**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. Principally, in this paper is proposed a replacement strategy and an elite population, which are designed to preserve diversity. The proposal is extended and integrated with DE to generate the DE with Enhanced Diversity Maintennce (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. Severals 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, which are present in the population to explore the search space. In this sense is similar to 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 metaheuristics [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 metaheuristics [12] show that properly balancing the exploration and intensification is particularly useful for avoiding premature convergence. 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 extended 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

Several extensions of DE that affect its exploration capabilities have been devised [4]. In this work, in order to properly show the benefits of our extension, our proposal is applied with the classic DE scheme (DE/rand/1/bin). However, 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. This section is devoted to summarize the classic DE variants and to introduce some of the most important terms used in 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 dimensionality 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 $\Omega = \prod_{j=1}^{D} [a_i, b_i]$.

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 $i$th vector of the population at the generation $G$ is denoted as $\mathbf{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, ..., X_{D,i,G}]$. 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 $j$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 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, differences are low and 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 similarity 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 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 several of the 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 a 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, the benefits of adaptation was also shown, where SaDE obtained the third place.

In 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.

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-$\Lambda_{CR}$. 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 n 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 (EBOwithC-MAR), 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, 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 the optimizer. For

instance, some proposals decrease the population size and in other cases the DE are modified to intensify further 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

Principally, our proposal is motivated by two remarkable works in the area of diversity-based EAs. First, several studies related with premature convergence, which were developed by Montgomery et al. [15]. They stablished several empirical studies to diagnose the premature convergence that appear in DE. The second work, provides significant improvements and a solid literature revision in the combinatorial optimization field, it was considered by Segura et al. [13]. Particularly, they incorporated a novel replacement strategy called *The Replacement with Multiobjective based Dynamic Diversity Control* (RMDDC) to relate the control of diversity with the stopping criterion and elapsed generation. 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 similar principles to RMDDC. This novel optimizer is called *Differential Evolution with Enhanced Diversity Maintenance* (DE-EDM) and its source code is freely available [1].

The main novelty of DE-EDM is the incorporation of a replacement strategy. The main principle of the novel replacement strategy is similar to the one used in RMDDC, i.e. individuals that contribute too little to diversity should not be accepted as target vectors of the next generation. At the same time, the solutions of high quality are recorded in the elite population. Moreover, in order to establish the minimum acceptable diversity contribution the stopping criterion and elapsed generations are taken into account.

The contribution to diversity is estimated with the Distance to Closest Neighbour metric (DCN), i.e. given a set of already selected individuals ($Survivors$),

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

the contribution of a non-selected one is calculated as the minimum distance to any of the individuals in $Survivors$. However, 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, a new population called the elite population is used.

**Algorithm 1** General scheme of DE-EDM

Randomly initialize the population of $NP$ individuals, where each one is uniformly distributed.<br>
$G = 0$<br>
**while** stopping criterion is nor satisfied **do**<br>
    **for** $i = 1$ to $NP$ **do**<br>
        Mutation: Generate the donor vector ($V_{i,G}$) according Eq. (1).<br>
        Crossover: Recombine the mutate vector ($U_{i,G}$) according Eq. (2).<br>
        Selection: Update the elite vector ($E_{i,G}$ instead of $X_{i,G}$) according Eq. (3).<br>
        Replacement: Select the target vectors ($X_{i,G+1}$) according to algorithm 2 .<br>
    $G = G + 1$

RMDDC is a replacement strategy that considers the maximization of the diversity contribution of each individual as an explicit objective. Then, the notion of Pareto dominance is used to select the survivors. It uses a dynamic threshold to prevent the selection of individuals with low contribution to diversity. One of the main weaknesses of this method is that its convergence is highly delayed. Thus, executions of several days were required to attain high-quality solutions. As a result our proposal incorporates two extensions to alleviate such a drawback. DE-EDM alters the replacement strategy of DE with the aim of controlling the balance between exploration and exploitation.

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 the each generation it must select the next parent population (target vectors). Additionally, it calculates the desired minimum distance $D_t$ given the current number of function evaluations. First, it joins the three populations in a set of current members. 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. In order to select the $NP$ survivors (target vectors) an iterative process is repeated. In each step the best individual in the *Current set*, i.e. the one with best objective function is selected to survive (i.e. moved to the *Survivor set*). Then, the individuals in the *Current set* with a DCN to the selected individuals lower than $D_t$ are transferred to the *Penalized set*. In case were the *Current set* is empty previous to the selection of $NP$ individuals, the *Survivor set* is filled by selecting in each step the Penalized individuals with the largest DCN to the *Survivor set*.

**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** $|Survivors| < NP$ OR $|Current| > 0$ **do**
    Select the best individual ($Current_{best}$) of $Current$.
    Find the individuals with lowest DCN to $D_t$ between $Current$ and $Current_{best}$ and move them to $Penalized$ (considering the normalized distance of Eq. (4)).
    Move the best individual $Current_{best}$ to $Survivors$.
**while** $|Survivors| < pop\_size$ **do**
    Select the individual $Penalized$ with maximum DCN.
    Move the individual from $Penalized$ to $Survivors$.
**return** $Survivors$

In order to complete the description is important to specify the way to calculate $D_t$ and the methodology to update the elite individuals. The remaining steps are maintained as in the classic DE variant. The value of $D_t$ should depend on the optimization stage. Specifically, this value should be reduced as the stopping criterion is approached. In our scheme, an initial $D_I$ value must be set. Then, a linear reduction of $D_t$ is done. Particularly, in this work, the stopping criterion is set by function evaluations (*nfes*). The reduction is calculated in such way that by the 95% of maximum number of evaluations the resulting $D_t$ value is 0. Therefore, in the rest 5% the behavior is similar to the classical DE. Thus, if $max\_nfes$ is the maximum number of evaluations and $nfes$ is the elapsed number of evaluations, $D_t$ can be calculated as $D_t = D_I - D_I * (nfes/(0.95 * max\_nfes))$. We consider a linear model reduction based in empirical studies of similar works [13], where they indicated that a linear decrement provides the most stable results.

Principally, the replacement phase considers a defined radius ($D_t$), which for simlicity is considered through the normalized euclidean distance (Eq. 4), however other distances could be implemented (e.g. mahalanobis distance). It is important to mention that the normalized distance is needed, thus each dimension is equally important, also this simplifies the setting of the $D_I$ parameter, since that the maximum distance is the unity, which is a fraction of the main space diagonal.

$$distance(x_{seed}, x_j) = \frac{\sqrt{\sum_{d=1}^{D}\left(\frac{x_{seed}^d - x_j^d}{max_d - min_d}\right)^2}}{\sqrt{D}} \tag{4}$$

Theoretically, the effect of the initial distance factor ($D_I$) affects directively the behavior of the algorithm. If this parameter is fixed too large, then at the first optimization stages the algorithm aims to maximize the DCN, thus the problems that should need a high level of diversity, which usually are the most difficult, are properly solved (e.g. deceptive problems). On the other hand, a low initial distance factor results in a lower level exploration, therefore it provokes a high level of exploitation. Depending in the optimization stage and the fitness landscape it could be benefical, principally in multi-modal problems where each basis of attraction seems to be promising and each one of them should have the suitable quantity of computationals efforts. It is interesting to take into account that our proposal could be traslated to the multi-modal fashion just setting a final distance factor, which should be strictly positive, however this trend is not considered in this work. Therefore, when the initial distance factor is setted several factors should be taken into account. One of them is the maximum number of evaluations, when the problem is setted with few number of evaluations, the initial distance factor should be lower, since the exploitation should be promoted. On the other hand, when the problem is configured with a large number of evaluations, then the initial distance factor should be higher. Also the number of vectors should be considered, since that they are directly realted with the diversity. Although our proposal should take into account the previously details, there exist an usual range where the results are enough stable, it is showed in the empirical analyzes.

Our proposal provides several advantages in constrast of the standard DE and some of the state-of-the-art algorithms, perhaps the most important is that it is not over-parametrized. Although several top-rank algorithms present good enough results, usually these algorithms need a tunning phase to fit the parameters, therefore in real application it could be infeasible.

Specifically, the standard DE recives two parameters, these are the crossover probability ($CR$) and the mutation factor ($F$). The first one is perhaps the most important according to several studies showed 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 not just aligned with a small number of search space axes, it also shows small displacements, which provokes a gradual and slow convergence that in some scenarios migh result in a robust behavior. On the other hand, high $CR$ values migh generate few quality solutions, however these solutions provoke large displacements and could improve significantly the solutions. According this, we employ both high and low $CR$ values as is showed in the equation 5.

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

In the same vein, the mutation factor $F$ is computed as follows. For each vector is sampled a $F$ value with a Cauchy distribution $Cauchy(0.5, 0.5 * nfes/max\_nfes)$. Particularly, it is based in several empirical results, since that at the first optimizaion stages the density function is located in 0.5. While the execution transcurs, the density function changes and the values generated are closest to 0.0 and to 1.0, since the Cauchy density function has heavly tails and the values are truncated to $[0.0, 1.0]$, thus it avoids stagnation in different stages of the algorithm.

Principally, our proposal is based in several ideas of the mentioned works, and are listed as follows:

- Is considered a threshold to control explicitly the convergence of the solutions.
- This threshold decreases over the algorithm's run.

– The selection operator is relaxed in the sense that it does not provokes premature convergence, all this through an elite population.

## 5 Experimental Study

In this section the experimental validation is carried out. Specifically is showed that controlling the diversity in a classic DE, is a way to improve further some of the results obtained by the state-of-the-art algorithms. 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 conformed by the algorithms that correspond to the first places of each year. Thus, the algorithms considered from the CEC 2016 are UMOEAs-II [27] and L-SHADE-EpSin [28] that are the first and second place respectively. Also the top algorithms from CEC 2017 are EBOwithCMAR [29] and jSO [30]. It is interesting to take into account that EBOwithCMAR is an improvement of the UMOEAs-II. Also, jSO and L-SHADE-EpSin are considered from the SHADE's family.

Given that all of them are stochastic algorithms, each execution was repeated 51 times with different seeds. The stopping criterion was set to 25,000,000 functions evaluations. We performed our evaluation following the guidelines of CEC benchmark competitions. According this, 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 specific parameterization of each one tested algorithm is as follows:

- **EBOwithCMAR**: For EBO 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**: Initial population size (N) $= 25log(D)\sqrt{D}$, historical memory size H= 5, initial mutation memory $M_F = 0.5$, initial probability memory $M_{CR} = 0.8$, maximum population size $= N$, minimum population size $=$ 4, initial p-best $= 0.25 * N$, final p-best $= 2$.
- **L-SHADE-EpSin**: Initial population size (N) $= 25log(D)\sqrt{D}$, historical memory size H= 5, initial mutation memory $M_F = 0.5$, initial probability memory $M_{CR} = 0.5$, initial memory frequency $\mu_F = 0.5$, maximum population size $= N$, minimum population size $= 4$, initial p-best $= 0.25 * N$, final p-best $= 2$, generations of local search $G_{LS} = 250$.
- **Diversity-DE**: Initial niche radius $D_I = 0.3$, population size $= 250$, $F = Cauchy(0.5, nfes/max\_nfes)$.

Our experimental analyzes has been performed in base of the error between the true optimal and the optimal obtained. In order to statistically compare the results, a similar guideline than the one proposed in [31] 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, if the mean and median obtained by $X$ are higher than the mean and median achieved by $Y$.

In the tables 1 and 2 are showed the summary of CEC 2016 and CEC 2017 respectively. The statistical tests indicate that the diversity DE algorithm provides significantly better results that the state-of-the-art algorithms in both benchmarks. Although that our proposal loses with the functions $\{f_6, f_7, f_{13}, f_{14}, f_{28}\}$ in CEC 2016 and $\{f_{12}, f_{16}, f_{18}\}$ for CEC 2017, it is important to take into account that our proposal provides acceptable and in some problems reach to the optimal. In fact based in a preliminary study this functions are solved at least one time with different configurations (radius niche and populations). The column named "Always Solved" indicates the number of functions that have a zero error in the 51 runs and the column named "At least one time solved" indicates the number of functions that reach to the optimal at least with one run. Almost all functions were solved in CEC 2017 with our proposal (28 functions) and more than a half in CEC 2016, however the state-of-the-art only were able to reach the optimal values in approximately a half of the functions in both years.

Based in the guideline of the CEC, the "Score" is computed as follows. The evaluation method combines two scores defined in the equation (6). Thus the final score is composed by the sum $Score = Score_1 + Score_2$.

$$\begin{aligned} Score_1 &= \left(1 - \frac{SE - SE_{min}}{SE}\right) \times 50, \\ Score_2 &= \left(1 - \frac{SR - SR_{min}}{SR}\right) \times 50, \end{aligned} \tag{6}$$

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 $SE = \sum_{i=1}^{30} error\_f_i$. Based in the final score the results provided for our proposal are superior in both years. Moreover, in both years the multi-operator algorithms have a superior score than the SHADE's algorithms. However it seems that the EBOwithCMAR algorithm could suffer an over-parametrization, since that it is deteriorated in CEC 2017 considering long-term executions. Principally, the winners of the CEC 2016 outperform to the winners of the CEC 2017 in long-term. Similarly, the winners of the CEC 2017 outperform to the winners of the CEC2016, this is a highlight that each algorithm is specifically tunned to short-term executions and for each

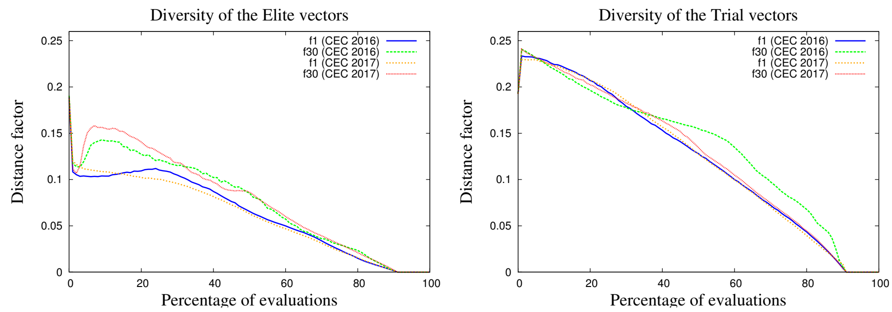


**Fig. 1** Average distance to the closest individual of the 51 executions with the problems $f_1$ and $f_{30}$ (CEC2016 and CEC2017). The initial distance factor considered corresponds to $D_I = 0.3$.

**Table 1** Summary results - CEC 2016

| Algorithm | Always solved | At least one time solved | Score | Statistical Tests | | |
|---|---|---|---|---|---|---|
| | | | | ↑ | ↓ | ⟷ |
| **EBOwithCMAR** | 8 | 14 | 64.88 | 26 | 2 | 49 |
| **jSO** | 9 | 17 | 51.29 | 38 | 41 | 41 |
| **UMOEAs-II** | 9 | 14 | 51.52 | 14 | 57 | 57 |
| **L-SHADE-Epsilon** | 7 | 13 | 56.10 | 42 | 22 | 56 |
| **Proposal** | 13 | 21 | 100.00 | 64 | 19 | 37 |

**Table 2** Summary results - CEC 2017

| Algorithm | Always solved | At least one time solved | Score | Statistical Tests | | |
|---|---|---|---|---|---|---|
| | | | | ↑ | ↓ | ⟷ |
| **EBOwithCMAR** | 11 | 15 | 26.20 | 28 | 36 | 56 |
| **jSO** | 8 | 19 | 36.66 | 27 | 39 | 54 |
| **UMOEAs-II** | 9 | 18 | 40.71 | 37 | 30 | 53 |
| **L-SHADE-Epsilon** | 8 | 15 | 35.37 | 7 | 62 | 51 |
| **Proposal** | 21 | 28 | 100.00 | 73 | 5 | 42 |

benchmark. In the figure 1 is showed the diversity of our proposal through the time of the Elite vectors (left side) and Trial vectors (right side). Despite the fact that the Elite vectors could lose the diversity, it shows that both in easy functions ($f_1$) and difficult functions ($f_{30}$) is implicitly maintained the diversity. On the other hand the trial vectors maintain correctly the diversity until the 95% of the total evaluations.

The error values between the best fitness values found in each run out of 51 runs and true optimal value are calculated and then best, worst, median, mean, standard deviation and success ratio of the error values are presented in each column in the tables 3 and 4. These tables show that the uni-modal functions and almost all the hybrid functions were solved. Approximately a half of the composition functions are solved with at least one run. However our proposal has problems solving the multi-modal functions, this can be provoked since that our proposal does not applies an advanced strategy to deal with the

**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 |

incremented distribution of difference vectors. Since that the algorithm finds some niches through the optimization process, the mutation provokes high displacements, that as result some regions are not analyzed properly. To deal with the previously issue, we suggest apply a matting restriction or implement a local search, which could further improve the convergence.

## 5.1 Sensitive analyses of the initial radius niche

In our proposal the diversity is explicitly promoted through several stages given an initial radius niche or distance factor $D_I$. Therefore, the robustness of this parameter is analyzed as follows. Based in the configurations of the experimental validation are executed several distance factors configurations ($D_I = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1\}$).

In the figure 2 is showed the average success ratio vs. the initial distance factor $D_I$. The main conclusions obtained are as follows:

- If the diversity is not promoted ($D_I = 0.0$) the performance of the algorithms is seriously implicated.
- In this scenario the ideal configuration is $D_I = 0.3$, although that the range $[0.1, 0.4]$ also provides quality solutions.

**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 |

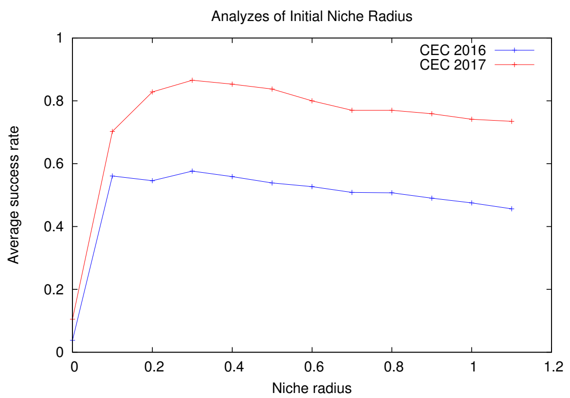


**Fig. 2** Average success rate with different initial distance factors in the benchmark of CEC 2016 and CEC 2017, is considered a population size of 250 and 25000000 function evaluations.

– If the diversity of the solutions increases (after a range) the quality of solutions is implicated.

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 Conclusion

From the experimental results in this paper, several conclusions can be drawn.

Firstly, from experimental investigation on the working mechanism, it can be seen that our proposal is able to relieve the premature convergence to several optimization levels. Secondly, our proposal is able to enhance the performance of DE algorithms, in particular when the search space is large. Third, it is also less sensitive to the parameter of population size, so our proposal can also be competitive even if the population size is small. Fourth, it seems that our proposal has some drawback in relation with the proportion of difference vectors.

For future work of this paper, two interesting issues should be addressed for our proposal. The first one is that explored areas in the search space should be avoided to save computing resources. Development an adaptive strategy for the distance factor should involve a more stable algorithm. Explore the possibility of implement a local search scheme with two goals, save function evaluations and tackle the current multi-modal problem. Applying our proposal to real-world problems should be an interesting topic. Based in several analyzes the mutation factor could be selected inside the distance factor, then develop a strategy where this parameter is no required. Generate a theoretical model to select the adequately population size given a initial distance factor.

## 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. S. Elsayed, N. Hamza, R. Sarker, in *Evolutionary Computation (CEC), 2016 IEEE Congress on* (IEEE, 2016), pp. 2966–2973
28. 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
29. A. Kumar, R.K. Misra, D. Singh, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 1835–1842
30. J. Brest, M.S. Maučec, B. Bošković, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 1311–1318
31. 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)