Int. J. Mach. Learn. & Cyber. (2019) 10:253–277
DOI 10.1007/s13042-017-0711-7

ORIGINAL ARTICLE

Adaptive guided differential evolution algorithm with novel mutation for numerical optimization

Ali Wagdy Mohamed[1] · **Ali Khater Mohamed**[2]

Received: 27 September 2016 / Accepted: 7 August 2017 / Published online: 17 August 2017

Abstract This paper presents adaptive guided differential evolution algorithm (AGDE) for solving global numerical optimization problems over continuous space. In order 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 the bottom 100*p*% individuals in the current population of size *NP* while the third vector is selected randomly from the middle [*NP-2*(100*p* %)] individuals. This new mutation scheme helps maintain effectively the balance between the global exploration and local exploitation abilities for searching process of the DE. Besides, a novel and effective adaptation scheme is used to update the values of the crossover rate to appropriate values without either extra parameters or prior knowledge of the characteristics of the optimization problem. In order to verify and analyze the performance of AGDE, Numerical experiments on a set of 28 test problems from the CEC2013 benchmark for 10, 30, and 50 dimensions, including a comparison with classical DE schemes and some recent evolutionary algorithms are executed. Experimental results indicate that in terms of robustness, stability and quality of the solution obtained, AGDE is significantly better than, or at least comparable to state-of-the-art approaches.

Keywords Evolutionary computation · Global optimization · Differential evolution · Novel mutation · Adaptive crossover

1 Introduction

Differential evolution (DE), proposed by Storn and Price [1, 2], is a stochastic population-based search method. It exhibits excellent capability in solving a wide range of optimization problems with different characteristics from several fields and many real-world application problems [3]. Similar to all other evolutionary algorithms (EAs), the evolutionary process of DE uses mutations, crossover and selection operators at each generation to reach the global optimum. Besides, it is one of the most efficient evolutionary algorithms (EAs) currently in use. In DE, each individual in the population is called target vector. Mutation is used to generate a mutant vector, which perturbs a target vector using the difference vector of other individuals in the population. After that, crossover operation generates the trial vector by combining the parameters of the mutation vector with the parameters of a parent vector selected from the population. Finally, according to the fitness value and selection operation determines which of the vectors should be chosen for the next generation by implementing a one-to-one completion between the generated trail vectors and the corresponding parent vectors [4, 5].The performance of DE basically depends on the mutation strategy, the crossover operator. Besides, The intrinsic control parameters (population size NP, scaling factor *F*, the crossover rate *CR*) play a vital role in balancing the diversity of population and convergence speed of the algorithm. The advantages are simplicity of implementation, reliable, speed and robustness [3]. Thus, it has been widely

✉ Ali Wagdy Mohamed
aliwagdy@gmail.com

Ali Khater Mohamed
ak.mohamed@mu.edu.sa

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

2 College of Science and Humanities, Majmaah University, P.O Box 66, Majmaah 11952, Kingdom of Saudi Arabia

applied in solving many real-world applications of science and engineering, such as {0–1} Knapsack Problem [6], financial markets dynamic modeling [7], feature selection [8], admission capacity planning in higher education [9, 10], and solar energy [11], for more applications, interested readers can refer to [12]. However, DE has many weaknesses, as all other evolutionary search techniques do w.r.t the NFL theorem. Generally, DE has a good global exploration ability that can reach the region of global optimum, but it is slow at exploitation of the solution [13]. Additionally, 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 [14]. Finally, the performance of DE deteriorates significantly when the problems of premature convergence and/or stagnation occur [14, 15]. Consequently, researchers have suggested many techniques to improve the basic DE. From the literature [12, 16], these proposed modifications, improvements and developments on DE focus on adjusting control parameters in an adaptive or self-adaptive manner while there are a few attempts in developing new mutations rule. In this paper, In order to overcome these drawbacks of DE, two evolutionary process components are proposed. In fact, one or both can be combined with the DE family of algorithms to enhance their search capabilities on difficult and complicated optimization problems.

1. First, to enhance global and local search capabilities and simultaneously increases the convergence speed of DE, a new directed mutation operator is introduced.
2. Second, simple adaptive scheme without extra parameters to update the values of CR in each generation are proposed. It guided the search process by the previous and current knowledge of their successful ratios that produced better trial vectors in the last generation.

AGDE has been tested on 28 benchmark test functions developed for the 2013 IEEE Congress on Evolutionary Computation (IEEE CEC2013) [17]. Extensive numerical experiments and comparisons indicate that the proposed (AGDE) algorithm is superior and competitive with four other recent evolutionary algorithms particularly in the case of high dimensional complex optimization problems with different characteristics. The rest of the paper is organized as follows. Section 2 gives a brief introduction to canonical DE algorithm, including its typical mutation operators, crossover, and selection operators. Section 3 reviews the related work. Next, in Sect. 4, the proposed (AGDE) algorithm is introduced. Section 5 computational results of testing benchmark functions and on the comparison with other techniques are reported. Section 6 discusses the effectiveness of the proposed modifications. Finally, conclusions and future works are drawn in Sect. 7.

2 Differential evolution

This section provides a brief summary of the basic differential evolution (DE) algorithm. In simple DE, generally known as DE/rand/1/bin [2, 18], an initial random population, denoted by P, consists of NP individual. Each individual is represented by the vector $x_i = (x_{1,i}, x_{2,i}, \ldots, x_{D,i})$, where D is the number of dimensions in solution space. Since the population will be varied with the running of evolutionary process, the generation times in DE are expressed by $G = 0,1, \ldots, Gmax$, where $Gmax$ is the maximal times of generations. For the ith individual of P at the G generation, it is denoted by$x_i^G = (x_{1,i}^G, x_{2,i}^G, \ldots, x_{D,i}^G)$. The lower and upper bounds in each dimension of search space are respectively recorded by$x_L = (x_{1,L}, x_{2,L}, \ldots, x_{D,L})$ and $x_U = (x_{1,U}, x_{2,U}, \ldots, x_{D,U})$. The initial population P^0 is randomly generated according to a uniform distribution within the lower and upper boundaries (x_L, x_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 [16, 19]. DE steps are discussed below:

2.1 Initialization

In order to establish a starting point for the optimization process, an initial population P^0 must be created. Typically, each jth component $(j = 1, 2, \ldots.., D)$ of the ith individuals $(i = 1, 2, \ldots.., NP)$ in the P^0 is obtained as follow:

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

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

2.2 Mutation

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

$$v_i^G = x_{r_1}^G + F \cdot (x_{r_2}^G - x_{r_3}^G), \; r_1 \neq r_2 \neq r_3 \neq i. \tag{2}$$

With randomly chosen indices $r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$. 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 [2], 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 (1). The other most frequently used mutations strategies are

$$\text{“DE/best/1” [19]: } v_i^G = x_{best}^G + F \cdot (x_{r_1}^G - x_{r_2}^G) \tag{3}$$

$$\text{“DE/best/2” [19]: } v_i^G = x_{best}^G + F \cdot (x_{r_1}^G - x_{r_2}^G) + F \cdot (x_{r_3}^G - x_{r_4}^G) \tag{4}$$

$$\text{“DE/rand/2” [19]: } v_i^G = x_{r_1}^G + F \cdot (x_{r_2}^G - x_{r_3}^G) + F \cdot (x_{r_4}^G - x_{r_5}^G) \tag{5}$$

$$\text{“DE/current-to-best/1” [19]: } v_i^G = x_i^G + F \cdot (x_{best}^G - x_i^G) + F \cdot (x_{r_1}^G - x_{r_2}^G) \tag{6}$$

$$\text{“DE/current-to-rand/1” [19]: } v_i^G = x_i^G + F \cdot (x_{r_1}^G - x_i^G) + F \cdot (x_{r_2}^G - x_{r_3}^G). \tag{7}$$

The indices r_1, r_2, r_3, r_4, r_5 are mutually different integers randomly generated within the range [1,2,…,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.

2.3 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.

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

where $rand_{j,i}$ ($i \in [1, NP]\, and\, j \in [1, D]$) 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.

2.4 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):

$$x_i^{G+1} = \begin{cases} u_i^G, f(u_i^G) \leqslant f(x_i^G) \\ x_i^G, \text{otherwise} \end{cases}. \tag{9}$$

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

3 Related work

Virtually, the performance of the canonical DE algorithm mainly depends on the chosen mutation/crossover strategies and the associated control parameters. Moreover, due to DE limitations as previously aforementioned in Sect. 1, during the past 15 years, many researchers have been working on the improvement of DE. Thus, many researchers have proposed novel techniques to overcome its problems and improve its performance [16]. A brief overview of these algorithms is proposed in this section. Firstly, many trial-and-errors experiments have been conducted to adjust the control parameters. Storn and Price [1] suggested that NP (population size) between 5D and 10D and 0.5 as a good initial value of F (mutation scaling factor). The effective value of F usually lies in a range between 0.4 and 1. The CR (crossover rate) is an initial good choice of CR = 0.1; however, since a large CR often accelerates convergence, it is appropriate to first try CR as 0.9 or 1 in order to check if a quick solution is possible. Gamperle at al. [20] recommended that a good choice for NP is between 3D and 8D, with F = 0.6 and CR lies in [0.3, 0.9]. However, Rokkonen et al. [21] concluded that F = 0.9 is a good compromise between convergence speed and convergence rate. Additionally, CR depends on the nature of the problem, so CR with a value between 0.9 and 1 is suitable for non-separable and multimodal objective functions, while a value of CR between 0 and 0.2 when the objective function is separable. On the other hand, due to the apparent contradictions regarding the rules for determining the appropriate values of the control parameters from the literature, some techniques have been designed with a view of adjusting control parameters in adaptive or self-adaptive manner instead of manual tuning procedure. Zaharie [22] introduced an adaptive DE (ADE) algorithm based on the idea of controlling the population diversity and implemented a multi-population approach. Liu and Lampinen [23] proposed a Fuzzy Adaptive Differential Evolution (FADE) algorithm. Fuzzy logic controllers were used to adjust F and CR. Numerical experiments and comparisons on a set of well known benchmark functions showed that the FADE Algorithm outperformed basic DE algorithm. Likewise, Brest et al. [24] proposed an efficient technique, named jDE, for self-adapting control parameter settings. This technique encodes the parameters into each individual and adapts them by means of evolution. The results showed that jDE is better than, or at least comparable to, the standard DE algorithm, (FADE) algorithm and other state of-the-art algorithms when considering the quality of the solutions obtained. In the same context, Omran et al. [25]

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

Begin
  G=0
  Create a random initial population $\vec{x}_i^G \; \forall i, i = 1, \ldots, NP$
  Evaluate $f(\vec{x}_i^G) \; \forall i, i = 1, \ldots, NP$
  For G=1 to Gmax **Do**
    For i=1 to NP **Do**
      Select randomly $r1 \neq r2 \neq r3 \neq i \in [1, NP]$
      j_{rand}= randint(1,D)
      For j=1 to D **Do**
        If ($rand_{i,j}[0,1] <$ CR **or** j= j_{rand}) **Then**
          $u_{i,j}^G = x_{r1,j}^G + F \cdot (x_{r2,j}^G - x_{r3,j}^G)$
        Else
          $u_{i,j}^G = x_{i,j}^G$
        End If
      End For
      If($f(\vec{u}_i^G) \leq f(\vec{x}_i^G)$)**Then**
        $\vec{x}_i^{G+1} = \vec{u}_i^G$
      Else
        $\vec{x}_i^{G+1} = \vec{x}_i^G$
      End If
    End For
    G=G+1
  End For
End

proposed a self-adaptive differential evolution (SDE) algorithm. F was self-adapted using a mutation rule similar to the mutation operator in the basic DE. The experiments conducted showed that SDE generally outperformed DE algorithms and other evolutionary algorithms. Qin et al. [26] introduced a self-adaptive differential evolution (SaDE). The main idea of SaDE is to simultaneously implement two mutation schemes: “DE/rand/1/bin” and “DE/best/2/bin” as well as adapt mutation and crossover parameters. The Performance of SaDE evaluated on a suite of 26 several benchmark problems and it was compared with the conventional DE and three adaptive DE variants. The experimental results demonstrated that SaDE yielded better quality solutions and had a higher success rate. In the same context, inspired by SaDE algorithm and motivated by the success of diverse self-adaptive DE approaches, Mallipeddi et al. [19] developed a self-adaptive DE, called EPSDE, based on ensemble approach. In EPSDE, a pool of distinct mutation strategies along with a pool of values for each control parameter coexists throughout the evolution process and competes to produce offspring. The performance of EPSDE was evaluated on a set of bound constrained problems and was compared with conventional DE and other state-of-the-art parameter adaptive DE variants. The resulting comparisons showed that EPSDE algorithm outperformed conventional DE and other state of-the-art parameter adaptive DE variants in terms of solution quality and robustness. Similarly, motivated by the important results obtained by other researchers during past years, Wang et al. [27] proposed a novel method, called composite DE (CoDE). This method used three trial vector generation strategies and three control parameter settings. It randomly combines them to generate trial vectors. The performance of CoDE has been evaluated on 25 benchmark test functions developed for IEEE CEC2005 and it was very competitive to other compared algorithms. Recently, super-fir multicriteria adaptive DE (SMADE) is proposed by Caraffini et al. [28], which is a Memetic approach based on the hybridization of the covariance matrix adaptive evolutionary strategy (CMAES) with modified DE, namely Multicriteria Adaptive DE (MADE). MADE makes use of four mutation/crossover strategies in adaptive manner which are rand/1/bin, rand/2/bin, rand-to-best/2/bin and current-to-rand/1. The control parameters CR and F are adaptively adjusted during the evolution. At the beginning of the optimization process, CMAES is used to generate a solution with a high quality which is then injected into the population of MADE. The performance of SMADE has been evaluated on 28 benchmark test functions developed for IEEE CEC2013 and the experimental results are very promising. On the other hand, improving the trial vector generation strategy has attracted many researches. In order to improve the convergence velocity of DE, Fan and Lampinen [18] proposed a trigonometric mutation scheme, which is considered local search operator, and combined it with DE/rand/1 mutation operator to design TDE algorithm. Zhang and Sanderson [29] 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 by updating control parameters in an adaptive manner. Simulation results show that JADE was better than, or at least competitive to, other classic or adaptive DE algorithms such as Particle swarm and other evolutionary algorithms from the literature in terms of convergence performance. Along the same lines, to overcome the premature convergence and stagnation problems observed in the classical DE, Islam et al. [30] proposed modified DE with p-best crossover, named MDE_$_p$BX. The novel mutation strategy is similar to DE/current-to-best/1 scheme. It selects the best vector from a dynamic group of q% of the randomly selected population members. Moreover, their crossover strategy uses a vector that is randomly selected from the p top ranking vectors according to their objective values in the current population (instead of the target vector) to enhance the inclusion of generic information from the elite individuals in the current generation. The parameter p is linearly reduced with generations to favors the exploration at the beginning of the search and exploitation during the later stages by gradually downsizing the elitist portion of the population. Das et al. [14] proposed two kinds of topological neighborhood models for DE in order to achieve better balance between its explorative and exploitative tendencies. In this method, called DEGL, two trial vectors are created by the global and local neighborhood-based mutation. These two trial vectors are combined to form the actual trial vector by using a weight factor. The performance of DEGL has been evaluated on 24 benchmark test functions and two real-world problems and showed very competitive results. In order to solve unconstrained and constrained optimization problems, Mohamed et al. [31, 32] proposed a novel directed mutation based on the weighted difference vector between the best and the worst individuals at a particular generation, is introduced. The new directed mutation rule is combined with the modified basic mutation strategy DE/rand/1/bin, where only one of the two mutation rules is applied with the probability of 0.5. The proposed mutation rule is shown to enhance the local search ability of the basic differential evolution (DE) and to get a better trade-off between convergence rate and robustness. Numerical experiments on well-known unconstrained and constrained benchmark test functions and five engineering design problems have shown that the new approach is efficient, effective and robust. Similarly, to enhance global and local search capabilities and simultaneously increases the convergence speed of DE, Mohamed [33] introduced a new triangular mutation rule based on the convex combination vector of the triplet defined by the three randomly chosen vectors and the difference vector between the best and the worst individuals among the three randomly selected vectors. In this algorithm, called IDE, the mutation rule is combined with the basic mutation strategy through a non-linear decreasing probability rule. A restart mechanism is also proposed to avoid premature convergence. IDE istested on a well-known set of unconstrained problems and shows its superiority to state-of-the-art differential evolution variants. Practically, it can be observed that the main modifications, improvements and developments on DE focus on adjusting control parameters in an adaptive or self-adaptive manner. However, a few enhancements have been implemented to modify the structure and/or mechanism of basic DE algorithm or to propose new mutation rules so as to enhance the local and global search ability of DE and to overcome the problems of stagnation or premature convergence.

4 Adaptive guided differential evolution algorithm with novel mutation for numerical optimization (AGDE)

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

4.1 Novel mutation scheme

DE/rand/1 is the fundamental mutation strategy developed by Storn and Price [2, 3], and is reported to be the most successful and widely used scheme in the literature [16]. 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 [26]. DE/rand/2 strategy, like the former scheme with extra two vectors that forms another difference vector, which might lead to better perturbation than one-difference-vector-based strategies [26]. 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 lead 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, in order 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 [19, 26, 27]. Contrarily, taking into consideration the weakness of existing greedy strategies, [29] introduced a new differential evolution (DE) algorithm, named JADE, to improve optimization performance by implementing a new mutation strategy "DE/current-to-*p*best" 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:

$$v_i^{G+1} = x_r^G + F \cdot (x_{p\_best}^G - x_{p\_worst}^G), \tag{10}$$

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 $p \in (0\%, 50\%)$, $F$is the mutation factors that are independently generated according to uniform distribution in (0.1,1). 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. (10), it can be observed that the incorporation of the objective function value in the mutation scheme has two benefits. Firstly, 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. Consequently, the global solution can be easily reached if all vectors follow the same directions of the best vectors besides they also follow the opposite directions of the worst vectors among the randomly selected vectors. 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 with different weights [34]. Thus, it is considerably used to explore the landscape of the objective function being optimized in different sub-region 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.

4.2 Parameter adaptation schemes in AGDE

The successful performance of DE algorithm is significantly dependent upon the choice of its two control parameters: The scaling factor F and crossover rate CR [3, 16, 25]. In Fact, they play 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 general, F is an important parameter that controls the evolving rate of the population i.e. it is closely related to the convergence speed [26]. 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 [8]. On the other hand, A large F value improve the exploration capability of the algorithm that can make the mutant vectors distribute widely in the search space and can increase the diversity of the population [27]. However, it may slow down the search [8] with respect to the scaling factors in the proposed algorithm, at each generation G, the scale factors F, of each individual target vector is independently generated according to uniform distribution in (0.1,1) 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 [19, 34]. The constant crossover (CR) practically controls the diversity of the population [27]. 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. Additionally, CR is usually more sensitive to problems with different characteristics such as unimodality and multimodality, separable and non-separable problems. 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 [21]. 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 uses the previous experience (of generating better solutions as offspring) to guide the adaptation of these parameters [13, 19, 24–27, 29, 31, 32, 35, 36]. On the other hand, other studies make use of full information of the distribution conditions of each vector-individual in successive populations dynamically through generations to generate suitable control parameters [37–41]. The presented work is differing completely from all other previous studies in the following major aspects.

1. AGDE uses new adaptation schemes to update *CR* during generations without using extra parameters or determining specific learning period like [26, 37].
2. AGDE uses a pre-determined specific candidate pool for generating values for CR based on other researchers' studies. Thus, AGDE benefits from the previous experience and results of other researches by focusing on choosing the most appropriate values during evolution process from narrow set of possible values of CR.

Therefore, taking into consideration all the above mentioned guideline information about the major control parameters of DE, a novel adaptation scheme is used for *CR*, respectively.

Crossover adaptation At each generation G, the crossover probability *CR* of each individual target vector is independently generated according to one of the following two uniform distributions.

$$(1)\ CR_1 \in [0.05, 0.15]\ ;\ (2)\ CR_2 \in [0.9, 1]. \tag{11}$$

In fact, the lower and upper limits of the ranges for these proposed sets are recommended by [21, 27] to deal effectively with problems with different characteristics such as uni-modality and multimodality, separable and non-separable problems. Obviously, at each generation G, these two sets are adaptively selected based on their experiences of generating promising solutions during the evolution process as follows:

If G = 1

$$CR_i^1 = \begin{cases} CR_1\,, \text{ if } u(0,1) \leqslant 1/2 \\ CR_2\,, \text{otherwise.} \end{cases}$$

Else (12)

$$CR_i^G = \begin{cases} CR_1\,, \text{ if } u(0,1) \leqslant p_1 \\ CR_2\,, \text{ if } p_1 < u(0,1) \leqslant p_1 + p_2 \end{cases}$$

End

Denote $p_j, j = 1, 2, ..., m$ as the probability of selecting jth set, where m the total number of sets and it is set to 2 and the sum of p_j is 1. p_j is initialized as $1/j$, i.e. 1/2. The roulette wheel selection method is used to select the appropriate set for each target vector based on the probability p_j. p_j is continuously updated during generations in the following manner:

$$p_j^{G+1} = ((G-1) \times p_j^{G-1} + ps_j^G)/G \tag{13}$$

$$ps_j^G = \frac{s_j^G}{\sum_{j=1}^{m} s_j^G}, \tag{14}$$

where

$$s_j^G = \frac{ns_j^G}{\sum_{G=1}^{G} ns_j^G + \sum_{G=1}^{G} nf_j^G} + \varepsilon, \tag{15}$$

where G is the generation counter, ns_j^G and nf_j^G are the respective numbers of the offspring vectors generated by the jth set that survive or fail in the selection operation in the last G generations; s_j^G is the success ratio of the trial vector generated by the jth set and successfully entering the next generations; ps_j^G is the probability of selecting jth set in the current generation. ε is a small constant value to avoid the possible null success ratios. ε is set to 0.01 in our experiments to prevent s_j^G from becoming zero. It is clearly from the Eq. (14) that the probability p_j is the weighted mean of both the previous and current success ratios to consider the complete history of the experience of a specific set during generations. Moreover, in case of null success ratios of both sets the p_j is reinitialized as 1/2 to overcome the possible of stagnation. The core idea of the fitness-based adaptation scheme for the crossover rate is based on the following two fundamental principles. First, 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 randomly 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 decrease as the vectors in the population is clustered because each individual gets closer to the best vector found so far. Consequently, in this stage, in order to advance the diversity and increase the convergence speed, the value of *CR* must be a large value. Second, as previously aforementioned, these two sets are designed as recommended in [21, 27] to deal effectively with problems with different characteristics such as unimodality and multimodality, separable and non-separable problems. The pseudocode of AGDE is presented in Fig. 2.

Generally, 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 adaptation schemes for adaptive change of the values of the crossover rate can excellently benefit from the previous and current success ratios during the evolution process which in turn can considerably balance the common trade-off between the population diversity and convergence speed.

5 Numerical experiments and comparisons

In this section the computational results of AGDE are discussed along with comparisons with other state-of-the-art algorithms.

5.1 Experiments setup

The performance of the proposed AGDE algorithm was tested on 28 benchmark functions proposed in the CEC 2013

Fig. 2 Description of AGDE algorithm

Begin
  G=0
  Create a random initial population $\vec{x}_i^G\ \forall i, i=1,\dots,NP$.
  Evaluate $f(\vec{x}_i^G)\ \forall i, i=1,\dots,NP$
  For G=1 to Gmax **Do**
    For i=1 to NP **Do**
      Generate $F = \mathrm{rand}(0.1,1)$
      Compute the (crossover rate) Cr_i according to Eq. (13).
      Randomly choose $x_{p\_best}^G$ as one of the 100p% best vectors (top individuals).
      Randomly choose $x_{p\_worst}^G$ as one of the 100p% worst vectors (bottom individuals).
      Randomly choose x_r^G as one of the (NP-2(100p %)) vectors (middle individuals).
        j_{rand}= randint(1,D),
        For j=1 to D **Do**
          If ($rand_{i,j}[0,1] < CR$ **or** $j = j_{rand}$) **Then**
            $v_i^{G+1} = x_r^G + F \cdot (x_{p\_best}^G - x_{p\_worst}^G)$
          Else
            $u_{i,j}^G = x_{i,j}^G$
          End If
        End For
        If($f(\vec{u}_i^G) \le f(\vec{x}_i^G)$)**Then**
          $\vec{x}_i^{G+1} = \vec{u}_i^G, (f(\vec{x}_i^{G+1}) = f(\vec{u}_i^G))$
            If($f(\vec{u}_i^G) \le f(\vec{x}_{best}^G)$)**Then**
            $\vec{x}_{best}^{G+1} = \vec{u}_i^G, (f(\vec{x}_{best}^{G+1}) = f(\vec{u}_i^G))$
            $ns_j^G = ns_j^G + 1$
            End
        Else
            $\vec{x}_i^{G+1} = \vec{x}_i^G$
            $nf_j^G = nf_j^G + 1$
        End If
      End For
      Generate p_j^{G+1}, according to Eq.(14) for the next generation
    G=G+1
  End For
End

special session on real-parameter optimization. A detailed description of these test functions can be found in [17]. These 28 test functions can be divided into three classes:

1. unimodal functions f_1–f_5;
2. basic multimodal functions f_6–f_{20};
3. hybrid composition functions f_{21}–f_{28}.

5.2 Parameter settings and involved algorithms

To evaluate the performance of algorithms, experiments were conducted on the test suite. We adopt the solution error measure (f(x) − f(x*)), where x is the best solution obtained by algorithms in one run and x* is well-known global optimum of each benchmark function. Error values and standard deviations smaller than 10^{-8} are taken as zero [17]. The dimensions (D) of function are 10, 30 and 50, respectively. The maximum number of function evaluations (FEs), the terminal criteria, is set to 10,000 × D all experiments for each function and each algorithm run 51 times independently. The population size in AGDE was set to 50. 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. To perform comprehensive evaluation, the experimental results are compared with recent two DE-based algorithms and three recent non DE-based algorithms. Therefore, AGDE was compared with recent five state-of-the-art optimization algorithms, namely: The super-fit multicriteria adaptive differential evolution (SMADE) [28], covariance matrix adaptive evolution strategy (CMAES) [43], the self-adaptive Differential Evolution with $_p$BX cross over (MDE_$_p$BX) [31], and the Cooperatively Coevolving Particle Swarms Optimizers (CCPSO2) [42] that are listed in Caraffini et al.et al.in addition to Creatively-oriented optimization algorithm (COOA) [42]. SMADE [28] and MDE_$_p$BX [31] have been briefly introduced in Sect. 3. CMAES proposed by Hansen and Ostermeier [43], is an evolution strategy (ES) based on completely derandomized self-adaptation. CCPSO2, proposed by Li and Yao [44], is an improved version of particle swarm optimization (PSO). CCPSO2 adopts a new PSO position update rule that relies on Cauchy and Gaussian distributions

Table 1 Results of AGDE in 10D

Function	$f(x^*)$	Best	Median	Mean	Worst	SD
1	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	0	0.00E+00	0.00E+00	4.88E−06	2.39E−04	3.34E−05
3	0	0.00E+00	6.17E−05	1.60E−02	1.94E−01	4.15E−02
4	0	0.00E+00	2.11E−08	3.50E−03	1.13E−01	1.68E−02
5	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6	0	0.00E+00	0.00E+00	3.66E+00	9.81E+00	4.79E+00
7	0	9.72E−05	2.39E−03	8.03E−03	1.32E−01	2.06E−02
8	0	2.02E+01	2.04E+01	2.04E+01	2.05E+01	6.80E−02
9	0	0.00E+00	1.56E+00	1.96E+00	5.12E+00	1.64E+00
10	0	0.00E+00	4.35E−02	4.41E−02	9.85E−02	2.32E−02
11	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12	0	4.49E+00	9.18E+00	9.46E+00	1.53E+01	2.23E+00
13	0	1.56E+00	1.18E+01	1.12E+01	1.97E+01	4.18E+00
14	0	0.00E+00	0.00E+00	1.22E−02	6.25E−02	2.50E−02
15	0	3.07E+02	8.38E+02	8.59E+02	1.16E+03	1.66E+02
16	0	7.50E−01	1.06E+00	1.06E+00	1.39E+00	1.72E−01
17	0	1.01E+01	1.01E+01	1.03E+01	1.41E+01	7.17E−01
18	0	2.42E+01	2.91E+01	3.00E+01	3.61E+01	3.21E+00
19	0	1.31E−01	3.92E−01	3.90E−01	4.86E−01	6.40E−02
20	0	1.45E+00	2.41E+00	2.40E+00	3.21E+00	4.06E−01
21	0	0.00E+00	4.00E+02	3.77E+02	4.00E+02	7.64E+01
22	0	2.11E+01	4.33E+01	4.57E+01	1.18E+02	1.64E+01
23	0	4.56E+02	8.42E+02	8.27E+02	1.10E+03	1.62E+02
24	0	1.23E+02	2.00E+02	2.00E+02	2.10E+02	1.13E+01
25	0	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.77E−02
26	0	1.05E+02	1.19E+02	1.32E+02	2.00E+02	3.31E+01
27	0	3.00E+02	3.00E+02	3.00E+02	3.00E+02	2.05E−03
28	0	1.00E+02	3.00E+02	2.96E+02	3.00E+02	2.80E+01

to sample new points in the search space, and a scheme to dynamically determine the coevolving subcomponent sizes of the variables. Virtually, CMAES and CCPSO2 represent the state-of-the-art in ES and PSO, respectively. According to the Google Scholar Citation, as of 12 July 2016, the number of citations of CMAES and CCPSO2 is 812 and 244, respectively, and their performances are very competitive. On the other hand, COOA is a novel optimization algorithm inspired by the creative thinking process. At first, a creativity-oriented optimization model (COOM) inspired by the creative thinking process is constructed and simplified from a computer science perspective. Based on COOM, specific mathematical operations for COOM to solve the continuous function optimization problems are designed. Then, a creativity-oriented optimization algorithm COOA is presented. To compare and analyze the solution quality from a statistical angle of different algorithms and to check the behavior of the stochastic algorithms García et al. [45], the results are compared using two non-parametric statistical hypothesis tests: (1) the Friedman test (to obtain the final rankings of different algorithms for all functions) and (2) multi-problem Wilcoxon signed-rank test (to check the differences between all algorithms for all functions); at a 0.05 significance level Caraffini at al. [28]. All the p values in this paper were computed using SPSS (the version is 20.00).

5.3 Experimental results and discussions

5.3.1 Results of the proposed approach (AGDE)

The statistical results of the AGDE on the benchmarks with 10, 30, and 50 dimensions are summarized in Tables 1, 2 and 3, respectively. It includes the known optimal solution for each test problem and the obtained best, median, mean, worst values and the standard deviations of error from optimum solution of the proposed AGDE over 51 runs for all 28 benchmark functions. Generally, from Tables 1, 2 and 3, it can be clearly seen that AGDE succeeded at solving, at least once, ten problems in 10D, seven problems in 30D, and three problems in 50D. Regarding Unimodal functions (f_1–f_5), all the unimodal functions are solved in 10 dimensions. In 30 dimensions, the optimum is detected in three cases, while

Table 2 Results of AGDE in 30D

Function	$f(x^*)$	BEST	MEDIAN	MEAN	WORST	SD
1	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	0	3.79E+03	2.28E+04	2.64E+04	5.99E+04	1.43E+04
3	0	0.00E+00	1.85E+00	3.07E+05	5.82E+06	1.05E+06
4	0	7.65E−02	1.35E+00	2.32E+00	2.40E+01	3.59E+00
5	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6	0	0.00E+00	1.18E−04	1.55E+00	2.64E+01	6.28E+00
7	0	4.27E−02	2.27E+00	3.98E+00	1.78E+01	4.22E+00
8	0	2.08E+01	2.09E+01	2.09E+01	2.10E+01	4.67E−02
9	0	7.56E+00	2.73E+01	2.54E+01	3.04E+01	5.64E+00
10	0	0.00E+00	2.71E−02	3.11E−02	1.20E−01	2.27E−02
11	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12	0	3.38E+01	9.75E+01	9.45E+01	1.43E+02	2.29E+01
13	0	3.88E+01	1.12E+02	1.10E+02	1.54E+02	2.19E+01
14	0	0.00E+00	4.16E−02	5.47E−02	1.25E−01	3.25E−02
15	0	4.85E+03	6.00E+03	5.99E+03	6.66E+03	4.03E+02
16	0	1.51E+00	2.35E+00	2.32E+00	2.74E+00	2.61E−01
17	0	3.04E+01	3.04E+01	3.04E+01	3.04E+01	2.12E−03
18	0	1.48E+02	1.84E+02	1.84E+02	2.12E+02	1.40E+01
19	0	2.46E+00	3.12E+00	3.07E+00	3.53E+00	2.60E−01
20	0	1.06E+01	1.15E+01	1.15E+01	1.22E+01	3.72E−01
21	0	2.00E+02	3.00E+02	3.00E+02	4.44E+02	7.94E+01
22	0	2.41E+02	6.17E+02	6.12E+02	8.26E+02	1.07E+02
23	0	4.96E+03	6.24E+03	6.14E+03	6.77E+03	4.58E+02
24	0	2.00E+02	2.05E+02	2.08E+02	2.40E+02	8.57E+00
25	0	2.45E+02	2.81E+02	2.76E+02	2.94E+02	1.43E+01
26	0	2.00E+02	2.00E+02	2.05E+02	3.56E+02	2.59E+01
27	0	3.20E+02	5.12E+02	5.53E+02	1.03E+03	2.04E+02
28	0	3.00E+02	3.00E+02	3.00E+02	3.00E+02	0.00E+00

Table 3 Results of AGDE in 50D

Function	$f(x^*)$	BEST	MEDIAN	MEAN	WORST	SD
1	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	0	3.64E+04	1.19E+05	1.27E+05	3.15E+05	6.11E+04
3	0	1.87E+01	7.94E+05	3.22E+06	3.93E+07	7.74E+06
4	0	4.05E−01	3.07E+00	4.40E+00	2.52E+01	4.25E+00
5	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6	0	4.34E+01	4.34E+01	4.34E+01	4.34E+01	0.00E+00
7	0	4.52E+00	1.86E+01	1.86E+01	3.77E+01	8.43E+00
8	0	2.11E+01	2.11E+01	2.11E+01	2.12E+01	3.26E−02
9	0	1.96E+01	5.52E+01	5.06E+01	5.96E+01	1.18E+01
10	0	1.97E−02	5.66E−02	6.12E−02	1.36E−01	2.99E−02
11	0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12	0	4.97E+01	2.16E+02	1.96E+02	2.98E+02	6.58E+01
13	0	8.45E+01	2.70E+02	2.62E+02	3.39E+02	4.98E+01
14	0	2.62E+02	3.86E+02	3.81E+02	5.28E+02	6.21E+01
15	0	1.17E+04	1.29E+04	1.29E+04	1.38E+04	3.81E+02
16	0	1.89E+00	3.27E+00	3.20E+00	3.76E+00	3.78E−01
17	0	6.81E+01	7.41E+01	7.39E+01	7.75E+01	2.07E+00
18	0	3.42E+02	3.83E+02	3.82E+02	4.16E+02	1.56E+01
19	0	7.21E+00	8.51E+00	8.53E+00	1.01E+01	6.01E−01
20	0	2.08E+01	2.16E+01	2.16E+01	2.22E+01	2.65E−01
21	0	2.00E+02	2.00E+02	5.89E+02	1.12E+03	4.27E+02
22	0	1.93E+03	2.45E+03	2.48E+03	3.05E+03	2.42E+02
23	0	1.11E+04	1.33E+04	1.31E+04	1.39E+04	5.37E+02
24	0	2.04E+02	2.41E+02	2.39E+02	2.64E+02	1.32E+01
25	0	2.83E+02	3.67E+02	3.60E+02	3.82E+02	2.41E+01
26	0	2.00E+02	2.00E+02	2.91E+02	4.43E+02	9.68E+01
27	0	6.70E+02	9.59E+02	1.10E+03	1.75E+03	3.39E+02
28	0	4.00E+02	4.00E+02	4.58E+02	3.34E+03	4.11E+02

the mean error ranges from 0.00E+00 to 3.07E+05 and the standard deviation ranges from 0.00E+00 to 1.05E+06. In 50 dimensions, the optimum is detected in two cases, while the mean error ranges from 0.00E+00 to 3.22E+06 and the standard deviation ranges from 0.00E+00 to 7.74E+06. As for the multimodal functions (f_6–f_{20}), in ten dimensions the optimum is detected in four cases, while the mean error ranges from 0.00E+00 to 8.59E+02 and the standard deviation ranges from 0.00E+00 to 1.66E+02. In 30 dimensions, the optimum is detected in four cases, while the mean error ranges from 0.00E+00 to 5.99E+03and the standard deviation ranges from 0.00E+00 to 4.03E+02. In 50 dimensions, the optimum is detected in one case, while the mean error ranges from 0.00E+00 to 1.29E+04 and the standard deviation ranges from 0.00E+00 to 3.81E+02. Finally, regarding the composition functions (f_{21}–f_{28}), the optimum is detected in one case, while the mean error ranges from 4.57E+01 to 8.27E+02, from 2.05E+02 to 6.14E+03, from 2.39E+02 to 1.31E+04, while the standard deviation ranges from 2.05E−03 to 1.88E+02 ,from 0.00E+00 to 4.58E+02, from 1.32E+01 to 5.37E+02, in 10, 30 and 50 dimensions, respectively. Therefore, it can be concluded that, apart from f_2 (in 30D) and f_2, f_3 (in 50D), AGDE is able to reach the global optimum in case of the unimodal functions. Regarding the multimodal functions, AGDE reach results which show small deviation from the global optimum and got very close to the optimum in all functions in 50D. On the other hand, regarding composition functions, apart from f22 (in 50D) and f23 (in 30D and 50D), AGDE got very close to the optimum in all functions for all the three dimensionalities. Additionally, in all functions for all the three dimensionalities, 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 the AGDE is a robust algorithm. Finally, due to insignificant difference between the results in three dimensions, it can be concluded that the performance of the AGDE 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 increases.

Table 4 Comparison between AGDE and various state-of-the-art methods on 10D problems

Function	COOA	SMADE	MDE-$_p$BX	CMAES	CCPSO2	AGDE
1	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	1.92E−04 ± 1.16E−03	**0.00E+00 ± 0.00E+00**
2	6.48E+03 ± 5.06E+03	**0.00E+00 ± 0.00E+00**	4.15E+02 ± 9.60E+02	**0.00E+00 ± 0.00E+00**	9.93E+05 ± 7.58E+05	4.88E−06 ± 3.34E−05
3	4.59E+04 ± 6.50E+04	2.48E−01 ± 1.23E+00	4.96E+03 ± 4.66E+04	5.69E−01 ± 1.81E+00	2.13E+07 ± 3.13E+07	**1.60E−02 ± 4.15E−02**
4	1.33E+01 ± 2.14E+01	**0.00E+00 ± 0.00E+00**	6.50E−02 ± 6.38E−01	**0.00E+00 ± 0.00E+00**	8.80E+03 ± 2.50E+03	3.50E−03 ± 1.68E−02
5	1.24E−04 ± 6.91E−05	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	2.94E−03 ± 8.06E−03	**0.00E+00 ± 0.00E+00**
6	**1.48E+00 ± 3.49E+00**	5.41E+00 ± 4.76E+00	6.18E+00 ± 4.73E+00	6.74E+00 ± 6.74E+00	1.52E+00 ± 2.91E+00	3.66E+00 ± 4.79E+00
7	1.07E+01 ± 1.11E+00	2.27E+00 ± 4.45E+00	5.63E+00 ± 7.94E+00	5.45E+08 ± 5.38E+09	3.27E+01 ± 8.31E+00	**8.03E−03 ± 2.06E−02**
8	**2.03E+01 ± 9.30E−02**	**2.03E+01 ± 1.03E−01**	2.05E+01 ± 1.06E−01	**2.03E+01 ± 1.32E−01**	2.04E+01 ± 7.66E−02	**2.03E+01 ± 6.80E−02**
9	**1.53E+00 ± 9.67E−01**	2.29E+00 ± 7.19E−01	2.37E+00 ± 1.41E+00	1.50E+01 ± 3.65E00	4.98E+00 ± 9.13E−01	1.96E+00 ± 1.64E+00
10	1.37E−02 ± 3.00E−02	1.42E−02 ± 9.58E−03	1.25E−01 ± 9.20E−02	**1.33E−02 ± 1.39E02**	1.47E+00 ± 6.44E−01	4.41E−02 ± 2.32E−02
11	1.36E+00 ± 5.21E+00	9.75E−02 ± 2.96E−01	2.48E+00 ± 1.61E+00	2.31E+02 ± 2.67E02	1.97E+00 ± 1.18E+00	**0.00E+00 ± 0.00E+00**
12	1.17E+01 ± 5.47E+00	**7.80E+00 ± 4.10E+00**	1.06E+01 ± 4.76E+00	3.49E+02 ± 3.81E02	2.64E+01 ± 7.93E+00	9.46E+00 ± 2.23E+00
13	1.19E+01 ± 8.61E+00	1.21E+01 ± 6.40E+00	2.01E+01 ± 7.94 + 00	2.94E+02 ± 3.99E+02	3.56E+01 ± 8.30E+00	**1.12E+01 ± 4.18E+00**
14	4.19E+01 ± 2.92E+01	3.64E+00 ± 4.39E+00	1.25E+02 ± 1.12E+02	1.88E+03 ± 4.25E+02	5.27E+01 ± 3.86E+01	**1.22E−02 ± 2.50E−02**
15	**4.04E+02 ± 2.76E+02**	7.36E+02 ± 2.60E+02	7.26E+02 ± 2.61E+02	1.80E+03 ± 3.92E+02	8.92E+02 ± 2.18E+02	8.59E+02 ± 1.66E+02
16	**4.76E−02 ± 4.14E−02**	4.04E−01 ± 3.14E−01	5.43E−01 ± 4.49E−01	4.51E−01 ± 5.06E−01	1.18E+00 ± 2.19E−01	1.06E+00 ± 1.72E−01
17	2.07E+01 ± 4.87E+00	**1.03E+01 ± 1.55E−01**	1.32E+01 ± 1.85E+00	9.55E+02 ± 3.42E+02	1.60E+01 ± 2.07E+00	**1.03E+01 ± 7.17E−01**
18	**1.48E+01 ± 6.53E+00**	2.46E+01 ± 4.68E+00	1.93E+01 ± 4.67E+00	9.01E+02 ± 3.10E+02	5.41E+01 ± 6.33E+00	3.00E+01 ± 3.21E+00
19	6.97E−01 ± 2.43E−01	3.95E−01 ± 1.25E−01	6.44E−01 ± 2.09E−01	1.19E+00 ± 5.00E−01	7.65E−01 ± 2.52E−01	**3.90E−01 ± 6.40E−02**
20	**2.33E+00 ± 5.92E−01**	2.65E+00 ± 4.48E−01	2.87E+00 ± 5.25E−01	4.68E+00 ± 3.79E−01	3.50E+00 ± 1.96E−01	2.40E+00 ± 4.06E−01
21	**2.90E+02 ± 1.22E+02**	3.83E+02 ± 5.50E+01	3.98E+02 ± 1.99E+01	3.68E+02 ± 8.71E+01	3.73E+02 ± 6.75E+01	3.77E+02 ± 7.64E+01
22	4.57E+02 ± 2.74E+02	4.93E+01 ± 5.33E+01	1.33E+02 ± 1.04E+02	2.32E+03 ± 4.25E+02	7.27E+01 ± 4.98E+01	**4.57E+01 ± 1.64E+01**
23	**5.11E+02 ± 2.89E+02**	5.78E+02 ± 3.16E+02	8.82E+02 ± 3.08E+02	2.25E+03 ± 4.48E+02	1.15E+03 ± 2.58E+02	8.27E+02 ± 1.62E+02
24	**1.38E+02 ± 3.07E+01**	2.02E+02 ± 1.76E+01	2.02E+02 ± 1.56E+01	3.83E+02 ± 1.57E+02	2.01E+02 ± 2.54E+01	2.00E+02 ± 1.13E+01
25	**1.94E+02 ± 3.40E+01**	2.02E+02 ± 1.91E+00	2.00E+02 ± 1.23E+01	2.62E+02 ± 4.92E+01	2.12E+02 ± 1.09E+01	2.00E+02 ± 2.77E−02
26	**1.26E+02 ± 2.25E+01**	**1.26E+02 ± 3.69E+01**	1.47E+02 ± 4.36E+01	2.57E+02 ± 1.13E+02	1.58E+02 ± 2.40E+01	1.32E+02 ± 3.31E+01
27	3.14E+02 ± 6.11E+01	3.37E+02 ± 5.23E+01	3.06E+02 ± 2.76E+01	4.23E+02 ± 1.35E+02	4.27E+02 ± 4.91E+01	**3.00E+02 ± 2.05E−03**
28	**2.00E+02 ± 1.02E+02**	3.17E+02 ± 6.87E+01	3.07E+02 ± 5.78E+01	1.21E+03 ± 1.21E+03	3.01E+02 ± 1.28E+02	2.96E+02 ± 2.80E+01

5.3.2 Comparison with state-of-the-art algorithms

The statistical results of the comparisons on the benchmarks with 10, 30, and 50 dimensions are summarized in Tables 4, 5 and 6, respectively. It includes the obtained best and the standard deviations of error from optimum solution of AGDE and other five state-of-the-art algorithms over 51 runs for all 28 benchmark functions. The results provided by these approaches were directly taken from Feng et al. [42]. The best results are marked in bold for all problems. Furthermore, Figs. 3, 4 and 5 depict the Friedman ranks of the five algorithms on each of the 28 functions with 10D, 30D and 50D, respectively. A smaller rank indicates a better performance of the corresponding algorithm on that problem. Each figure consists of three sub-figures, corresponding to the three categories of function in the 28 CEC-2013 benchmark functions: uni-modal functions (f_1–f_5), multi-modal functions (f_6–f_{20}) and composition functions (f_{21}–f_{28}).

Firstly, the performance of AGDE and other competitive algorithms on the functions of different dimensions is discussed. Table 7 lists the average ranks of the five algorithms for D = 10, 30 and 50, respectively. The best ranks are marked in bold and the second ranks are underlined. It can be clearly seen from Table 7 that, AGDE get the first ranking among the five algorithms in 10-dimensional functions, followed by COOA, SMADE, MDE-$_p$BX, CMAES and CCPSO2. Regarding 30D, AGDE and SMADE obtain the best ranking, followed by COOA, MDE-$_p$BX, CCPSO2 and CMAES. Regarding 50D, COOA obtains the best ranking, followed by SMADE, AGDE, MDE-$_p$BX, CCPSO2 and CMAES. Additionally, Average ranks for all algorithms across all problems and all dimensions are also presented. It can be clearly concluded that COOA and AGDE obtain almost the same best ranking, followed by SMADE, MDE-$_p$BX, CCPSO2 and CMAES. Table 8 summarizes the statistical analysis results of applying Wilcoxon's test between AGDE and other five compared algorithms on the functions with different dimensions. R^+ represents the sum ranks for the problems in which AGDE outperformed the comparison algorithm. The p values under the significance level are shown in bold. From the results shown in Table 8, we can see that AGDE provides higher R^+ values than R^- in all the cases with exception to SMADE in 30D and 50D cases and COOA in 50D case only. Precisely, we can draw

Table 5 Comparison between AGDE and various state-of-the-art methods on 30D problems

Function	COOA	SMADE	MDE-pBX	CMAES	CCPSO2	AGDE
1	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
2	1.03E+05 ± 1.68E+05	**0.00E+00 ± 0.00E+00**	9.56E+04 ± 6.16E+04	**0.00E+00 ± 0.00E+00**	9.95E+05 ± 5.24E+05	2.64E+04 ± 1.43E+04
3	7.84E+06 ± 8.38E+06	9.82E+03 ± 4.94E+04	1.80E+07 ± 3.12E+07	**1.53E+01 ± 1.23E+02**	5.59E+08 ± 5.57E+08	3.07E+05 ± 1.05E+06
4	6.67E+02 ± 1.24E+03	**0.00E+00 ± 0.00E+00**	1.32E+01 ± 5.46E+01	**0.00E+00 ± 0.00E+00**	5.57E+04 ± 2.06E+04	2.32E+00 ± 3.59E+00
5	6.67E−04 ± 2.57E−04	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	2.62E−08 ± 6.05E−08	**0.00E+00 ± 0.00E+00**
6	1.31E+01 ± 1.19E+01	2.67E+00 ± 7.85E+00	1.99E+01 ± 2.22E+01	3.05E+00 ± 8.33E+00	2.19E+01 ± 2.27E+01	**1.55E+00 ± 6.28E+00**
7	4.19E+01 ± 1.43E+01	3.25E+01 ± 1.61E+01	5.70E+01 ± 1.77E+01	9.83E+03 ± 6.44E+04	1.15E+02 ± 3.11E+01	**3.98E+00 ± 4.22E+00**
8	**2.09E+01 ± 1.04E−03**	2.10E+01 ± 4.80E−02	2.11E+01 ± 5.94E−02	**2.09E+01 ± 6.66E−02**	2.10E+01 ± 4.60E−02	**2.09E+01 ± 4.67E−02**
9	**1.60E+01 ± 1.88E+00**	2.23E+01 ± 3.57E+00	2.22E+01 ± 4.80E+00	4.45E+01 ± 6.99E+00	2.84E+01 ± 2.08E+00	2.54E+01 ± 5.64E+00
10	4.56E−02 ± 6.08E−02	1.84E−02 ± 1.34E−02	1.64E−01 ± 1.20E−01	**1.73E−02 ± 1.25E−02**	1.48E−01 ± 6.90E−02	3.11E−02 ± 2.27E−02
11	1.01E+01 ± 2.14E+01	1.09E+01 ± 4.18E+00	4.62E+01 ± 1.44E+01	9.47E+01 ± 2.36E+02	1.19E−01 ± 2.90E−01	**0.00E+00 ± 0.00E+00**
12	**5.36E+01 ± 1.91E+01**	5.72E+01 ± 1.70E+01	6.94E+01 ± 2.00E+01	7.11E+02 ± 9.66E+02	2.12E+02 ± 5.24E+01	9.45E+01 ± 2.29E+01
13	1.13E+02 ± 3.46E+01	1.28E+02 ± 3.50E+01	1.49E+02 ± 3.66E+01	1.64E+03 ± 1.66E+03	2.44E+02 ± 3.44E+01	**1.10E+02 ± 2.19E+01**
14	2.03E+03 ± 5.62E+02	1.33E+02 ± 1.27E+02	1.17E+03 ± 3.95E+02	5.21E+03 ± 7.37E+02	4.48E+00 ± 2.87E+00	**5.47E−02 ± 3.25E−02**
15	**2.65E+03 ± 4.90E+02**	4.10E+03 ± 8.47E+02	3.95E+03 ± 6.57E+02	5.37E+03 ± 6.73E+02	3.85E+03 ± 4.52E+02	5.99E+03 ± 4.03E+02
16	1.29E−01 ± 1.22E−01	1.31E−01 ± 7.57E−02	1.25E+00 ± 6.19E−01	**1.26E−01 ± 8.87E−02**	2.16E+00 ± 3.76E−01	2.14E+00 ± 2.61E−01
17	1.37E+2 ± 1.95E+01	3.48E+01 ± 1.52E+00	7.05E+01 ± 1.24E+01	3.95E+03 ± 7.89E+02	3.07E+01 ± 3.03E+00	**3.04E+01 ± 2.12E−03**
18	1.15E+02 ± 2.91E+01	8.33E+01 ± 2.06E+01	**8.26E+01 ± 1.89E+01**	4.23E+03 ± 8.31E+02	2.31E+02 ± 5.43E+01	1.84E+02 ± 1.40E+01
19	3.93E+00 ± 1.25E+00	2.55E+00 ± 5.18E−01	9.54E+00 ± 5.54E+00	3.66E+00 ± 9.52E−01	**7.77E−01 ± 1.58E−01**	3.07E+00 ± 2.60E−01
20	1.06E+01 ± 1.77E−01	**1.05E+01 ± 8.07E−01**	1.07E+01 ± 7.75E−01	1.50E+01 ± 6.45E−02	1.35E+01 ± 5.50E−01	1.15E+01 ± 3.72E−01
21	2.92E+02 ± 7.30E+01	3.27E+02 ± 8.65E+01	3.40E+02 ± 7.62E+01	3.05E+02 ± 9.01E+01	**2.37E+02 ± 6.71E+01**	3.00E+02 ± 7.94E+01
22	3.05E+03 ± 3.11E+02	1.79E+02 ± 4.50E+01	1.17E+03 ± 4.92E+02	6.97E+03 ± 1.06E+03	**9.87E+01 ± 6.70E+01**	6.12E+02 ± 1.07E+02
23	**2.96E+03 ± 5.41E+02**	4.22E+03 ± 8.74E+02	4.70E+03 ± 7.70E+02	6.76E+03 ± 6.82E+02	4.99E+03 ± 6.31E+02	6.14E+03 ± 4.58E+02
24	**2.02E+02 ± 7.10E+00**	2.32E+02 ± 2.57E+01	2.31E+02 ± 8.60E+00	8.19E+02 ± 6.15E+02	2.80E+02 ± 6.34E+00	2.08E+02 ± 8.57E+00
25	**2.40E+02 ± 6.46E+01**	2.78E+02 ± 9.90E+00	2.79E+02 ± 1.38E+01	3.46E+02 ± 1.45E+02	2.98E+02 ± 6.94E+00	2.76E+02 ± 1.43E+01
26	**2.00E+02 ± 5.44E−02**	2.15E+02 ± 5.25E+01	2.26E+02 ± 5.15E+01	5.51E+02 ± 5.14E+02	**2.00E+02 ± 6.76E−01**	2.05E+02 ± 2.59E+01
27	**3.74E+02 ± 5.91E+01**	6.47E+02 ± 1.37E+02	6.50E+02 ± 1.04E+02	8.53E+02 ± 2.42E+02	1.04E+03 ± 8.09E+01	5.53E+02 ± 2.04E+02
28	**300E+02 ± 2.02E−05**	3.88E+02 ± 3.23E+02	3.09E+02 ± 1.50E+02	1.96E+03 ± 3.40E+03	4.35E+02 ± 5.10E+02	**3.00E+02 ± 0.00E+00**

the following conclusions: AGDE outperforms MDE-pBX, CMAES and CCPSO2 significantly, in all dimensions, respectively, and it is comparable with other algorithms in 50D. Furthermore, the average performance of the five algorithms on the benchmark functions over all dimensions is analyzed. The average Friedman ranks per function over all dimensions is depicted in Table 9, and the best results are shown in bold. The *p* value computed through Friedman test is 0.00E+00. Thus, it can be concluded that there is a significant difference between the performances of the algorithms. The performance of the six algorithms can be sorted by the average ranks into the following order: COOA, AGDE, SMADE, CCPSO2, MDE-pBX, and CMAES. Additionally, due to the importance of the multiple-problem statistical analysis, Table 10 summarizes the statistical analysis results of applying Wilcoxon's test between AGDE and other five compared algorithms. From the results shown in Table 10, we can see that AGDE provides higher R^+ values than R^- in all the cases. Precisely, we can draw the following conclusions: AGDE outperforms MDE-pBX, CCPSO2 and CMAES significantly, and it is comparable with COOA and SMADE. Finally, in order to analyze the performance of all algorithms on different categories of functions (uni-modal functions, multi-modal functions and composition functions). Using the results listed in Table 9, the average Friedman ranks on different categories of functions are shown in Table 11. The results show that AGDE performs differently on different categories of functions although AGDE is in the second place on the uni-modal, multimodal and composition functions. In fact, it can be obviously seen from Table 9 that, AGDE is first and second in 2 and 2 out of 5 unimodal functions, 7 and 1 out of 15 multimodal functions and 1 and 5 out of 8 composition functions, respectively. Taking into consideration that these test functions are much harder than others since each of them is composed of many sub-functions. Overall, AGDE performs second best on different categories of functions. Table 12 summarizes the statistical analysis results of applying Wilcoxon's test between AGDE and other five compared algorithms on different categories of functions. From the results shown in Table 12, we can see that AGDE provides higher R^+ values than R^- in all the cases with exception to SMADE and CMEAS in ten uni-modal functions, COOA and SMADE in multimodal functions and COOA in composition functions, respectively. Precisely, we can draw the following conclusions: AGDE performs significantly CMAES on both multi-modal and

Table 6 Comparison between AGDE and various state-of-the-art methods on 50D problems

Function	COOA	SMADE	MDE-$_p$BX	CMAES	CCPSO2	AGDE
1	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
2	3.85E+05 ± 1.34E+05	**0.00E+00 ± 0.00E+00**	4.37E+05 ± 1.64E+05	**0.00E+00 ± 0.00E+00**	1.85E+06 ± 9.33E+05	1.27E+05 ± 6.11E+04
3	4.31E+07 ± 2.84E+07	3.81E+05 ± 1.35E+06	8.45E+07 ± 1.46E+08	**9.64E+02 ± 4.49E+03**	1.98E+09 ± 2.09E+09	3.22E+06 ± 7.74E+06
4	3.96E+02 ± 3.20E+02	**0.00E+00 ± 0.00E+00**	3.05E+01 ± 6.62E+01	**0.00E+00 ± 0.00E+00**	1.00E+05 ± 3.57E+04	4.40E+00 ± 4.25E+00
5	8.72E−04 ± 3.73E−04	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
6	4.34E+01 ± 1.74E+01	**4.30E+01 ± 6.28E+00**	5.48E+01 ± 2.12E+01	4.32E+01 ± 7.10E+00	4.35E+01 ± 1.36E+01	4.34E+01 ± 0.00E+00
7	7.60E+01 ± 9.45E+00	4.32E+01 ± 1.66E+01	6.59E+01 ± 1.06E+01	4.19E+01 ± 1.70E+01	1.37E+02 ± 2.31E+01	**1.86E+01 ± 8.43E+00**
8	**2.10E+01 ± 5.46E−02**	2.11E+01 ± 3.85E−02	2.12E+01 ± 4.44E−02	2.11E+01 ± 9.75E−02	2.11E+01 ± 4.49E−02	**2.10E+01 ± 3.26E−02**
9	**3.45E+01 ± 2.65E+00**	4.36E+01 ± 4.06E+00	4.32E+01 ± 7.71E+00	7.66E+01 ± 7.77E+00	5.79E+01 ± 4.39E+00	5.06E+01 ± 1.18E+01
10	2.32E−02 ± 7.14E−02	2.47E−02 ± 1.48E−02	1.34E−01 ± 1.23E−01	**2.24E−02 ± 1.49E−02**	1.24E−01 ± 4.62E−02	6.12E−02 ± 2.99E−02
11	2.03E+01 ± 3.89E+01	4.81E+01 ± 1.49E+01	1.24E+02 ± 2.87E+01	2.19E+02 ± 4.56E+02	4.31E−01 ± 5.74E−01	**0.00E+00 ± 0.00E+00**
12	**1.30E+02 ± 3.57E+01**	1.57E+02 ± 4.52E+01	1.58E+02 ± 3.25E+01	2.25E+03 ± 1.37E+03	4.46E+02 ± 7.92E+01	1.96E+02 ± 6.58E+01
13	3.04E+02 ± 3.92E+01	3.35E+02 ± 5.63E+01	3.24E+02 ± 4.74E+01	3.36E+03 ± 1.09E+03	5.49E+02 ± 6.67E+01	**2.62E+02 ± 4.98E+01**
14	5.07E+03 ± 6.96E+02	3.41E+02 ± 2.05E+02	2.65E+03 ± 8.86E+02	8.82E+03 ± 1.04E+03	**6.45E+00 ± 3.20E+00**	3.81E+02 ± 6.21E+01
15	**5.72E+03 ± 5.81E+02**	8.54E+03 ± 9.77E+02	7.46E+03 ± 7.95E+02	9.09E+03 ± 9.43E+02	7.95E+03 ± 7.11E+02	1.29E+04 ± 3.81E+02
16	8.72E−02 ± 1.77E−01	8.96E−02 ± 4.24E−02	1.75E+00 ± 7.40E−01	**8.01E−02 ± 4.72E−02**	2.39E+00 ± 5.90E−01	3.20E+00 ± 3.78E−01
17	2.27E+02 ± 4.35E+01	6.57E+01 ± 5.27E+00	1.75E+02 ± 3.72E+01	6.97E+03 ± 1.07E+03	**5.14E+01 ± 2.84E−01**	7.39E+01 ± 2.07E+00
18	2.36E+02 ± 4.76E+01	1.93E+02 ± 3.46E+01	**1.85E+02 ± 3.40E+01**	7.08E+03 ± 9.14E+02	4.94E+02 ± 1.08E+02	3.82E+02 ± 1.56E+01
19	9.67E+00 ± 3.45E+00	5.43E+00 ± 1.07E+00	4.25E+01 ± 2.66E+01	6.32E+00 ± 1.18E+00	**1.40E+00 ± 2.19E−01**	8.53E+00 ± 6.01E−01
20	2.08E+01 ± 2.28E−01	**1.92E+01 ± 8.86E−01**	2.00E+01 ± 9.04E−01	2.50E+01 ± 9.73E−02	2.28E+01 ± 7.85E−01	2.16E+01 ± 2.65E−01
21	**3.16E+02 ± 3.23E+02**	8.46E+02 ± 3.43E+02	9.22E+02 ± 3.06E+02	8.12E+02 ± 3.73E+02	3.27E+02 ± 2.64E+02	5.89E+02 ± 4.27E+02
22	6.03E+03 ± 1.08E+03	3.39E+02 ± 2.24E+02	3.09E+03 ± 9.98E+02	1.19E+04 ± 1.26E+03	**7.58E+01 ± 8.58E+01**	2.48E+03 ± 2.42E+02
23	**4.75E+03 ± 7.78E+02**	9.89E+03 ± 1.90E+03	8.88E+03 ± 1.20E+03	1.18E+04 ± 8.52E+02	1.05E+04 ± 1.11E+03	1.31E+04 ± 5.37E+02
24	**2.02E+02 ± 9.46E+00**	3.00E+02 ± 1.20E+01	2.87E+02 ± 1.47E+01	1.64E+03 ± 1.05E+03	3.56E+02 ± 9.89E+00	2.39E+02 ± 1.32E+01
25	**2.75E+02 ± 1.01E+01**	3.68E+02 ± 1.36E+01	3.69E+02 ± 1.78E+01	4.94E+02 ± 1.88E+02	3.96E+02 ± 1.19E+01	3.60E+02 ± 2.41E+01
26	2.22E+02 ± 5.10E+01	2.91E+02 ± 9.70E+01	3.50E+02 ± 7.93E+01	6.04E+02 ± 7.11E+02	**2.09E+02 ± 3.92E+01**	2.91E+02 ± 9.68E+01
27	**7.75E+02 ± 5.47E+01**	1.18E+03 ± 1.67E+02	1.24E+03 ± 1.56E+02	1.28E+03 ± 2.51E+02	1.79E+03 ± 8.78E+01	1.10E+03 ± 3.39E+02
28	**4.00E+02 ± 2.65E+02**	1.07E+03 ± 1.27E+03	4.33E+02 ± 3.26E+02	3.27E+03 ± 5.60E+03	6.33E+02 ± 8.95E+02	4.58E+02 ± 4.11E+02

composition functions and CCPSO2 on both nui-modal and multimodal functions. In other cases, there do not exist significance under the current significance level of 0.05.

From the above results, comparisons and discussion, the proposed AGDE algorithm is of better searching quality, efficiency and robustness for solving unconstrained global optimization problems. It is clear that the proposed AGDE algorithm performs well and it has shown its outstanding superiority with separable, non-separable, unimodal and multimodal 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, its performance is superior and competitive with the performance of the-state-of-the-art well-known algorithms. Moreover, compared to the complicated structures and number of methods and number of control parameters used in other algorithms such as SMADE that uses four mutation Strategies combined with well-known CMAES optimizer with adaptive crossover and scaling factors and COOA with sophisticated mathematical model and many parameters, we can see that AGDE is very simple and easy to be implemented and programmed in many programming languages. It only uses very simple adaptive crossover rate without extra parameters and novel one mutation rule with one parameter. Thus, it neither increases the complexity of the original DE algorithm nor the number of control parameters.

Finally, it is clearly visible that the proposed modifications play a vital role and has a significant impact on improving the convergence speed of AGDE algorithm for most problems. The AGDE algorithm has a considerable ability to maintain its convergence rate, improve its diversity as well as advance its local tendency through a search process. Accordingly, the main benefits of the proposed modifications are the remarkable balance between the exploration capability and exploitation tendency through the optimization process. This balance leads to superior performance with fast convergence speed and the extreme robustness over the entire range of benchmark functions which are the weak points of all evolutionary algorithms.

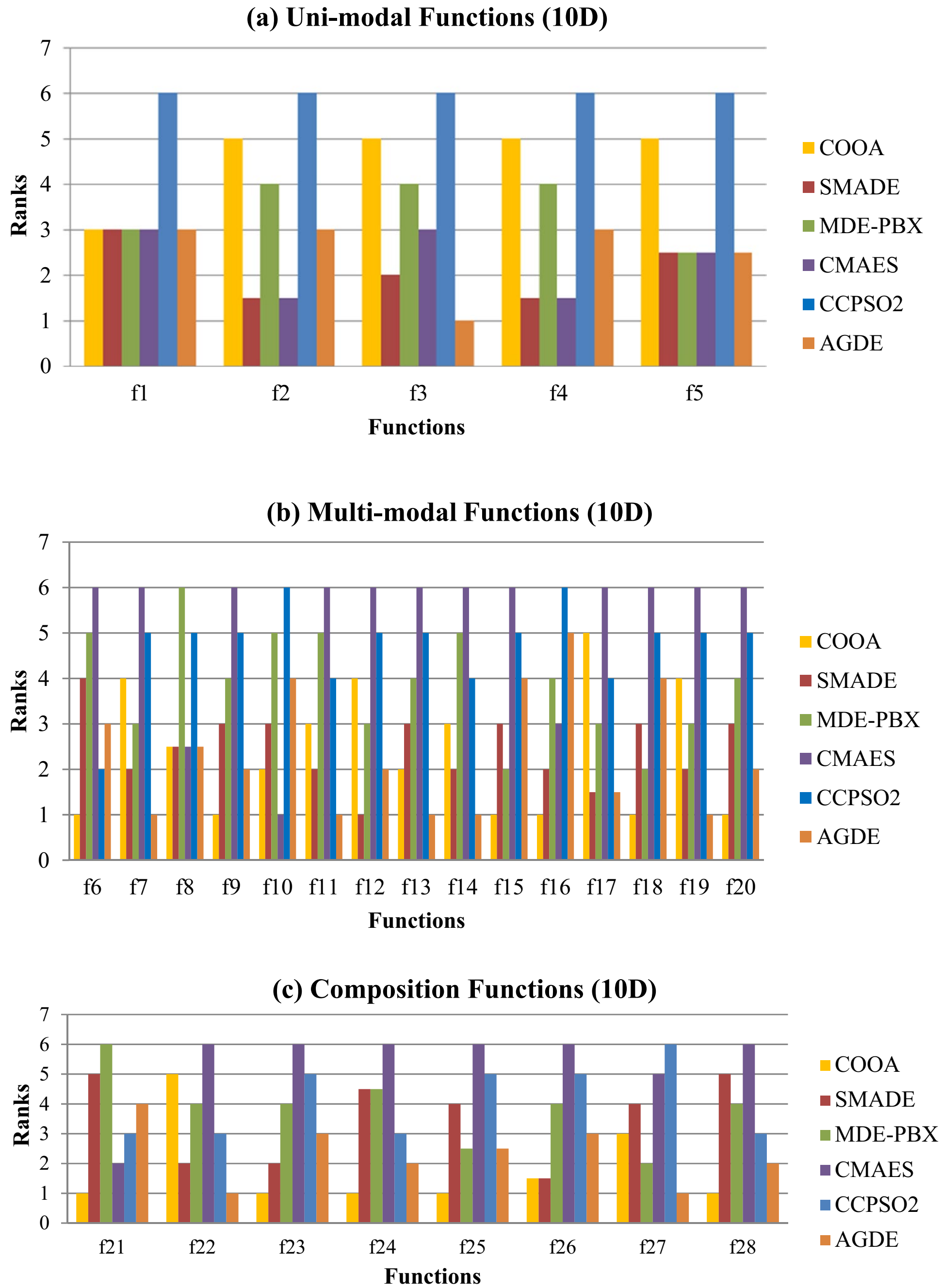
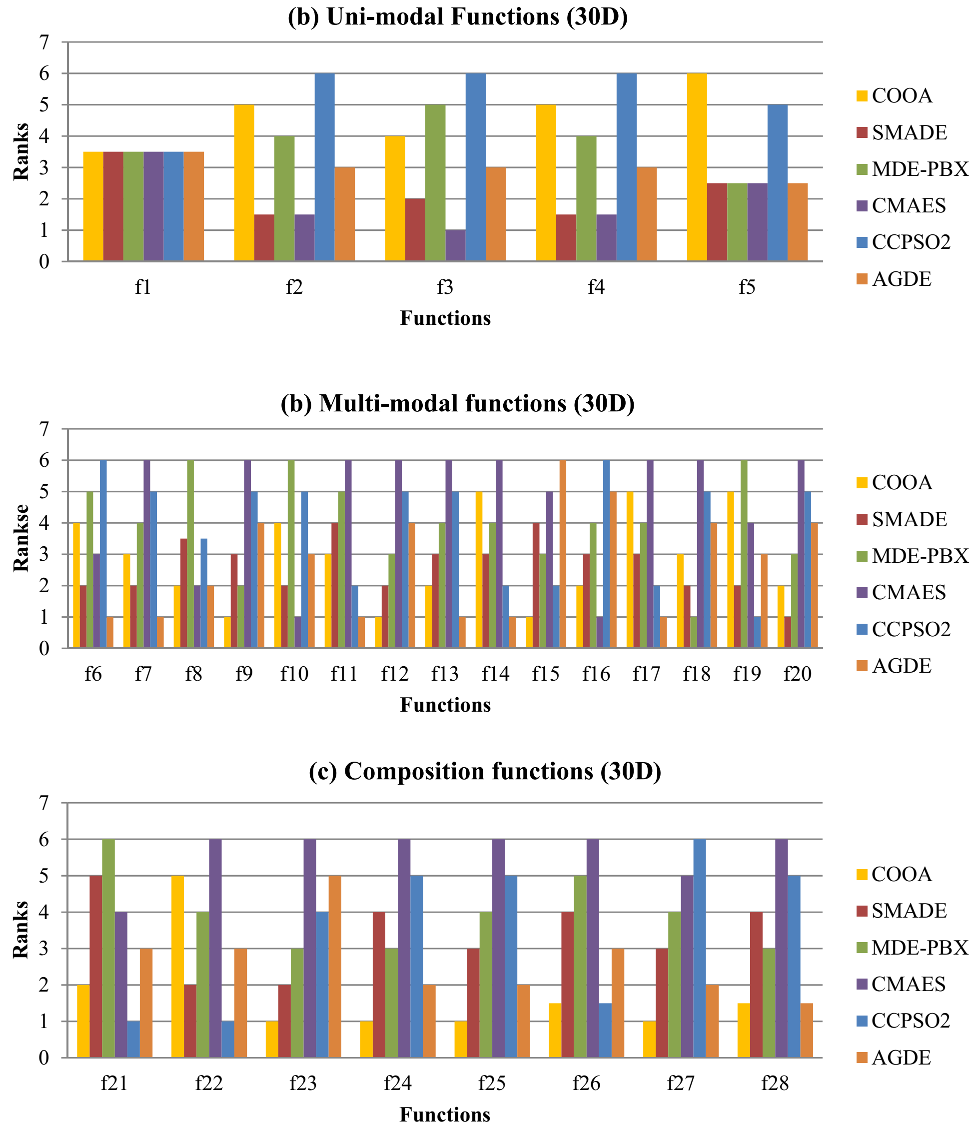


Fig. 3 Summary of Friedman ranks in 10D

6 AGDE parametric study

In fact, the performance of AGDE is mainly dependent on the selection of the group size p in new directed mutation scheme and the adaptive crossover mechanism. Firstly, the group size p remains an open problem when solving any given single-objective optimization problem. However, some empirical guidelines can be derived based on the fact that, if the value of group size p is smaller, the exploration capability of the proposed mutation is considerably hampered while enhances the exploitation tendency. Thus, in many cases, the population may lose its diversity and global exploration ability within a relatively small number of generations. Therefore, the algorithm may be trapped at some local optimum that is not the actual global optimum. This is due to the fact that, with the value of p being small, the target vectors are always attracted toward the same best or second positions found

Fig. 4 Summary of Friedman ranks in 30D



so far by the entire population and following the same few possible directions from the worst to the best top and bottom vectors. On the other hand, a large value of p increases the risk of losing the exploitative tendency of the algorithm while significantly enhances the explorative capability of the algorithm. The reason is that, the target vector will be attracted toward the same many possible directions from the worst to the best vectors which in turn effectively explore the search space to reach better regions and to escape suboptimal peaks. Nonetheless, the convergence performance may get deteriorated on some functions. Secondly, regarding the adaptive cross over scheme, it is proposed for adjusting the crossover rate during optimization process. Therefore, the question is whether AGDE can be improved by random selection of the suggested two sets of *CR*. In this section, in order to investigate the impact of the proposed modifications, some experiments are conducted. Five different versions of AGDE has been tested and compared against the proposed one. These five versions of AGDE have been compared with the proposed one on 28 benchmark functions benchmark on 30 dimensions.

1. *Version 1* to study the effectiveness of the proposed new mutation scheme, the proposed self-adaptive scheme for crossover rate is combined with novel mutation rule with $p=20\%$ i.e. $p.NP=10$. This version denotes the AGDE-1.
2. *Version 2* to study the effectiveness of the proposed new mutation scheme, the proposed self-adaptive scheme for crossover rate is combined with novel mutation rule with $p=30\%$ i.e. $p.NP=15$. This version denotes the AGDE-2.
3. *Version 3* to study the effectiveness of the proposed new mutation scheme, the proposed self-adaptive scheme

Fig. 5 Summary of Friedman ranks in 50D

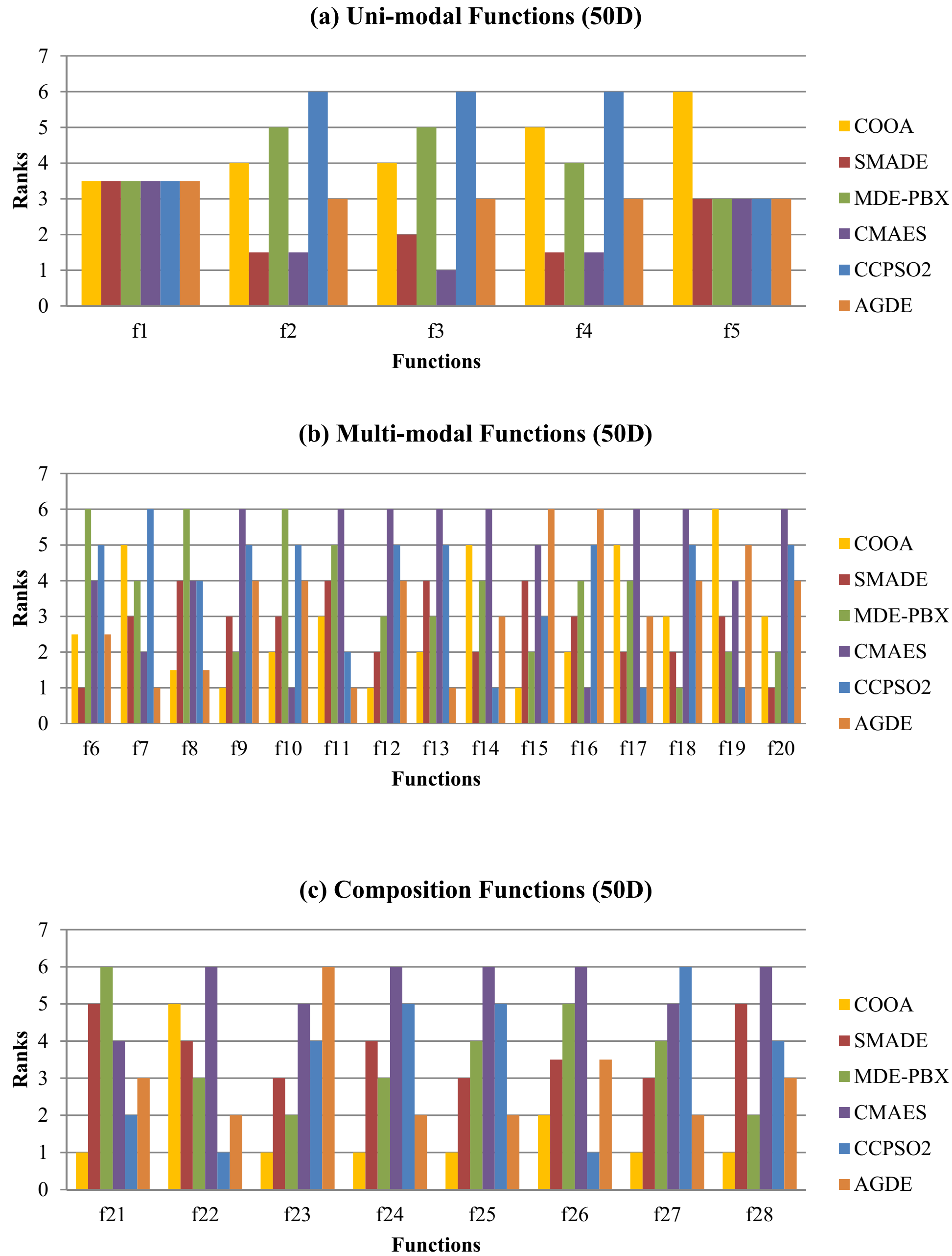


Table 7 Average ranks for D = 10, 30 and 50, respectively

Dimensions	COOA	SMADE	MDE-$_p$BX	CMAES	CCPSO2	AGDE
D = 10	2.607	2.696	3.804	4.750	4.786	**2.357**
D = 30	2.876	**2.768**	3.964	4.464	4.090	**2.768**
D = 50	**2.804**	2.964	3.696	4.411	3.946	3.179
Mean rank	**2.762**	2.810	3.821	4.542	4.274	2.768

The best ranks are marked in bold and the second ranks are underlined

for crossover rate is combined with novel mutation rule with $p = 40\%$ i.e. $p.NP = 20$. This version denotes the AGDE-3.

4. *Version 4* to study the effectiveness of the proposed new mutation scheme, the proposed self-adaptive scheme for crossover rate is combined with novel mutation rule

Table 8 Wilcoxon's test between DAEP and other algorithms for D = 10, 30 and 50, respectively

AGDE vs	COOA	SMADE	MDE-$_p$BX	CMAES	CCPSO2
D = 10					
R^+	191	201.5	279	325	358
R^-	187	123.5	46	26	20
P value	0.962	0.294	**0.002**	**0.000**	**0.000**
D = 30					
R^+	186	162	258	250.5	254
R^-	139	189	93	74.5	97
P value	0.527	0.732	**0.036**	**0.018**	**0.046**
D = 50					
R^+	166	113	254	227	200
R^-	185	187	97	98	125
P value	0.809	0.290	**0.046**	0.083	0.313

using best and worst-so-far vectors in the entire population, i.e. $p.NP = 1$. This means that it uses the global best vector and the global worst vector instead of using top and bottom 100p% individuals in the current population. This version denotes the AGDE-4.

5. *Version 5* to study the effectiveness of the proposed self-adaptation scheme for crossover rate, the suggested intervals for CR are randomly generated with constant probability 1/2, during generations, i.e. adaptive selection of the intervals vs random selection of the intervals. They are combined with the proposed mutation scheme. This version denotes the AGDE-5.

Firstly, the overall comparison results of the AGDE algorithm against its versions (AGDE-1 AGDE-2, AGDE-3, AGDE-4 and AGDE-5) on the benchmarks with 30 dimensions are summarized in Table 13. It includes the obtained best and the standard deviations of error from optimum solution of AGDE, AGDE-1 AGDE-2, AGDE-3, AGDE-4 and AGDE-5 algorithms over 51 runs for all 28 benchmark functions. The best results are marked in bold for all problems. From Table 13, it can be obviously seen that AGDE, AGDE-1, AGDE-2, AGDE-3 and AGDE-5 exhibit better

Table 9 Average ranks per function over all dimensions

Function	COOA	SMADE	MDE-$_p$BX	CMAES	CCPSO2	AGDE
f_1	**3.33**	**3.33**	**3.33**	**3.33**	4.33	**3.33**
f_2	4.67	**1.50**	4.33	**1.50**	6.00	3.00
f_3	4.33	2.00	4.67	**1.67**	6.00	2.33
f_4	5.00	**1.50**	4.00	**1.50**	6.00	3.00
f_5	5.67	**2.67**	**2.67**	**2.67**	4.67	**2.67**
f_6	2.50	2.33	5.33	4.33	4.33	**2.17**
f_7	4.00	2.33	3.67	4.67	5.33	**1.00**
f_8	**2.00**	3.33	6.00	2.83	4.17	**2.00**
f_9	**1.00**	3.00	2.67	6.00	5.00	3.33
f_{10}	2.67	2.67	5.67	**1.00**	5.33	3.67
f_{11}	3.00	3.33	5.00	6.00	2.67	**1.00**
f_{12}	2.00	**1.67**	3.00	6.00	5.00	3.33
f_{13}	2.00	3.33	3.67	6.00	5.00	**1.00**
f_{14}	4.33	2.33	4.33	6.00	2.33	**1.67**
f_{15}	**1.00**	3.67	2.33	5.33	3.33	5.33
f_{16}	**1.67**	2.67	4.00	**1.67**	5.67	5.33
f_{17}	5.00	2.17	3.67	6.00	2.33	**1.83**
f_{18}	2.33	2.33	**1.33**	6.00	5.00	4.00
f_{19}	5.00	**2.33**	3.67	4.67	**2.33**	3.00
f_{20}	2.00	**1.67**	3.00	6.00	5.00	3.33
f_{21}	**1.33**	5.00	6.00	3.33	2.00	3.33
f_{22}	5.00	2.67	3.67	6.00	**1.67**	2.00
f_{23}	**1.00**	2.33	3.00	5.67	4.33	4.67
f_{24}	**1.00**	4.17	3.50	6.00	4.33	2.00
f_{25}	**1.00**	3.33	3.50	6.00	5.00	2.17
f_{26}	**1.67**	3.00	4.67	6.00	2.50	3.17
f_{27}	**1.67**	3.33	3.33	5.00	6.00	**1.67**
f_{28}	**1.17**	4.67	3.00	6.00	4.00	2.17
Mean	**2.79**	3.18	4.14	4.55	3.46	2.97

The best ranks are marked in bold and the second ranks are underlined

Table 10 Wilcoxon's test between AGDE and other algorithms over all dimensions

AGDE vs	COOA	SMADE	MDE-$_p$BX	CMAES	CCPSO2
R^+	171	179.5	280	264.5	262
R^-	154	171.5	71	35.5	44
P value	0.819	0.919	**0.008**	**0.001**	**0.000**

Table 12 Wilcoxon's test between AGDE and other algorithms on different categories of functions

AGDE vs	COOA	SMADE	MDE-$_p$BX	CMAES	CCPSO2
Uni-modal functions					
R^+	10	0	6	0	15
R^-	0	6	0	6	0
P value	0.066	0.102	0.109	0.102	**0.042**
Multi-modal functions					
R^+	51.5	50	87	89	105
R^-	53.5	70	33	15	15
P value	0.950	0.570	0.125	**0.020**	**0.010**
Composition functions					
R^+	6	28	29.5	28	26
R^-	22	8	6.5	0	10
P value	0.176	0.161	0.106	**0.180**	0.263

high quality results for all test functions than AGDE-4. Obviously, AGDE-4 was the weaker in performance. Generally, the AGDE algorithm produces 12, 14, 14, 25 and 15 significantly better, 6, 4 ,6, 2 and 5 equal and 10, 10, 8, 1 and 8 slightly worse results than the AGDE-1, AGDE-2, AGDE-3, AGDE-4 and AGDE-5 algorithms, respectively. Thus, as expected, it can be concluded that a small value of *p*, less than 0.1, and a large value of *p*, greater than *0.3* significantly deteriorates the performance of AGDE-4 and AGDE-3, respectively. The former is too greedy to maintain the diversity of the population due to the loss of exploration capability of the AGDE-4 algorithm while the latter causes slow or bad convergence performance of AGDE-3 on some test functions f_2, f_3, f_4, f_6, f_7 due to the lack of exploitation tendency. However, the positive effect of increasing the value of *p* is that the performance of AGDE-1,AGDE-2 AGDE-3 are slightly better than AGDE on some composition functions as they need more exploration capability. Specifically, it shows that AGDE works best with $p \in [10, 30\%]$ that keep balance between the exploration and exploitation capabilities. Besides, the proposed self-adaptation scheme for crossover rate is effective as AGDE is better than AGDE-5. However, AGDE-5 has slightly performed better than AGDE in 8 test functions. Therefore, it is recommended that the proposed scheme for CR needs some improvement to produce superior performance. Secondly, the performance of AGDE and its versions on the functions of 30D is statistically validated. Table 14 summarizes the statistical analysis results of applying multiple-problem Wilcoxon's test between AGDE and other compared five versions for 30D problems. From Table 14, we can see that AGDE obtains higher R^+ values than R^- values in all cases with exception of AGDE-1and AGDE-2 versions, where AGDE-1 gets higher R^- values than R^+ values all cases with exception of AGDE-2 version. Besides, AGDE-2 obtains higher R^+ values than R^- values in comparison with AGDE-4 and AGDE-5. Furthermore, AGDE-3 obtains higher R^+ values than R^- values in comparison with AGDE-4 and AGDE-5. Finally, as can be seen All versions obtain higher R^+ values than R^- values in comparison with AGDE-4 version. According to the Wilcoxon's test at $\alpha = 0.05$ and 0.1, the significance difference can be observed in five cases only (i.e. AGDE vs AGDE-4, AGDE-1 vs AGDE-4, AGDE-2 vs AGDE-4 and AGDE-3 vs AGDE-4, AGDE-5 vs AGDE-4), which means that all these versions are significantly better than AGDE-4. However, the insignificant difference can be observed in the remaining cases which mean that these five versions are competitive. Besides, Table 15 lists the average ranks of all algorithms according to Friedman test for $D = 30$. The P value computed through Friedman test is 0.00E+00. Thus, it can be concluded that there is a significant difference between the performances of the algorithms. It can be clearly seen from Table 15 that, AGDE get the first ranking among all algorithms in 30-dimensional functions, followed by AGDE-1, AGDE-3, AGDE-2, AGDE-5 and AGDE-4. Furthermore, in order to analyze the convergence behavior of each compared algorithm, the convergence characteristics in terms of the best fitness value of the median run of each algorithm for 28 functions with dimension 30 is illustrated in Fig. 6, and it shows that ADGE and its versions exhibit similar performance on two unimodal functions f_1 and f_5, three multimodal function f_8, f_{18}, f_{19} and two composition functions f_{21} and f_{28}. Furthermore, as expected, all

Table 11 Average ranks on uni-modal functions, multi-modal functions, composition functions, respectively

The function category	COOA	SMADE	MDE-$_p$BX	CMAES	CCPSO2	AGDE
Uni-modal	3.767	**2.667**	3.400	3.367	4.233	<u>3.050</u>
Multi-modal	**2.733**	3.078	4.278	4.667	3.389	<u>3.033</u>
Composition	**2.292**	3.688	4.333	5.083	3.125	<u>2.813</u>

The best ranks are marked in bold and the second ranks are underlined

Table 13 Comparison between AGDE, AGDE-1, AGDE-2, AGDE-3, AGDE-4 and AGDE-5 on 30D problems

Function	AGDE	AGDE-1	AGDE-2	AGDE-3	AGDE-4	AGDE-5
1	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
2	2.64E+04 ± 1.43E+04	**2.57E+04 ± 1.48E+04**	3.24E+04 ± 1.57E+04	6.33E+04 ± 3.22E+04	1.12E+05 ± 1.04E+05	4.96E+04 ± 2.46E+04
3	3.07E+05 ± 1.05E+06	**2.45E+05 ± 8.40E+05**	2.49E+05 ± 8.54E+05	1.11E+06 ± 2.76E+06	2.22E+06 ± 5.17E+06	4.27E+05 ± 1.29E+06
4	2.32E+00 ± 3.59E+00	**2.07E+00 ± 2.72E+00**	5.98E+00 ± 7.40E+00	4.39E+01 ± 5.36E+01	8.64E+01 ± 8.53E+01	5.08E−01 ± 8.26E−01
5	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
6	**1.55E+00 ± 6.28E+00**	1.66E+00 ± 6.25E+00	6.35E+00 ± 4.43E+00	1.31E+01 ± 9.05E+00	6.74E+00 ± 8.99E+00	**1.55E+00 ± 6.28E+00**
7	3.98E+00 ± 4.22E+00	**3.06E+00 ± 2.59E+00**	4.53E+00 ± 3.43E+00	6.66E+00 ± 6.43E+00	1.41E+01 ± 1.06E+01	6.32E+00 ± 6.25E+00
8	**2.09E+01 ± 4.67E−02**	**2.09E+01 ± 5.42E−02**	2.10E+01 ± 6.05E−02	**2.09E+01 ± 4.89E−02**	2.10E+01 ± 4.92E−02	**2.09E+01 ± 5.57E−02**
9	2.54E+01 ± 5.64E+00	2.30E+01 ± 7.71E+00	1.97E+01 ± 8.31E+00	**1.72E+01 ± 7.42E+00**	2.68E+01 ± 3.59E+00	1.79E+01 ± 7.95E+00
10	**3.11E−02 ± 2.27E−02**	3.25E−02 ± 1.85E−02	4.35E−02 ± 2.35E−02	4.94E−02 ± 2.80E−02	5.29E−02 ± 2.86E−02	3.29E−02 ± 2.72E−02
11	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	3.87E−03 ± 2.73E−02	**0.00E+00 ± 0.00E+00**
12	9.45E+01 ± 2.29E+01	7.83E+01 ± 2.83E+01	5.49E+01 ± 1.99E+01	**4.99E+01 ± 1.33E+01**	1.11E+02 ± 1.71E+01	6.25E+01 ± 2.46E+01
13	1.10E+02 ± 2.19E+01	1.11E+02 ± 1.84E+01	1.08E+02 ± 2.06E+01	1.01E+02 ± 2.42E+01	1.20E+02 ± 2.38E+01	**9.00E+01 ± 2.41E+01**
14	**5.47E−02 ± 3.25E−02**	8.19E−02 ± 1.67E−01	1.91E−01 ± 3.35E−01	1.62E−01 ± 2.63E−01	9.72E+01 ± 4.56E+01	3.60E+01 ± 1.66E+01
15	**5.99E+03 ± 4.03E+02**	6.12E+03 ± 3.31E+02	6.10E+03 ± 3.18E+02	6.08E+03 ± 5.42E+02	6.05E+03 ± 3.04E+02	6.10E+03 ± 2.73E+02
16	**2.14E+00 ± 2.61E−01**	2.36E+00 ± 2.69E−01	2.39E+00 ± 2.26E−01	2.33E+00 ± 2.68E−01	2.34E+00 ± 2.46E−01	2.29E+00 ± 2.88E−01
17	**3.04E+01 ± 2.12E−03**	**3.04E+01 ± 1.50E−02**	**3.04E+01 ± 1.54E−02**	**3.04E+01 ± 5.93E−03**	3.38E+01 ± 1.93E+00	3.38E+01 ± 8.68E−01
18	1.84E+02 ± 1.40E+01	1.85E+02 ± 1.18E+01	1.85E+02 ± 1.10E+01	1.82E+02 ± 1.05E+01	1.86E+02 ± 1.26E+01	**1.81E+02 ± 1.30E+01**
19	3.07E+00 ± 2.60E−01	3.12E+00 ± 2.56E−01	3.05E+00 ± 2.66E−01	**2.99E+00 ± 2.97E−01**	3.14E+00 ± 3.47E−01	3.38E+00 ± 3.80E−01
20	**1.15E+01 ± 3.72E−01**	1.16E+01 ± 3.35E−01	1.16E+01 ± 3.73E−01	**1.15E+01 ± 3.82E−01**	1.18E+01 ± 3.79E−01	1.16E+01 ± 3.63E−01
21	3.00E+02 ± 7.94E+01	3.03E+02 ± 7.27E+01	3.05E+02 ± 8.06E+01	3.06E+02 ± 7.53E+01	3.23E+02 ± 8.75E+01	**2.91E+02 ± 7.07E+01**
22	6.12E+02 ± 1.07E+02	5.72E+02 ± 1.30E+02	4.53E+02 ± 1.96E+02	**3.35E+02 ± 1.67E+02**	6.89E+02 ± 1.02E+02	8.35E+02 ± 1.37E+02
23	6.14E+03 ± 4.58E+02	6.21E+03 ± 3.95E+02	6.27E+03 ± 5.59E+02	6.21E+03 ± 6.02E+02	**6.13E+03 ± 3.21E+02**	6.37E+03 ± 3.73E+02
24	2.08E+02 ± 8.57E+00	**2.07E+02 ± 6.84E+00**	**2.07E+02 ± 6.63E+00**	2.13E+02 ± 9.38E+00	2.17E+02 ± 1.45E+01	2.12E+02 ± 6.27E+00
25	2.76E+02 ± 1.43E+01	2.69E+02 ± 1.78E+01	2.65E+02 ± 1.58E+01	2.62E+02 ± 1.23E+01	2.86E+02 ± 6.20E+00	**2.56E+02 ± 1.17E+01**
26	2.05E+02 ± 2.59E+01	2.07E+02 ± 2.78E+01	**2.00E+02 ± 1.49E−03**	2.02E+02 ± 1.58E+01	2.05E+02 ± 2.37E+01	2.12E+02 ± 3.62E+01
27	5.53E+02 ± 2.04E+02	5.49E+02 ± 1.72E+02	**5.21E+02 ± 1.30E+02**	5.85E+02 ± 1.30E+02	8.21E+02 ± 2.00E+02	5.49E+02 ± 1.20E+02
28	**3.00E+02 ± 0.00E+00**	**3.00E+02 ± 0.00E+00**	**3.00E+02 ± 0.00E+00**	**3.00E+02 ± 0.00E+00**	**3.00E+02 ± 0.00E+00**	**3.00E+02 ± 0.00E+00**

Table 14 Results of multiple-problem Wilcoxon's test AGDE, AGDE-1, AGDE-2, AGDE-3, AGDE-4 and AGDE-5 for 30D problems

Algorithm	R^+	R^-	p value	α = 0.05	α = 0.1
AGDE vs AGDE-1	106	147	0.506	No	No
AGDE vs AGDE-2	146	154	0.909	No	No
AGDE vs AGDE-3	162	91	0.249	No	No
AGDE vs AGDE-4	336	15	0.000	Yes	Yes
AGDE vs AGDE-5	176.5	99.5	0.242	No	No
AGDE-1 vs AGDE-2	115	116	0.986	No	No
AGDE-1 vs AGDE-3	129.5	123.5	0.922	No	No
AGDE-1 vs AGDE-4	322.5	55.5	0.001	Yes	Yes
AGDE-1 vs AGDE-5	144	109	0.570	No	No
AGDE-2 vs AGDE-3	148	152	0.954	No	No
AGDE-2 vs AGDE-4	306	45	0.001	Yes	Yes
AGDE-2 vs AGDE-5	181	95	0.191	No	No
AGDE-3 vs AGDE-4	324	54	0.001	Yes	Yes
AGDE-3 vs AGDE-5	132	168	0.607	No	No
AGDE-4 vs AGDE-5	84	267	0.020	Yes	Yes

algorithms converge to better or global solution faster than AGDE-4 in all cases as it may be stagnated or it prematurely converges to local optima in middle or later stage of the search process. Besides, AGDE-5 converges slower than AGDE in most cases while AGDE-5 converges faster than AGDE on 1 unimodal functions f_4, five multimodal functions f_6, f_9, f_{12}, f_{13} and f_{20} and 1 composition functions f_{25}. On the other hand, AGDE-3 is slightly faster than AGDE, AGDE-1 and AGDE-2 on two multimodal functions f_9 and f_{12} and two composition functions f_{22} and f_{25} while converges slower than AGDE ,AGDE-1 and AGDE-2 versions on three unimodal functions f_2, f_3, f_4, four multimodal functions f_6, f_7, f_{11} and f_{14} and one composition function f_{27}. Generally, the convergence behavior of AGDE, AGDE-1 and AGDE-2 has been approximately similar on all functions. Therefore, the direct comparison, statistical analysis and convergence behavior support the previous empirical guidelines.

It is clear that the proposed modifications play a vital role and has a significant impact on improving the convergence speed of AGDE algorithm for most problems. The AGDE algorithm has a considerable ability to maintain its convergence rate, improve its diversity as well as advance its local tendency through a search process. Thus, after the

Table 15 Average ranking of AGDE, AGDE-1, AGDE-2, AGDE-3, AGDE-4 and AGDE-5 according to Friedman test for D = 30

Rank	Algorithm	Mean ranking
1	AGDE	2.73
2	AGDE-1	3.11
3	AGDE-3	3.27
4	AGDE-2	3.29
5	AADE-5	3.32
6	AADE-4	5.29

above analysis and discussion, the proposed algorithms AGDE, AGDE-1, AGDE-2, AGDE-3 and AGDE-5 show competitive performance in terms of quality of solution, efficiency, convergence rate and robustness. They are superior to AGDE-4 algorithm that is basically based on the global best vector and global worst vector in the entire population. Accordingly, the main benefits of the proposed modifications are the remarkable balance between the exploration capability and exploitation tendency through the optimization process. This balance leads to superior performance with fast convergence speed and the extreme robustness over the entire range of benchmark functions which are the weak points of all evolutionary algorithms.

7 Conclusions

In order to enhance the overall performance of basic DE algorithm, introducing new mutation rules combined with advanced adaptive and/or self adaptive schemes for crossover rate and scaling factor is a must although it is a challenging task. In this paper, a new DE algorithm, named AGDE, is proposed for solving global numerical optimization problems over continuous space. In the proposed algorithm, a new mutation operator is introduced. It uses two random chosen vectors of the top and bottom 100*p*% individuals in the current population of size *NP* while the third vector is selected randomly from the middle [*NP-2*(100*p* %)] individuals. Besides, a novel and effective adaptation scheme is used to update the values of the crossover rate to appropriate values without either extra parameters or prior knowledge of the characteristics of the optimization problem. The proposed novel approach to mutation operator is shown to enhance the global and local search capabilities and to increase the convergence speed of the new algorithm compared with classical scheme. In order to test the effectiveness of AGDE, it is applied to solve the CEC-2013 real-parameter benchmark optimization problems. Experimental results are compared with five state-of-the-art algorithms: COOA, SMADE, MDE-PBX, CMEAS, CCPSO2 and one of its variants. In order to statistically analyze the performance

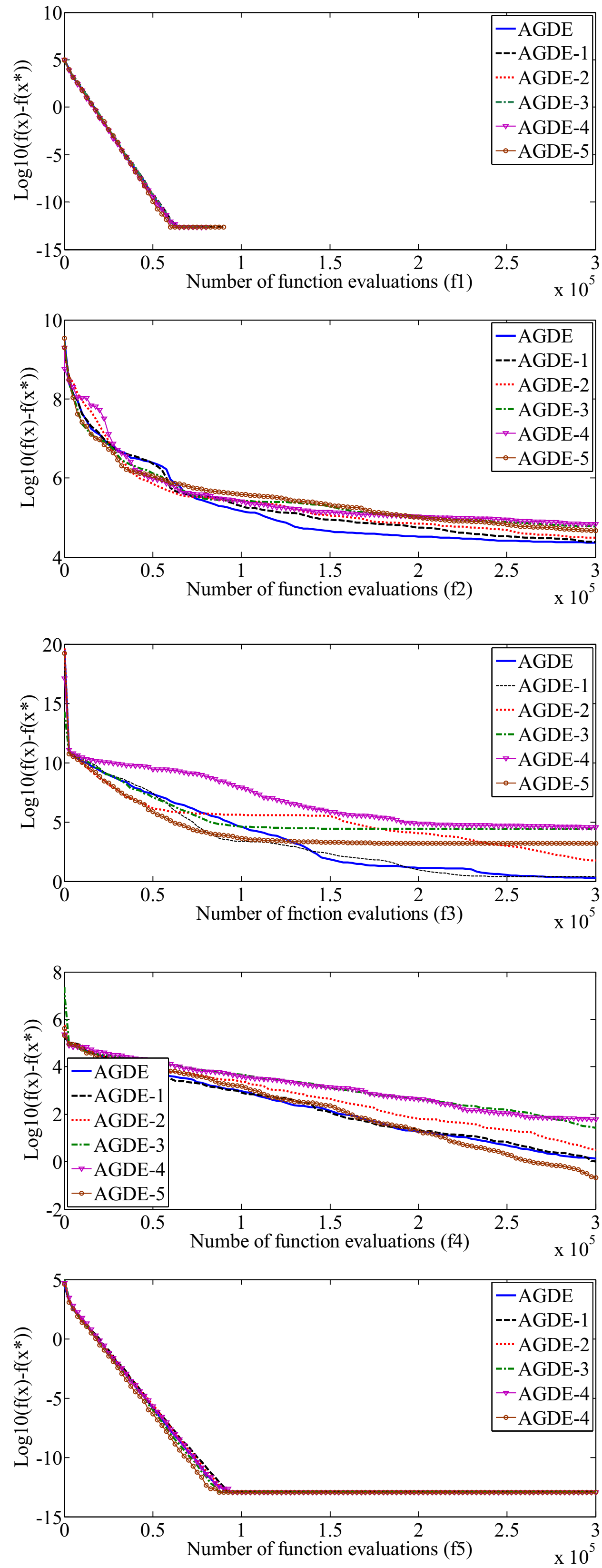


Fig. 6 Convergence graph (*median curves*) of AGDE, AGDE-1, AGDE-2, AGDE-3, AGDE-4 and AGDE-5 on 30-dimensional test functions f_1–f_{28}

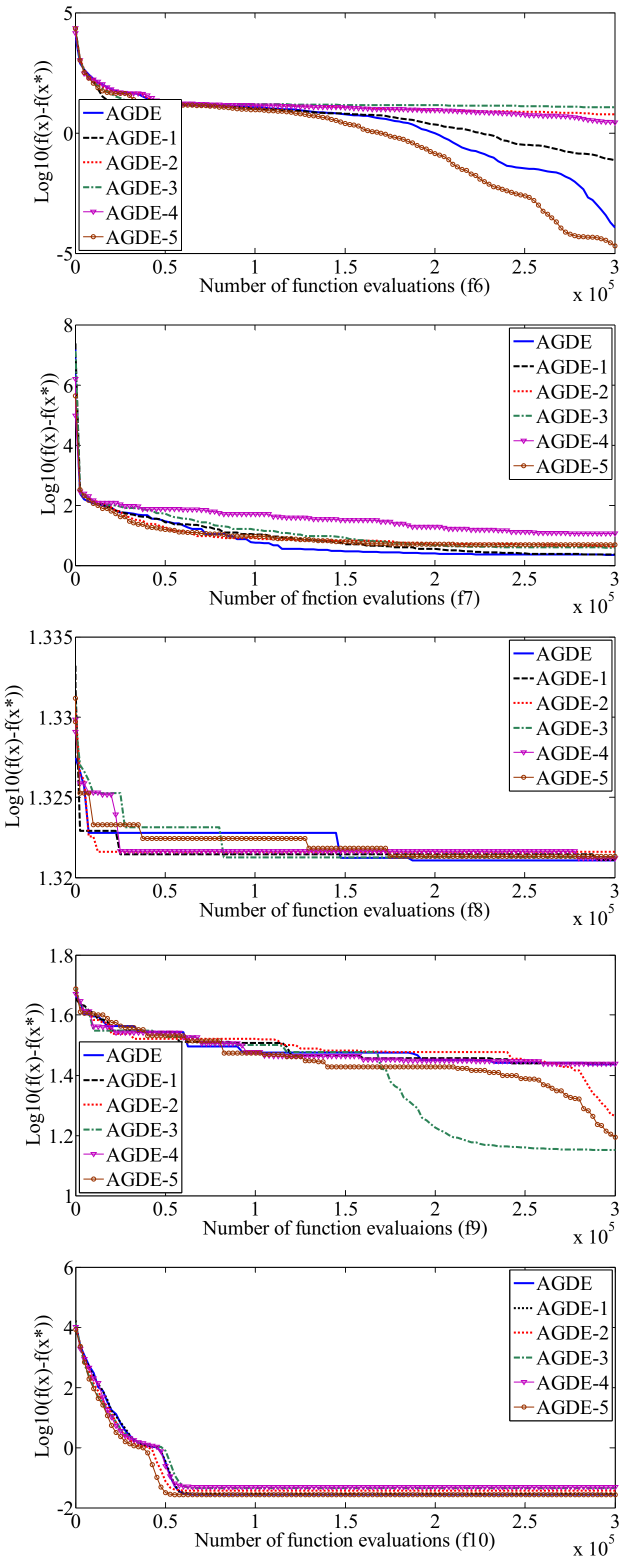


Fig. 6 (continued)

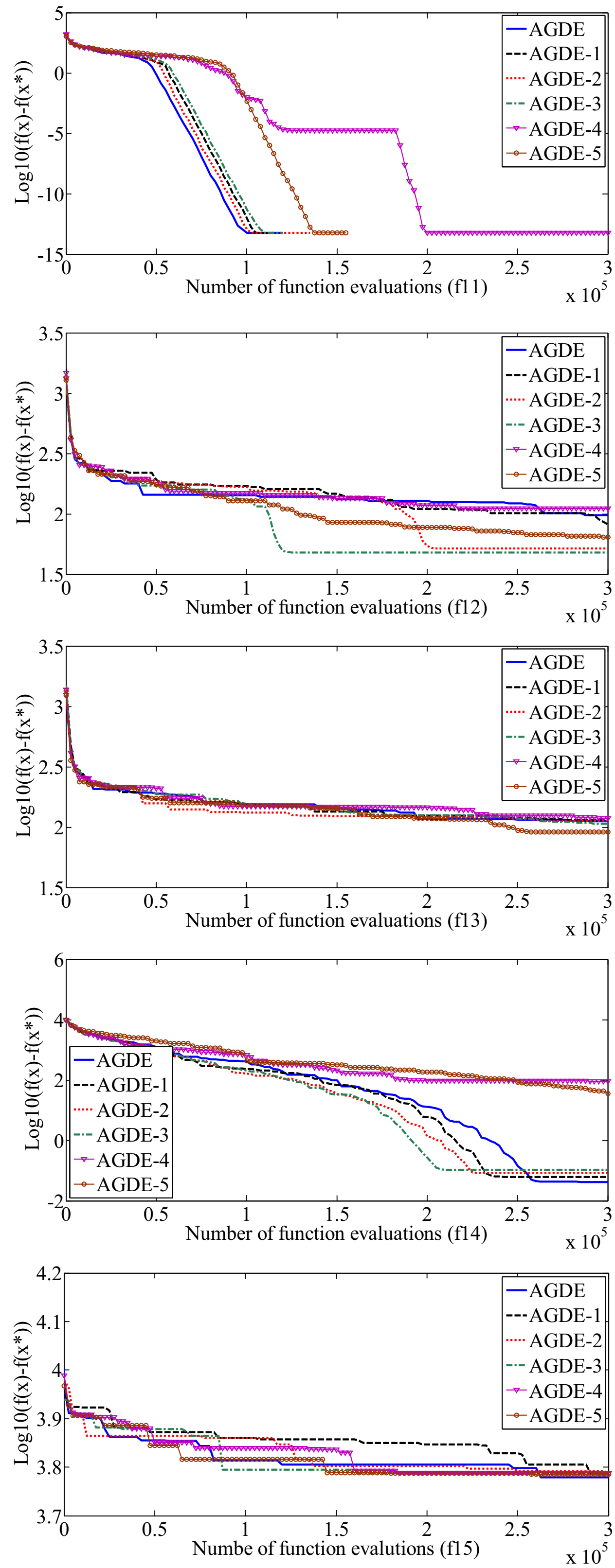


Fig. 6 (continued)

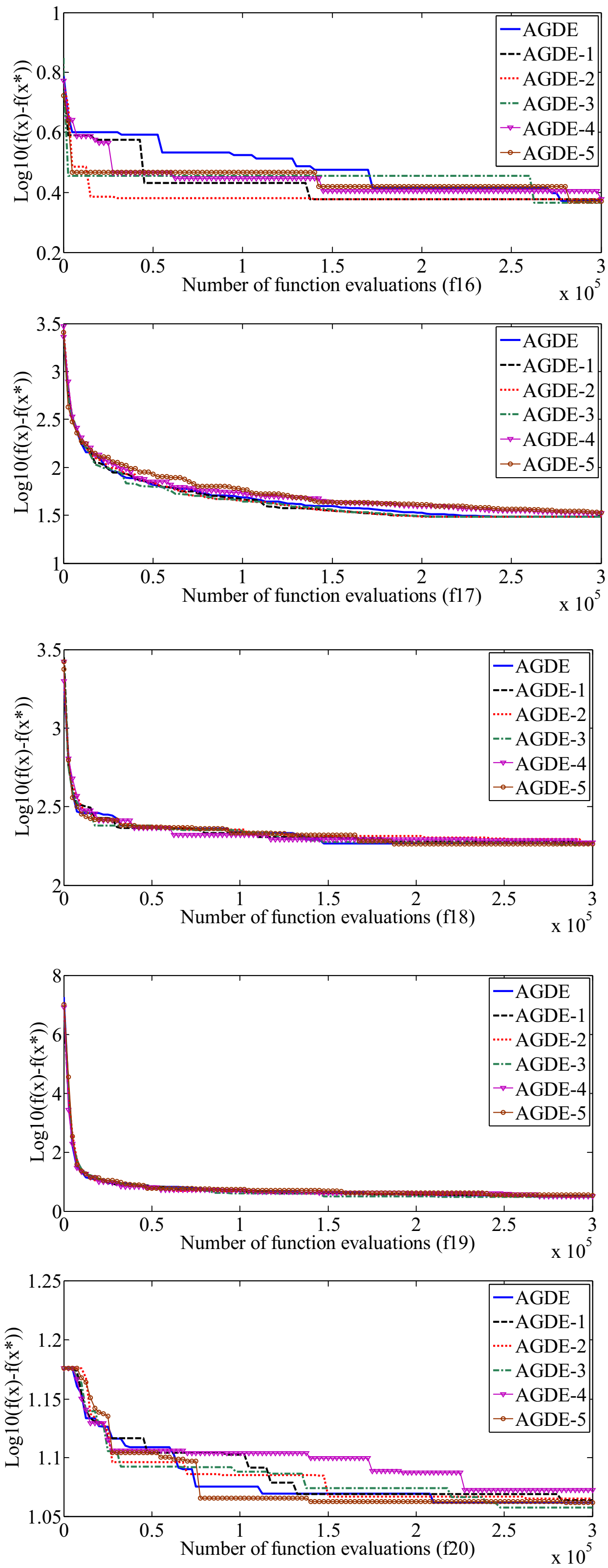


Fig. 6 (continued)

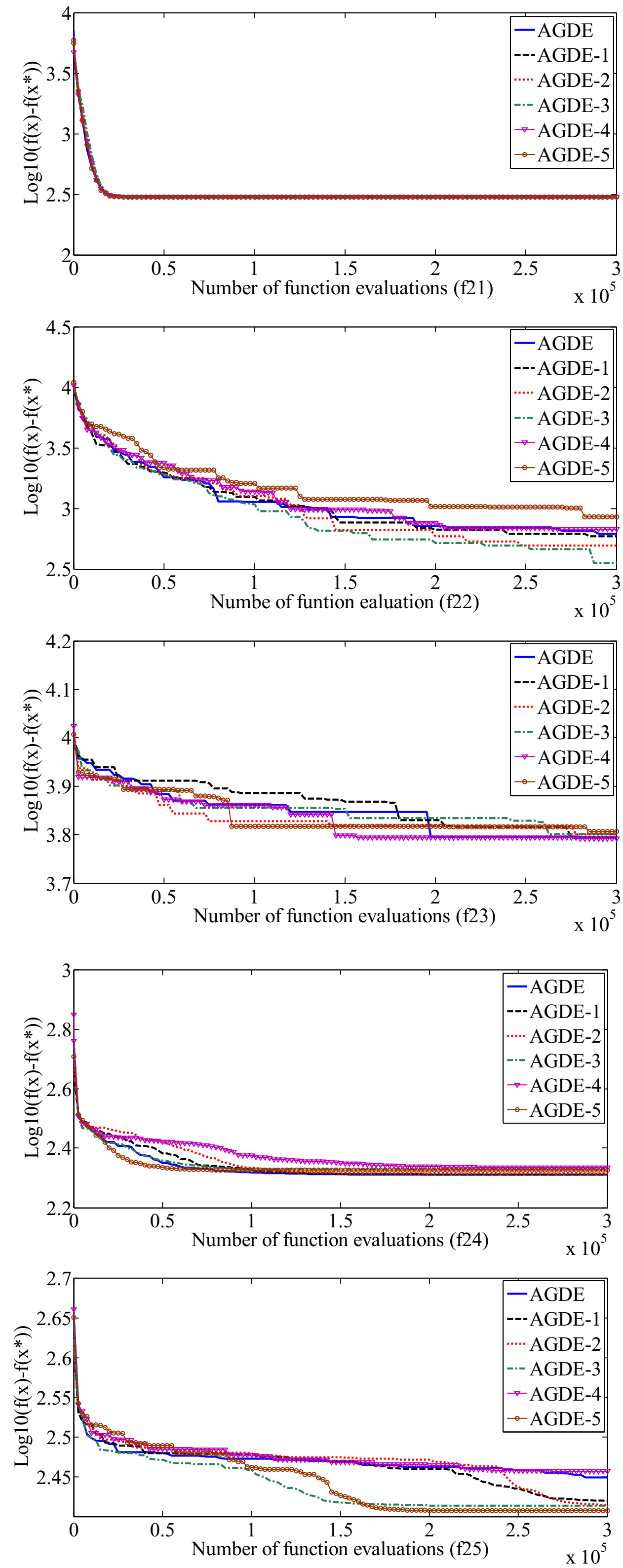


Fig. 6 (continued)

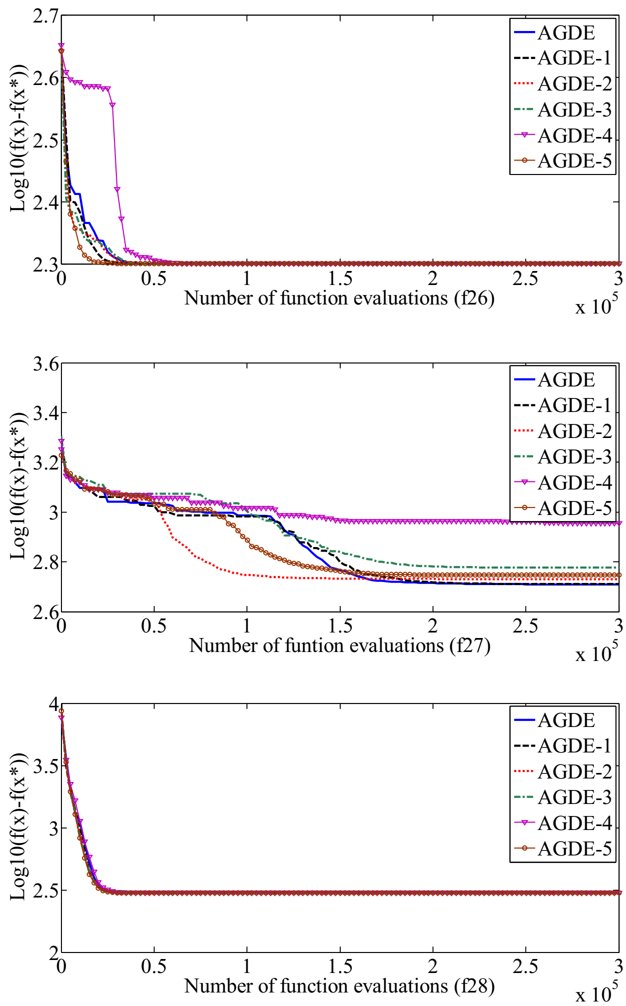


Fig. 6 (continued)

of DEAP, two non-parametric tests (the Friedman test and Wilcoxon's test) are used with the significance level of 0.05. As a summary of results, the performance of the AGDE algorithm was statistically superior to and competitive with other recent and well-known state-of-the-art algorithms in the majority of functions and for different dimensions especially for composition functions. Virtually, it is easily implemented and has been proven to be a reliable approach for real parameter optimization. In addition to its very promising performance, the effectiveness and benefits of the proposed modifications used in AGDE were experimentally investigated and compared. It was found that AGDE works best with $p \in [10\%, 30\%]$ that keep balance between the exploration and exploitation capabilities. Moreover, experimental results indicate that the proposed adaptive scheme for crossover rate plays a vital role in promoting the effectiveness of AGDE. However, much effort is needed to improve it to enhance AGDE performance.

Several current and future works can be developed from this study. Firstly, Current research efforts focus on how to modify the AGDE algorithm for handling constrained and multi-objective optimization problems as well as solving practical engineering optimization problems and real world applications. Secondly, concerning the improvement of AGDE, it would be very interesting to propose another adaptive AGDE version by combining other mutation strategies. Future research studies may focus on applying the algorithm to solve high dimensions or large scale global optimization problems. Another possible direction is integrating the proposed mutation scheme with other self-adaptive DE variants like SMADE and MDE-pBX and many others. Finally, hybridizing the AGDE algorithm with other powerful metaheuristics, such as CMAES, CCPSO2 are thought to be promising direction. The Matlab source code of AGDE is available upon request. The Matlab source code of AGDE can be downloaded from https://sites.google.com/view/optimization-project/files.

Acknowledgements The author would like to thank Deanship of Scientific Research at Majmaah University for supporting this work.

Compliance with ethical standards

Conflict of interest Ali Wagdy Mohamed and Ali Khater Mohamed declare that he has no conflict of interest.

Human participants or animals This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. 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. http://icsi.berkeley.edu/~storn/litera.html. Accessed 16 June 2016
2. 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
3. Price K, Storn R, Lampinen J (2005) Differential evolution: a practical approach to global optimization. Springer, Heidelberg
4. Li X, Yin M (2014) Modified differential evolution with self-adaptive parameters method. J Comb Optim 31(2):546–576
5. Wang Y, Li H-X, Huang T, Li L (2014) Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. Appl Soft Comput 18:232–247
6. Zhu H, He Y, Tsang E, Xi-zhao W (2017) Discrete differential evolution for the discounted {0–1} knapsack problem. J Bio Inspir Comput (**Accepted in June 2017**)
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. Dong C-R, Ng WWY, Wang X-Z, Chan PPK, Yeung DS (2014) An improved differential evolution and its application to

determining feature weights in similarity-based clustering. Neurocomputing 146:95–103. 10.1016/j.neucom.2014.04.065
9. El-Quliti SA, Ragab AH, Abdelaal R et al (2015) Anonlinear goal programming model for university admission capacity planning with modified differential evolution algorithm.Math Probl Eng 2015:13
10. 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. Comput Intell Neurosci 2016:14
11. El-Quliti SA, Mohamed AW (2016) Alarge-scale nonlinear mixedbinary goal programming model to assess candidate locations for solar energy stations: an improved binary differential evolution algorithm with a case study. J Comput Theor Nanosci 13(11):7909–7921
12. Greenwood GW (2009) Using differential evolution for subclass of graph theory problems. IEEE Trans Evol Comput 13(5):1190–1192
13. Noman N, Iba H (2008) Accelerating differential evolution using an adaptive local search. IEEE Trans Evol Comput 12(1):107–125
14. 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
15. Lampinen J, Zelinka I (2000) On stagnation of the differential evolution algorithm. In: Matoušek R, Ošmera P, editors. Proceedings of Mendel 2000, 6th international conference on soft computing, pp 76–83
16. Das SS, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31
17. Liang JJ, Qin BY, Suganthan PN, Hernndez-Diaz AG (2013) Problem definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization, Zhengzhou University/Nanyang Technological University, Zhengzhou, China/Singapore, Technical Report 201212
18. Fan HY, Lampinen J (2003) A trigonometric mutation operation to differential evolution. J Glob Optim 27(1):105–129
19. 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
20. Gamperle R, Muller SD, Koumoutsakos P (2002) A parameter study for differential evolution. In: Gremla A, Mastorakis NE (eds) Advances in intelligent systems, fuzzy systems, evolutionary computation. WSEAS Press, Interlaken, pp 293–298
21. Ronkkonen J, Kukkonen S, Price KV (2005) Real parameter optimization with differential evolution. In: Proceedings of the IEEE congress evolutionary computation (CEC-2005), vol 1. IEEE Press, Piscataway, pp 506–513
22. Zaharie D (2003) Control of population diversity and adaptation in differential evolution algorithms. In: Matousek R, Osmera P (eds) 9th international conference on soft computing proceedings of mendel 2003, pp 41–46
23. Liu J, Lampinen J (2005) A fuzzy adaptive differential evolution algorithm. Soft Comput 9(6):448–462
24. 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
25. Omran M, Salman A, Engelbrecht A (2005) Self-adaptive differential evolution. Lect Notes Artif Intell 3801:192–199
26. 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
27. 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
28. Caraffini F, Neri F, Cheng J, Zhang G, Picinail L, Iacca G, Mininno E (2013) Super-fit multicriteria adaptive differential evolution. In: 2013 IEEE congress on evolutionary computation (CEC). IEEE, New York, pp 1678–1685
29. Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 13(5):945–958
30. Islam S, Das S, Ghosh S, Roy S, Suganthan PN (2012) An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. Syst Man Cybern Part B Cybern IEEE Trans 42(2):482–500
31. 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
32. Mohamed AW (2017) A novel differential evolution algorithm for solving constrained engineering optimization problems. J Intell Manuf. doi:10.1007/s10845-017-1294-6
33. Mohamed AW (2015) An improved differential evolution algorithm with triangular mutation for global numerical optimization. Comput Ind Eng 85:359–375
34. Feoktistov V (2006) Differential evolution: in search of solutions. Springer, Berlin
35. Wang Y, Liu Z-Z, Li J, Li H-X, Wang J (2017) On the selection of solutions for mutation in differential evolution. Front Comput Sci. doi:10.1007/s11704-016-5353-5 **(in press)**
36. Wang Y, Liu Z-Z, Li J, Li H-X, Yen GG (2016) Utilizing cumulative population distribution information in differential evolution. Appl Soft Comput 48:329–346
37. Mohamed AW, Hadi AA, Fattouh AM, Jambi KM (2017) LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. In: 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastian, pp 145–152. doi:10.1109/CEC.2017.7969307
38. Mohamed AW (2017) Solving large-scale global optimization problems using enhanced adaptive differential evolution algorithm. Complex Intell Syst 1–27. doi:10.1007/s40747-017-0041-0
39. Ghosh A, Das S, Chowdhury A, Giri R (2011) An improved differential evolution algorithm with fitness-based adaptation of the control parameters. Inf Sci 181(18):3749–3765
40. Ali MM, Törn A (2004) Population set based global optimization algorithms: some modifications and numerical studies. Comput Oper Res 31:1703–1725
41. Zhang X, Chen W, Dai C, Cai W (2010) Dynamic multi-group self-adaptive differential evolution algorithm for reactive power optimization. Int J Electr Power 32:351–357
42. Feng X, Zou R, Yu H (2015) A novel optimization algorithm inspired by the creative thinking process. Soft Comput 19(10):2955–2972
43. Hansen N, Műller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolut Comput 11(1):1–18
44. Li X, Yao X (2012) Cooperatively coevolving particle swarms for large scale optimization. IEEE Trans Evol Comput 16(2):210–224
45. 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