k = 1, \dots, 11$ .
07. Evaluate  $f(\vec{x}_i^G) \forall i, i = 1, \dots, NP$ 
08. For G=1 to GEN Do
09.   For i=1 to NP Do
10.     Compute the (crossover rate)  $Cr_i$  according to procedure (1).
11.     If (rand[0,1]<=0.5) Then (Use New Directed Mutation Scheme)
12.       Randomly choose  $X_{p\_best}^G$  as one of the 100 $p$ % best vectors (top individuals).
13.       Randomly choose  $X_{p\_worst}^G$  as one of the 100 $p$ % worst vectors (bottom individuals).
14.       Randomly choose  $X_r^G$  as one of the (NP-2(100 $p$  %)) vectors (middle individuals).
15.        $j_{rand} = \text{randint}(1, D)$ ,  $F_1 = \text{rand}(0, 1)$ 
16.       For j=1 to D Do
17.         If (randi[0,1] < CR or j =  $j_{rand}$ ) Then
18.            $V_i^{G+1} = X_r^G + F_1 \cdot (X_{p\_best}^G - X_r^G) + F_2 \cdot (X_r^G - X_{p\_worst}^G)$ 
19.         Else
20.            $u_{i,j}^{G+1} = X_{i,j}^G$ 
21.         End If
22.       End For
23.       Else (Use Basic Mutation Scheme)
24.         Select randomly  $r1 \neq r2 \neq r3 \neq i \in [1, NP]$ 
25.          $j_{rand} = \text{randint}(1, D)$ ,  $F = \text{rand}(0, 1)$ 
26.         For j=1 to D Do
27.           If (randi[0,1] < CR or j =  $j_{rand}$ ) Then
28.              $u_{i,j}^{G+1} = X_{r1,j}^G + F \cdot (X_{r2,j}^G - X_{r3,j}^G)$ 
29.           Else
30.              $u_{i,j}^{G+1} = X_{i,j}^G$ 
31.           End If
32.         End For
33.       End If
34.       If ( $f(u_i^{G+1}) \leq f(\vec{x}_i^G)$ ) Then
35.          $\vec{x}_i^{G+1} = \vec{u}_i^{G+1}$ , ( $f(\vec{x}_i^{G+1}) = f(\vec{u}_i^{G+1})$ )
36.       If ( $f(\vec{u}_i^{G+1}) \leq f(\vec{x}_{best}^G)$ ) Then
37.          $\vec{x}_{best}^{G+1} = \vec{u}_i^{G+1}$ , ( $f(\vec{x}_{best}^{G+1}) = f(\vec{u}_i^{G+1})$ )
38.       End

```

Fig. 3 Description of EADE algorithm

been experimentally investigated and tuned by us. For separable functions F_1 – F_3 in CEC'2010 and F_1 , F_4 , F_5 and F_6 in CEC'2008, CR is chosen to be 0.05 as they are separable

functions. Regarding CEC'2010, EADE was compared to DE-based algorithms that were all tested on this test suite in this competition. These algorithms are:

```

39.          CR_Flag_List [ i ] =1
           The relative change improvement ratio (RCIR) is updated
40.          CR_Ratio_List[k]= CR_Ratio_List[k] + (1 -  $\frac{\min(|f(\vec{u}_i^{G+1})|, |f(\vec{x}_i^G)|)}{\max(|f(\vec{u}_i^{G+1})|, |f(\vec{x}_i^G)|)}$ ))
41.          Else
42.           $\vec{x}_i^{G+1} = \vec{x}_i^G$ 
43.          CR_Flag_List [ i ] =0
44.          End If
45.        End For
46.      G=G+1
47.    End For
48.  End

```

Fig. 3 continued

- Cooperative Co-evolution with Delta Grouping for Large-Scale Non-separable Function Optimization (DECC-DML) [28].
- Large-scale optimization by Differential Evolution with Landscape modality detection and a diversity archive (LMDEa) [34].
- Large-Scale Global Optimization using Self-adaptive Differential Evolution Algorithm (jDElsgo) [35].
- DE Enhanced by Neighborhood Search for Large-Scale Global Optimization (SDENS) [36].

Besides, Memetic algorithm based on local search chains for large-scale continuous global optimization (MA-SW-Chains) algorithm, which is non-DE-based algorithm, as it won the CEC'2010 LSGO competition [37].

On the other hand, regarding CEC'2008, EADE was compared to different evolutionary algorithms that were all tested on this test suite in this competition or recently. These algorithms are:

- A competitive swarm optimizer for large-scale optimization (CEO) [39].
- A social learning particle swarm optimization algorithm for scalable optimization (SL-PSO) [40].
- Cooperatively co-evolving particle swarms for large-scale optimization (CCPSO2) [50].
- A simple modification in CMA-ES achieving linear time and space complexity (sep-CMA-ES) [51].
- Solving large-scale global optimization using improved particle swarm optimizer (EPUS-PSO) [52].
- Multilevel cooperative co-evolution for large-scale optimization (MLCC) [31].
- Dynamic multi-swarm particle swarm optimizer with local search for large-scale global Optimization (DMS-L-PSO) [53].

Among these seven algorithms, CEO [39] is the most recently proposed state-of-the-art for large-scale optimiza-

tion, which belongs to the cooperative co-evolution (CC) framework [26] for large-scale optimization. In the proposed CSO, neither the personal best position of each particle nor the global best position (or neighborhood best positions) is involved in updating the particles. Instead, a pairwise competition mechanism is introduced, where the particle that loses the competition will update its position by learning from the winner. To understand the search behavior of the proposed CSO, a theoretical proof of convergence is also provided. Similarly, a social learning PSO (SL-PSO) [40] has also been proposed belonging to the CC framework. Unlike classical PSO variants, each particle in the proposed SL-PSO learns from any better particles (termed demonstrators) in the current swarm. In addition, to ease the burden of parameter settings, the proposed SL-PSO adopts a dimension-dependent parameter control method. CCPSO2 also belongs to the cooperative co-evolution (CC) framework [50] for large-scale optimization, where a random grouping strategy is adopted based on the idea of divide-and-conquer [26]. The sep-CMA-ES is an extension of the original CMA-ES algorithm [51], which has been shown more efficient and fairly scalable to some high-dimensional test functions up to 1000-D. EPUS-PSO is another PSO variant which adjusts the swarm size according to the search results [52], and DMS-L-PSO is the DMS-PSO enhanced with a local search operator [53].

To compare the solution quality from a statistical angle of different algorithms and to check the behavior of the stochastic algorithms [54], the results are compared using multi-problem Wilcoxon signed-rank test at a 0.05 significance level. Wilcoxon signed-rank test is a non-parametric statistical test that allows us to judge the difference between paired scores when it cannot make the assumption required by the paired-sample t test, such as that the population should be normally distributed, where R^+ denotes the sum of ranks for the test problems in which the first algorithm performs better than the second algorithm (in the first column), and

R^- represents the sum of ranks for the test problems in which the first algorithm performs worse than the second algorithm (in the first column). Larger ranks indicate larger performance discrepancy. The numbers in Better, Equal, and Worse columns denote the number of problems in which the first algorithm is better than, equal, or worse than the second algorithm. As a null hypothesis, it is assumed that there is no significance difference between the mean results of the two samples. Whereas the alternative hypothesis is that there is significance in the mean results of the two samples, the number of test problems $N = 20$ for $1.25e+05$, $6.00e+05$, and $3.00e+006$ Function evaluations for CEC'2010, while the number of test problems $N = 7$ for $5.00e+005$, $2.50E+06$, and $5.00e+006$ Function evaluations with $D = 100$, $D = 500$, $D = 1000$ for CEC'2008 and 5% significance level. Use the smaller of the values of the sums as the test value and compare it with the critical value or use the p value and compare it with the significance level. Reject the null hypothesis if the test value is less than or equal to the critical value or if the p value is less than or equal to the significance level (5%). Based on the result of the test, one of three signs (+, −, and \approx) is assigned for the comparison of any two algorithms (shown in the last column), where (+) sign means the first algorithm is significantly better than the second, (−) sign means that the first algorithm is significantly worse than the second, and (\approx) sign means that there is no significant difference between the two algorithms. In addition, to obtain the final rankings of different algorithms for all functions, the Friedman test is used at a 0.05 significance level. All the p values in this paper were computed using SPSS (the version is 20.00).

To perform comprehensive evaluation and to assess the effectiveness of the proposed self-adaptive crossover rate scheme and new mutation scheme, another version of EADE, named EADE*, has been tested and compared against EADE and other DE-based algorithms. EADE* is the same as EADE except that the new mutation scheme is only used.

Experimental results and discussions

In this section, we compare directly the mean results obtained by EADE and EADE* with the ones obtained by LMDEa [34], SDENS [36], jDElsgo [35], DECC-DML [28], and MA-SW-chains [37] for CEC'2010. Tables 1, 2, 3 contain the results obtained by all algorithms in $1.2e+05$, $6.0e+05$, and $3.0e+06$ function evaluations (FEs), respectively. For remarking the best algorithm, best mean for each function is highlighted in boldface. From these tables, we have highlighted the following direct comparisons and conclusions:

- For many test functions, the worst results obtained by the proposed algorithms are better than the best results obtained by other algorithms with all FEs.
- For many test functions, there are continuous improvement in the results obtained by our proposed algorithms, especially EADE and EADE*, with all FEs, while the results with FEs = $6.0E+05$ are very close to the results with FEs = $3.0E+06$ obtained by some of the compared algorithms which indicate that our proposed approaches are scalable enough and can balance greatly the exploration and exploitation abilities for solving high-dimensional problems until the maximum FEs are reached.
- For many functions, the remarkable performance of EADE and EADE* with FEs = $1.20E+05$, and FEs = $6.0E+05$ compared to the performance of other algorithms shows its fast convergence behavior. Thus, our proposed algorithms can perform well and achieve good results within limited number of function evaluations which is very important issue when dealing with real-world problems.
- EADE and EADE* got very close to the optimum of single-group m-non-separable multi-modal functions F_6 in all statistical results with $1.20E+05$ FEs.
- EADE and LMDEa, among all other algorithms, got very close to the optimum in all runs of single-group m-non-separable multi-modal functions F_8 with $3.0E+06$ FEs.
- The performance of EADE and EADE* performs well in all types of problems which indicate that it is less affected than the most of other algorithms by the characteristics of the problems.

Furthermore, compared to the complicated structures and number of methods and number of control parameters used in other algorithms, we can see that our proposed EADE and EADE* are very simple and easy to be implemented and programmed in many programming languages. They only use very simple self-adaptive crossover rate with two parameters and a novel mutation rule with one parameters and basic mutation. Thus, they neither increase the complexity of the original DE algorithm nor the number of control parameters. To investigate and compare the performance of the proposed algorithms EADE and EADE* against other algorithms in statistical sense, multi-problem Wilcoxon signed-rank test at a significance level 0.05 is performed on mean errors of all problems with ($1.25E+05$ FES, $6.00E+05$ FES, and $3.00E+06$ FES), and the results are presented in Tables 4, 5, and 6, respectively. Where R^+ is the sum of ranks for the functions in which first algorithm outperforms the second algorithm in the row, and R^- is the sum of ranks for the opposite. From Table 4 and 5, it can be obviously seen that EADE and EADE* are significantly better than SDENS, jDElsgo, and DECC-DML algorithms. Moreover, there is no significant difference between EADE*, LMDEa, and EADE algorithm. However, MA-SW chains are significantly better than EADE and EADE* algorithms. Finally,

Table 1 Experimental comparisons between EADE, EADE*, and state-of-the-art algorithms, FES = 1.20E+05

	F_1	F_2	F_3	F_4	F_5	F_6	F_7
EADE							
Best	3.45E+07	7.51E+03	6.79E+00	8.85E+12	2.48E+08	1.86E+01	1.14E+09
Median	3.96E+07	7.67E+03	7.00E+00	1.30E+13	2.95E+08	1.89E+01	1.94E+09
Worst	4.54E+07	7.92E+03	7.32E+00	3.13E+13	3.11E+08	2.14E+01	3.38E+09
Mean	3.94E+07	7.70E+03	7.00E+00	1.48E+13	2.87E+08	1.93E+01	2.11E+09
Std	2.80E+06	1.08E+02	1.51E−01	6.09E+12	1.80E+07	9.98E−01	8.17E+08
EADE*							
Best	3.21E+07	7.94E+03	6.83E+00	3.94E+12	4.58E+07	2.00E+01	1.28E+08
Median	3.58E+07	8.19E+03	7.08E+00	5.27E+12	1.56E+08	2.03E+01	2.63E+08
Worst	4.07E+07	8.32E+03	7.73E+00	1.01E+13	2.39E+08	2.04E+01	1.06E+09
Mean	3.61E+07	8.16E+03	7.12E+00	6.29E+12	1.52E+08	2.03E+01	4.22E+08
Std	2.44E+06	1.25E+02	2.28E−01	2.01E+12	6.45E+07	1.06E−01	3.10E+08
LMDEa							
Best	4.40E+08	9.68E+03	1.43E+01	2.60E+13	2.35E+08	4.09E+04	6.05E+09
Median	4.92E+08	9.84E+03	1.51E+01	6.08E+13	2.98E+08	6.44E+04	1.46E+10
Worst	6.07E+08	1.02E+04	1.55E+01	9.40E+13	3.41E+08	1.53E+05	2.34E+10
Mean	5.08E+08	9.89E+03	1.51E+01	6.25E+13	2.94E+08	6.90E+04	1.52E+10
Std	4.76E+07	1.37E+02	2.50E−01	1.72E+13	2.51E+07	2.30E+04	3.94E+09
SDENS							
Best	3.93E+09	1.16E+04	1.99E+01	3.90E+13	3.14E+08	9.88E+05	3.07E+10
Median	4.74E+09	1.19E+04	2.01E+01	4.60E+13	3.32E+08	2.03E+06	3.57E+10
Worst	6.19E+09	1.20E+04	2.02E+01	7.90E+13	3.41E+08	2.39E+06	4.70E+10
Mean	5.01E+09	1.19E+04	2.01E+01	5.10E+13	3.29E+08	1.84E+06	3.75E+10
Std	9.18E+08	9.89E+01	1.17E−01	1.46E+13	1.04E+07	4.77E+05	5.46E+09
jDElsgo							
Best	2.78E+09	1.06E+04	1.81E+01	8.06E+13	2.98E+08	3.36E+06	2.89E+10
Median	3.72E+09	1.09E+09	1.88E+01	1.43E+14	3.38E+08	4.24E+06	5.40E+10
Worst	4.89E+09	1.13E+04	1.97E+01	2.30E+14	3.75E+08	4.84E+06	7.23E+10
Mean	3.70E+09	1.09E+04	1.87E+01	1.40E+14	3.39E+08	4.26E+06	5.39E+10
Std	5.11E+08	1.75E+02	4.46E−01	3.69E+13	1.82E+07	3.91E+05	1.07E+10
DECC-DML							
Best	2.28E+08	5.51E+03	8.22E+00	3.80E+13	1.43E+08	1.25E+06	2.65E+09
Median	2.85E+08	5.76E+03	9.71E+00	6.40E+13	2.85E+08	1.96E+06	5.50E+09
Worst	7.02E+08	5.96E+03	1.01E+01	1.20E+14	5.21E+08	2.00E+07	1.17E+10
Mean	4.09E+08	5.75E+03	9.51E+00	6.76E+13	3.00E+08	2.70E+06	5.97E+09
Std	1.75E+08	1.35E+02	5.55E−01	2.02E+13	9.31E+07	3.62E+06	2.49E+09
MA-SW-chains							
Best	2.15E+07	3.32E+03	1.13E+01	1.22E+12	9.35E+07	2.02E+01	4.54E+06
Median	2.76E+07	3.75E+03	1.15E+01	2.04E+12	2.64E+08	2.08E+01	4.91E+06
Worst	3.51E+07	1.00E+04	1.22E+01	3.35E+12	3.42E+08	1.16E+06	5.71E+06
Mean	2.83E+07	5.09E+03	1.16E+01	2.12E+12	2.52E+08	8.14E+04	4.90E+06
Std	3.06E+06	2.38E+03	2.68E−01	6.21E+11	6.49E+07	2.84E+05	2.59E+05

Table 1 continued

	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
EADE							
Best	4.32E+07	2.05E+09	1.25E+04	7.47E+01	2.66E+06	5.06E+06	4.09E+09
Median	4.54E+07	2.67E+09	1.29E+04	9.71E+01	3.00E+06	9.22E+06	4.49E+09
Worst	1.04E+08	3.38E+09	1.33E+04	1.34E+02	4.13E+06	1.47E+07	5.41E+09
Mean	5.31E+07	2.63E+09	1.29E+04	1.01E+02	3.10E+06	9.03E+06	4.59E+09
Std	2.04E+07	3.30E+08	2.46E+02	1.94E+01	4.04E+05	2.49E+06	3.92E+08
EADE*							
Best	1.28E+07	1.58E+09	6.36E+03	1.71E+02	1.41E+06	1.40E+07	3.05E+09
Median	9.20E+07	1.98E+09	6.58E+03	1.87E+02	1.54E+06	2.04E+07	3.71E+09
Worst	1.54E+08	2.55E+09	7.07E+03	2.06E+02	1.78E+06	1.41E+08	4.14E+09
Mean	8.05E+07	2.02E+09	6.63E+03	1.88E+02	1.56E+06	3.71E+07	3.67E+09
Std	4.12E+07	2.87E+08	1.85E+02	9.52E+00	1.08E+05	3.30E+07	2.99E+08
LMDEa							
Best	6.79E+07	4.06E+09	1.22E+04	2.16E+02	2.67E+06	2.11E+07	7.54E+09
Median	1.71E+08	5.25E+09	1.27E+04	2.26E+02	2.84E+06	2.87E+07	9.59E+09
Worst	4.46E+08	6.33E+09	1.34E+04	2.29E+02	3.05E+06	4.58E+07	1.16E+10
Mean	1.88E+08	5.21E+09	1.27E+04	2.25E+02	2.85E+06	2.90E+07	9.64E+09
Std	9.84E+07	5.87E+08	2.67E+02	2.90E+00	1.07E+05	5.51E+06	8.85E+08
SDENS							
Best	6.05E+08	1.13E+10	1.37E+04	2.27E+02	2.71E+06	1.70E+10	1.42E+10
Median	6.23E+08	1.52E+10	1.38E+04	2.27E+02	2.83E+06	1.91E+10	1.73E+10
Worst	1.20E+09	1.89E+10	1.42E+04	2.28E+02	3.29E+06	2.01E+10	2.31E+10
Mean	7.71E+08	1.56E+10	1.39E+04	2.27E+02	2.95E+06	1.88E+10	1.84E+10
Std	2.27E+08	2.77E+09	2.51E+02	3.49E−01	2.37E+05	1.07E+09	3.56E+09
jDElsgo							
Best	1.04E+09	1.43E+10	1.31E+04	2.02E+02	2.76E+06	2.23E+09	1.95E+10
Median	2.29E+09	1.59E+10	1.43E+04	2.20E+02	3.18E+06	3.70E+09	2.32E+10
Worst	5.42E+09	2.07E+10	1.51E+04	2.26E+02	3.65E+06	5.44E+09	2.76E+10
Mean	2.39E+09	1.64E+10	1.43E+04	2.19E+02	3.15E+06	3.76E+09	2.32E+10
Std	9.13E+08	1.73E+09	4.38E+02	5.92E+00	2.19E+05	1.04E+09	2.03E+09
DECC-DML							
Best	2.23E+09	4.09E+09	1.32E+04	1.02E+02	4.07E+06	1.09E+08	1.26E+10
Median	4.92E+09	4.91E+09	1.39E+04	1.22E+02	4.68E+06	1.82E+08	1.37E+10
Worst	1.35E+10	5.54E+09	1.45E+04	1.70E+02	5.35E+06	3.72E+08	1.51E+10
Mean	5.57E+09	4.89E+09	1.38E+04	1.24E+02	4.70E+06	2.11E+08	1.37E+10
Std	2.56E+09	3.77E+08	3.24E+02	1.38E+01	2.99E+05	9.68E+07	6.86E+08
MA-SW-chains							
Best	3.30E+07	4.48E+08	3.62E+03	5.01E+01	2.20E+05	7.64E+05	8.16E+08
Median	4.17E+07	5.60E+08	4.15E+03	6.41E+01	2.40E+05	9.04E+05	8.81E+08
Worst	8.55E+08	6.45E+08	1.00E+04	7.13E+01	2.62E+05	1.11E+06	1.04E+09
Mean	1.21E+08	5.54E+08	5.12E+03	6.31E+01	2.40E+05	9.13E+05	8.95E+08
Std	2.11E+08	5.20E+07	2.20E+03	5.53E+00	1.26E+04	8.09E+04	6.60E+07

Table 1 continued

	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
EADE						
Best	1.36E+04	2.68E+02	3.12E+06	4.89E+09	2.22E+07	7.66E+09
Median	1.40E+04	3.27E+02	3.68E+06	7.15E+09	2.51E+07	1.01E+10
Worst	1.46E+04	3.74E+02	4.84E+06	1.00E+10	3.04E+07	1.26E+10
Mean	1.40E+04	3.21E+02	3.81E+06	7.00E+09	2.54E+07	9.97E+09
Std	2.93E+02	3.73E+01	4.48E+05	1.47E+09	2.23E+06	1.30E+09
EADE*						
Best	6.86E+03	3.91E+02	1.84E+06	1.48E+10	6.28E+06	1.68E+10
Median	7.59E+03	3.99E+02	2.07E+06	1.81E+10	7.26E+06	2.83E+10
Worst	1.21E+04	4.05E+02	2.26E+06	3.07E+10	8.11E+06	4.08E+10
Mean	7.92E+03	3.98E+02	2.04E+06	1.96E+10	7.18E+06	2.75E+10
Std	1.21E+03	3.65E+00	1.10E+05	4.53E+09	5.79E+05	5.74E+09
LMDEa						
Best	1.32E+04	4.14E+02	4.21E+06	2.38E+09	8.75E+06	2.40E+09
Median	1.37E+04	4.17E+02	4.70E+06	2.95E+09	9.96E+06	3.52E+09
Worst	1.42E+04	4.18E+02	5.06E+06	3.97E+09	1.10E+07	4.80E+09
Mean	1.37E+04	4.16E+02	4.65E+06	3.07E+09	1.00E+07	3.57E+09
Std	2.66E+02	7.76E−01	2.37E+05	4.01E+08	5.88E+05	5.43E+08
SDENS						
Best	1.36E+04	4.15E+02	3.84E+06	2.00E+11	1.19E+07	2.39E+11
Median	1.45E+04	4.15E+02	4.25E+06	2.09E+11	1.57E+07	2.62E+11
Worst	1.45E+04	4.15E+02	4.98E+06	2.35E+11	2.31E+07	2.82E+11
Mean	1.43E+04	4.15E+02	4.31E+06	2.11E+11	1.67E+07	2.61E+11
Std	3.72E+02	1.08E−01	4.04E+05	1.27E+10	3.71E+06	1.49E+10
jDElsgo						
Best	1.44E+04	4.09E+02	4.28E+06	5.25E+10	2.07E+07	5.80E+10
Median	1.55E+04	4.17E+02	4.79E+06	6.37E+10	2.88E+07	8.14E+10
Worst	1.59E+04	4.24E+02	5.71E+06	8.76E+10	3.56E+07	1.11E+11
Mean	1.54E+04	4.17E+02	4.85E+06	6.60E+10	2.85E+07	7.99E+10
Std	3.33E+02	3.28E+00	3.53E+05	9.47E+09	3.38E+06	1.25E+10
DECC-DML						
Best	1.58E+04	3.22E+02	7.48E+06	1.56E+09	1.77E+07	2.04E+09
Median	1.65E+04	3.73E+02	8.77E+06	3.30E+09	2.23E+07	3.93E+09
Worst	1.73E+04	4.28E+02	1.01E+07	4.06E+09	2.72E+07	5.09E+09
Mean	1.65E+04	3.75E+02	8.81E+06	3.08E+09	2.20E+07	3.84E+09
Std	3.61E+02	3.60E+01	6.86E+05	7.84E+08	2.36E+06	7.72E+08
MA-SW-chains						
Best	3.94E+03	2.01E+02	5.80E+05	2.22E+04	3.23E+06	2.01E+03
Median	4.29E+03	2.12E+02	6.78E+05	5.18E+04	3.63E+06	2.22E+03
Worst	9.63E+03	2.31E+02	7.50E+05	8.33E+04	4.05E+06	4.69E+03
Mean	4.83E+03	2.13E+02	6.78E+05	5.14E+04	3.63E+06	2.43E+03
Std	1.51E+03	9.19E+00	3.52E+04	1.64E+04	1.94E+05	5.43E+02

Table 2 Experimental comparisons between EADE, EADE*, and state-of-the-art algorithms, FES = 6.00E+05

	F_1	F_2	F_3	F_4	F_5	F_6	F_7
EADE							
Best	1.11E−02	1.94E+03	1.06E−04	4.63E+11	4.08E+07	1.59E+01	2.39E+05
Median	1.38E−02	2.06E+03	2.01E−04	9.89E+11	6.37E+07	1.98E+01	4.24E+05
Worst	2.19E−02	2.15E+03	4.90E−04	1.56E+12	1.06E+08	2.01E+01	6.62E+05
Mean	1.45E−02	2.06E+03	2.18E−04	9.17E+11	6.58E+07	1.96E+01	4.18E+05
Std	3.17E−03	5.46E+01	1.04E−04	3.40E+11	1.70E+07	1.02E+00	9.99E+04
EADE*							
Best	3.46E−02	2.21E+03	1.28E+00	3.54E+11	4.88E+07	1.65E+01	2.83E+05
Median	4.57E−02	2.69E+03	1.63E+00	1.04E+12	5.77E+07	2.04E+01	3.72E+05
Worst	1.26E−01	2.90E+03	1.87E+00	1.63E+12	8.56E+07	1.03E+06	6.42E+05
Mean	5.45E−02	2.68E+03	1.56E+00	9.84E+11	6.25E+07	4.11E+04	3.99E+05
Std	2.32E−02	1.88E+02	1.92E−01	3.84E+11	1.04E+07	2.05E+05	8.65E+04
LMDEa							
Best	3.00E+02	3.23E+03	6.17E−01	1.65E+12	3.68E+07	4.73E+00	2.37E+07
Median	4.32E+02	3.35E+03	9.23E−01	4.23E+12	6.17E+07	5.50E+00	5.03E+07
Worst	6.90E+02	3.53E+03	1.10E+00	9.52E+12	1.50E+08	6.46E+00	1.34E+08
Mean	4.59E+02	3.37E+03	9.16E−01	4.49E+12	7.21E+07	5.60E+00	5.85E+07
Std	1.09E+02	6.88E+01	1.10E−01	1.81E+12	2.74E+07	3.91E−01	2.99E+07
SDENS							
Best	3.82E+06	7.00E+03	5.13E+00	8.47E+12	1.51E+08	1.38E+01	5.73E+09
Median	4.59E+06	7.12E+03	6.27E+00	1.53E+13	1.83E+08	1.53E+01	7.73E+09
Worst	1.95E+07	7.17E+03	6.76E+00	2.85E+13	2.12E+08	1.74E+01	1.36E+10
Mean	7.87E+06	7.09E+03	6.12E+00	1.72E+13	1.81E+08	1.53E+01	9.28E+09
Std	5.94E+06	6.76E+01	6.30E−01	6.68E+12	2.29E+07	1.18E+00	3.44E+09
jDElsgo							
Best	7.04E+04	3.67E+03	9.70E−01	7.89E+12	1.42E+08	2.20E+01	3.36E+09
Median	8.71E+04	3.93E+03	1.18E+00	1.29E+13	1.87E+08	3.24E+01	6.41E+09
Worst	1.23E+05	4.20E+03	1.58E+00	2.67E+13	2.30E+08	3.14E+02	1.10E+10
Mean	8.99E+04	3.95E+03	1.22E+00	1.39E+13	1.88E+08	5.97E+01	6.43E+09
Std	1.39E+04	1.32E+02	1.38E−01	4.60E+12	2.31E+07	5.81E+01	2.123+09
DECC-DML							
Best	6.95E+01	2.51E+03	1.06E−02	7.92E+12	1.42E+08	4.59E+01	3.14E+08
Median	4.63E+02	2.64E+03	1.83E−02	1.51E+13	2.85E+08	1.09E+02	5.42E+08
Worst	1.22E+03	2.78E+03	2.20E−02	3.29E+13	5.20E+08	1.98E+07	9.17E+08
Mean	6.02E+02	2.64E+03	1.81E−02	1.61E+13	2.99E+08	7.94E+05	5.84E+08
Std	4.11E+02	5.88E+01	3.08E−03	6.19E+12	9.31E+07	3.97E+06	1.68E+08
MA-SW-chains							
Best	8.52E+02	2.36E+03	3.44E+00	4.29E+11	3.68E+07	3.61E+00	6.33E+04
Median	1.55E+03	2.68E+03	3.83E+00	5.75E+11	2.59E+08	1.78E+01	7.78E+05
Worst	7.28E+03	2.97E+03	4.60E+00	7.42E+11	3.24E+08	1.16E+06	4.61E+06
Mean	2.24E+03	2.67E+03	3.84E+00	5.79E+11	2.17E+08	8.14E+04	8.35E+05
Std	1.71E+03	1.63E+02	2.13E−01	6.46E+10	8.56E+07	2.84E+05	9.08E+05

Table 2 continued

	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
EADE							
Best	2.58E+07	1.79E+08	3.11E+03	1.58E+02	3.40E+05	3.24E+03	7.66E+08
Median	3.61E+07	2.20E+08	3.65E+03	1.67E+02	3.82E+05	5.06E+03	7.76E+08
Worst	1.28E+08	2.62E+08	4.02E+03	1.77E+02	4.35E+05	1.85E+04	8.80E+08
Mean	5.95E+07	2.21E+08	3.66E+03	1.68E+02	3.89E+05	6.21E+03	8.13E+08
Std	3.40E+07	2.28E+07	2.10E+02	6.33E+00	2.89E+04	3.96E+03	5.65E+07
EADE*							
Best	9.90E+06	1.81E+08	3.73E+03	1.56E+02	3.25E+05	5.17E+03	6.75E+08
Median	3.44E+07	2.27E+08	4.09E+03	1.72E+02	3.99E+05	1.89E+04	7.34E+08
Worst	9.69E+07	4.29E+08	4.46E+03	1.84E+02	4.78E+05	4.79E+04	8.01E+08
Mean	4.64E+07	2.40E+08	4.05E+03	1.73E+02	4.00E+05	2.01E+04	7.33E+08
Std	2.99E+07	5.32E+07	1.49E+02	7.50E+00	3.52E+04	1.20E+04	3.85E+07
LMDEa							
Best	3.19E+07	2.26E+08	7.82E+03	4.18E+01	3.91E+05	1.48E+03	6.49E+08
Median	3.30E+07	2.76E+08	9.68E+03	6.70E+01	4.51E+05	2.07E+03	7.55E+08
Worst	3.49E+07	3.25E+08	1.02E+04	1.14E+02	5.17E+05	5.72E+03	1.01E+09
Mean	3.32E+07	2.71E+08	9.65E+03	6.91E+01	4.50E+05	2.34E+03	7.70E+08
Std	8.55E+05	2.40E+07	4.76E+02	1.71E+01	3.21E+04	9.90E+02	7.45E+07
SDENS							
Best	4.64E+07	1.78E+09	1.02E+04	2.25E+02	1.25E+06	4.37E+05	3.91E+09
Median	6.40E+07	2.13E+09	1.09E+04	2.26E+02	1.30E+06	6.67E+05	5.02E+09
Worst	1.09E+08	2.88E+09	1.15E+04	2.26E+02	1.42E+06	7.64E+05	6.93E+09
Mean	7.41E+07	2.23E+09	1.10E+04	2.26E+02	1.32E+06	6.43E+05	5.14E+09
Std	2.73E+07	3.70E+08	4.59E+02	3.83E-01	5.98E+04	1.10E+05	9.89E+08
jDElsgo							
Best	3.57E+07	1.45E+09	7.66E+03	7.73E+01	8.61E+05	2.90E+04	3.61E+09
Median	4.65E+07	1.64E+09	8.69E+03	1.14E+02	9.35E+05	4.95E+04	4.11E+09
Worst	1.39E+08	1.82E+09	9.49E+03	1.49E+02	9.88E+05	9.02E+04	4.72E+09
Mean	6.82E+07	1.66E+09	8.67E+03	1.17E+02	9.39E+05	5.32E+04	4.10E+09
Std	3.53E+07	8.29E+07	3.99E+02	1.87E+01	2.96E+04	1.70E+04	2.89E+08
DECC-DML							
Best	4.19E+07	2.82E+08	1.25E+04	4.00E-01	3.64E+06	8.29E+02	9.82E+08
Median	1.15E+08	3.85E+08	1.30E+04	7.09E-01	4.22E+06	1.71E+03	1.18E+09
Worst	2.38E+08	4.21E+08	1.36E+04	1.72E+00	4.65E+06	1.47E+04	1.29E+09
Mean	1.24E+08	3.73E+08	1.30E+04	7.66E-01	4.19E+06	3.15E+03	1.17E+09
Std	5.40E+07	3.13E+07	2.93E+02	2.81E-01	2.18E+05	3.09E+03	8.20E+07
MA-SW-chains							
Best	3.42E+06	6.93E+07	2.79E+03	2.77E+01	1.39E+03	1.08E+03	1.51E+08
Median	1.90E+07	8.08E+07	3.25E+03	3.79E+01	1.64E+03	3.06E+03	1.70E+08
Worst	6.11E+08	1.00E+08	3.54E+03	5.15E+01	1.91E+03	1.07E+04	1.95E+08
Mean	6.13E+07	8.18E+07	3.22E+03	3.83E+01	1.63E+03	4.34E+03	1.69E+08
Std	1.27E+08	8.36E+06	1.85E+02	7.23E+00	1.53E+02	3.21E+03	1.17E+07

Table 2 continued

	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
EADE						
Best	3.97E+03	2.74E+02	9.31E+05	1.67E+04	7.60E+06	7.13E+03
Median	4.34E+03	3.24E+02	1.04E+06	4.47E+04	1.15E+07	9.68E+03
Worst	9.42E+03	3.58E+02	1.14E+06	6.22E+04	1.50E+07	1.24E+04
Mean	4.96E+03	3.21E+02	1.04E+06	4.39E+04	1.12E+07	9.73E+03
Std	1.75E+03	2.21E+01	5.41E+04	1.29E+04	2.47E+06	1.64E+03
EADE*						
Best	4.40E+03	3.44E+02	7.51E+05	1.70E+05	2.73E+06	2.12E+05
Median	4.63E+03	3.57E+02	8.07E+05	5.63E+05	2.93E+06	4.61E+05
Worst	4.81E+03	3.95E+02	8.98E+05	1.47E+06	3.19E+06	3.73E+07
Mean	4.63E+03	3.63E+02	8.10E+05	5.81E+05	2.95E+06	3.13E+06
Std	1.39E+02	1.55E+01	3.95E+04	3.75E+05	1.32E+05	9.47E+06
LMDEa						
Best	1.14E+04	4.09E+02	1.27E+06	6.11E+03	2.91E+06	3.33E+03
Median	1.21E+04	4.13E+02	1.38E+06	1.80E+04	3.15E+06	3.99E+03
Worst	1.24E+04	4.14E+02	1.48E+06	3.24E+04	3.40E+06	4.56E+03
Mean	1.20E+04	4.13E+02	1.38E+06	1.90E+04	3.17E+06	4.02E+03
Std	3.04E+02	1.21E+00	5.84E+04	6.80E+03	1.44E+05	2.94E+02
SDENS						
Best	7.32E+03	4.13E+02	1.96E+06	1.65E+08	4.92E+06	1.36E+08
Median	1.18E+04	4.13E+02	2.02E+06	1.86E+08	5.39E+06	2.78E+08
Worst	1.26E+04	4.14E+02	2.29E+06	3.00E+08	6.18E+06	3.52E+08
Mean	1.03E+04	4.13E+02	2.07E+06	2.02E+08	5.41E+06	2.69E+08
Std	2.29E+03	3.49E−01	1.17E+05	5.02E+07	4.31E+05	7.57E+07
jDElsgo						
Best	1.10E+04	2.67E+02	1.81E+06	6.95E+05	4.16E+06	5.98E+05
Median	1.22E+04	2.97E+02	1.97E+06	1.04E+06	5.48E+06	1.05E+06
Worst	1.26E+04	3.53E+02	2.10E+06	1.39E+06	1.22E+07	1.66E+06
Mean	1.20E+04	2.99E+02	1.95E+06	1.03E+06	6.09E+06	1.01E+06
Std	5.30E+02	1.91E+01	6.54E+04	2.08E+05	1.65E+06	2.48E+05
DECC-DML						
Best	1.53E+04	3.46E+00	6.50E+06	5.64E+03	1.54E+07	1.43E+03
Median	1.59E+04	8.65E+00	7.29E+06	1.48E+04	1.84E+07	1.67E+03
Worst	1.67E+04	4.28E+02	7.99E+06	3.96E+04	2.46E+07	2.02E+03
Mean	1.59E+04	4.47E+01	7.27E+06	1.74E+04	1.87E+07	1.69E+03
Std	3.63E+02	1.16E+02	3.77E+05	8.26E+03	1.99E+06	1.58E+02
MA-SW-chains						
Best	2.95E+03	8.51E+01	3.59E+04	1.80E+03	1.29E+06	1.02E+03
Median	3.19E+03	9.71E+01	4.29E+04	3.89E+03	1.42E+06	1.18E+03
Worst	3.45E+03	1.26E+02	5.01E+04	1.61E+04	1.58E+06	1.65E+03
Mean	3.19E+03	1.02E+02	4.31E+04	5.53E+03	1.41E+06	1.21E+03
Std	1.46E+02	1.42E+01	3.42E+03	3.94E+03	7.44E+04	1.42E+02

Table 3 Experimental comparisons between EADE, EADE*, and state-of-the-art algorithms, FES = 3.0E+06

	F_1	F_2	F_3	F_4	F_5	F_6	F_7
EADE							
Best	1.53E−23	3.80E+02	5.68E−14	4.94E+10	4.48E+07	1.89E+01	3.31E−03
Median	1.23E−22	4.17E+02	6.39E−14	8.91E+10	8.66E+07	1.90E+01	3.60E−02
Worst	2.75E−21	4.53E+02	6.39E−14	2.69E+11	1.69E+08	1.91E+01	1.92E+00
Mean	4.70E−22	4.16E+02	6.25E−14	1.08E+11	8.79E+07	1.90E+01	2.11−001
Std	8.65E−22	2.07E+01	2.62E−15	6.56E+10	3.11E+07	7.06E−02	4.87E−01
EADE*							
Best	1.09E−22	7.43E+02	1.35E+00	6.80E+10	5.57E+07	1.95E+01	1.54E−02
Median	3.27E−22	8.39E+02	1.68E+00	1.33E+11	7.96E+07	1.96E+01	3.88E−01
Worst	1.19E−21	8.97E+02	1.74E+00	2.25E+11	1.19E+08	1.96E+01	1.85E+00
Mean	4.18E−22	8.27E+02	1.60E+00	1.37E+11	8.48E+07	1.96E+01	4.18E−01
Std	3.12E−22	4.89E+01	1.47E−01	3.79E+10	1.80E+07	2.22E−02	4.81E−01
LMDEa							
Best	2.42E−24	5.31E+02	7.86E−14	1.14E+11	3.68E+07	4.00E−09	3.79E−02
Median	7.36E−24	6.87E+02	8.79E−01	2.06E+11	6.07E+07	4.02E−09	1.91E−01
Worst	1.55E−22	8.62E+02	1.09E+00	3.15E+11	1.13E+08	1.01E+00	6.42E−01
Mean	1.35E−23	6.97E+02	6.44E−01	2.08E+11	6.62E+07	2.63E−01	2.45E−01
Std	2.91E−23	8.22E+01	4.46E−01	5.89E+10	2.02E+07	4.22E−01	1.68E−01
SDENS							
Best	1.75E−06	2.14E+03	1.23E−05	3.26E+12	7.66E+07	1.53E−04	6.36E+07
Median	2.54E−06	2.17E+03	2.35E−05	3.72E+12	1.17E+08	1.76E−04	8.57E+07
Worst	1.16E−05	2.39E+03	5.50E−05	8.99E+12	1.52E+08	2.57E−04	2.39E+08
Mean	5.73E−06	2.21E+03	2.70E−05	5.11E+12	1.18E+08	2.02E−04	1.20E+08
Std	4.46E−06	8.95E+01	1.54E−05	2.16E+12	2.88E+07	4.29E−05	6.56E+07
jDElsgo							
Best	4.78E−20	1.09E−11	1.63E−12	3.09E+10	7.42E+07	7.14E−09	2.69E−05
Median	6.63E−20	4.69E−11	2.35E−12	8.28E+10	9.82E+07	7.22E−09	1.04E−04
Worst	2.24E−19	1.12E+00	2.24E−11	1.34E+11	1.24E+08	2.10E−07	3.19E−01
Mean	8.86E−20	1.25E−01	3.81E−12	8.06E+10	9.72E+07	1.70E−08	1.31E−02
Std	4.51E−20	3.45E−01	5.02E−12	3.08E+10	1.44E+07	4.03E−08	6.38E−02
DECC-DML							
Best	9.05E−27	1.62E+02	1.10E−13	1.38E+12	1.42E+08	3.55E−09	7.09E+07
Median	1.22E−25	2.12E+02	1.14E−13	3.32E+12	2.85E+08	7.11E−09	1.23E+08
Worst	7.12E−25	2.94E+02	1.35E−13	6.89E+12	5.20E+08	1.98E+07	4.82E+08
Mean	1.93E−25	2.17E+02	1.18E−13	3.58E+12	2.99E+08	7.93E+05	1.39E+08
Std	1.86E−25	2.98E+01	8.22E−15	1.54E+12	9.31E+07	3.97E+06	7.72E+07
MA-SW-chains							
Best	3.18E−15	7.04E+02	3.34E−13	3.04E+11	2.89E+07	8.13E−07	3.35E−03
Median	1.50E−14	7.90E+02	6.11E−13	3.54E+11	2.31E+08	1.60E+00	9.04E+01
Worst	8.15E−14	9.37E+02	1.58E−12	3.97E+11	2.90E+08	1.16E+06	2.68E+02
Mean	2.10E−14	8.10E+02	7.28E−13	3.53E+11	1.68E+08	8.14E+04	1.03E+02
Std	1.99E−14	5.88E+01	3.40E−13	3.12E+10	1.04E+08	2.84E+05	8.70E+01

Table 3 continued

	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
EADE							
Best	2.40E−10	3.08E+07	2.53E+03	1.02E+02	1.67E+04	7.18E+02	1.28E+08
Median	2.97E−05	3.59E+07	2.64E+03	1.15E+02	2.77E+04	9.89E+02	1.47E+08
Worst	1.63E−03	4.29E+07	2.78E+03	1.22E+02	4.12E+04	1.36E+03	1.61E+08
Mean	2.26E−04	3.67E+07	2.62E+03	1.14E+02	2.80E+04	1.01E+03	1.46E+08
Std	4.41E−04	3.48E+06	8.12E+01	6.43E+00	5.72E+03	1.83E+02	9.60E+06
EADE*							
Best	5.49E−08	2.82E+07	2.78E+03	1.07E+02	2.75E+04	5.73E+02	1.19E+08
Median	1.50E−03	3.22E+07	3.01E+03	1.23E+02	4.92E+04	1.04E+03	1.35E+08
Worst	3.99E+06	3.86E+07	3.24E+03	1.34E+02	7.08E+04	1.39E+03	1.54E+08
Mean	2.66E+05	3.29E+07	3.00E+03	1.23E+02	4.98E+04	1.06E+03	1.35E+08
Std	1.03E+06	2.94E+06	1.07E+02	8.88E+00	1.32E+04	2.10E+02	1.12E+07
LMDEa							
Best	7.80E−05	2.25E+07	2.45E+03	6.49E−11	1.35E+04	4.56E+02	7.65E+07
Median	3.29E−04	2.65E+07	2.76E+03	2.74E+00	1.81E+04	5.64E+02	8.67E+07
Worst	9.20E−04	2.98E+07	3.97E+03	5.40E+01	2.49E+04	9.38E+02	1.03E+08
Mean	3.61E−04	2.64E+07	2.80E+03	1.19E+01	1.83E+04	5.95E+02	8.63E+07
Std	2.33E−04	1.89E+06	2.84E+02	1.50E+01	2.62E+03	1.06E+02	6.30E+06
SDENS							
Best	3.96E+07	4.77E+08	5.78E+03	2.20E+02	3.80E+05	1.16E+03	1.61E+09
Median	4.09E+07	5.75E+08	7.03E+03	2.21E+02	3.95E+05	1.80E+03	1.86E+09
Worst	9.35E+07	6.38E+08	7.37E+03	2.22E+02	4.97E+05	4.13E+03	2.30E+09
Mean	5.12E+07	5.63E+08	6.87E+03	2.21E+02	4.13E+05	2.19E+03	1.88E+09
Std	2.12E+07	5.78E+07	5.60E+02	5.09E−01	4.28E+04	1.03E+03	2.33E+08
jDElsgo							
Best	3.40E−03	2.36E+07	2.10E+03	1.27E+00	8.32E+03	4.79E+02	1.28E+08
Median	1.25E+06	3.04E+07	2.66E+03	1.95E+01	1.18E+04	6.91E+02	1.72E+08
Worst	8.08E+06	4.22E+07	3.30E+03	5.81E+01	1.71E+04	1.02E+03	2.02E+08
Mean	3.15E+06	3.11E+07	2.64E+03	2.20E+01	1.21E+04	7.11E+02	1.69E+08
Std	3.27E+06	5.00E+06	3.19E+02	1.53E+01	2.04E+03	1.37E+02	2.08E+07
DECC-DML							
Best	7.34E+05	4.51E+07	1.21E+04	1.63E−13	3.46E+06	6.19E+02	1.54E+08
Median	1.57E+07	5.97E+07	1.24E+04	1.78E−13	3.81E+06	1.06E+03	1.89E+08
Worst	1.21E+08	7.09E+07	1.30E+04	2.03E−13	4.11E+06	2.09E+03	2.22E+08
Mean	3.46E+07	5.92E+07	1.25E+04	1.80E−13	3.80E+06	1.14E+03	1.89E+08
Std	3.56E+07	4.71E+06	2.66E+02	9.88E−15	1.50E+05	4.31E+02	1.49E+07
MA-SW-chains							
Best	1.54E+06	1.19E+07	1.81E+03	2.74E+01	2.65E−06	3.86E+02	2.79E+07
Median	3.43E+06	1.40E+07	2.07E+03	3.75E+01	3.50E−06	1.07E+03	3.09E+07
Worst	1.80E+08	1.62E+07	2.28E+03	5.11E+01	4.98E−06	2.92E+03	3.67E+07
Mean	1.41E+07	1.41E+07	2.07E+03	3.80E+01	3.62E−06	1.25E+03	3.11E+07
Std	3.68E+07	1.15E+06	1.44E+02	7.35E+00	5.92E−07	5.72E+02	1.93E+06

Table 3 continued

	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}
EADE						
Best	2.93E+03	2.86E+02	1.33E+05	1.74E+03	1.20E+06	1.93E+03
Median	3.20E+03	2.99E+02	1.55E+05	2.27E+03	1.30E+06	2.06E+03
Worst	3.41E+03	3.12E+02	1.66E+05	3.16E+03	1.38E+06	2.39E+03
Mean	3.18E+03	3.00E+02	1.52E+05	2.26E+03	1.29E+06	2.10E+03
Std	1.33E+02	5.81E+00	1.14E+04	3.63E+02	8.25E+04	1.33E+02
EADE*						
Best	3.18E+03	2.89E+02	1.23E+05	2.07E+03	1.35E+06	1.72E+03
Median	3.50E+03	3.00E+02	1.59E+05	2.21E+03	1.47E+06	2.17E+03
Worst	3.78E+03	3.10E+02	1.86E+05	2.94E+03	1.59E+06	2.53E+03
Mean	3.50E+03	3.00E+02	1.59E+05	2.28E+03	1.47E+06	2.20E+03
Std	1.54E+02	6.07E+00	2.28E+04	3.04E+02	8.18E+04	2.15E+02
LMDEa						
Best	5.15E+03	3.75E+02	1.92E+05	1.34E+03	4.07E+05	1.11E+03
Median	5.64E+03	3.85E+02	2.13E+05	1.65E+03	4.43E+05	1.38E+03
Worst	6.34E+03	4.00E+02	2.43E+05	2.24E+03	4.93E+05	1.60E+03
Mean	5.63E+03	3.87E+02	2.14E+05	1.68E+03	4.42E+05	1.38E+03
Std	2.81E+02	5.24E+00	1.47E+04	2.09E+02	1.85E+04	1.16E+02
SDENS						
Best	7.14E+03	4.03E+02	8.78E+05	1.16E+04	7.57E+05	9.81E+02
Median	7.32E+03	4.09E+02	1.14E+06	3.32E+04	8.02E+05	9.83E+02
Worst	7.44E+03	4.10E+02	1.18E+06	4.51E+04	1.19E+06	1.02E+03
Mean	7.32E+03	4.08E+02	1.08E+06	3.08E+04	8.80E+05	9.90E+02
Std	9.63E+01	2.53E+00	1.11E+05	1.22E+04	1.59E+05	1.62E+01
jDElsgo						
Best	5.20E+03	7.30E+01	7.75E+04	1.31E+03	2.39E+05	1.24E+03
Median	5.78E+03	1.46E+02	1.00E+05	1.88E+03	2.77E+05	1.55E+03
Worst	6.84E+03	2.00E+02	1.28E+05	2.57E+03	3.21E+05	1.83E+03
Mean	5.84E+03	1.44E+02	1.02E+05	1.85E+03	2.74E+05	1.53E+03
Std	4.48E+02	3.43E+01	1.26E+04	3.18E+02	2.12E+04	1.32E+02
DECC-DML						
Best	1.48E+04	2.74E−13	5.65E+06	1.64E+03	1.30E+07	9.69E+02
Median	1.53E+04	3.20E−13	6.55E+06	2.21E+03	1.59E+07	9.75E+02
Worst	1.62E+04	1.27E+00	7.63E+06	7.52E+03	2.16E+07	1.10E+03
Mean	1.54E+04	5.08E−02	6.54E+06	2.47E+03	1.59E+07	9.91E+02
Std	3.59E+02	2.54E−01	4.63E+05	1.18E+03	1.72E+06	3.51E+01
MA-SW-chains						
Best	2.56E+03	8.51E+01	1.04E+00	7.83E+02	2.49E+05	9.25E+02
Median	2.72E+03	9.44E+01	1.26E+00	1.19E+03	2.85E+05	1.06E+03
Worst	2.96E+03	1.24E+02	1.63E+00	2.55E+03	3.32E+05	1.21E+03
Mean	2.74E+03	9.98E+01	1.24E+00	1.30E+03	2.85E+05	1.07E+03
Std	1.22E+02	1.40E+01	1.25E−01	4.36E+02	1.78E+04	7.29E+01

Table 4 Results of multiple-problem Wilcoxon’s test for EADE and EADE* versus LMDEa, SDENS, jDElsgo, and DECC-DML over all functions at a 0.05 significance level with (1.25E+05 FES)

Algorithm	R^+	R^-	p value	Better	Equal	Worse	Dec.
EADE vs EADE*	77	133	0.296	9	0	11	≈
EADE vs LMDEa	148	62	0.108	14	0	6	≈
EADE vs SDENS	193	17	0.001	18	0	2	+
EADE vs jDElsgo	210	0	0.000	20	0	0	+
EADE vs DECC-DML	162	48	0.033	16	0	4	+
EADE vs MA-SW-chains	22	188	0.002	3	0	17	–
EADE* vs LMDEa	162	48	0.033	17	0	3	+
EADE* vs SDENS	210	0	0.000	20	0	0	+
EADE* vs jDElsgo	210	0	0.000	20	0	0	+
EADE* vs DECC-DML	164	46	0.028	15	0	5	+
EADE* vs MA-SW-chains	35	175	0.009	4	0	16	–

Table 5 Results of multiple-problem Wilcoxon’s test for EADE and EADE* versus LMDEa, SDENS, jDElsgo, and DECC-DML over all functions at a 0.05 significance level with (6.00E+05 FES)

Algorithm	R^+	R^-	p value	Better	Equal	Worse	Dec.
EADE vs EADE*	116	94	0.681	13	0	7	≈
EADE vs LMDEa	130	80	0.351	12	0	8	≈
EADE vs SDENS	198	12	0.001	18	0	8	+
EADE vs jDElsgo	190	20	0.002	17	0	3	+
EADE vs DECC-DML	182	28	0.004	15	0	5	+
EADE vs MA-SW-chains	71	139	0.204	7	0	13	≈
EADE* vs LMDEa	147	63	0.117	13	0	7	≈
EADE* vs SDENS	172	38	0.000	13	0	7	+
EADE* vs jDElsgo	183	27	0.004	15	0	5	+
EADE* vs DECC-DML	203	7	0.012	19	0	1	+
EADE* vs MA-SW-chains	61	149	0.10	6	0	14	≈

Table 6 Results of multiple-problem Wilcoxon’s test for EADE and EADE* versus LMDEa, SDENS, jDElsgo, and DECC-DML over all functions at a 0.05 significance level with (3.00E+06 FES)

Algorithm	R^+	R^-	p value	Better	Equal	Worse	Dec.
EADE vs EADE*	138	52	0.084	15	1	4	≈
EADE vs LMDEa	83	127	0.411	9	0	11	≈
EADE vs SDENS	188	22	0.002	17	0	3	+
EADE vs jDElsgo	81	129	0.370	7	0	13	≈
EADE vs DECC-DML	186	24	0.002	15	0	5	+
EADE vs MA-SW-chains	98	112	0.794	10	0	10	≈
EADE* vs LMDEa	51	159	0.044	4	0	16	–
EADE* vs SDENS	185	25	0.003	16	0	4	+
EADE* vs jDElsgo	66	144	0.145	4	0	16	≈
EADE* vs DECC-DML	182	28	0.004	14	0	6	+
EADE* vs MA-SW-chains	79	131	0.332	7	0	13	≈

from Table 5, it can be obviously seen that EADE and EADE* are significantly better than SDENS and DECC-DML algorithms, EADE* is significantly worse than LMDEa algorithm. Besides, there is no significant difference between EADE*, LMDEa, and jDElsgo and EADE. From Tables 4 and 5, it is noteworthy that EADE* is better than all DE s algorithms (LMDEa, SDENS, jDElsgo, and DECC-DML).

Moreover, from Table 6, EADE* outperforms SDENS and DECC-DML algorithms and it is competitive with jDElsgo algorithm which indicate that the new mutation scheme helps to maintain effectively the balance between the global exploration and local exploitation abilities for searching process of the DE during the search process. EADE outperforms SDENS and DECC-DML algorithms, and it is competi-

Table 7 Average ranks for all algorithms across all problems and 1.2e+05, 6.0e+05, and 3.0e+06 function evaluations (FEs)

Rank	Algorithm	1.2e+05	6.0e+05	3.0e+06	Mean ranking
1	EADE	3.00	<u>2.90</u>	3.33	<u>3.08</u>
2	EADE*	<u>2.65</u>	3.35	4.53	3.51
3	LMDEa	4.05	3.45	3.20	3.57
4	SDENS	5.75	6.08	5.85	5.89
5	jDElsgo	6.40	5.08	2.78	4.75
6	DECC-DML	4.70	4.65	5.03	4.79
7	MA-SW-chains	1.45	2.50	<u>3.30</u>	2.42

Table 8 Experimental comparisons between EADE and state-of-the-art algorithms, $D = 100$

	EADE	CEO	SL-PSO	CCPSO2	sep-CMA-ES	EPUS-PSO	MLCC	DMS-L-PSO
F_1								
Mean	0.00E+00	9.11E−29	1.09E−27	7.73E−14	9.02E−15	7.47E−01	6.82E−14	0.00E+00
Std	0.00E+00	1.10E−28	3.50E−28	3.23E−14	5.53E−15	1.70E−01	2.32E−14	0.00E+00
F_2								
Mean	3.58E−03	3.35E+01	9.45E−06	6.08E+00	2.31E+01	1.86E+01	2.53E+01	3.65E+00
Std	3.49E−03	5.38E+00	4.97E−06	7.83E+00	1.39E+01	2.26E+00	8.73E+00	7.30E−01
F_3								
Mean	9.36E+01	3.90E+02	5.74E+02	4.23E+02	4.31E+00	4.99E+03	1.50E+02	2.83E+02
Std	5.10E+01	5.53E+02	1.67E+02	8.65E+02	1.26E+01	5.35E+03	5.72E+01	9.40E+02
F_4								
Mean	0.00E+00	5.60E+01	7.46E+01	3.98E−02	2.78E+02	4.71E+02	4.39E−13	1.83E+02
Std	0.00E+00	7.48E+00	1.21E+01	1.99E−01	3.43E+01	5.94E+01	9.21E−14	2.16E+01
F_5								
Mean	0.00E+00	0.00E+00	0.00E+00	3.45E−03	2.96E−04	3.72E−01	3.41E−14	0.00E+00
Std	0.00E+00	0.00E+00	0.00E+00	4.88E−03	1.48E−03	5.60E−02	1.16E−14	0.00E+00
F_6								
Mean	1.42E−14	1.20E−14	2.10E−14	1.44E−13	2.12E+01	2.06E+00	1.11E−13	0.00E+00
Std	0.00E+00	1.52E−15	5.22E−15	3.06E−14	4.02E−01	4.40E−01	7.87E−15	0.00E+00
F_7								
Mean	−1.17E+03	−7.28E+05	−1.48E+03	−1.50E+03	−1.39E+03	−8.55E+02	−1.54E+03	−1.14E+03
Std	1.83E+01	1.88E+04	1.90E+01	1.04E+01	2.64E+01	1.35E+01	2.52E+00	8.48E+00

tive with jDElsgo, LMDEa, and MA-SW-chains algorithms. Furthermore, the performance of all algorithms is analyzed using all function evaluations (Fes) and different categories of functions. Therefore, the mean aggregated rank of all the 6 algorithms across all problems (20) and all 1.2e+05, 6.0e+05, and 3.0e+06 function evaluations (FEs) is presented in Table 7. The best ranks are marked in bold and the second ranks are underlined. From Table 7, it can be clearly concluded that MA-SW-chains is the best followed by EADE as second best among all algorithms while EADE* is ranked third. Note that the main contribution of this study is to propose a DE framework, and not to propose a “Best” algorithm or competitor to defeat other state-of-the-art algorithms. However, it is noteworthy to mentioning that the performance of EADE considerably increases as the num-

ber of functions evaluation increases from 1.25E+05 to 3.00E+06 which means that it benefits from extra FES. Therefore, it can be obviously observed from Tables 4, 5, and 6 that EADE is inferior to MA-SW chains for 17, 13, and 10 functions in 1.25E+05, 6.00E+05, and 3.00E+06 FES, respectively. Thus, it can be concluded that the inferiority of the EADE algorithm against MA-SW chains algorithm considerably decreases as the FEs increases.

On the other hand, regarding CEC’2008 benchmark functions, Tables 8, 9, and 10 contain the results obtained by all algorithms in $D = 100$, $D = 500$, and $D = 1000$, respectively. It includes the obtained best and the standard deviations of error from optimum solution of EADE and other seven state-of-the-art algorithms over 25 runs for all 7 benchmark functions. The results provided by these approaches

Table 9 Experimental comparisons between EADE and state-of-the-art algorithms, $D = 500$

	EADE	CEO	SL-PSO	CCPSO2	sep-CMA-ES	EPUS-PSO	MLCC	DMS-L-PSO
F_1								
Mean	0.00E+00	6.57E−23	7.24E−24	7.73E−14	2.25E−14	8.45E+01	4.30E−13	0.00E+00
Std	0.00E+00	3.90E−24	2.05E−25	3.23E−14	6.10E−15	6.40E+00	3.31E−14	0.00E+00
F_2								
Mean	7.28E+01	2.60E+01	3.47E+01	5.79E+01	2.12E+02	4.35E+01	6.67E+01	6.89E+01
Std	4.35E+00	2.40E+00	1.03E+00	4.21E+01	1.74E+01	5.51E−01	5.70E+00	2.01E+00
F_3								
Mean	1.02E+03	5.74E+02	6.10E+02	7.24E+02	2.93E+02	5.77E+04	9.25E+02	4.67E+07
Std	106E+02	1.67E+02	1.87E+02	1.54E+02	2.59E+01	8.04E+03	1.73E+02	5.87E+06
F_4								
Mean	2.54E+01	3.19E+02	2.72E+03	3.98E−02	2.18E+03	3.49E+03	1.79E−11	1.61E+03
Std	6.65E+00	2.16E+01	3.25E+02	1.99E−01	1.51E+02	1.12E+02	6.31E−11	1.04E+02
F_5								
Mean	3.10E−16	2.22E−16	3.33E−16	1.18E−03	7.88E−04	1.64E+00	2.13E−13	0.00E+00
Std	4.68E−17	0.00E+00	0.00E+00	4.61E−03	2.82E−03	4.69E−02	2.48E−14	0.00E+00
F_6								
Mean	4.26E−14	4.13E−13	1.46E−13	5.34E−13	2.15E+01	6.64E+00	5.34E−13	2.00E+02
Std	0.00E+00	1.10E−14	2.95E−15	8.61E−14	3.10E−01	4.49E−01	7.01E−14	9.66E−02
F_7								
Mean	−4.41E+03	−1.97E+06	−5.94E+03	−7.23E+03	−6.37E+03	−3.51+03	−7.43E+03	−4.20E+03
Std	4.11E+01	4.08E+04	1.72E+02	4.16E+01	7.59E+01	2.10E+01	8.03E+00	1.29E+01

Table 10 Experimental comparisons between EADE and state-of-the-art algorithms, $D = 1000$

	EADE	CEO	SL-PSO	CCPSO2	sep-CMA-ES	EPUS-PSO	MLCC	DMS-L-PSO
F_1								
Mean	4.06E−028	1.09E−21	7.10E−23	5.18E−13	7.81E−15	5.53E+02	8.46E−13	0.00E+00
Std	5.73E−028	4.20E−23	1.40E−24	9.61E−14	1.52E−15	2.86E+01	5.01E−14	0.00E+00
F_2								
Mean	8.97E+01	4.15E+01	8.87E+01	7.82E+01	3.65E+02	4.66E+01	1.09E+02	9.15E+01
Std	2.77E+00	9.74E−01	5.25E+00	4.25E+01	9.02E+00	4.00E−01	4.75E+00	7.14E−01
F_3								
Mean	2.15E+03	1.01E+03	1.04E+03	1.33E+03	9.10E+02	8.37E+05	1.80E+03	8.98E+09
Std	1.51E+02	3.02E+01	5.14E+01	2.63E+02	4.54E+01	1.52E+05	1.58E+02	4.39E+08
F_4								
Mean	1.54E+02	6.89E+02	5.89E+02	1.99E−01	5.31E+03	7.58E+03	1.37E−10	3.84E+03
Std	7.54E+00	3.10E+01	9.26E+00	4.06E−01	2.48E+02	1.51E+02	3.37E−10	1.71E+02
F_5								
Mean	4.88E−16	2.26E−16	4.44E−16	1.18E−03	3.94E−04	5.89E+00	4.18E−13	0.00E+00
Std	6.08E−17	2.18E−17	0.00+E00	3.27E−03	1.97E−03	3.91E−01	2.78E−14	0.00E+00
F_6								
Mean	5.75E−14	1.21E−12	3.44E−13	1.02E−12	2.15E+01	1.89E+01	1.06E−12	7.76E+00
Std	2.24E−15	2.64E−14	5.32E−15	1.68E−13	3.19E−01	2.49E+00	7.68E−14	8.92E−02
F_7								
Mean	−7.99E+03	−3.83E+06	−1.30E+04	−1.43E+04	−1.25E+04	−6.62E+03	−1.47E+04	−7.50E+03
Std	3.09E+01	4.82E+04	1.04E+02	8.27E+01	9.36E+01	3.18E+01	1.51E+01	1.63E+01

Table 11 Results of multiple-problem Wilcoxon's test for EADE versus state-of-the-art algorithms over all functions at a 0.05 significance level with ($D = 100$)

Algorithm	R^+	R^-	p value	Better	Equal	Worse	Dec.
EADE vs CEO	13	8	0.6	4	1	2	≈
EADE vs SL-PSO	13	8	0.6	4	1	2	≈
EADE vs CCPSO2	21	7	0.237	6	0	1	≈
EADE vs Sep-CMA-ES	17	11	0.612	5	0	2	≈
EADE vs EPUS-PSO	28	0	0.018	7	0	0	+
EADE vs MLCC	21	7	0.237	6	0	1	≈
EADE vs DMS-L-PSO	14	1	0.080	4	1	2	≈

were directly taken from references [39,40]. For remarking the best algorithm, best mean for each function is highlighted in boldface.

As shown in Table 8, EADE is able to find the global optimal solution consistently in 4 out of 7 test functions over 25 runs with the exception of test functions (F_2 , F_3 , and F_7). With respect to F_2 , although the optimal solution is not reached, the best result achieved is very close to the global optimal solution which can be verified by the very small function error and standard deviation. Regarding F_3 , the narrow valley from local optimal to global optimal present is a challenge to all algorithms that prevent EADE from finding global solution. In addition, EADE gets trapped in local optima on f_7 as all other compared algorithms with exception to CEO which provide the best mean, although its global optimal solution is unknown. From the results presented in Tables 9 and 10, for 500D and 1000D problems, a similar trend as was observed in 100D is continued here. It is still able to provide the same competitive results. In general, it can be observed that EADE, CEO, and SL-PSO do significantly better than the others on most functions in different dimensions. On the other hand, EPUS-PSO performs poorly on all the functions in all dimensions.

In addition, in all functions for all the three dimensionalities, EADE provides very small standard deviation which means that the differences between mean and median are small even in the cases when the final results are far away from the optimum, regardless of the dimensions. That implies that EADE is a robust algorithm. Moreover, due to insignificant difference between the results in three dimensions, it can be concluded that the performance of the EADE algorithm slightly diminishes and it is still more stable and robust against the curse of dimensionality, i.e., it is overall steady as the dimensions of the problems increase. Obviously, it can be deduced that the performance of the proposed EADE on large-scale optimization problems is surprisingly good, because there is no specific mechanism for large-scale optimization such as the divide-and-conquer or the CC framework adopted in EADE. However, the good scalability of EADE is due to the following two reasons. First, the new mutation scheme helps to maintain effectively the

balance between the global exploration and local exploitation abilities for searching process of the DE needed for handling large-scale problems. Second, the proposed novel self-adaptive scheme for gradual change of the values of the crossover rate considerably balances the common trade-off between the population diversity and convergence speed might have contributed to the scalability. In fact, further investigation and experimental analysis of the performance of EADE on solving large-scale optimization are needed. Furthermore, to investigate and compare the performance of the proposed algorithms EADE against other algorithms in statistical sense, the multi-problem Wilcoxon signed-rank and Friedman tests between EADE and others in 100D, 500D, and 1000D are summarized in Tables 11, 12, 13, and 14, respectively. Where R^+ is the sum of ranks for the functions in which first algorithm outperforms the second algorithm in the row, and R^- is the sum of ranks for the opposite.

From Table 11, we can see that EADE obtains higher R^+ values than R^- in all cases, while slightly lower R^+ value than R^- value in comparison with SaDE. However, from Tables 12 and 13, in the cases of EFADE versus CEO, SL-PSO, and CCPSO2, they get higher R^- than R^+ values. The reason is that EADE gains the performance far away of what these three algorithms do on function F_7 , resulting in higher ranking values. According to the Wilcoxon's test at $\alpha = 0.05$, the significance difference can only be observed in EFADE versus EPUS-PSO case. Besides, Table 14 lists the average ranks EADE and other algorithms according to Friedman test for $D = 100, 500$, and 1000 , respectively. The best ranks are marked in bold and the second ranks are underlined. The p value computed through Friedman test is 0.01, 0.48, and 0.47, respectively. Thus, it can be concluded that there is a significant difference between the performances of the algorithms. It can be clearly seen from Table 14 that EADE gets the first ranking among all algorithms in 100-dimensional functions, followed by CEO and SL-PSO. Regarding 500D and 1000D problems, CEO gets the first ranking, followed by SL-PSO and EADE. Furthermore, the performance of all algorithms is analyzed using all dimensions and different categories of functions. Therefore, the mean aggregated rank of all the 8 algorithms across all problems (7) and all

Table 12 Results of multiple-problem Wilcoxon's test for EADE* versus state-of-the-art algorithms over all functions at a 0.05 significance level with ($D = 500$)

Algorithm	R^+	R^-	p value	Better	Equal	Worse	Dec.
EADE vs CEO	9	19	0.398	3	0	4	≈
EADE vs SL-PSO	13	15	0.866	4	0	3	≈
EADE vs CCPSO2	6	22	0.176	3	0	4	≈
EADE vs Sep-CMA-ES	17	11	0.612	5	0	2	≈
EADE vs EPUS-PSO	22	6	0.063	6	0	1	≈
EADE vs MLCC	6	22	0.176	3	0	4	≈
EADE vs DMS-L-PSO	18	3	0.116	4	1	2	≈

Table 13 Results of multiple-problem Wilcoxon's test for EADE* versus state-of-the-art algorithms over all functions at a 0.05 significance level with ($D = 1000$)

Algorithm	R^+	R^-	p value	Better	Equal	Worse	Dec.
EADE vs CEO	9	19	0.398	3	0	4	≈
EADE vs SL-PSO	9	19	0.398	3	0	4	≈
EADE vs CCPSO2	6	22	0.176	3	0	4	≈
EADE vs Sep-CMA-ES	17	11	0.612	5	0	2	≈
EADE vs EPUS-PSO	25	3	0.063	6	0	1	≈
EADE vs MLCC	10	18	0.499	3	0	4	≈
EADE vs DMS-L-PSO	25	3	0.063	5	0	2	≈

Table 14 Average ranks for all algorithms across all problems with $D = 100$, $D = 500$ and $D = 1000$

Rank	Algorithm	$D = 100$	$D = 500$	$D = 1000$	Mean ranking
1	EADE	2.57	3.93	3.86	<u>3.45</u>
2	CEO	3.64	2.43	2.86	2.98
3	SL-PSO	3.93	<u>3.71</u>	<u>3.29</u>	3.64
4	CCPSO2	5.14	4.36	4.00	4.5
5	Sep-CMA-ES	5.43	5.29	5.71	5.47
6	EPUS-PSO	7.43	6.86	6.86	7.05
7	MLCC	4.29	4.21	4.43	4.31
8	DMS-L-PSO	<u>3.57</u>	5.21	5.00	4.59

dimensions (100D, 500D, and 1000D) is presented in Table 12. From Table 12, it can be clearly concluded that CEO is the best followed by EADE as second best among all algorithms, while SL-PSO is ranked third. Finally, it is noteworthy to highlighting that EADE has shown comparable performance to MLCC, CCPSO2, and DMS-PSO, the three algorithms originally designed for solving large-scale optimization problems. Plus, it also significantly outperforms sep-CMA-ES and EPUS-PSO algorithms.

Overall, from the above results, comparisons, and discussion, the proposed EADE algorithm is of better searching quality, efficiency, and robustness for solving unconstrained large-scale global optimization problems. It is clear that the proposed EADE and EADE* algorithms perform well and it has shown its outstanding superiority with separable, non-separable, unimodal, and multi-modal functions with shifts in dimensionality, rotation, multiplicative noise in fitness, and composition of functions. Consequently, its performance is not influenced by all these obstacles. Contrarily, it greatly

keeps the balance the local optimization speed and the global optimization diversity in challenging optimization environment with invariant performance. Besides, it can be obviously concluded from direct and statistical results that EADE and EADE* are powerful algorithms, and its performance is superior and competitive with the performance of the-state-of-the-art well-known DE-based algorithms.

Conclusion

To efficiently concentrate the exploitation tendency of some sub-region of the search space and to significantly promote the exploration capability in whole search space during the evolutionary process of the conventional DE algorithm, an enhanced adaptive Differential Evolution (EADE) algorithm for solving large-scale global numerical optimization problems over continuous space was presented in this paper. The proposed algorithm introduces a new mutation rule. It uses

two random chosen vectors of the top and bottom $100p\%$ individuals in the current population of size NP , while the third vector is selected randomly from the middle ($NP-2p$) individuals. The mutation rule is combined with the basic mutation strategy $DE/rand/1/bin$, where only one of the two mutation rules is applied with the probability of 0.5. Furthermore, we propose a novel self-adaptive scheme for gradual change of the values of the crossover rate that can excellently benefit from the past experience of the individuals in the search space during evolution process which, in turn, can considerably balance the common trade-off between the population diversity and convergence speed. The proposed mutation rule was shown to enhance the global and local search capabilities of the basic DE and to increase the convergence speed. The algorithm has been evaluated on the standard high-dimensional benchmark problems. The comparison results between EADE and EADE* and the other four state-of-art DE-based algorithms that were all tested on this test suite on the IEEE congress on Evolutionary competition in 2008 and 2010 indicate that the proposed algorithm and its version are highly competitive algorithms for solving large-scale global optimization problem. The experimental results and comparisons showed that the EADE and EADE* algorithms performed better in large-scale global optimization problems with different types and complexity; they performed better with regard to the search process efficiency, the final solution quality, the convergence rate, and robustness, when compared with other algorithms. In fact, the performance of the EADE and EADE* algorithm was statistically superior to and competitive with other recent and well-known DEs algorithms. Finally, to the best of our knowledge, this is the first study that uses all these different types of approaches (12) to carry out evaluation and comparisons on CEC'2008 and CEC'2010 benchmark problems. Virtually, this study aims to prove that EADE is a competitive and an efficient approach as well as being superior to the most recent techniques in the field of large-scale optimization. Several current and future works can be developed from this study. First, current research effort focuses on how to control the scaling factors by self-adaptive mechanism and develop another self-adaptive mechanism for crossover rate. In addition, the new version of EADE combined with Cooperative Co-evolution (CC) framework is being developed and will be experimentally investigated soon. Moreover, future research will investigate the performance of the EADE algorithm in solving constrained and multi-objective optimization problems as well as real-world applications such as data mining and clustering problems. In addition, large-scale combinatorial optimization problems will be taken into consideration. Another possible direction is integrating the proposed novel mutation scheme with all compared and other self-adaptive DE variants plus combining the proposed self-adaptive crossover with other DE mutation schemes. In

addition, the promising research direction is joining the proposed mutation with evolutionary algorithms, such as genetic algorithms, harmony search, and particle swarm optimization, as well as foraging algorithms such as artificial bee colony, bees algorithm, and ant colony optimization. The MATLAB source code of EADE is available upon request.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Tang K, Li X, Suganthan PN, Yang Z, Weise T (2009) Benchmark functions for the CEC 2010 special session and competition on large scale global optimization. Technical report. Nature Inspired Computation and Applications Laboratory, USTC, China
2. Tang K, Yao X, Suganthan PN, MacNish C, Chen YP, Chen CM, Yang Z (2007) Benchmark functions for the CEC2008 special session and competition on large scale global optimization. Technical Report for CEC 2008 special issue, Nov. 2007
3. Engelbrecht AP (2002) Computational intelligence: an introduction, 2nd edn. Wiley, USA. ISBN 978-0-470-03561-0
4. Storn R, Price K (1995) Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012. ICSI
5. El-Quliti SA, Ragab AH, Abdelaal R et al (2015) Anonlinear goal programming model for university admission capacity planning with modified differential evolution algorithm. *Mathematical Probl Eng* 2015:13
6. El-Qulity SA, Mohamed AW (2016) A generalized national planning approach for admission capacity in higher education: a nonlinear integer goal programming model with a novel differential evolution algorithm. *Computational Intell Neurosci* 2016:14
7. Hachicha N, Jarboui B, Siarry P (2011) A fuzzy logic control using a differential evolution algorithm aimed at modeling the financial market dynamics. *Inf Sci* 181(1):79–91
8. El-Quliti SA, Mohamed AW (2016) A large-scale nonlinear mixed-binary goal programming model to assess candidate locations for solar energy stations: an improved binary differential evolution algorithm with a case study. *J Computational Theoretical Nanosci* 13(11):7909–7921
9. Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. *IEEE Trans Evol Comput* 15(1):4–31
10. Noman N, Iba H (2008) Accelerating differential evolution using an adaptive local search. *IEEE Trans Evol Comput* 12(1):107–125
11. Das S, Abraham A, Chakraborty UK, Konar A (2009) Differential evolution using a neighborhood based mutation operator. *IEEE Trans Evol Comput* 13(3):526–553
12. Lampinen J, Zelinka I (2000) On stagnation of the differential evolution algorithm, In: Proceedings of Mendel 6th international conference on soft computing, pp 76–83
13. Mohamed AW, Sabry HZ (2012) Constrained optimization based on modified differential evolution algorithm. *Inf Sci* 194(2012):171–208
14. Mohamed AW, Sabry HZ, Khorshid M (2012) An alternative differential evolution algorithm for global optimization. *J Adv Res* 3(2):149–165

15. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans Evol Comput* 13(2):398–417
16. Zhang JQ, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. *IEEE Trans Evol Comput* 13(5):945–958
17. Mallipeddi R, Suganthan PN, Pan QK, Tasgetiren MF (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl Soft Comput* 11(2):1679–1696
18. Mohamed AW (2015) An efficient modified differential evolution algorithm for solving constrained non-linear integer and mixed-integer global optimization problems. *Int J Mach Learn Cyber*. doi:10.1007/s13042-015-0479-6
19. Omran MGH, Salman A, Engelbrecht AP (2005) Self-adaptive differential evolution. In: *Computational intelligence and security*, PT 1. Proceedings Lecture notes in artificial intelligence, pp 192–199
20. Mohamed AW, Sabry HZ, Farhat A (2011) Advanced differential evolution algorithm for global numerical optimization. In: *Proceedings of the IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE'11)*, pp 156–161. Penang, Malaysia, December 2011.
21. Mohamed AW, Sabry HZ, Abd-Elaziz T (2013) Real parameter optimization by an effective differential evolution algorithm. *Egypt Inform J* 14(1):37–53
22. Mohamed AW (2017) A novel differential evolution algorithm for solving constrained engineering optimization problems. *J Intell Manuf*. doi:10.1007/s10845-017-1294-6
23. Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
24. Fan HY, Lampinen J (2003) A trigonometric mutation operation to differential evolution. *J Glob Optim* 27(1):105–129
25. Price KV, Storn RM, Lampinen JA (2005) *Differential evolution: a practical approach to global optimization*, 1st edn. Springer, New York
26. Potter AM, Jong KAD (1994) A cooperative co-evolutionary approach to function optimization. In: *Proceedings of the third international conference on parallel problem solving from the nature*, pp 249–257. Springer-Verlag
27. Yang Z, Tang K, Yao X (2008) Large scale evolutionary optimization using cooperative co-evolution. *Inf Sci* 178(15):2985–2999
28. Omidvar MN, Li XD, Yao X (2010) Cooperative co-evolution with delta grouping for large scale non-separable function optimization. In: *2010 IEEE Congress on evolutionary computation (CEC), 2010*, pp 1762–1769
29. Liu Y, Yao X, Zhao Q, Higuchi T (2001) Scaling up fast evolutionary programming with cooperative co-evolution. In: *Proceedings of the IEEE World Congress on computational intelligence, 2001*, pp 1101–1108
30. Yang Z, Tang K, Yao X (2007) Differential evolution for high-dimensional function optimization. In: *Proceedings of the IEEE World Congress on computational intelligence, 2007*, pp 3523–3530
31. Yang Z, Tang K, Yao X (2008) Multilevel cooperative coevolution for large scale optimization. In: *Proceedings of the IEEE World Congress on computational intelligence, 2008*, pp 1663–1670
32. Yang P, Tang K, Yao X (2017) Turning high-dimensional optimization into computationally expensive optimization. *IEEE Trans Evol Comput*. doi:10.1109/TEVC.2017.2672689
33. Tang R-L, Wu Z, Fang Y-J (2016) Adaptive multi-context cooperatively coevolving particle swarm optimization for large-scale problems. *Soft Comput* 1–20: doi:10.1007/s00500-016-2081-6
34. Takahama T, Sakai S (2012) Large scale optimization by differential evolution with landscape modality detection and a diversity archive. In: *Proceedings of 2012 IEEE Congress on evolutionary computation*, pp 2842–2849
35. Brest J, Zamuda A, Bošković B, Fister I, Maučec MS (2010) Large scale global optimization using self-adaptive differential evolution algorithm. In: *IEEE World Congress on computational intelligence*, pp 3097–3104
36. Wang H, Wu Z, Rahnamayan S, Jiang D (2010) Sequential DE enhanced by neighborhood search for large scale global optimization. In: *Proceedings of the IEEE Congress on evolutionary computation, 2010*, pp 4056–4062
37. Molina D, Lozano M, Herrera F (2010) MA-SW-Chains: Memetic algorithm based on local search chains for large scale continuous global optimization. In: *IEEE Congress on evolutionary computation*, pp 1–8. IEEE, Barcelona, Spain
38. Kabán A, Bootkrajang J, Durrant RJ (2015) Toward large-scale continuous eda: a random matrix theory perspective. *Evol Comput* 24(2):255–291
39. Cheng R, Jin Y (2015) A competitive swarm optimizer for large scale optimization. *IEEE Trans Cybern* 45(2):191–205
40. Cheng R, Jin Y (2015) A social learning particle swarm optimization algorithm for scalable optimization. *Inf Sci* 291:43–60
41. Wang H, Wu Z, Rahnamayan S (2011) Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems. *Soft Comput* 15(11):2127–2140
42. Dong W, Chen T, Tino P, Yao X (2013) Scaling up estimation of distribution algorithms for continuous optimization. *IEEE Trans Evol Comput* 17(6):797–822
43. Feoktistov V (2006) *Differential evolution, in search of solutions*, vol 5. Springer, New York
44. 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
45. Ronkkonen J, Kukkonen S, Price KV (2005) Real parameter optimization with differential evolution. In: *Proceedings of the IEEE Congress on evolutionary computation, 2005*, pp 506–513
46. Brest J, Greiner S, Bošković B, Mernik M, Žumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans Evol Comput* 10(6):646–657
47. Weber M, Neri F, Tirronen V (2011) A study on scale factor in distributed differential evolution. *Inf Sci* 181(12):2488–2511
48. Sarker RA, Elsayed SM, Ray T Differential evolution with dynamic parameters selection for optimization problems. *IEEE Trans Evol Comput* doi:10.1109/TEVC.2013.2281528
49. Elsayed SM, Sarker RA, Essam DL (2013) An improved self-adaptive differential evolution for optimization problems. *IEEE Trans Ind Eng* 9(1):89–99
50. Li X, Yao Y (2011) Cooperatively coevolving particle swarms for large scale optimization. *IEEE Trans Evol Comput* 16(2):1–15
51. Ros R, Hansen N (2008) A simple modification in CMA-ES achieving linear time and space complexity. In: Rudolph G (ed) *LNCS: Parallel problem solving from nature (PPSN X)*, vol 5199. Springer, New York, pp 296–305
52. Hsieh S-T, Sun T-Y, Liu C-C, Tsai S-J (2008) Solving large scale global optimization using improved particle swarm optimizer. In: *Proceedings of IEEE Congress on evolutionary computation, IEEE, 2008*, pp 1777–1784
53. Zhao S-Z, Liang JJ, Suganthan PN, Tasgetiren MF (2008) Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. In: *Proceedings of IEEE Congress on evolutionary computation, IEEE, 2008*, pp 3845–3852
54. García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: a case study on the CEC'2005 special session on real parameter optimization. *J Heuristics* 15:617–644