Noname manuscript No.
(will be inserted by the editor)

Differential Evolution with Enhanced Diversity Maintenance

Joel Chacón Castillo · Carlos Segura

Received: date / Accepted: date

Abstract Differential Evolution (DE) is a popular population-based metaheuristic 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 proposed. The proposal is integrated with DE to generate the DE with Enhanced Diversity Maintenance (DE-EDM). The main novelty is the use of a dynamic balance between exploration and exploitation to adapt the proposal 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 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 [1] and applied in many fields, such as in science, economic and engineering [2]. Among them, Differential Evolution (DE) [3] is one of the most effective strategies to deal with continuous optimization. In fact, it has

Joel Chacón Castillo
Centro de Investigación en Matemticas, Guanajuato, Mexico
E-mail: joel.chacon@cimat.mx

Carlos Segura
Centro de Investigación en Matemticas, Guanajuato, Mexico
E-mail: carlos.segura@cimat.mx

been the winning strategy of several optimization competitions [4]. Similarly to other EAs, DE is inspired by the natural evolution process and it involves the application of mutation, recombination 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 [5] and the Controlled Random Search (CRS) [6] optimizers.

In spite of the effectiveness of DE, there exists several weaknesses that have been detected and partially solved by extending the standard variant [4]. Among them, the sensitivity to its parameters [7], the appearance of stagnation due to the reduced exploration capabilities [8, 9] and premature convergence [10] 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 [1], 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.

Since the appearance of DE, some criticism appeared because of its incapability to maintain a large enough diversity due to the use of a selection with high pressure [8]. Thus, several extensions of DE to deal with premature convergence have been devised such as parameter adaptation [10], auto-enhanced population diversity [11] and selection strategies with a lower selection pressure [8]. Some of the last studies on design of population-based meta-heuristics [12] 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 [13]. 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, the method was designed with the aim of inducing a larger degree of intensification.

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 section 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 section 4. The experimental validation, which includes comparisons against state-of-the-art approaches, is shown in section 5. Finally, our conclusions and some lines of future work are given in section 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 and it has been extensively used to generate more complex DE variants [4]. In fact, our proposal also extends the classic variant. However, in order to perform fair comparisons, our experimental validation takes into account state-of-the-art approaches that incorporate more complex components and even algorithms not belonging to the DE field.

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 $\mathbf{X} = [x_1, x_2, ..., x_D]$, 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 $f : \Omega \subseteq \Re^D \to \Re$. 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 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 ith vector of the population at the generation G is denoted as $\mathbf{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, ..., X_{D,i,G}]$. 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 NP vectors. Since there is commonly no information about the performance of different regions, uniform random generators are usually applied. Hence, the jth component of the ith 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 and 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:

$$\mathbf{V}_{i,G} = \mathbf{X}_{r1,G} + F \times (\mathbf{X}_{r2,G} - \mathbf{X}_{r3,G}) \quad r1 \neq r2 \neq r3 \tag{1}$$

The indices $r1, r2, r3 \in [1, NP]$ are different integers randomly chosen from the range $[1, NP]$. 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 $\mathbf{U_{i,G}} = [u_{1,i,G}, u_{2,i,G}, ..., u_{D,i,G}]$. The most typical crossover is the *binomial* one, which operates as follows:

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

where $rand_{i,j}[0, 1]$ 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:

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

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 in Differential Evolution

DE is highly susceptible to the loss of diversity due to the greedy strategy applied in the selection phase. However, several analyses to better deal with this issue have been carried out. Since the general implications of each parameter on the diversity are known, one of the alternatives is to theoretically estimate proper values for the DE parameters [10]. Differently, some analyses regarding the effects of the norm of the difference vectors used in the mutation have also been performed [14]. Such analyses and additional empirical studies regarding the crossover allowed to conclude that some kind of movements might be disallowed to delay the convergence [15]. In this last study the kind of accepted movements varies along the run. Specifically, it discards movements with a size 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 [16]. 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, the restrictions apply to the differences appearing in the reproduction phase.

A different alternative operates by altering the selection operator [8]. 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 [11]. Strategies with similar principles but with different disturbance schemes have also been devised [17].

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 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 define 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 offer 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 [18], 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 [19], 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 the top rank. This, together with the fact that the performance of 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 [20]. 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 [21], 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 have increased considerably. In the 2014 competition on real parameter optimization [22], 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. In order to get a better degree between exploration and exploitation it dynamically reduces the population size. In the 2015 competition based on learning [23], 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 [23], 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 the two last ones were again variants of SHADE. In fact, variants of SHADE have also excelled in the learning-based competitions [24].

In the CEC 2017 case [25], 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 have 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 [26]. Note that by controlling the diversity, the degree between exploration and exploitation can be properly altered automatically. As a result, parameter adaptation or modifications to alter the probability of different movements are not included in our proposal. We consider that some of these modification might be beneficial but they 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 [15], which presents several empirical analyses that confirm issues related to premature convergence in DE. The second work, by Segura et al. [13], 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 conclusions 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. 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 solutions is not modified at all (lines 5 and 6). The novelty is the incorporation of an elite population (E) and a novel diversity-based replacement strategy. In order to select the members of the elite population, the original greedy replacement of DE is used (line 7). On the other way, the replacement strategy (line 8), which is in charge of selecting the next population members, follows the same principle that guided the design of RMDDC, i.e. individuals 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 parent population (X). In order

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

Algorithm 1 General scheme of DE-EDM

Randomly initialize the population of NP individuals, where each one is uniformly distributed.

$G = 0$
while stopping criterion is not satisfied **do**
  for $i = 1$ to NP **do**
    Mutation: Generate the mutant vector ($V_{i,G}$) according to Eq. (1).
    Crossover: use recombination to generate the trial vector ($U_{i,G}$) according to Eq. (2).
    Selection: Update the elite vector ($E_{i,G}$ instead of $X_{i,G}$) according to Eq. (3).
  Replacement: Select the target vectors (X_{G+1}) according to Algorithm 2 .
  $G = G + 1$

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 population is also considered as an input of the replacement strategy.

Our replacement strategy (see Algorithm 2) operates as follows. It receives as input the parent population (target vectors), the offspring population (trial vectors), and the elite population. In each generation it must select the NP vectors of the next parent population. First, it calculates the desired minimum distance D_t given the current number of elapsed function evaluations (line 2). Then, it joins the three populations in a set of current members (line 3). The current members set contains vectors that might be selected to survive. Then, the set of survivors and penalized individuals are initialized to the empty set (line 4). In order to select the NP survivors (next parent population) an iterative process is repeated (lines 5 - 13). In each step the best individual in the *Current set*, i.e. the one with best objective function is selected to survive, i.e. it is moved to the *Survivor set* (line 6 - 8). Then, individuals in the *Current set* with a distance metric lower than D_t are transferred to the *Penalized set* (line 9). The way to calculate the distance between two individuals is by using the normalized Euclidean distance described in Eq. 4, where D is the dimension of the problem, and a_d, b_d are the minimum and maximum bounds of each dimension (d). In cases where the *Current set* is empty previous to the selection of NP individuals, the *Survivor set* is filled by selecting in each step the individual in *Penalized* with the largest distance to the closest individual in the *Survivor set* (lines 10 - 13).

$$distance(x_i, x_j) = \frac{\sqrt{\sum_{d=1}^{D}\left(\frac{x_i^d - x_j^d}{b_d - a_d}\right)^2}}{\sqrt{D}} \tag{4}$$

In order to complete the description it is important to specify the way to calculate D_t and the methodology to update the elite individuals. All the remaining steps are maintained as in the classic DE variant. The value of D_t is used to alter the degree between exploration and explotation so it should depend on the optimization stage. Specifically, this value should be reduced as

Algorithm 2 Replacement Phase

Input: *Population* (*target vectors*), *Offspring* (*trial vectors*), and *Elite*
Update $D_t = D_I - D_I * (nfes/(0.95 * max\_nfes))$
$Current = Population \cup Offspring \cup Elite$.
$Survivors = Penalized = \emptyset$.
while $|Survivors| < NP$ And $|Current| > 0$ **do**
    Selected = Select the best individual of *Current*.
    Remove *Selected* from *Current*.
    Copy *Selected* to *Survivors*.
    Find the individuals from *Current* with a distance to *Selected* lower than D_t and move them to *Penalized*. Normalized distance is considered (Eq. 4).
while $|Survivors| < NP$ **do**
    Selected = Select the individual from *Penalized* with the largest distance to the closest individual in *Survivors*.
    Remove *Selected* from *Penalized*.
    Copy *Selected* to *Survivors*.
return *Survivors*

the stopping criterion is reached with the aim of promoting explotation. In our scheme, an initial value for D_t (D_I) must be set. Then, similarly than in [13], 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 95% of maximum number of evaluations the resulting D_t value is 0. Therefore, in the remaining 5% diversity is not considered at all. Thus, if max_nfes is the maximum number of evaluations and $nfes$ is the elapsed number of evaluations, D_t can be calculated as $D_t = D_I - D_I * (nfes/(0.95 * max\_nfes))$.

The initial distance (D_I) heavily 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 population, so a proper exploration is performed which is very important in some kinds of problems such as highly multimodal and deceptive ones. Thus, the effect of premature convergence might be alleviated. A too large D_I might induce too much exploration so a proper exploitation phase is not performed. In the opposite case, a too low D_I might avoid the exploration phase, so avoiding local optima is more difficult. 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 unimodal 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.

Similarly that 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. [26]. These authors empirically proved that extremes CR values leads 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 successful. According to this, we employ both high and low CR values as it is showed in Eq. 5.

$$CR = \begin{cases} Normal(0.2, 0.1), & \text{if} \quad rand[0,1] \leq 0.5 \\ Normal(0.9, 0.1), & \text{otherwise} \end{cases} \tag{5}$$

Following the principles of several SHADE variants [27, 28], the function evaluations are considered in the random generation of the mutation factor F. Particularly, each F is sampled through a Cauchy distribution (Eq. 6).

$$Cauchy(0.5, 0.5 * nfes/max\_nfes) \tag{6}$$

Therefore, at the first optimization stages, F values near to 0.5 are generated. 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 resampled. One of the effects of this approach is to increase the probability of generating large F-values as the execution progresses with the aim of avoiding a fast convergence.

5 Experimental Study

In this section the experimental validation is presented. 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. The state-of-the-art is composed by the algorithms that attained the first places of each year competition. Additionally, the standard DE was included. Thus, the algorithms considered from the CEC 2016 are UMOEAs-II [29] and L-SHADE-EpSin [27] that achieved the first and second place respectively. Similarly, the top algorithms from CEC 2017 are EBOwithCMAR [30] and jSO [31]. It is interesting to remark that EBOwithCMAR is considered as an improvement of the UMOEAs-II. Additionally, jSO and L-SHADE-EpSin belong to the SHADE's family. All these algorithms are tested with both benchmarks as it is suggested by [32].

Given that all of them are stochastic algorithms, each execution was repeated 51 times with different seeds. In every case, the stopping criterion was set to $25,000,000$ functions evaluations. We performed our evaluation following the guidelines of CEC benchmark competitions. Thus, if the gap between the values of the best solution found and the optimal solution was 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 $S_2 = 4 + \lfloor 3log(D) \rfloor$, $\mu = \frac{PS}{2}$, $\sigma = 0.3$, 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.
– **Standard-DE**: population size = 250 (operators as DE-EDM).

Our experimental analyses have been performed in base of 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 [33] 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 obtained by X are higher than the mean and median achieved by Y.

In tables 1 and 2 a summary of the results obtained for CEC 2016 and CEC 2017 are shown, 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 to optimality at least in one run. Practically all functions (28 of them) of the CEC 2017 benchmark were solved with our proposal at least one time. Additionally, 21 functions of the CEC 2016 were also solved. This constrast 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 superioriy of DE-EDM, pair-wise statisticall test were used. The column tagged with the symbol ↑ shows the number of times that the superiority of each method could be confirmed, whereas the column tagged with the symbol ↓ count the number of cases where the method was inferior.

Finally, the number comparisons whose differences were not significant are shown in the column tagged with the symbol $\longleftrightarrow$. The statistical tests indicate that the DE-EDM attained the best results in both years. The number of wins in CEC 2016 and CEC 2017 were 77 and 88 respectively. Also the number of losts were of 25 and 6 respectively. Additionally, the last place attained in both years was by the L-SHADE-Epsilon with 20 wins in 2016 and 7 wins in 2017. The last column tagget with "Score" shows the analyses proposed in the CEC's competitions. Particularly, the evaluation method combines two scores defined in the equation (7). Thus the final score is composed by the sum $Score = Score_1 + Score_2$.

$$\begin{aligned} Score_1 &= \left(1 - \frac{SE - SE_{min}}{SE}\right) \times 50, \\ Score_2 &= \left(1 - \frac{SR - SR_{min}}{SR}\right) \times 50, \end{aligned} \tag{7}$$

Here, SE_{min} is the minimal sum of errors from all the algorithms, and SE is the sum of error values $SE = \sum_{i=1}^{30} error\_f_i$. Also, SR_{min} is the minimal sum of ranks from all the algorithms, namely the sum of each rank in each function for the considered algorithms $SE = \sum_{i=1}^{30} error\_f_i$. Principally, our proposal attained the best scores (100.00) in both years, showing its superiority. Additionally, the Standard-DE attained good enough results, in fact it got the third and second places in CEC 2016 and CEC 2017 respectively. This shows that the performance of the state-of-the-art algorithms is different considering long-term executions. Specifically, although that in CEC 2017 the L-SHADE-Epsilon algorithm got the lowest number of wins in the statistical test it showed a competitive score. This might occurs since that the statistical scores considers both mean and median errors. Morever, the score considers a rank and mean based in the error.

Giving that our proposal is based in the explicitly control of the diversity and with the aim of a better understanding of its behavior in the figure 1 is showed the diversity through the elapsed function evaluatons. Particularly, the DE-EDM was executed with the functions f_1 and f_{30}. Specifically, based in their properties the first function is easily solved (unimodal) and the second is one of the most difficult (hybrid). In the left side is showed the diversity of the Elite population. Although there are not constraints in the Elite population to lost the diversity, it seems that in both functions f_1 and f_{30} the diversity is implicitly maintained. Similarly, in the right side is showed the diversity of the trial vectors. It shows that the diversity is explicitly maintained as is desired (i.e. until the 95% of the total function evaluations).

In order, to provide comparable results of our proposal, in the tables 3 and 4 are reported the best, worst, median, mean, standard deviation and success ratio. Particularly, these tables show that the uni-modal were solved by our proposal. Also, several simple multi-modal functions were adequatelly approximated. Principally, our proposal solved several complex functions (e.g. Composition Functions) that were not solved by the state-of-the-art.

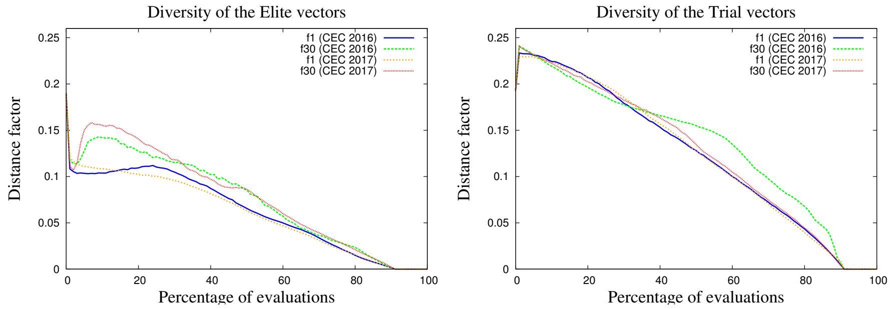

Fig. 1 Average DCN of the 51 executions with the problems f_1 and f_{30} (CEC 2016 and CEC 2017). The initial distance factor considered corresponds to $D_I = 0.3$.

Table 1 Summary results - CEC 2016

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

Table 2 Summary results - CEC 2017

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

5.1 Empirical analyses of the initial distance factor

In our proposal the diversity is explicitly promoted through several stages, which are controlled with the initial distance factor D_I. Therefore, the effect of this parameter is analysed in detail. Particularly, the general configuration of the experimental validation is taken into account. Thus, several initial distance factors were considered ($D_I = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1\}$).

In the figure 2 is showed the average success ratio vs. the initial distance factor D_I. The most relevant points are described as follows:

- If the diversity is not promoted ($D_I = 0.0$) the performance of the algorithms is seriously implicated.

Table 3 Results for DE based diversity CEC 2016 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	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

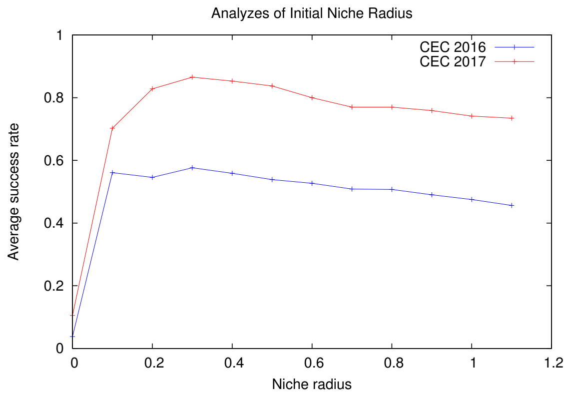

Fig. 2 Average success rate with different initial distance factors in the benchmark of CEC 2016 and CEC 2017, is considered a population size of 250 and 25,000,000 function evaluations.

– In this scenario the ideal configuration is $D_I = 0.3$, although that the range $[0.1, 0.4]$ also provides quality solutions.
– If the diversity of the solutions increases (after a range) the quality of solutions is implicated.

Table 4 Results for DE based diversity 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

Finally, its important stand out that the solutions are less affected by the population size, however there is still present a relation between the D_I and the population size.

6 Conclusions and future works

Based in the experimental results carried out in this work, the next conclusions can be drawn. Principally, the premature convergence can be treated adequately through an advanced replacement phase and considering an elite population. Also, our proposal is able to enhance the performance of the state-of-the-art, as well the standard-DE in long-term executions. Additionally, the DE-EDM only requires the population size and the initial distance factor, which is considered an advantage since that several recient algorithms are over-parameterized. Finally, differently to the DE algorithms, our proposal is not extremely affected by small population sizes.

Several extesions could be explored as alternateive or to further improve our proposal, the most important are described as follows. Firstly, some strategies that involves the expensive optimization field should be analysed, thus several functions evaluations could be avoided to save resources. Additionally, a self-adaptive scheme of the initial distance factor should be develop to provide a stable behavior. Aslo, to explore the possibility of involve a local search scheme. Thus, the irregular fitness landscapes could be solved adequately (e.g. multi-modal problems). Applying our proposal to real-world problems could provide high quality solutions (e.g. deal with combinatorial problems). Generate a theoretical model to select the adequately population size given a initial distance factor. Finally, implement the replacement procedure to the Estimation of Distribution Algorithms seems a promising field.

References

1. E. Talbi, *Metaheuristics: From Design to Implementation.* Wiley Series on Parallel and Distributed Computing (Wiley, 2009). URL `https://books.google.com.mx/books?id=SIsa6zi5XV8C`
2. U.K. Chakraborty, *Advances in differential evolution*, vol. 143 (Springer, 2008)
3. R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, Journal of global optimization **11**(4), 341 (1997)
4. S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE transactions on evolutionary computation **15**(1), 4 (2011)
5. J.A. Nelder, R. Mead, A simplex method for function minimization, The computer journal **7**(4), 308 (1965)
6. W. Price, Global optimization by controlled random search, Journal of Optimization Theory and Applications **40**(3), 333 (1983)
7. J. Zhang, A.C. Sanderson, Jade: adaptive differential evolution with optional external archive, IEEE Transactions on evolutionary computation **13**(5), 945 (2009)
8. Â.A. Sá, A.O. Andrade, A.B. Soares, S.J. Nasuto, in *AISB 2008 Convention Communication, Interaction and Social Intelligence*, vol. 1 (2008), vol. 1, p. 57
9. J. Lampinen, I. Zelinka, et al., in *Proceedings of MENDEL* (2000), pp. 76–83
10. D. Zaharie, in *Proc. of MENDEL*, vol. 9 (2003), vol. 9, pp. 41–46
11. M. Yang, C. Li, Z. Cai, J. Guan, Differential evolution with auto-enhanced population diversity, IEEE transactions on cybernetics **45**(2), 302 (2015)
12. M. Črepinšek, S.H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: A survey, ACM Computing Surveys **45**(3), 35:1 (2013)
13. C. Segura, C.A.C. Coello, E. Segredo, A.H. Aguirre, A novel diversity-based replacement strategy for evolutionary algorithms, IEEE transactions on cybernetics **46**(12), 3233 (2016)
14. J. Montgomery, in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on* (IEEE, 2009), pp. 2833–2840
15. J. Montgomery, S. Chen, in *Evolutionary Computation (CEC), 2012 IEEE Congress on* (IEEE, 2012), pp. 1–8
16. A. Bolufé-Röhler, S. Estévez-Velarde, A. Piad-Morffis, S. Chen, J. Montgomery, in *Evolutionary Computation (CEC), 2013 IEEE Congress on* (IEEE, 2013), pp. 40–47
17. L. Zhao, C.j. Sun, X.c. Huang, B.x. Zhou, in *Control Conference (CCC), 2016 35th Chinese* (IEEE, 2016), pp. 2784–2787
18. P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization, KanGAL report **2005005**, 2005 (2005)
19. K. Tang, X. Yáo, P.N. Suganthan, C. MacNish, Y.P. Chen, C.M. Chen, Z. Yang, Benchmark functions for the cec2008 special session and competition on large scale global optimization, Nature Inspired Computation and Applications Laboratory, USTC, China **24** (2007)
20. C. Segura, C.A.C. Coello, A.G. Hernández-Díaz, Improving the vector generation strategy of differential evolution for large-scale optimization, Information Sciences **323**, 106 (2015)

21. S. Das, P.N. Suganthan, 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)
22. J. Liang, B. Qu, P. Suganthan, 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)
23. J. Liang, B. Qu, P. Suganthan, Q. Chen, Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization, Technical Report201411A, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore (2014)
24. B. Liu, Q. Chen, Q. Zhang, J. Liang, P. Suganthan, B. Qu, in *Technical Report* (2013)
25. G. Wu, R. Mallipeddi, P. Suganthan, 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)
26. J. Montgomery, S. Chen, in *Evolutionary Computation (CEC), 2010 IEEE Congress on* (IEEE, 2010), pp. 1–8
27. N.H. Awad, M.Z. Ali, P.N. Suganthan, R.G. Reynolds, in *Evolutionary Computation (CEC), 2016 IEEE Congress on* (IEEE, 2016), pp. 2958–2965
28. J. Brest, M.S. Maučec, B. Bošković, in *Evolutionary Computation (CEC), 2016 IEEE Congress on* (IEEE, 2016), pp. 1188–1195
29. S. Elsayed, N. Hamza, R. Sarker, in *Evolutionary Computation (CEC), 2016 IEEE Congress on* (IEEE, 2016), pp. 2966–2973
30. A. Kumar, R.K. Misra, D. Singh, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 1835–1842
31. J. Brest, M.S. Maučec, B. Bošković, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 1311–1318
32. D. Molina, F. Moreno-García, F. Herrera, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 805–812
33. J.J. Durillo, A.J. Nebro, C.A.C. Coello, J. Garcia-Nieto, F. Luna, E. Alba, A Study of Multiobjective Metaheuristics When Solving Parameter Scalable Problems, IEEE Transactions on Evolutionary Computation **14**(4), 618 (2010)