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 is proposed a replacement strategy that with an elite population aims to preserve diversity explicitly. The proposal is extended and 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

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

1 Introduction

2 Literature Review

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 [1], 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 [2], 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 [3]. 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 [4], 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 [5], 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 [6], 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 [6], 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 [7].

In the CEC 2017 case [8], 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 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 [9]. 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 [10], which presents several empirical analyses that confirm issues related to premature convergence in DE. The second work, by Segura et al. [11], provides significant improvements in the combinatorial optimization field by

Algorithm 1 General scheme of DE-EDM

```
1: Randomly initialize the population of NP individuals, where each one is uniformly distributed.
2: G = 0
3: while stopping criterion is not satisfied do
4:    for i = 1 to NP do
5:       Mutation: Generate the donor vector (V_{i,G}) according Eq. (??).
6:       Crossover: Recombine the mutate vector (U_{i,G}) according Eq. (??).
7:       Selection: Update the elite vector (E_{i,G} instead of X_{i,G}) according Eq. (??).
8:       Replacement: Select the target vectors (X_{i,G+1}) according to Algorithm 2 .
9:    G = G + 1
```

developing a novel replacement strategy called *The 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) 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 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

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

Algorithm 2 Replacement Phase

1: Input: $Population$ ($target\ vectors$), $Offspring$ ($trial\ vectors$), and $Elite$
2: Update $D_t = D_I - D_I * (nfes/(0.95 * max\_nfes))$
3: $Current = Population \cup Offspring \cup Elite$.
4: $Survivors = Penalized = \emptyset$.
5: **while** $|Survivors| < NP$ And $|Current| > 0$ **do**
6: $Selected$ = Select the best individual of $Current$.
7: Remove $Selected$ from $Current$.
8: Copy $Selected$ to $Survivors$.
9: Find the individuals from $Current$ with a distance to $Selected$ lower than D_t and move them to $Penalized$. Normalized distance is considered (Eq. 1).
10: **while** $|Survivors| < NP$ **do**
11: $Selected$ = Select the individual from $Penalized$ with the largest distance to the closest individual in $Survivors$.
12: Remove $Selected$ from $Penalized$.
13: Copy $Selected$ to $Survivors$.
14: **return** $Survivors$

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 decribed in the equation 1, 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*.

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

In order to complete the description 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 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 [11], a linear reduction of D_t is performed by taking into account the elepased function evaluations and stopping criterion. 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 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. [9]. 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 quality solutions with a lower probability. However, these transformations provoke large displacements that could improve significantly the solutions. According to this, we employ both high and low CR values as it is showed in Eq. 2.

$$CR = \begin{cases} Normal(0.2, 0.1), & \text{if} \quad rand[0,1] \leq 0.5 \\ Normal(0.9, 0.1), & \text{otherwise} \end{cases} \tag{2}$$

Following the principles of several SHADE variants [12, 13], the function evaluations are considered in the random generation of the mutation factor F. Particularly, each F is sampled through a Cauchy distribution (Eq. 3).

$$Cauchy(0.5, 0.5 * nfes/max\_nfes) \tag{3}$$

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. Thus, the algorithms considered from the CEC 2016 are UMOEAs-II [14] and L-SHADE-EpSin [12] that achieved the first and second place respectively. Similarly, the top algorithms from CEC 2017 are EBOwithCMAR [15] and jSO [16]. It is interesting to remark that EBOwith-CMAR 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 [17].

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.

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 [18] 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 $\uparrow$ shows the number of times that the superiority of each method could be confirmed, whereas the column tagged with the symbol $\downarrow$ count the number of cases where the method was inferior. Finally, the number comparisons where differences were not significant are shown in the column tagged with the symbol $\longleftrightarrow$.

Based in the guideline of the CEC, the "Score" is computed as follows. The evaluation method combines two scores defined in the equation (4). Thus the final score is composed by the sum $Score = Score_1 + Score_2$.

$$\begin{aligned} Score_1 &= \left(1 - \frac{SE - SE_{min}}{SE}\right) \times 50, \\ Score_2 &= \left(1 - \frac{SR - SR_{min}}{SR}\right) \times 50, \end{aligned} \tag{4}$$

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. On the other hand, the UMOEAs-II has the second best score in CEC 2017, and the EBOwithCMAR has the second best score in the CEC 2016. The same effect occurs in the third place with the SHADE's algorithms jSO and L-SHADE-EpSin. Thus, its evident that the performance of the algorithms is different with long-term executions. Also, this might be a highlight that the SHADE's algorithms are improved by the multi-operator algorithms. However, it seems that the multi-operator could be sensible with the parameterization. So, the winners of the CEC 2016 outperform to the winners of the CEC 2017 in long-term, and the winners of the CEC 2017 outperform to the winners of the CEC 2016.

Based in several experiments one drawback of the L-SHADE-EpSin and EBOwithCMAR resides in the strategy to adapt the mutation factor F. These algorithms take into account the elapsed evaluations to adapt the mutation factor, however this strategy does not considers long-term executions. Thus,

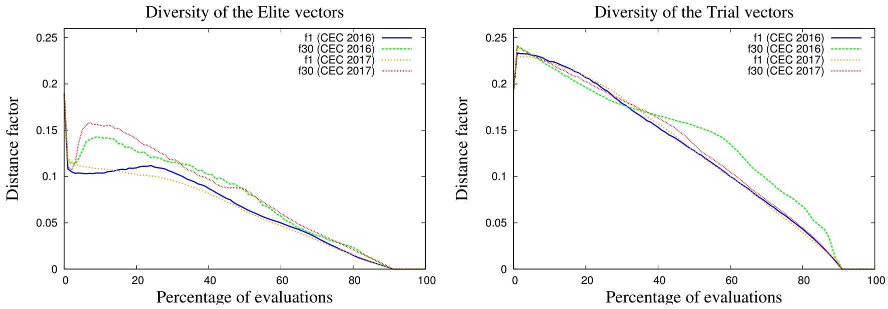

Fig. 1 Average DCN of the 51 executions with the problems f_1 and f_{30} (CEC 2016 and CEC 2017). The initial distance factor considered corresponds to $D_I = 0.3$.

Table 1 Summary results - CEC 2016

Algorithm	Always solved	At least one time solved	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
DE-EDM	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
DE-EDM	21	28	100.00	73	5	42

the adaptation of this parameter is unstable. In several experiments we try tuning the mutation factor, however the results were not significantly different. Broadly speaking, the results of the state-of-the-art shows that each algorithm was specifically tuned to short-term executions and for their respectively benchmark.

The mechanism of the DE-EDM should maintain the diversity properly, with the aim of have a better understanding of the methodology proposed, in the figure 1 is showed the diversity through the time elapsed of DE-EDM with two functions: f_1 and f_{30} respectively. While the first function is easily solved, the second is one of the most difficult. In the left side are showed the Elite vectors, although there are not constraints in the Elite vector to lost the diversity, it seems that in both functions f_1 and f_{30} the diversity is implicitly maintained. On the other hand the trial vectors maintain the diversity a is desired (i.e. until the 95% of the total function evaluations).

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

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 seems problematic solving the multi-modal functions, this can be provoked since that it does not applies an advanced strategy to deal with the incremented distribution of difference vectors. Due that the DE-EDM some several attraction basis 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 Empirical analyzes of the initial distance factor

In our proposal the diversity is explicitly promoted through several stages, which are controlled with the initial distance factorD_I. Therefore, the robustness of this parameter is analyzed as follows. In general, the configuration of the experimental validation is taken into account. Particularly, several initial dis-

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

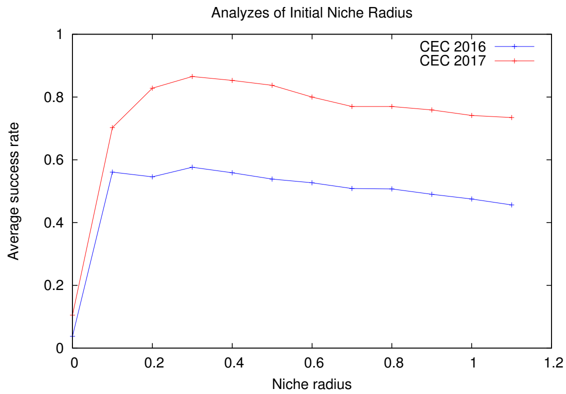

Fig. 2 Average success rate with different initial distance factors in the benchmark of CEC 2016 and CEC 2017, is considered a population size of 250 and 25,000,000 function evaluations.

tance factor where 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.
- 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.

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.

First, from experimental research on the working mechanism, it can be seen that our proposal is able to relieve the premature convergence to several optimization levels. Second, 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 be competitive even if the population size is small. Fourth, it seems that our proposal has some drawbacks in relation with the proportion of difference vectors, due that the displacement of the vectors is directly related with the diversity promoted in the population. Thus, in some cases the minimum displacement bounded by the diversity that is explicitly promoted in the population.

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 to involve a local search scheme with two goals, save the function evaluations and consider irregular fitness landscapes (e.g. multi-modal problems). Applying our proposal to real-world problems should be an interesting topic. 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. 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)
2. 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)
3. 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)

4. 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)
5. 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)
6. 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)
7. B. Liu, Q. Chen, Q. Zhang, J. Liang, P. Suganthan, B. Qu, in *Technical Report* (2013)
8. 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)
9. J. Montgomery, S. Chen, in *Evolutionary Computation (CEC), 2010 IEEE Congress on* (IEEE, 2010), pp. 1–8
10. J. Montgomery, S. Chen, in *Evolutionary Computation (CEC), 2012 IEEE Congress on* (IEEE, 2012), pp. 1–8
11. 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)
12. 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
13. J. Brest, M.S. Maučec, B. Bošković, in *Evolutionary Computation (CEC), 2016 IEEE Congress on* (IEEE, 2016), pp. 1188–1195
14. S. Elsayed, N. Hamza, R. Sarker, in *Evolutionary Computation (CEC), 2016 IEEE Congress on* (IEEE, 2016), pp. 2966–2973
15. A. Kumar, R.K. Misra, D. Singh, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 1835–1842
16. J. Brest, M.S. Maučec, B. Bošković, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 1311–1318
17. D. Molina, F. Moreno-García, F. Herrera, in *Evolutionary Computation (CEC), 2017 IEEE Congress on* (IEEE, 2017), pp. 805–812
18. 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)