Optimization Letters
https://doi.org/10.1007/s11590-019-01454-5

ORIGINAL PAPER

Differential evolution with enhanced diversity maintenance

Joel Chacón Castillo[1] · **Carlos Segura**[1]

Received: 20 September 2018 / Accepted: 11 July 2019

Abstract
Differential evolution (DE) is a popular population-based meta-heuristic that has been successfully used in complex optimization problems. Premature convergence is one of the most important drawbacks that affects its performance. In this paper, a novel replacement strategy that combines the use of an elite population and a mechanism to preserve diversity explicitly is devised. The proposal is integrated with DE to generate the DE with enhanced diversity maintenance. The main novelty is the use of a dynamic balance between exploration and exploitation to adapt the optimizer to the requirements of the different optimization stages. Experimental validation is carried out with several benchmark tests proposed in competitions of the well-known IEEE Congress on Evolutionary Computation. Top-rank algorithms of each competition, as well as other diversity-based schemes, are used to illustrate the usefulness of the proposal. The new method avoids premature convergence and significantly improves further the results obtained by state-of-the-art algorithms.

Keywords Diversity · Differential evolution · Premature convergence

1 Introduction

Evolutionary algorithms (EAs) are one of the most widely used techniques to deal with complex optimization problems. Several variants of these strategies have been devised [30] and applied in many fields, such as in science, economic and engineering [5]. Among them, Differential Evolution (DE) [28] is one of the most effective strategies to deal with continuous optimization. In fact, it has been the winning strategy of several optimization competitions [8]. Similarly to other EAs, DE is inspired by the natural evolution process and it involves the application of mutation, recombination

✉ Joel Chacón Castillo
joel.chacon@cimat.mx

Carlos Segura
carlos.segura@cimat.mx

[1] Centro de Investigación en Matemáticas, Guanajuato, Mexico

Published online: 16 July 2019

and selection. The main peculiarity of DE is that it considers the differences among vectors that are present in the population to explore the search space. In this sense it is similar to the Nelder-Mead [23] and the Controlled Random Search (CRS) [24] optimizers.

In spite of the effectiveness of DE, there exists several weaknesses that have been detected and partially solved by extending the standard variant [8,18]. Among them, the sensitivity to its parameters [36], the appearance of stagnation due to the reduced exploration capabilities [14,25] and premature convergence [35] are some of the most well-known issues. This last one issue is tackled in this paper. Note that, attending to the proper design of population-based meta-heuristics [30], special attention must be paid to attain a proper balance between exploration and exploitation. A too large exploration degree prevents the proper intensification of the best located regions, usually resulting in a too slow convergence. Differently, an excessive exploitation degree provokes loss of diversity meaning that only a limited number of regions are sampled. In the case of DE, since its inception some criticism appeared because of its incapability to maintain a large enough diversity due to the use of a selection with high pressure [25]. Thus, several extensions of DE to deal with premature convergence have been devised such as parameter adaptation [35], auto-enhanced population diversity [34] and selection strategies with a lower selection pressure [25]. Some of the last studies on design of population-based meta-heuristics [6] show that explicitly controlling the diversity to properly balance the exploration and intensification degree is particularly useful. Specifically, in the field of combinatorial optimization some novel replacement strategies that dynamically alter the balance between exploration and exploitation have appeared [27]. The main principle of such proposals is to use the stopping criterion and elapsed generations to bias the decisions taken by the optimizers with the aim of promoting exploration in the initial stages and exploitation in the last ones. Probably their main weakness is that the time required to obtain high-quality solution increases. Our novel proposal, which is called DE with Enhanced Diversity Maintenance (DE- EDM), integrates a similar principle into DE. However, in order to avoid the excessive growth of computational requirements typical of diversity-based replacement strategies, two modifications that induce a larger degree of intensification are included.

The rest of the paper is organized as follows. Some basic concepts of DE and a review of works related to diversity within DE are given in Sect. 2. Section 3 presents an analysis about the algorithms with best performance on the last continuous optimization contests held at the IEEE Congress on Evolutionary Computation. More emphasis is given on the variants based on DE. Our proposal is described in Sect. 4. The experimental validation, which includes comparisons against state-of-the-art approaches, is detailed in Sect. 5. Finally, our conclusions and some lines of future work are given in Sect. 6.

2 Literature review

2.1 Differential Evolution: Basic Concepts

This section is devoted to summarize the classic DE variant and to introduce some of the most important terms used in the DE field. The classic DE scheme is called

the DE/rand/1/bin, which has been extensively used to generate more complex DE variants [8]. In fact, our proposal also extends the classic variant. DE was originally proposed as a direct search method for single-objective continuous optimization. The variables governing a given problem performance are given as a vector like $\mathbf{X} = [x_1, x_2, \ldots, x_D]$, where D is the dimension of the problem. In continuous optimization, each x_i is a real number and usually box-constraints are given, i.e. there is a lower bound (a_i) and upper bound (b_i) for each variable. The aim of the optimization process is to obtain the vector $\mathbf{X}^*$ which minimizes a given objective function, mathematically denoted by $f : \Omega \subseteq \Re^D \rightarrow \Re$. In the box-constrained case $\Omega = \prod_{j=1}^{D}[a_j, b_j]$.

DE is a population-based stochastic algorithm, so it iteratively evolves a multi-set of candidate solutions. In DE such candidate solutions are usually called vectors. In the basic DE variant for each member of the population—they are called *target vectors*—a new *mutant vector* is created. Then, the mutant vector is combined with the target vector to generate a *trial vector*. Finally, a selection phase is applied to choose the survivors. In this way, several generations are evolved until a stopping criterion is reached. The i-th vector of the population at the generation G is denoted as $\mathbf{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G}]$. In the following more details are given for each component of DE.

2.1.1 Initialization

DE usually starts the optimization process with a randomly initiated population of N vectors. Since there is commonly no information about the performance of different regions, uniform random generators are usually applied. Hence, the j-th component of the i-th vector is initialized as $x_{j,i,0} = a_j + rand_{i,j}[0, 1](b_j - a_j)$, where $rand_{i,j}[0, 1]$ is an uniformly distributed random number lying between 0 and 1.

2.1.2 Mutation

For each target vector a mutant vector is created. Several ways of performing such a process have been proposed. In the classic DE variant the rand/1 strategy is applied. In this case, the mutant vector $V_{i,G}$ is created as follows:

$$\mathbf{V}_{i,G} = \mathbf{X}_{r_1,G} + F \times (\mathbf{X}_{r_2,G} - \mathbf{X}_{r_3,G}) \quad r_1 \neq r_2 \neq r_3 \tag{1}$$

The indices $r_1, r_2, r_3 \in [1, N]$ are mutually different integers randomly chosen from the range $[1, N]$. In addition, they are all different from the index i. It is important to take into account that the difference between vectors is scaled with the number F, which is usually defined in the interval [0.4, 1]. The scaled difference is added to a third vector, meaning that when diversity decreases and consequently differences are low, mutant vectors are similar to target vectors. As a result, maintaining some degree of diversity is specially important in DE.

2.1.3 Crossover

In order to combine information of different candidate solutions and with the aim of increasing diversity, the crossover operator is applied. Specifically, each target vector $\mathbf{X}_{i,G}$ is mixed with its corresponding mutant vector $V_{i,G}$ to generate the trial vector $\mathbf{U_{i,G}} = [u_{1,i,G}, u_{2,i,G}, \ldots, u_{D,i,G}]$. The most typical crossover is the *binomial* one, which operates as follows:

$$\mathbf{U}_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if}(rand[0,1] \leq CR \quad or \quad j = j_{rand}) \\ x_{j,i,G}, & \text{otherwise} \end{cases} \tag{2}$$

where $rand[0, 1]$ is a uniformly distributed random number, j_{rand} is a randomly chosen index which ensures that $\mathbf{U}_{i,G}$ inherits at least one component from $\mathbf{V}_{i,G}$ and $CR \in [0, 1]$ is the crossover rate.

2.1.4 Selection

Finally, a greedy selection is performed to determine the survivors of the next generation. Each trial vector is compared with its corresponding target vector and the best one survives:

$$\mathbf{X}_{i,G+1} = \begin{cases} \mathbf{U}_{i,G}, & \text{if} \quad f(\mathbf{U}_{i,G}) \leq f(\mathbf{X}_{i,G}) \\ \mathbf{X}_{i,G}, & \text{otherwise} \end{cases} \tag{3}$$

Hence, each population member either gets better or remains with the same objective value in each generation. Since members never deteriorate, it is considered to be a selection with high pressure. Note that in case of a tie, the trial vector survives.

2.2 Diversity preservation in evolutionary algorithms

Most EAs start with a set of diverse candidate solutions, and as the generations evolve, such a diversity is reduced [6]. This reduction on diversity might lead to premature convergence, which in fact is a common drawback in EAs. Thus, several variants to deal with this issue have been designed [11]. Depending on the component of the EA that is modified, these methods are usually classified in one the following groups [6]: *selection-based*, *population-based*, *crossover/mutation-based*, *fitness-based*, and *replacement-based*. Recently, the replacement-based strategies have attained quite good performance, therefore this section is devoted to this kind of methods. Particularly, two different strategies that are used to validate our proposal are discussed. One of the first techniques categorized as replacement-based are the *crowding* methods. A quite popular realization is the *Restricted Tournament Selection* [12] (RTS) strategy. In RTS after a new individual (C) is created, CF individuals from the current population are randomly selected. Then, C and its most similar individual—from those in the selected set—compete for a place in the population using a traditional binary tournament, i.e. the best one survives.

Other strategies are based on considering the diversity to calculate a fitness value for the replacement stage. Specifically, in the *Hybrid Genetic Search with Adaptive Diversity Control* (HGSADC) [32], the individuals—union of parents and offspring—are sorted by their contribution to diversity and by their original cost. In order to calculate the contribution to diversity of an individual, its mean distance to the closest N_{Close} individuals is calculated. For an individual I, the ranking in terms of diversity is denoted as $RD(I)$, whereas the ranking for the original cost is denoted as $RC(I)$. Then, the rankings are combined to generate the biased fitness value $BF(I)$ using Eq. 4. In each step of the replacement phase, the individual with the lowest biased fitness is erased and the ranks are re-calculated. This process is performed until the desired population size is attained. It is important to remark that this scheme requires the setting of the parameters: N_{Close} and N_{Elite}, whereas $N_{population}$ refers to the number of individuals that has not been erased yet by the replacement scheme.

$$BF(I) = RC(I) + \left(1 - \frac{N_{Elite}}{N_{population}}\right) RD(I) \tag{4}$$

2.3 Diversity in differential evolution

DE is highly susceptible to the loss of diversity, partially due to the greedy strategy applied in the selection phase. Thus, several analyses to better deal with this issue have been carried out. Since the general implications of each DE parameter on the diversity are known, one of the alternatives is to theoretically estimate proper values for the DE parameters [35]. Differently, some analyses regarding the effects of the magnitude of the difference vectors used in the mutation have also been performed [20]. Such analyses and additional empirical studies regarding the crossover allowed to conclude that some kind of movements should be disallowed to delay the convergence [22]. In this last study the kind of accepted movements varies along the run. Specifically, it discards movements with a norm below a threshold and this threshold decreases taking into account the elapsed generations. Other ways of altering the kind of accepted movements have been proposed [2]. Note that these kinds of methods have similarities with our proposal in the sense that decisions are biased by the number of elapsed generations. However, our method operates on the replacement strategy and not on the mutation phase. Moreover, these methods do not consider explicitly the differences appearing on the whole population. Instead, restrictions apply to the differences appearing in the reproduction phase.

A different alternative operates by altering the selection operator [25]. Particularly, the selection pressure is relaxed through a probabilistic selection to maintain the population diversity and consequently to allow escaping from basin of attraction of local optima. Since it considers the fitness to establish the acceptance probabilities, it is very sensitive to scale transformations. In this case, decisions are not biased by the elapsed generations. Finally, in the *Auto-Enhanced Population Diversity* (AEPD), the diversity is explicitly measured and it triggers a mechanism to diversify the population when a too low diversity is detected [34]. Strategies with similar principles but with different disturbance schemes have also been devised [37].

Note that DE variants with best performance in competitions do not apply these modifications and that most of these extensions have not been implemented in the most widely used optimization frameworks. As a result, these extensions are not so widely used in the community in spite of their important benefits for some cases.

3 Performance in IEEE CEC contests

In recent years, several contests have been organized at the IEEE CEC to facilitate comparisons among optimizers. Such contests usually define a set of optimization functions with different features and complexities, so analyzing the results through the years offers insights about which are the principles and algorithms that provide more advantages. This section is devoted to summarize the methods and ideas with more contributions, focusing the efforts on DE variants with the aim of detecting design tendencies on the DE field.

In CEC 2005 competition on real parameter optimization [29], classical DE attained the second rank and the self-adaptive DE variant called SaDE obtained the third rank in 10 dimensions. However, they performed poorly with more than 30 dimensions. Subsequently, in the 2008 competition on large scale global optimization [31], a self-adaptive DE (jDEdynNP-F) reached the third place, confirming the importance of parameter adaptation. In fact, in other kinds of competitions such as in the 2006 constrained optimization one, the benefits of adaptation was also shown, where SaDE obtained the third place. In a subsequent competition in large-scale optimization (CEC 2010), DE variants did not reach top ranks. This, together with the fact that several DE variants performed properly only in low-dimensionality, is an indicator of the weaknesses of DE in large scale problems. In fact, some of the reasons of the curse of dimensionality were analyzed in [26]. Thus, it is known that there is room for improvement in terms of scalability, although dealing with large-scale optimization is out of the scope of this paper. Finally, in CEC 2011 competition with real world optimization problems [7], hybrid algorithms including DE have performed properly. For instance, the second place was obtained by the hybrid DE called DE-Λ_{CR}. Again a Self-adaptive Multi-Operator DE (SAMODE) performed properly and obtained the third place.

In recent years, adaptive variants have also stood out. However, the complexity of the best schemes has increased considerably. In the 2014 competition on real parameter optimization [15], the first place was reached by the Linear Population Size Reduction Success-History Based Adaptive DE (L-SHADE). Similarly to other adaptive variants, this proposal adapts the internal parameters of DE and the success-history based variants are currently very well-known strategies. Additionally, in order to get a better degree between exploration and exploitation it dynamically reduces the population size. In the 2015 competition based on learning [16], a variant of the previous approach obtained the first place. Additionally, two DE variants with parameter adaptation attained the second and third place.

In this paper, experimental validation is focused on the CEC 2016 and CEC 2017 competitions in real parameter optimization. In the case of 2016 [16], the first place was reached with the United Multi-Operator Evolutionary Algorithm (UMOEAs-II).

This approach is not a DE scheme but some of the DE operators are taken into account. The second place was reached by Ensemble Sinusoidal Differential Covariance Matrix Adaptation with Euclidean Neighborhood (L-SHADE-EpSin) and the third place was attained by the Improved L-SHADE (iL-SHADE). Note that these two approaches were again variants of SHADE. In fact, variants of SHADE have also excelled in the learning-based competitions [17].

In the CEC 2017 case [33], the first place was obtained by the Effective Butterfly Optimizer with Covariance Matrix Adapted Retreat Phase (EBOwithCMAR), which is not a DE variant. EBOwithCMAR is an extension of UMOEAs-II. The second place was reached by jSO, which is an improvement of iL-SHADE. Finally, the L-SHADE-EpSin, again a variant of SHADE, attained the third place.

Attending to the features of the different approaches, the following trend is detected:

- Typically, the parameters are altered during the run with the aim of adapting the optimizer to the requirements of the different optimization stages.
- In some of the last algorithms, the adaptation considers the stopping criterion and elapsed generations to bias the decisions taken by the optimizer. For instance, some proposals decrease the population size and in other cases DE is modified to further intensify in last stages.
- The overall complexity of the winners has increased significantly. Particularly, several variants include modifications to perform promising movements with a higher probability, for instance by including the principles of the Covariance Matrix Adaptation scheme.

Our proposal takes the previous conclusions into consideration. However, our hypothesis is that for long-term executions simpler variants with explicit control of diversity are enough to excel and that some of the proposed modifications might be counter-productive. For instance, it is known that the parameter adaptation might provoke some improper movements that might affect performance in the long term [21]. Note that by controlling the diversity, the degree between exploration and exploitation can be properly altered automatically. As a result, parameter adaptation is not included in our proposal. Instead, taking into account the performance of some SHADE variants, a dynamic parameterization that relates the parameter values with the elapsed function evaluations is taken into account. We consider that incorporating an adaptive parameterization might be beneficial but it should be included carefully.

4 Proposal

Our proposal is motivated by two main works in the area of control of diversity in EAs. The first one is the empirical study developed by Montgomery et al [22], which presents several empirical analyses that confirm issues related to premature convergence in DE. The second work, by Segura et al. [27], provides significant improvements in the combinatorial optimization field by developing a novel replacement strategy called *Replacement with Multi-objective based Dynamic Diversity Control* (RMDDC) that relates the control of diversity with the stopping criterion and elapsed generations. Important benefits were attained by methods including RMDDC, so given the conclu-

Algorithm 1 General scheme of DE-EDM

1: Randomly initialize the N target vectors, where each one is uniformly distributed.
2: $G = 0$
3: **while** stopping criterion is not satisfied **do**
4: **for** $i = 1$ to N **do**
5: Mutation: Generate the mutant vector ($V_{i,G}$) according to Eq. (1).
6: Crossover: Use recombination to generate the trial vector ($U_{i,G}$) according to Eq. (2).
7: Selection: Update the elite vector ($E_{i,G}$ instead of $X_{i,G}$) according to Eq. (3).
8: Replacement: Select the target vectors (X_{G+1}) according to Algorithm 2.
9: $G = G + 1$

sions of these previous works, the proposal of this paper is a novel DE variant that includes an explicit mechanism that follows some of the principles of RMDDC. Since this is applied in the context of DE, the parent population and offspring population are referred to as target vectors and trial vectors respectively, and the term vector is used with the same meaning as individual. This novel optimizer is called *Differential Evolution with Enhanced Diversity Maintenance* (DE- EDM) and its source code is freely available.[1]

The core of DE- EDM (see Algorithm 1) is quite similar to the standard DE. In fact, the way of creating new trial vectors is not modified at all (lines 5 and 6). The novelty is the incorporation of elite vectors (E) and a novel diversity-based replacement strategy. The former, records the vectors that have the best objective function value in relation with each target vector, therefore there are N elite vectors, which must be considered as a multi-set since repeated individuals might appear. In order to select the members of the elite vectors, the original greedy replacement of DE is used (line 7). In the case of the replacement strategy (line 8), which is in charge of selecting the next target vectors, it follows the same principle that guided the design of RMDDC, i.e. vectors that contribute too little to diversity should not be accepted as members of the next generation. In this way, the greedy selection strategy of DE is not used to maintain the target vectors (X). In order to establish the minimum acceptable diversity contribution to be selected, the stopping criterion and elapsed generations are taken into account. One of the main weaknesses of RMDDC is that its convergence is highly delayed. Thus, in order to promote a faster convergence than in RMDDC two modifications are performed. First, no concepts of the multi-objective field are applied, instead a more greedy selection is taken into account. Second, the elite vectors are also considered as an input of the replacement strategy.

Our replacement strategy (see Algorithm 2) operates as follows. It receives as input the target vectors, the trial vectors, and the elite vectors. In each generation it must select the N trial vectors of the next generation. First, it calculates a desired minimum distance between selected vectors (D_t) given the current number of elapsed function evaluations (line 2). Then, it joins the three multi-set of vectors in a multi-set of current vectors (line 3). Then, the multi-set of survivors and penalized vectors are initialized to the empty set (line 4). In order to select the N survivors (next target vectors) an iterative process is repeated (lines 5–13). In each step the best vector in *Current*, i.e. the one with best objective function is selected to survive, i.e. it is moved to *Survivors*

[1] The code in C++ can be downloaded in the next link https://github.com/joelchaconcastillo/Diversity_DE_Research.git/.

Algorithm 2 Replacement Phase

1: Input: $Target\ vectors$, $Trial\ vectors$, and $Elite$
2: Update $D_t = D_I - D_I * (nfes/(0.90 * max\_nfes))$
3: $Current = Target \cup Trial \cup Elite$.
4: $Survivors = Penalized = \emptyset$.
5: **while** $|Survivors| < N$ And $|Current| > 0$ **do**
6: $Selected$ = Select the best vector of $Current$.
7: Remove $Selected$ from $Current$.
8: Copy $Selected$ to $Survivors$.
9: Find the vectors from $Current$ with a distance to $Selected$ lower than D_t and move them to $Penalized$. Normalized distance is considered (Eq. 5).
10: **while** $|Survivors| < N$ **do**
11: $Selected$ = Select the vector from $Penalized$ with the largest distance to the closest vector in $Survivors$.
12: Remove $Selected$ from $Penalized$.
13: Copy $Selected$ to $Survivors$.
14: **return** $Survivors$

(line 6–8). Then, the vectors in *Current* with a distance lower than D_t to the selected vector are transferred to *Penalized* (line 9). The way to calculate the distance between two vectors is by using the normalized Euclidean distance described in Eq. 5, where D is the dimension of the problem, and a_d, b_d are the minimum and maximum bounds of dimension d. In cases where *Current* is empty previous to the selection of N vectors, *Survivor* is filled by selecting in each step the vector in $Penalized$ with the largest distance to the closest vector in *Survivor* (lines 10–13).

$$distance(X, Y) = \frac{\sqrt{\sum_{d=1}^{D} \left(\frac{X_d - Y_d}{b_d - a_d}\right)^2}}{\sqrt{D}} \tag{5}$$

In order to complete the description it is important to specify the logic behind the way of calculating D_t. The value of D_t is used to alter the degree between exploration and exploitation so it should depend on the optimization stage. Specifically, this value should be reduced as the stopping criterion is reached with the aim of promoting exploitation. In our scheme, an initial value for D_t (D_I) must be set. Then, similarly than in [27], a linear reduction of D_t is performed by taking into account the elapsed function evaluations and stopping criterion. Particularly, in this work, the stopping criterion is set by function evaluations. The reduction is calculated in such a way that by the 90% of maximum number of evaluations the resulting D_t value is 0. Therefore, in the remaining 10% diversity is not considered at all, meaning that intensification is promoted. Thus, if max_nfes is the maximum number of evaluations and $nfes$ is the elapsed number of evaluations, D_t is calculated as $D_t = D_I - D_I * (nfes/(0.90 * max\_nfes))$.

The initial distance (D_I) affects the performance of DE- EDM. If this parameter is fixed large enough, then at the first optimization stages the algorithm aims to maximize the diversity of the target vectors, so a proper exploration is performed which is very important in some kinds of problems such as highly multi-modal and deceptive ones. Thus, the effect of premature convergence might be alleviated. However, a too large D_I might induce too much exploration resulting in an improper exploitation phase. In the opposite case, a too low D_I might result in an improper exploration phase, thus hindering the avoidance of local optima. Depending on the kind of fitness landscape and

stopping criterion, the optimal D_I might vary. For instance, deceptive and highly multi-modal problems usually require larger values than uni-modal problems. However, in our proposal, D_I is not adapted to each problem, instead an experimental study to check the robustness of different D_I value is attached in the experimental validation section.

In the same way that in the standard DE, in DE- EDM the crossover probability (CR) and the mutation factor (F) must be set. The first one is perhaps the most important for the performance according to several studies developed by Montgomery et al. [21]. These authors empirically proved that extreme CR values lead to vastly different search behaviors. They explained that low CR values result in a search that is aligned with a small number of search space axes and induce small displacements. This provokes a gradual and slow convergence that in some scenarios might result in a robust behavior. Additionally, high CR values might generate higher quality solutions with a lower probability. However, these transformations provoke large displacements that could improve significantly the solutions when they are successful. According to this, we employ both high and low CR values as is indicated in Eq. 6.

$$CR = \begin{cases} Normal(0.2, 0.1), & \text{if } \ rand[0, 1] \leq 0.5 \\ Normal(0.9, 0.1), & \text{otherwise} \end{cases} \tag{6}$$

Following the principles of several SHADE variants [1,3], the elapsed function evaluations are considered in the random generation of the mutation factor F. Particularly, each F is sampled from a Cauchy distribution (Eq. 7).

$$Cauchy(0.5, 0.5 * nfes/max\_nfes) \tag{7}$$

Therefore, at the initial optimization stages, F values near to 0.5 are generated with a high probability. Then, as the execution advances, the density function suffers a gradual transformation and the variance is increased, meaning that values outside the interval [0.0, 1.0] are generated with a higher probability. In the cases when values larger than 1.0 are generated, the value 1.0 is used. In the case of generating a negative value, the F is re-sampled. This means that the probability of generating large F-values increases as the execution progresses. The principle behind this decision is to help in the avoidance of fast convergence.

5 Experimental validation

This section presents the experimental study carried out to validate the performance of DE- EDM. Specifically, we show that by explicitly controlling the diversity in DE, results of state-of-the-art algorithms are improved further. Particularly, the benchmarks of CEC 2016 and CEC 2017 are considered. Each one of them is composed of thirty different problems, meaning that the validation is performed with a set of large and diverse functions. The kind of problems of each benchmark is divided in uni-modal, simple multi-modal, hybrid and composition functions. Table 1 shows the problems

Table 1 Problems grouped by properties for the CEC 2016 and CEC 2017 benchmarks

Type function	CEC 2016	CEC 2017
Uni-modal	f_1–f_3	f_1–f_3
Simple multi-modal	f_4–f_{16}	f_4–f_{10}
Hybrid (multi-modal)	f_{17}–f_{22}	f_{11}–f_{20}
Composition (multi-modal)	f_{23}–f_{30}	f_{21}–f_{30}

belonging to each category for the CEC 2016 and CEC 2017 benchmarks. In the hybrid functions, the variables are randomly divided into some sub-components and each one is related to basic functions. In the same line, the composition functions merge the properties of several sub-functions and maintains continuity around each local optima. Also the local optima which has the smallest bias value is the global optimum. The search space is bounded by the range $\Omega = \prod_{j=1}^{D}[-100, 100]^D$.

The experimental validation takes into account the algorithms that attained the first places of each year competition, as well as the standard DE. Additionally, comparisons against other diversity-based schemes are included. The algorithms considered from the CEC 2016 are UMOEAs-II [10] and L-SHADE-EpSin [1], which achieved the first and second places respectively. Similarly, the top algorithms from CEC 2017 are taken into account, i.e. EBOwithCMAR [13] and jSO [4]. Following the recommendations given in [19], all these algorithms are tested with both benchmarks.

Given that the optimizers taken into account are stochastic algorithms, each execution was repeated 51 times with different seeds. In every case, the stopping criterion was set to 25×10^6 functions evaluations. In addition, problems were configured by setting $D = 10$. The validation follows the guidelines of CEC benchmark competitions and the statistical tests proposed in [9] are also included. Note that, as it is usual in these competitions, when the gap between the values of the best solution found and the optimal solution is 10^{-8} or smaller, the error is treated as 0. The parameterization indicated by the authors was used in every algorithm and it is as follows:

- **EBOwithCMAR**: For EBO, the maximum population size of $S_1 = 18D$, minimum population size of $S_1 = 4$, maximum population size of $S_2 = 146.8D$, minimum population size of $S_2 = 10$, historical memory size H = 6. For CMAR Population size $S_3 = 4 + 3log(D)$, $\sigma = 0.3$, CS = 50, probability of local search $pl = 0.1$ and $cfe_{ls} = 0.4 * FE_{max}$.
- **UMOEAs-II**: For MODE, maximum population size of $S_1 = 18D$, minimum population size of $S_1 = 4$, size memory H = 6. For CMA-ES Population size $S_2 = 4 + \lfloor 3log(D) \rfloor$, $\mu = \frac{N}{2}$, $\sigma = 0.3$, CS = 50. For local search, $cfe_{ls} = 0.2 * FE_{max}$.
- **jSO**: Maximum population size $= 25log(D)\sqrt{D}$, historical memory size H = 5, initial mutation memory $M_F = 0.5$, initial probability memory $M_{CR} = 0.8$, minimum population size = 4, initial p-best $= 0.25 * N$, final p-best = 2.
- **L-SHADE-EpSin**: Maximum population size $= 25log(D)\sqrt{D}$, historical memory size H = 5, initial mutation memory $M_F = 0.5$, initial probability memory $M_{CR} = 0.5$, initial memory frequency $\mu_F = 0.5$, minimum population size = 4, initial p-best $= 0.25 * N$, final p-best = 2, generations of local search $G_{LS} = 250$.

- **DE-EDM**: $D_I = 0.3$, population size $= 250$, CR $\sim Normal(\{0.2, 0.9\}, 0.1)$, F $\sim Cauchy(0.5, 0.5 * nfes/max_{nfes})$.
- **Standard-DE**: population size $= 250$ (operators as DE- EDM), CR $\sim Normal(\{0.2, 0.9\}, 0.1)$, F $\sim Cauchy(0.5, 0.5 * nfes/max_{nfes})$.

Our experimental analyses is based on the error, i.e. the difference between the optimal solution and the best obtained solution. In order to statistically compare the results, a similar guideline than the one proposed in [9] was used. First a Shapiro-Wilk test was performed to check whatever or not the values of the results followed a Gaussian distribution. If, so, the Levene test was used to check for the homogeneity of the variances. If samples had equal variance, an ANOVA test was done; if not, a Welch test was performed. For non-Gaussian distributions, the non parametric Kruskal-Wallis test was used to test whether samples are drawn from the same distribution. An algorithm X is said to win algorithm Y when the differences between them are statistically significant, and the mean and median error obtained by X are lower than the mean and median achieved by Y.

$$\begin{aligned} Score_1 &= \left(1 - \frac{SE - SE_{min}}{SE}\right) \times 50, \\ Score_2 &= \left(1 - \frac{SR - SR_{min}}{SR}\right) \times 50, \end{aligned} \tag{8}$$

Tables 2 and 3 offer a summary of the results obtained for CEC 2016 and CEC 2017, respectively. The column tagged with "Always Solved" shows the number of functions where a zero error was obtained in the 51 runs. Additionally, column tagged with "At least one time solved" shows the number of functions that were solved at least in one run. Practically all functions (28 of them) of the CEC 2017 benchmark were solved with DE- EDM at least one time. Additionally, 21 functions of the CEC 2016 were also solved. This contrasts with the results obtained by state-of-the-art algorithms. They were able to reach optimal values in significantly less functions. In order to confirm the superiority of DE- EDM, the pair-wise statistical tests already described were used. The column tagged with the symbol ↑ shows the number of cases where the superiority of each method could be confirmed, whereas the column tagged with

Table 2 Summary of the results obtained by DE- EDM and state-of-the-art algorithms—CEC 2016

Algorithm	Always solved	At least one time solved	Statistical tests			Score
			↑	↓	⟷	
DE-EDM	13	21	77	25	48	100.00
UMOEAs-II	9	14	51	31	68	62.45
Standard-DE	11	19	50	46	54	56.29
jSO	9	17	47	51	52	55.43
EBOwithCMAR	8	14	35	56	59	50.28
L-SHADE-Epsilon	7	13	20	71	59	50.12

Table 3 Summary of the results obtained by DE- EDM and state-of-the-art algorithms—CEC 2017

Algorithm	Always solved	At least one time solved	Statistical tests			Score
			↑	↓	⟷	
DE-EDM	21	28	88	6	56	100.00
Standard-DE	12	21	56	29	65	42.91
EBOwithCMAR	9	18	34	46	70	37.14
L-SHADE-Epsilon	8	19	7	81	62	32.78
jSO	8	15	29	55	66	29.30
UMOEAs-II	11	15	43	40	67	26.89

the symbol ↓ counts the number of cases where the method was inferior. Finally, the number comparisons with not significant differences are shown in the column tagged with the symbol ⟷. The results of the statistical tests show that DE- EDM attained the best results in both years. The number of victories in CEC 2016 and CEC 2017 were 77 and 88, whereas the number of losses were 25 and 6, respectively. DE- EDM is the approach with the largest number of victories and lowest number of losses in both benchmarks, confirming the superiority of the proposal. The last column—tagged with "Score"—considers the official score of CEC's competitions. Particularly, the raking of the algorithms is attained by taking into account the two scores defined in Eq. (8). Then, the final score is calculated as the sum $Score = Score_1 + Score_2$. In Eq. (8), the SE of an algorithm is the sum of the mean error values obtained in the 30 benchmark functions, i.e. $SE = \sum_{i=1}^{30} error\_f_i$. Then, SE_{min} is the minimal SE from all the algorithms. In order to calculate SR and SR_{min}, algorithms are sorted in each function in base of the attained mean error. Then, a rank is assigned to each algorithm in base of such an ordering. Finally, the SR of a method is the sum of the ranks obtained for each function and SR_{min} is the minimal SR from all the algorithms.

Note that in base of this definition, the best attainable score is 100. This happens when a given approach obtains both SR_{min} and SE_{min}. DE- EDM attained the best attainable score in both years, which confirms its clear superiority when compared both with state-of-the-art and standard DE. In these long-term executions, standard DE attained the third and second places in the problems of the CEC 2016 and CEC 2017, respectively. This means that the performance of the state-of-the-art algorithms is not so impressive in long-term executions.

Since our proposal is based on the explicit control of diversity, Fig. 1 shows the evolution of the mean of the diversity calculated as the mean distance to the closest vector with the aim of better understanding its behavior. Particularly, functions f_1 and f_{30} were selected for this analysis because they have quite different features (easy uni-modal vs. complex multi-modal). The left side shows the diversity of the Elite population. It is remarkable that, while there are no direct constraints in the Elite population related to diversity, the diversity is implicitly maintained. The right side shows the diversity of the target vectors. As expected, diversity decreases in a gradual way and a degree of diversity is maintained until the 90% of the total function evaluations is reached.

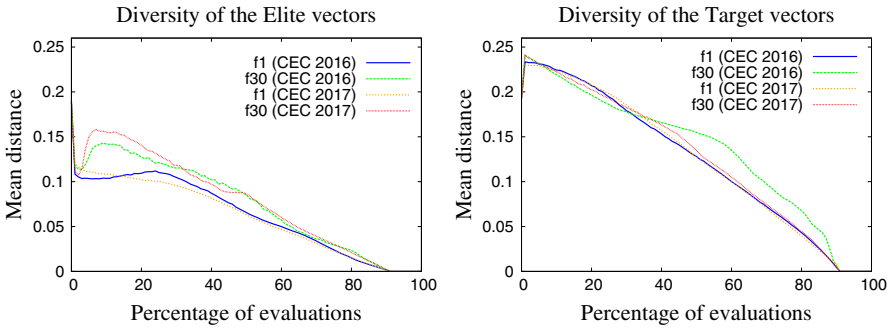

Fig. 1 Mean distance to the closest vector of the 51 executions with the problems f_1 and f_{30} (CEC 2016 and CEC 2017). The initial distance factor considered corresponds to $D_I = 0.3$

Finally, in order to provide comparable results of our proposal, Tables 4 and 5 report the best, worst, median, mean, standard deviation and success rate for both benchmarks. These tables show that all the uni-modal problems were solved by our proposal. Additionally, several simple and even some of the most complex multi-modal functions were optimally solved. In fact, several complex functions that had never been solved by state-of-the-art could be solved by DE- EDM.

5.1 Comparison of diversity replacement-based schemes

In order to better validate the advantages provided by the replacement phase proposed in this paper, comparisons against two additional diversity replacement-based strategies were developed. Particularly, the replacement-based methods taken into consideration are the *Restricted Tournament Selection* (RTS) and the *Hybrid Genetic Search with Adaptive Diversity Control* (HGSADC). These replacement-based strategies were incorporated to the standard DE framework. The three variants were executed with the same parameterization. Note that both RTS and HGSADC require additional parameters. Based on preliminary analyses, the RTS was executed with a sample size $CF = 25$, and HGSADC was executed with $N_{Close} = 1$, and $N_{Elite} = 8$. The remaining configuration follows the same parameterization previously defined. Tables 6 and 7 summarize the results attained for both benchmarks with the same meaning that in the previous experiment. Again, DE- EDM attained the best results in both years. It is clear that DE- EDM attained a significantly higher $Score$ than the remaining diversity-based methods. Note that the main difference of our proposal in constrast to the schemes RTS and HGSADC lies in the fact that they provide modifications with the aim of delaying convergence but in an indirect way, meaning that premature convergence might not always be avoided. Second, while these methods have been quite successful in comparison to Evolutionary Algorithms with a high selection pressure, DE already incorporates a replacement strategy that follows some of the principles of crowding. Thus, advantages of incorporating RTS and HGSADC in DE are diminished. Therefore, making decisions by taking into account the stopping criterion and elapsed generations seems to be the key to properly avoid premature convergence in long-term executions.

Table 4 Results for DE- EDM in the CEC 2016 problems

	Best	Worst	Median	Mean	Std.	Succ. rate
f_1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_6	0.00E+00	3.60E−02	4.00E−03	7.39E−03	1.15E−02	3.92E−01
f_7	2.00E−02	1.02E−01	5.90E−02	5.77E−02	4.93E−02	0.00E+00
f_8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{10}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{11}	0.00E+00	6.00E−02	0.00E+00	5.88E−03	1.90E−02	9.02E−01
f_{12}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{13}	1.00E−02	8.00E−02	5.00E−02	4.67E−02	2.60E−02	0.00E+00
f_{14}	1.00E−02	5.00E−02	3.00E−02	2.82E−02	2.13E−02	0.00E+00
f_{15}	0.00E+00	4.70E−01	2.20E−01	1.99E−01	1.55E−01	1.96E−02
f_{16}	4.00E−02	1.50E−01	8.00E−02	8.47E−02	4.96E−02	0.00E+00
f_{17}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{18}	0.00E+00	2.00E−02	1.00E−02	7.65E−03	6.32E−03	3.14E−01
f_{19}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{20}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{21}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{22}	0.00E+00	3.00E−02	0.00E+00	3.73E−03	2.76E−02	7.65E−01
f_{23}	0.00E+00	1.00E+02	0.00E+00	2.55E+01	5.10E+01	7.45E−01
f_{24}	0.00E+00	6.90E−01	0.00E+00	2.61E−02	1.33E−01	9.61E−01
f_{25}	1.00E+02	1.00E+02	1.00E+02	1.00E+02	0.00E+00	0.00E+00
f_{26}	8.00E−02	1.00E+02	5.29E+01	5.20E+01	3.19E+01	0.00E+00
f_{27}	2.50E−01	9.10E−01	5.40E−01	5.60E−01	2.92E−01	0.00E+00
f_{28}	0.00E+00	3.57E+02	3.43E+02	2.76E+02	1.60E+02	1.96E−01
f_{29}	1.00E+02	1.00E+02	1.00E+02	1.00E+02	0.00E+00	0.00E+00
f_{30}	1.84E+02	1.84E+02	1.84E+02	1.84E+02	3.25E−02	0.00E+00

5.2 Empirical analyses of the initial distance factor

In our proposal the diversity is explicitly promoted and the total amount of diversity maintained in the population depends on the initial distance factor D_I. Therefore, the effect of this parameter on the quality is analyzed in this section. Particularly, the same scheme previously mentioned was taken into account. However, several initial distance factors were considered ($D_I = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1\}$).

Figure 2 shows the mean success rate attained for both benchmarks when considering different D_I values. The most relevant conclusions are:

Table 5 Results for DE- EDM in the CEC 2017 problems

	Best	Worst	Median	Mean	Std.	Succ. ratio
f_1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{10}	0.00E+00	1.20E−01	0.00E+00	1.65E−02	3.39E−02	7.45E−01
f_{11}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{12}	0.00E+00	2.20E−01	0.00E+00	6.37E−02	1.76E−01	6.67E−01
f_{13}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{14}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{15}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{16}	0.00E+00	2.10E−01	0.00E+00	2.47E−02	7.27E−02	8.82E−01
f_{17}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{18}	0.00E+00	1.00E−02	0.00E+00	1.96E−03	4.47E−03	8.04E−01
f_{19}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{20}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{21}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{22}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{23}	0.00E+00	3.00E+02	0.00E+00	3.49E+01	1.03E+02	8.82E−01
f_{24}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{25}	0.00E+00	1.00E+02	0.00E+00	3.92E+00	2.00E+01	9.61E−01
f_{26}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{27}	0.00E+00	3.87E+02	3.87E+02	2.05E+02	2.68E+02	1.96E−02
f_{28}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{29}	1.45E+02	2.26E+02	2.18E+02	1.99E+02	4.21E+01	0.00E+00
f_{30}	3.95E+02	3.95E+02	3.95E+02	3.95E+02	2.10E−01	0.00E+00

Table 6 Summary of the results obtained with different replacement strategies—CEC 2016

Algorithm	Always solved	At least one time solved	Statistical tests			Score
			↑	↓	↔	
DE-EDM	13	21	51	1	8	100.00
RTS	2	7	3	47	10	19.74
HGSADC	3	15	21	27	12	44.12

Table 7 Summary of the results obtained with different replacement strategies—CEC 2017

Algorithm	Always solved	At least one time solved	Statistical tests			Score
			↑	↓	⟷	
DE-EDM	21	28	49	0	11	100.00
RTS	4	12	2	49	9	30.91
HGSADC	6	18	23	25	12	40.86

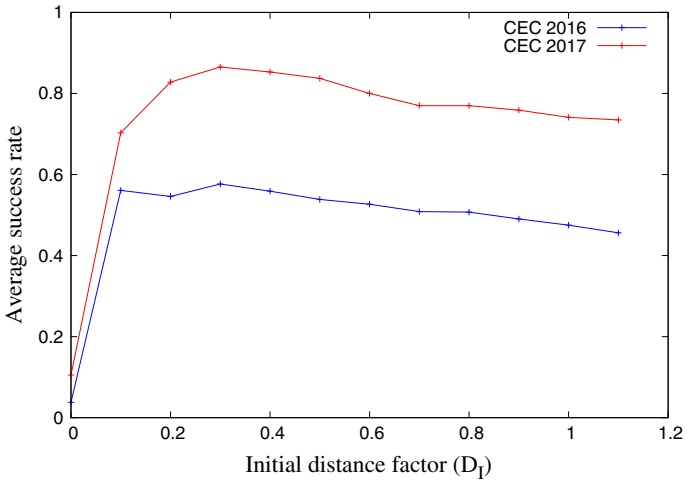

Fig. 2 Mean success rate for different D_I in the CEC 2016 and CEC 2017 benchmarks with a population size equal to 250 and 25×10^6 function evaluations

- If diversity is not promoted ($D_I = 0.0$) there is an important degradation in the performance.
- The performance is quite robust in the sense that a large range of D_I values provide good enough results. For instance, values in the range $[0.2, 0.6]$ provide high-quality solutions.
- To a certain extent, if the initial distance factor (D_I) is very large (values larger than 0.6), the quality of solutions is affected.

To summarize, DE- EDM incorporates a novel parameter which is important in its performance. However, results are quite robust in the sense that even if it is not tuned for each problem, high-quality results can be obtained and that a large range of values provide competitive results.

6 Conclusions and future work

Premature convergence is one of the most important drawbacks of DE. This paper proposes a novel variant of DE, the *Differential Evolution with Enhanced Diversity Maintenance* (DE- EDM), which incorporates a novel replacement phase that is based

on a recently proposed strategy (RMDDC) that was successfully used with memetic algorithms in the combinatorial optimization field. In order to attain a faster convergence than in RMDDC two modifications are proposed for the integration with DE. First, no concepts of the multi-objective field are applied, meaning that a more greedy strategy is devised. Second, an elite population is incorporated. Experimental validation shows the important benefits obtained by DE- EDM in long-term executions both when compared with a classic DE variant and with state-of-the-art proposals. DE- EDM requires setting a new parameter, which is called the initial distance factor. The obtained quality depends on the proper specification of this parameter. However, fixed values in a large range could be used for properly dealing with benchmarks of the CEC 2016 and CEC 2017 competitions, meaning that a quite robust proposal is obtained.

Several extensions might be explored to further improve our proposal. First, with the aim of facilitating the application of DE- EDM, adaptive and/or self-adaptive schemes might be applied for setting the initial distance factor, as well as for other DE parameters. Second, with the aim of reducing the number of evaluations required to attain high-quality solutions, some of the strategies that are used in the field of optimization with expensive functions might be integrated into our proposal. Finally, the incorporation of a local search engine mainly in the last stages of the optimization might bring additional benefits.

Acknowledgements Authors acknowledge the financial support from CONACyT through the "Ciencia Básica" Project No. 285599.

References

1. Awad, N.H., Ali, M.Z., Suganthan, P.N., Reynolds, R.G.: An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. In: 2016 IEEE Congress on Evolutionary Computation (CEC'16), pp. 2958–2965. IEEE (2016)
2. Bolufé-Röhler, A., Estévez-Velarde, S., Piad-Morffis, A., Chen, S., Montgomery, J.: Differential evolution with thresheld convergence. In: 2013 IEEE Congress on Evolutionary Computation (CEC'13), pp. 40–47. IEEE (2013)
3. Brest, J., Maučec, M.S., Bošković, B.: iL-SHADE: improved L-SHADE algorithm for single objective real-parameter optimization. In: 2016 IEEE Congress on Evolutionary Computation (CEC'16), pp. 1188–1195. IEEE (2016)
4. Brest, J., Maučec, M.S., Bošković, B.: Single objective real-parameter optimization: algorithm jSO. In: 2017 IEEE Congress on Evolutionary Computation (CEC'17), pp. 1311–1318. IEEE (2017)
5. Chakraborty, U.K.: Advances in Differential Evolution, vol. 143. Springer, Berlin (2008)
6. Črepinšek, M., Liu, S.H., Mernik, M.: Exploration and exploitation in evolutionary algorithms: a survey. ACM Comput. Surv. **45**(3), 35:1–35:33 (2013)
7. Das, S., Suganthan, P.N.: Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems. Jadavpur University, Nanyang Technological University, Kolkata (2010)
8. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evol. Comput. **15**(1), 4–31 (2011)
9. Durillo, J.J., Nebro, A.J., Coello, C.A.C., Garcia-Nieto, J., Luna, F., Alba, E.: A study of multiobjective metaheuristics when solving parameter scalable problems. IEEE Trans. Evol. Comput. **14**(4), 618–635 (2010)

10. Elsayed, S., Hamza, N., Sarker, R.: Testing united multi-operator evolutionary algorithms-ii on single objective optimization problems. In: 2016 IEEE Congress on Evolutionary Computation (CEC'16), pp. 2966–2973. IEEE (2016)
11. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: Foundations of Genetic Algorithms, vol. 2, pp. 187–202. Elsevier, Amsterdam (1993)
12. Harik, G.R.: Finding multimodal solutions using restricted tournament selection. In: ICGA, pp. 24–31 (1995)
13. Kumar, A., Misra, R.K., Singh, D.: Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase. In: 2017 IEEE Congress on Evolutionary Computation (CEC'17), pp. 1835–1842. IEEE (2017)
14. Lampinen, J., Zelinka, I., et al.: On stagnation of the differential evolution algorithm. In: Proceedings of MENDEL, pp. 76–83 (2000)
15. Liang, J., Qu, B., Suganthan, P.: Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore (2013)
16. Liang, J., Qu, B., Suganthan, P., Chen, Q.: Problem Definitions and Evaluation Criteria for the CEC 2015 Competition on Learning-Based Real-parameter Single Objective Optimization. Technical Report 201411A, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore (2014)
17. Liu, B., Chen, Q., Zhang, Q., Liang, J., Suganthan, P., Qu, B.: Problem definitions and evaluation criteria for computational expensive optimization. In: Technical Report (2013)
18. Locatelli, M., Vasile, M.: (Non) convergence results for the differential evolution method. Optim. Lett. **9**(3), 413–425 (2015)
19. Molina, D., Moreno-García, F., Herrera, F.: Analysis among winners of different IEEE CEC competitions on real-parameters optimization: Is there always improvement? In: 2017 IEEE Congress on Evolutionary Computation (CEC'17), pp. 805–812. IEEE (2017)
20. Montgomery, J.: Differential evolution: difference vectors and movement in solution space. In: 2009 IEEE Congress on Evolutionary Computation, pp. 2833–2840 (2009). https://doi.org/10.1109/CEC.2009.4983298
21. Montgomery, J., Chen, S.: An analysis of the operation of differential evolution at high and low crossover rates. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2010)
22. Montgomery, J., Chen, S.: A simple strategy for maintaining diversity and reducing crowding in differential evolution. In: 2012 IEEE Congress on Evolutionary Computation (CEC'12), pp. 1–8 (2012)
23. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965)
24. Price, W.: Global optimization by controlled random search. J. Optim. Theory Appl. **40**(3), 333–348 (1983)
25. Sá, Â.A., Andrade, A.O., Soares, A.B., Nasuto, S.J.: Exploration vs. exploitation in differential evolution. In: AISB 2008 Convention Communication, Interaction and Social Intelligence, vol. 1, p. 57 (2008)
26. Segura, C., Coello, C.A.C., Hernández-Díaz, A.G.: Improving the vector generation strategy of differential evolution for large-scale optimization. Inf. Sci. **323**, 106–129 (2015)
27. Segura, C., Coello, C.A.C., Segredo, E., Aguirre, A.H.: A novel diversity-based replacement strategy for evolutionary algorithms. IEEE Trans. Cybern. **46**(12), 3233–3246 (2016)
28. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1997)
29. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL Rep. **2005005**, 2005 (2005)
30. Talbi, E.: Metaheuristics: From Design to Implementation. Wiley Series on Parallel and Distributed Computing. Wiley, New York (2009)
31. Tang, K., Yáo, X., Suganthan, P.N., MacNish, C., Chen, Y.P., Chen, C.M., Yang, Z.: Benchmark Functions for the CEC2008 Special Session and Competition on Large Scale Global Optimization, vol. 24. Nature Inspired Computation and Applications Laboratory, USTC, China (2007)
32. Vidal, T., Crainic, T.G., Gendreau, M., Prins, C.: A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Comput. Oper. Res. **40**(1), 475–489 (2013)

33. Wu, G., Mallipeddi, R., Suganthan, P.: Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report (2016)
34. Yang, M., Li, C., Cai, Z., Guan, J.: Differential evolution with auto-enhanced population diversity. IEEE Trans. Cybern. **45**(2), 302–315 (2015)
35. Zaharie, D.: Control of population diversity and adaptation in differential evolution algorithms. Proc. MENDEL **9**, 41–46 (2003)
36. Zhang, J., Sanderson, A.C.: Jade: adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput. **13**(5), 945–958 (2009)
37. Zhao, L., Sun, C., Huang, X., Zhou, B.: Differential evolution with strategy of improved population diversity. In: 2016 35th Chinese eControl Conference (CCC), pp. 2784–2787. IEEE (2016)