A Comparative Study of Differential Evolution Variants for Global Optimization

Efrén Mezura-Montes
Lab. Nacional de Informática Avanzada (LANIA)
Rébsamen 80 Xalapa, Veracruz, 91000 MEXICO
emezura@lania.mx

Jesús Velázquez-Reyes and
Carlos A. Coello Coello
CINVESTAV-IPN, Computer Science Section
Av. IPN No. 2508, México D.F. 07300, MEXICO
jesus.velazquez@gmail.com
ccoello@cs.cinvestav.mx

ABSTRACT

In this paper, we present an empirical comparison of some Differential Evolution variants to solve global optimization problems. The aim is to identify which one of them is more suitable to solve an optimization problem, depending on the problem's features and also to identify the variant with the best performance, regardless of the features of the problem to be solved. Eight variants were implemented and tested on 13 benchmark problems taken from the specialized literature. These variants vary in the type of recombination operator used and also in the way in which the mutation is computed. A set of statistical tests were performed in order to obtain more confidence on the validity of the results and to reinforce our discussion. The main aim is that this study can help both researchers and practitioners interested in using differential evolution as a global optimizer, since we expect that our conclusions can provide some insights regarding the advantages or limitations of each of the variants studied.

Categories and Subject Descriptors

G.1.6 [**Mathematics of Computing**]: Numerical Analysis—*global optimization, unconstrained optimization*; J.2 [**Computer Applications**]: Physical Sciences and Engineering—*Engineering*

General Terms

Algorithms

Keywords

Differential Evolution, Global Optimization

1. INTRODUCTION

Evolutionary Algorithms (EAs) have been widely used to solve optimization problems [11, 6, 4]. Differential Evolution (DE) [14] is an EA proposed to solve optimization problems, mainly in continuous search spaces. As traditional EAs, several optimization problems have been successfully solved by using DE [13].

DE is an evolutionary (direct-search) algorithm which has been mainly used to solve optimization problems. DE shares similarities with traditional EAs. However it does not use binary encoding as a simple genetic algorithm [7] and it does not use a probability density function to self-adapt its parameters as an Evolution Strategy [18]. Instead, DE performs mutation based on the distribution of the solutions in the current population. In this way, search directions and possible stepsizes depend on the location of the individuals selected to calculate the mutation values.

There is a nomenclature scheme developed to reference the different DE variants. The most popular is called *"DE/rand/1/bin"*, where "DE" means Differential Evolution, the word "rand" indicates that individuals selected to compute the mutation values are chosen at random, "1" is the number of pairs of solutions chosen and finally "bin" means that a binomial recombination is used (in Section 5 we will omit the word "DE" at the beginning, focusing only in the differences in type of selection and recombination operator used). The corresponding algorithm of this variant is presented in Figure 1 and applications of it can be found in [1, 10, 8].

The "CR" parameter controls the influence of the parent in the generation of the offspring. Higher values mean less influence of the parent. The "F" parameter scales the influence of the set of pairs of solutions selected to calculate the mutation value (one pair in the case of the algorithm in Figure 1).

It is important to note that, increasing either the population size or the number of pairs of solutions to compute the mutation values will also increase the diversity of possible movements, promoting the exploration of the search space. However, the probability to find the correct search direction decreases considerably. Then, the balance between the population size and the number of differences determines the efficiency of the algorithm [5]. Besides this balance, another important factor when using DE is the selection of the variant. Each one varies the way mutation is computed and also the type of recombination operator. In fact, other authors have shown that other version, like *"DE/best/2/exp"* is very suitable to solve digital filter design problems [19, 14]. This

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

paper concentrates in the selection of the variant. The main motivation of this work is twofold: (1) to find the most competitive approach when solving a set of test problems with different features and (2) to identify which variant is more suitable to solve a problem, depending of its features. We will use a set of well-known test problems in our experiments. In fact, we know from the No Free Lunch Theorems for search [21] that using such a limited set of functions does not guarantee, in any way, that an algorithm (or a variant in our case) that performs well on them, will necessarily be competitive in a different set of problems. However, our goal is to give interested users some insights about which variant could be more convenient, based on the type of problem to be solved, and also, to analyze which of the different DE variants performs better in a common benchmark.

The problem of our interest is the global optimization problem, and it can be stated as to:

$$\text{Find } \vec{x} \text{ which optimizes } f(\vec{x}) \tag{1}$$

where $\vec{x} \in \mathbb{R}^n$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_n]^T$, and each x_i, $i = 1, \ldots, n$ is bounded by lower and upper limits $L_i \leq x_i \leq U_i$. The aim of this work is to use test problems with a high dimensionality ($n = 30$).

```
1     Begin
2        G=0
3        Create a random initial population $\vec{x}_{i,G}$ $\forall i$, $i = 1, \ldots, NP$
4        Evaluate $f(\vec{x}_{i,G})$ $\forall i$, $i = 1, \ldots, NP$
5        For G=1 to MAX_GEN Do
6           For i=1 to NP Do
7 ⇒            Select randomly $r_1 \neq r_2 \neq r_3$ :
8 ⇒            $j_{rand} = \text{randint}(1, D)$
9 ⇒            For j=1 to D Do
10 ⇒              If ($rand_j[0,1) < CR$ or $j = j_{rand}$) Then
11 ⇒                 $u_{i,j,G+1} = x_{r_3,j,G} + F(x_{r_1,j,G} - x_{r_2,j,G})$
12 ⇒              Else
13 ⇒                 $u_{i,j,G+1} = x_{i,j,G}$
14 ⇒              End If
15 ⇒           End For
16             If ($f(\vec{u}_{i,G+1}) \leq f(\vec{x}_{i,G})$) Then
17                $\vec{x}_{i,G+1} = \vec{u}_{i,G+1}$
18             Else
19                $\vec{x}_{i,G+1} = \vec{x}_{i,G}$
20             End If
21          End For
22          $G = G + 1$
23       End For
24    End
```

Figure 1: "DE/rand/1/bin" algorithm. randint(min,max) is a function that returns an integer number between min and max. rand$[0, 1)$ is a function that returns a real number between 0 and 1. Both are based on a uniform probability distribution. "NP", "MAX_GEN", "CR" and "F" are user-defined parameters. "D" is the dimensionality of the problem. Steps pointed with arrows change from variant to variant.

The paper is organized as follows: In Section 2, we provide a review of the related work. Afterwards, in Section 3 we describe the eight variants used in our experiments. Section 4 includes the details of our experimental design and also the results obtained. Later, in Section 5 we discuss the obtained results and finally in Section 6 we draw some conclusions and we define the future work.

2. RELATED WORK

Babu and Munawar [2] compared the performance of several DE variants to solve the optimal design of shell-and-tube heat exchangers. Their conclusions showed that *"DE/best/*/*"* variants provided better results than *"DE/rand/*/*"*. They also found that *"DE/best/1/*"* was the best when solving the heat exchangers problem. Qin and Suganthan [15] proposed a Self-adaptive DE. The aim of their work was to allow DE to switch between two variants: *"DE/rand/1/bin"* and *"DE/best/2/bin"* and also to adapt the "F" and "CR" values. The approach performed well on several benchmark problems.

Vesterstrom and Thomsen [20] compared *"DE/rand/1/bin"* with two Particle Swarm Optimization (PSO) versions and a simple EA on 23 benchmark problems for global unconstrained optimization. In their study, they observed a clear superior performance provided by *"DE/rand/1/bin"* compared with those provided by PSO and the simple EA. Rönkkönen et al. [16] also extensively tested *"DE/rand/1/bin"* on 25 benchmark problems. The results were competitive except on hybrid composition functions. Kukkonen and Lampinen [9] extended *"DE/rand/1/bin"* to solve, besides global optimization problems, constrained optimization and multiobjective optimization problems. The results obtained in two benchmark functions were better than those obtained by a similar algorithm used to solve several types of optimization problems.

Zhang and Xie [23] proposed a DE-PSO hybrid to solve optimization problems by incorporating the *"DE/rand/1/bin"* mutation operator inside the PSO algorithm. The results showed a better performance of the Hybrid approach with respect to those provided by two PSO-based approaches.

Noman and Iba [12] used two DE variants: *"DE/rand/1/exp"* and *"DE/best/1/exp"* coupled with a crossover-based local search. They tested their memetic variants on five benchmark problems with different dimensionalities and concluded that the use of local search accelerates convergence on DE, especially using a Simplex Crossover [12].

Bui et al. [3] proposed two variations of the *"DE/rand/1/bin"*, The first one consisted on generating the "F" parameter at random instead of being fixed and storing search directions in another population in order to evolve better directions in parallel. The second variation also maintains a population of directions but in this case, one individual of the pair of solutions chosen to compute mutation was taken from the best individuals in the population an the other solution is taken from the rest of solutions. The results obtained did not provide a clear winner between both approaches.

3. DIFFERENT VERSIONS TESTED

Unlike the previous approaches presented in Section 2, our study focuses on empirically comparing standard DE variants in order to detect which one of them is more suitable to solve an unconstrained optimization problem depending on the problem's features. Furthermore, we want to find which variant provides the most competitive performance in all problems, regardless of the variety in features in the set of problems solved.

In our study we selected eight different DE variants. The modifications from variant to variant are in the recombination operator used (steps 9 to 15 in Figure 1) and also in the way individuals are selected to calculate the mutation vector (step 7 in Figure 1).

We selected four variants whose recombination operator is discrete, always using two individuals: the original parent and the DE mutation vector (step 11 in Figure 1). Two discrete recombination operators are used: binomial and exponential. The main difference between them is that for binomial recombination, each variable value of the offspring is taken at every time from one of the two parents, based on the "CR" parameter value. On the other hand, in the exponential recombination, each variable value of the offspring is taken from the first parent until a random number surpasses the "CR" value, from this point, all the offspring variable values will be taken from the second parent. These variants are: *"DE/rand/1/bin"*, *"DE/rand/1/exp"*, *"DE/best/-1/bin"* and *"DE/best/1/exp"* [14]. The "rand" variants select all the individuals to compute mutation at random and the "best" variants use the best solution in the population besides the random ones.

We selected two variants with arithmetic recombination, which, unlike discrete recombination, is rotation invariant. These are *"DE/current-to-rand/1"* and variant *"DE/current-to-best/1"* [14]. The only difference between them is that the first selects the individuals for mutation at random and the second one uses the best solution in the population besides random solutions.

We also included in this comparison the variant *"DE/-rand/2/dir"* [5], which incorporates objective function information to the mutation and recombination operators. The aim of this approach is to guide the search to promising areas faster than traditional DE. Their authors argue that the best results are obtained when the number of pairs of solutions is two [5]; therefore, we adopted this value only for this variant.

Finally, we included a variant with a combined discrete-arithmetic recombination, the *"DE/current-to-rand/1/bin"* [14].

Each variant's implementation details are summarized in Table 1.

4. EXPERIMENTS AND RESULTS

The experimental design consists on testing the eight variants selected under similar conditions on a set of problems with a high dimensionality and different features. From [22], we selected 13 of them whose dimensionality can be scaled and it was fixed to 30 in all cases. The details of the test problems are presented in the Appendix at the end of the paper and a summary of their main features (modality, separability) is presented in Table 3. Please note that we consider the Rosenbrock function as a multimodal function based on the results summarized in [17]. We selected this set of problems because it is a well-known benchmark in the literature to test EAs for global optimization. All test functions have an optimum value at zero except for $f08$. In order to show similar results, the description of $f08$ was adjusted to have its optimum value at zero by just adding the optimal value for the function with 30 decision variables (12569.486618164879).

We fixed the stopping criterion as an error value (with respect to the global optimum value) of 1×10^{-12}. We also fixed the maximum number of evaluations to $120,000$. We decided to use this number of evaluations because we wanted to allow DE variants to have time to explore the search space, but always considering a lower number of evaluations than such required by approaches based on other EAs (which perform about $500,000$ evaluations) [22]. In this way, we can show that DE requires less evaluations to solve this set of well-known problems. The algorithm will stop before the number of evaluations is reached only if the tolerance error with respect to the global optimum is obtained. The parameters for all DE variants were: population size NP $= 60$, maximum number of generations $= 2000$. As discussed in Section 1, a large population affects the ability of the approach to find the correct search direction. Then we decided to work with a moderated and fixed population size in all experiments.

Based on previous experiments we empirically defined a range for the parameter $F \in [0.3, 0.9]$; this value is generated anew at each generation for all eight approaches. The same random values were used for the K value adopted in the arithmetic recombination from variants *"current-to-rand/1"*, *"current-to-best/1"* and *"current-to-rand/1/bin"*. The CR parameter was tuned for each pair variant-problem (only in variants where "CR" was needed, see Table 1) due to the high sensibility of DE to it. 11 different values for the "CR" parameter were tested $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ for each pair model-problem. 50 independent runs were performed for each combination model-problem-"CR" value. Based on the obtained results, a bootstrap test was computed for each combination variant-problem-"CR" value in order to determine the confidence interval for the mean statistic. The "CR" value corresponding to the best confidence interval (out of the 10 different "CR" values) was chosen to be used in our experiments and they are presented in Table 2. The study of DE's sensitivity to this parameter is beyond the scope of this work. However, we noted that, in most cases, values close to the extreme values $0.8 - 0.9$ or $0.0 - 0.3$ were the most common. This may suggest that the influence of the parent must be either significant or very small.

For all variants (except *"DE/rand/2/dir"*), we fixed the number of pairs for the mutation vector to 1. This was because we observed that in most cases, if more pairs are used, the convergence speed increases but it also increases the probability to get trapped in local optima. We also noted that similar results can be obtained with the same number of evaluations by just using one pair of solutions instead of using more than one pair with a larger population size. 100 independent runs were performed per variant per test function.

For the sake of an easier analysis, the results were grouped by features of the test problems. In Table 4 we present the mean objective function value obtained for each variant in each unimodal and separable test function. In Table 6, we show the mean objective function results for function $f03$ (unimodal and nonseparable) and also for $f08$ and $f09$ (multimodal and separable). Finally, in Table 5 the mean objective function values for the multimodal and nonseparable functions are presented.

5. DISCUSSION

5.1 Unimodal and Separable Problems

The results for the unimodal and separable functions (Table 4) show that the best results were provided by *"rand/-1/bin"*, *"best/1/bin"*, *"current-to-rand/1/bin"* and *"rand/-2/dir"*. In all cases, these variants reached the global opti-

Nomenclature	**Variant**
rand/p/bin	$u_{i,j} = \begin{cases} x_{r_3,j} + F \cdot \sum_{k=1}^{p}(x_{r_1^p,j} - x_{r_2^p,j}) & \text{if } U_j(0,1) < CR \text{ or } j = j_r \\ x_{i,j} & \text{otherwise} \end{cases}$
rand/p/exp	$u_{i,j} = \begin{cases} x_{r_3,j} + F \cdot \sum_{k=1}^{p}(x_{r_1^p,j} - x_{r_2^p,j}) & \text{from } U_j(0,1) < CR \text{ or } j = j_r \\ x_{i,j} & \text{otherwise} \end{cases}$
best/p/bin	$u_{i,j} = \begin{cases} x_{best,j} + F \cdot \sum_{k=1}^{p}(x_{r_1^p,j} - x_{r_2^p,j}) & \text{if } U_j(0,1) < CR \text{ or } j = j_r \\ x_{i,j} & \text{otherwise} \end{cases}$
best/p/exp	$u_{i,j} = \begin{cases} x_{best,j} + F \cdot \sum_{k=1}^{p}(x_{r_1^p,j} - x_{r_2^p,j}) & \text{from } U_j(0,1) < CR \text{ or } j = j_r \\ x_{i,j} & \text{otherwise} \end{cases}$
current-to-rand/p	$\vec{u}_i = \vec{x}_i + K \cdot (\vec{x}_{r_3} - \vec{x}_i) + F \cdot \sum_{k=1}^{p}(\vec{x}_{r_1^p} - \vec{x}_{r_2^p})$
current-to-best/p	$\vec{u}_i = \vec{x}_i + K \cdot (\vec{x}_{best} - \vec{x}_i) + F \cdot \sum_{k=1}^{p}(\vec{x}_{r_1^p} - \vec{x}_{r_2^p})$
current-to-rand/p/bin	$u_{i,j} = \begin{cases} \vec{x}_{i,j} + K \cdot (\vec{x}_{r_3,j} - \vec{x}_{i,j}) + F \cdot \sum_{k=1}^{p}(\vec{x}_{r_1^p,j} - \vec{x}_{r_2^p,j}) & \text{if } U_j(0,1) < CR \text{ or } j = j_r \\ x_{i,j} & \text{otherwise} \end{cases}$
rand/2/dir	$\vec{v}_i = \vec{v}_1 + \frac{F}{2}(\vec{v}_1 - \vec{v}_2 + \vec{v}_3 - \vec{v}_4)$ where $f(\vec{v}_1) < f(\vec{v}_2)$ and $f(\vec{v}_3) < f(\vec{v}_4)$

Table 1: DE variants used in our experiments j_r is a random integer number generated between $[0, n]$, where n is the number of variables of the problem. $U_j(0,1)$ is a real number generated at random between 0 an 1. Both numbers are generated using a uniform distribution. In our experiments we always use $p = 1$

	Problem												
Variant	$f01$	$f02$	$f03$	$f04$	$f05$	$f06$	$f07$	$f08$	$f09$	$f10$	$f11$	$f12$	$f13$
rand/1/bin	0.9	0.2	0.9	0.8	0.9	0.0	0.0	0.0	0.0	0.9	0.9	0.0	0.1
rand/1/exp	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.8	0.1	0.1	0.1	0.1
best/1/bin	0.2	0.3	0.8	0.3	0.3	0.0	0.0	0.0	0.0	0.3	0.1	0.1	0.2
best/1/exp	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
current-to-rand/1/bin	0.5	0.8	0.9	0.5	0.5	0.1	0.0	0.0	0.0	0.7	0.6	0.4	0.5

Table 2: Different "CR" values used in our experiments for each pair variant-function.

mum in problems $f01$, $f02$, $f06$ and $f07$. Test function $f04$ was not solved by any approach. The best result for this problem was provided by *"best/1/bin"*. The poorest performances in all these functions were provided by *"best/1/exp"*, *"current-to-best/1"* and *"current-to-rand/1"*. It is interesting that *"best/1/bin"* is the most competitive, whereas *"best/1/exp"* provided a consistent poor performance in all functions. This result suggests an important influence of the recombination type (binomial over exponential in this case) when solving unimodal and separable functions. Furthermore, the "rand" and "best" variants seem to be very suitable coupled with a binomial recombination when solving this type of problems; however, selecting the best solution to compute the mutation vector is slightly more convenient for this set of unimodal and separable problems because *"best/1/bin"* provided the best results in the hardest problem to solve ($f04$).

5.2 Unimodal/Nonseparable and Multimodal/Separable Problems

Only the *"best/1/bin"* (Table 6) was able to solve problem $f03$ (unimodal and nonseparable) and also problems $f08$ and $f09$ (multimodal and separable). The second best performance was provided by *"rand/1/bin"* and *"rand/2/dir"* because they solved consistently problems $f08$ and $f09$ (however, *"rand/1/bin"* provided a clearly better result than the provided by *"rand/2/dir"* in problem $f03$). The poorest performances were provided again by *"best/1/exp"*, *"current-to-best/1"* and *"current-to-rand/1"*. It is worth noting that, as in the previous set of problems, binomial recombination showed a better performance over the exponential recombination, even in combination with other mutation types, like in *"current-to-rand/1/bin"*.

5.3 Multimodal and Nonseparable Problems

For the hardest problems to solve (Table 5), the obtained results showed that *"rand/2/dir"* was the most competitive, reaching the global optimum in four of five problems ($f10$, $f11$, $f12$ and $f13$). The second best performance was provided by *"rand/1/bin"* which got the global optimum consistently in three problems ($f10$, $f12$ and $f13$). The third best performance was provided by *"best/1/bin"* and *"current-to-rand/1/bin"*. The worst results were provided again by *"best/1/exp"*, *"current-to-best/1"* and *"current-to-rand/1"*. It was also very clear that none of the variants was able to solve problem $f05$, where the best mean result was provided by *"rand/1/exp"*. For this set of problems, the binomial recombination provided better results than the exponential recombination, but the positive effect is not as marked as it was in the previous sets of problems. On the other hand, the "rand" selection seems to be more suitable for this set of multimodal and nonseparable problems. This behavior was expected, because, by selecting individuals at random and not necessarily picking the best one, search directions will be more diverse and the ability to avoid local optimum solutions will also increase.

5.4 Remarks

Based on the overall results in Tables 4, 6 and 5, the most competitive variants were *"best/1/bin"*, *"rand/1/bin"* *"rand/2/dir"* and *"current-to-rand/1/bin"*. On the other hand, the worst overall results were consistently provided by variants *"best/1/exp"*, *"current-to-best/1"* and *"current-to-rand/1"*. Variant *"rand/1/exp"* was also very irregular at providing good overall results. From all the above mentioned we can highlight the following:

- Variants *"current-to-best/1"* and *"current-to-rand/1"*,

	Modality	
Separable	*Unimodal*	*Multimodal*
YES	• **f01 (sphere)** Simple quadratic function (n-dimensional sphere) • **f02 (Sch. 2.22)** Discontinuous function. • **f04 (Sch. 2.21)** Discontinuous function. • **f06** Quadratic discontinuous function. • **f07** Polynomial function with initial noise.	• **f08 (Sch. 2.26)** Nonpolynomial function with several local optima. • **f09 (Rastrigin)** Nonpolynomial function.
NO	• **f03 (Sch. Double Sum)** Quadratic function.	• **f05 (Rosenbrock)** Polynomial function. • **f10 (Ackley)** Continuous exponential function. • **f11 (Griewangk)** Continuous polynomial function. • **f12 (GPF-1)** Irregular and discontinuous function. • **f13 (GPF-2)** Irregular and discontinuous function.

Table 3: Main features of the 13 test functions used in the experiments.

	Function				
Variant	*f01*	*f02*	*f04*	*f06*	*f07*
rand/1/bin	**0.0**	**0.0**	1.9521	**0.0**	**0.0**
rand/1/exp	**0.0**	**0.0**	3.7584	0.843360	**0.0**
best/1/bin	**0.0**	**0.0**	0.0017	**0.0**	**0.0**
best/1/exp	407.972	3.291	1.701872	2737.8458	0.070545
current-to-best/1	0.54148	4.842	4.233736	1.394	**0.0**
current-to-rand/1	0.69966	3.503	3.298563	1.767	**0.0**
current-to-rand/1/bin	**0.0**	**0.0**	0.149514	**0.0**	**0.0**
rand/2/dir	**0.0**	**0.0**	0.044199	**0.0**	**0.0**

Table 4: Mean objective function value of 100 independent runs for each DE variant when solving the unimodal and separable test functions $f01$, $f02$, $f04$, $f06$, $f07$. A result in boldface means that the global optimum value has been reached with the tolerance established.

which use arithmetic recombination, have problems to explore the search space (clearly noted when solving multimodal functions). This suggests that this combination of arithmetic recombination with the DE mutation based on differences is not very suitable to solve problems with high dimensionality.

- *"best/1/bin"* and *"rand/1/bin"* are clearly much better than *"best/1/exp"*, *"rand/1/exp"*. This behavior seems to be due to the fact that in the exponential recombination, not all of the corners of the hypercube formed by the mutation vector and the current parent can be sampled, regardless of the "CR" value used. In fact, from the 2^n possible solutions (where n is the number of decision variables of the problem) only $n+1$ solutions can be generated. Binomial recombination can generate all these 2^n solutions, but not with the same probability. This difference is evident in our experiments were $n = 30$.

- *"best/1/bin"* is the most competitive for unimodal & separable and unimodal & nonseparable test problems. However, for multimodal functions this variant provides competitive results only when the function is separable.

- For multimodal and nonseparable functions, *"rand/2/dir"* is more suitable, due to its ability to incorporate information of the fitness of the individuals in the mutation process. In general, the variant *"rand/*/*"* provides a better performance, because of the random way of finding new search directions.

5.5 Confidence Intervals

As in our study we based the comparison of results in the mean statistic, we considered very advisable to calculate the 95%-confidence mean intervals for the four most competitive variants: *"rand/1/bin"*, *"best/1/bin"*, *"current-to-rand/1/bin"* and *"rand/2/dir"*. In this way, the statistical results will get more statistical significance.

We considered confidence intervals useful because they provide a good approximation of the mean performance of the different variants in this set of problems but with statistical support and not based on a single sample of results.

We performed a bootstrap process using 1000 re-samples on our 100-runs original sample. The results are summarized in Table 7.

As can be noted, the four variants provide similar results.

	Function				
Variant	*f05*	*f10*	*f11*	*f12*	*f13*
rand/1/bin	19.577895	**0.0**	0.001117	**0.0**	**0.0**
rand/1/exp	6.696064	0.080037	0.000075	**0.0**	**0.0**
best/1/bin	30.390870	**0.0**	0.000722	**0.0**	0.000226
best/1/exp	132621.5	9.3961	5.9278	1293.0262	2584.85
current-to-best/1	30.984666	0.270788	0.219391	0.891301	0.038622
current-to-rand/1	31.702063	0.164786	0.184920	0.464829	5.169196
current-to-rand/1/bin	24.260535	**0.0**	**0.0**	0.001007	0.000114
rand/2/dir	30.654916	**0.0**	**0.0**	**0.0**	**0.0**

Table 5: Mean objective function value of 100 **independent runs for each DE variant when solving the multimodal and nonseparable test functions** $f05$, $f10$, $f11$, $f12$, $f13$. **A result in boldface means that the global optimum value has been reached with the tolerance established.**

	Variant			
Function	*rand/1/bin*	*best/1/bin*	*current-to-rand/1/bin*	*rand/2/dir*
f01	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)
f02	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)
f04	(1.487,2.453)	(0.0011,0.0024)	(0.1037,0.2053)	(0.0116,0.0874)
f06	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)
f07	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)
f08	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)
f09	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)
f03	(0.0206,0.0282)	(**0.0,0.0**)	(0.0001,0.0003)	(24.394,37.523)
f05	(16.594,22.773)	(26.399,34.643)	(24.115,24.398)	(28.278,33.651)
f10	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0,0.0**)
f11	(0.0005,0.0017)	(0.0002,0.0013)	(**0.0,0.0**)	(**0.0,0.0**)
f12	(**0.0,0.0**)	(**0.0,0.0**)	(**0.0**,0.0031)	(**0.0,0.0**)
f13	(**0.0,0.0**)	(**0.0**,0.0005)	(**0.0**,0.0003)	(**0.0,0.0**)

Table 7: 95% **confidence intervals for the variants:** ***rand/1/bin*****,** ***best/1/bin*****,** ***current-to-rand/1/bin*****,** ***rand/2/dir*** **in the** 13 **test functions. A result in boldface indicates that the global optimum was reached.**

	Function		
Variant	*f03*	*f08*	*f09*
rand/1/bin	0.024305	**0.0**	**0.0**
rand/1/exp	0.000004	3.124056	97.753938
best/1/bin	**0.0**	**0.0**	**0.0**
best/1/exp	10.607806	2114.961277	40.003971
current-to-best/1	0.471730	6240.387323	98.205432
current-to-rand/1	0.903563	6484.055540	92.263070
current-to-rand/1/bin	0.000232	0.000001	**0.0**
rand/2/dir	30.112881	**0.0**	**0.0**

Table 6: Mean objective function value of 100 **independent runs for each DE variant when solving the functions:** $f03$, **unimodal and nonseparable and** $f08, f09$ **, multimodal and separable functions. A result in boldface means that the global optimum value has been reached with the tolerance established.**

None of them is able to solve problem $f05$ (multimodal and nonseparable), the best approximation was obtained by *"rand/1/bin"*. On the other hand, the only variant which could solve problem $f03$ (unimodal and nonseparable) was *"best/1/bin"*. In the remaining functions, no significant differences that help us to determine the best one could be found.

5.6 Convergence Analysis

As a final experiment, in order to detect which variant is the most competitive, we calculated the mean percentage out of the total 120,000 evaluations required by each of the four most competitive variants to reach the global optimum with a smaller tolerance value fixed to 1×10^{-16}. The mean value is calculated based on 100 independent runs. The results are shown in Table 8.

For the unimodal and separable functions, the fastest convergence was provided by *"best/1/bin"* (only in function $f04$ was surpassed by *"rand/2/dir"* but only by 2%). For the multimodal and separable problems, the four variants provided a similar behavior in function $f08$, but, again, *"best/1/bin"* was faster in the other function ($f09$). The same performance was obtained in problem $f03$ (unimodal and nonseparable). Finally, for the multimodal and nonseparable problems, *"current-to-rand/1/bin"* provided a slightly faster convergence in three out of five problems ($f10$, $f11$ and $f12$) than the provided by the *"best/1/bin"*, which indeed showed the fastest convergence in problem $f13$. The overall results suggest that *"best/1/bin"* provides a faster convergence to the global optima regardless of the features of the problem to be solved. However, if the problem is multimodal and nonseparable, *"current-to-rand/1/bin"* is slightly faster.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an empirical comparison of some DE variants to solve global optimization problems. Eight different variants were implemented and extensively tested on 13 well-known benchmark problems. The results obtained showed that the most competitive approach, regardless the characteristics of the problem to be solved was the *"best/1/bin"* (using always the best solution to find search directions and also binomial recombination), based on quality and robustness of results. However, when solving unimodal and separable functions, the use of information of the objective function value to compute mutation provided by

	Variant			
Function	*rand/1/bin*	*best/1/bin*	*current-to-rand/1/bin*	*rand/2/dir*
f01	68%	**30%***	42%	48%
f02	100%	**35%***	64%	70%
f04	100%	100%	100%	**98%***
f06	23%	**12%***	**12%***	27%
f07	40%	**18%***	36%	39%
f08	100%	100%	100%	100%
f09	91%	**78%***	100%	90%
f03	100%	**99%***	100%	100%
f05	100%	100%	100%	**86%***
f10	100%	99%	**82%***	100%
f11	68%	51%	**43%***	50%
f12	76%	45%	**38%***	54%
f13	73%	**37%***	39%	56%

Table 8: Comparison of the percentage of the total evaluations required to reach the global optimum with a precision of 1×10^{-16} by variants: *rand/1/bin*, *best/1/bin*, *current-to-rand/1/bin*, *rand/2/dir* on each of the 13 test functions. The maximum number of evaluations was set to 120,000. The lowest percentage is marked with a "*" and boldface.

the *"rand/2/dir"* also provides competitive results. When the problem is unimodal and nonseparable *"best/1/bin"* was the only one to solve the problem. This variant also solved quite well multimodal and separable problems. However, *"rand/1/bin"* and *"rand/2/dir"* provided a similar good performance on this type of problem. Finally, for multimodal and nonseparable problems, the *"rand/2/dir"* was the most competitive and slightly faster to converge to the global optimum.

As future paths of research, we will analyze the effect of dimensionality in each variant using more than 30 variables per problem. After that, we will perform a more careful study about the strong influence of the "CR" parameter in the performance of the different DE variants. Also, we will try to combine the strengths of the competitive variants into a better and more complete approach and compare it against state-of-the-art techniques. Finally, we also plan to extend the capabilities of the algorithm to solve constrained optimization problems.

Acknowledgments

The third author acknowledges support from CONACyT through project number 42435-Y.

7. REFERENCES

[1] J. Alvarez-Gallegos, C. A. C. Villar, and E. A. P. Flores. Evolutionary Dynamic Optimization of a Continuously Variable Transmission for Mechanical Efficiency Maximization. In A. Gelbukh, Álvaro de Albornoz, and H. Terashima-Marín, editors, *MICAI 2005: Advances in Artificial Intelligence*, pages 1093–1102, Monterrey, México, November 2005. Springer. Lecture Notes in Artificial Intelligence Vol. 3789,.

[2] B. V. Babu and S. A. Munawar. Optimal Design of Shell-and-Tube Heat Exchangers by Different Strategies of Differential Evolution. Technical Report PILANI-333 031, Department of Chemical Engineering Birla Institute of Technology and Science, Rajasthan, India., 2001.

[3] L. T. Bui, Y. Shan, F. Qi, and H. A. Abbass. Comparing Two Versions of Differential Evolution in Real Parameter Optimization. Technical Report TR-ALAR-200504009, School of Information Technology and Electrical Engineering, University of New South Wales, Northcott Drive, Campbell, Canberra, ACT 2600, Australia, 2005.

[4] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, June 2002. ISBN 0-3064-6762-3.

[5] V. Feoktistov and S. Janaqi. Generalization of the Strategies in Differential Evolution. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS 2004), 2004, Santa Fe, New Mexico, USA*, page 165a, New Mexico, USA, April 2004. IEEE Computer Society.

[6] D. Goldberg. *The Design of Innovation.* Kluwer Academic Publishers, New York, June 2002. ISBN 1-4020-7098-5.

[7] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.

[8] T. Krink, B. Filipiĕ, G. B. Fogel, and R. Thomsen. Noisy Optimization Problems - A Particular Challenge for Differential Evolution? In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2004)*, volume 1, pages 332–339, Piscataway, New Jersey, June 2004. IEEE Service Center.

[9] S. Kukkonen and J. Lampinen. GDE3: The third Evolution Step of Generalized Differential Evolution. In *Proceedings of the Congress on Evolutionary Computation 2005 (CEC'2005)*, volume 1, pages 239–246, Piscataway, New Jersey, September 2005. Edinburgh, UK,, IEEE Service Center.

[10] J. Lampinen. A Constraint Handling Approach for the Diifferential Evolution Algorithm. In *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, volume 2, pages 1468–1473, Piscataway, New Jersey, May 2002. IEEE Service Center.

[11] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, third edition, 1996.

[12] N. Noman and H. Iba. Enhancing Differential Evolution Performance with Local Search for High Dimensional Function Optimization. In H.-G. e. Beyer, editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2005)*, pages 967–974, New York, June 2005. Washington DC, USA, ACM Press.

[13] K. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution : A Practical Approach to Global Optimization.* Springer-Verlag, 2005. ISBN: 3-540-20950-6.

[14] K. V. Price. An introduction to differential evolution. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 79–108. Mc Graw-Hill, UK, 1999.

[15] A. Qin and P. Suganthan. Self-adaptive Differential Evolution Algorithm for Numerical Optimization. In *Proceedings of the Congress on Evolutionary Computation 2005 (CEC'2005)*, volume 1, pages 630–636, Piscataway, New Jersey, September 2005. Edinburgh, UK,, IEEE Service Center.

[16] J. Rönkkönen, S. Kukkonen, and K. V. Price. Real-Parameter Optimization with Differential Evolution. In *Proceedings of the Congress on Evolutionary Computation 2005 (CEC'2005)*, volume 1, pages 567–574, Piscataway, New Jersey, September 2005. Edinburgh, UK,, IEEE Service Center.

[17] S. Yun-Wei and Q. Yu-Huan. A Note on the Extended Rosenbrock Function *Evolutionary Computation* 14(1): 119-126, 2006.

[18] H.-P. Schwefel. *Evolution and Optimization Seeking.* John Wiley & Sons, New York, 1995.

[19] R. Storn. Differential evolution design of an IIR-filter with requirements for magnitude and group delay. In *Proceedings of IEEE International Conference on Evolutionary Computation*

(ICEC'96), pages 268–273, New York, NY, USA, May 1996. IEEE.

[20] J. Vesterstrom and R. Thomsen. A Comparative Study of Differential Evolution Particle Swarm Optimization and Evolutionary Algorithms on Numerical Benchmark Problems. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2004)*, volume 3, pages 1980–1987, Piscataway, New Jersey, June 2004. Portland Oregon, USA, IEEE Service Center.

[21] David H. Wolpert and William G. Macready. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation* 1(1): 67-82, 1997.

[22] X. Yao, Y. Liu, K.-H. Liang, and G. Lin. Fast Evolutionary Algorithms. In G. Rozenberg, T. Bäck, and A. Eiben, editors, *Advances in Evolutionary Computing: Theory and Applications*, pages 45–94. Springer-Verlag, New York, NY, USA, 2003.

[23] W. J. Zhang and X.-F. Xie. DEPSO: Hybrid Particle Swarm with Differential Evolution Operator. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC'2003)*, volume 4, pages 3816–3821. Washington DC, USA, IEEE, October 2003.

APPENDIX

A. TEST FUNCTIONS

The details of the 13 test functions used in this paper are the following [22]:

$f01$ - Sphere Model

$$f_1(x) = \sum_{i=1}^{30} x_i^2$$

$$-100 \leq x_i \leq 100$$
$$min(f_1) = f_1(0, ..., 0) = 0$$

$f02$ - Schwefel's Problem 2.22

$$f_2(x) = \sum_{i=1}^{30} |x_i| + \prod_{i=1}^{30} |x_i|$$

$$-10 \leq x_i \leq 10$$
$$min(f_2) = f_2(0, ..., 0) = 0$$

$f03$ - Schwefel's Problem 1.2

$$f_3(x) = \sum_{i=1}^{30} \left(\sum_{j=1}^{i} x_j \right)^2$$

$$-100 \leq x_i \leq 100$$
$$min(f_3) = f_3(0, ..., 0) = 0$$

$f04$ - Schwefel's Problem 2.21

$$f_4(x) = max_i\{|x_i|, 1 \leq i \leq 30\}$$

$$-100 \leq x_i \leq 100$$
$$min(f_4) = f_4(0, ..., 0) = 0$$

$f05$ - Generalized Rosenbrock's Function

$$f_5(x) = \sum_{i=1}^{29} |100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2|$$

$$-30 \leq x_i \leq 30$$
$$min(f_5) = f_5(1, ..., 1) = 0$$

$f06$ - Step Function

$$f_6(x) = \sum_{i=1}^{30} (\lfloor x_i + 0.5 \rfloor)^2$$

$$-100 \leq x_i \leq 100$$
$$min(f_6) = f_6(0, ..., 0) = 0$$

$f07$ - Quartic Function with Noise

$$f_7(x) = \sum_{i=1}^{30} i x_i^4 + random[0, 1)$$

$$-1.28 \leq x_i \leq 1.28$$
$$min(f_7) = f_7(0, ..., 0) = 0$$

$f08$ - Generalized Schwefel's Problem 2.26

$$f_8(x) = \sum_{i=1}^{30} \left(x_i sin(\sqrt{|x_i|}) \right)$$

$$-500 \leq x_i \leq 500$$
$$min(f_8) = f_8(420.9687, ..., 420.9687) = -12569.5$$

$f09$ - Generalized Rastrigin's Function

$$f_9(x) = \sum_{i=1}^{30} [x_i^2 - 10cos(2\pi x_i) + 10]$$

$$-5.12 \leq x_i \leq 5.12$$
$$min(f_9) = f_9(0, ..., 0) = 0$$

$f10$ - Ackley's Function

$$f_{10}(x) = -20exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30} x_i^2)}\right) - exp\left(\frac{1}{30}\sum_{i=1}^{30} cos(2\pi x_i)\right) + 20 + e$$

$$-32 \leq x_i \leq 32$$
$$min(f_{10}) = f_{10}(0, ..., 0) = 0$$

$f11$ - Generalized Griewank's Function

$$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$-600 \leq x_i \leq 600$$
$$min(f_{11}) = f_{11}(0, ..., 0) = 0$$

$f12, f13$ - Generalized Penalized Functions

$$f_{12}(x) = \frac{\pi}{30}\left\{10sin^2(\pi y_1) + \sum_{i=1}^{29}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} +$$
$$\sum_{i=1}^{30} u(x_i, 10, 100, 4)$$

$$-50 \leq x_i \leq 50$$
$$min(f_{12}) = f_{12}(1, ..., 1) = 0$$

$$f_{13}(x) = 0.1\left\{sin^2(\pi 3 x_1) + \sum_{i=1}^{29}(x_i - 1)^2[1 + sin^2(3\pi x_{i+1})] + \right.$$
$$\left.(x_n - 1)^2[1 + sin^2(2\pi x_{30})]\right\} +$$
$$\sum_{i=1}^{30} u(x_i, 5, 100, 4)$$

$$-50 \leq x_i \leq 50$$
$$min(f_{13}) = f_{13}(1, ..., 1) = 0$$

where:

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$

$$y_i = 1 + \frac{1}{4}(x_i + 1)$$