Incorporating More Scaled Differences to Differential Evolution

Miguel Cárdenas-Montes(✉)

Centro de Investigaciones Energéticas Medioambientales y Tecnológicas,
Madrid, Spain
miguel.cardenas@ciemat.es

Abstract. Differential Evolution is an evolutionary algorithm composed of vectors and based on the application of scaled differences of two vectors over a third one, being all of them different. The variants of this algorithm propose different types of vectors for the scaled difference, and different number of scaled differences, to alter differently-selected vectors. The successful track of Differential Evolution has propitiated numerous variants. These variants use a limited number of vectors for forming the scaled differences and, in general, only one vector for receiving these differences. In this work, new variants with scaled differences using all the population vectors are proposed. These variants are confronted to a wide set of fitness functions and to a set of Differential Evolution variants.

Keywords: Differential evolution · Performance · Optimization

1 Introduction

Since it was proposed in 1995 [1], Differential Evolution (DE) has produced a successful track of applications to real-parameter optimization. DE is a population-based algorithm, in which its elements are termed vectors. It is based on the application of scaled differences of two of more vectors to other different vector, termed base vector. The weight is termed mutation factor, F This successful track has required the creation of new DE-variants [6].

DE-variants have been created through diverse mechanisms: the creation of scaled differences with two or four vectors randomly selected, the variation of the criterion for selecting some of the vectors involved in the scaled differences, or the selection criterion for the base vector. However, in all cases, the scaled differences are not larger than two[1].

In this work, exploratory studies about efficiently enlarging the number of scaled differences are undertaken. Two approaches for incorporating more scaled differences are presented. The first one is termed DE/rand/all, whereas the second one is termed DE/rand/alldiff (Sect. 2.3).

[1] Each scaled difference involves the selection of a pair of vectors. Therefore, two scaled differences mean the selection of four vectors.

F.J. Martínez de Pisón et al. (Eds.): HAIS 2017, LNAI 10334, pp. 101–112, 2017.
DOI: 10.1007/978-3-319-59650-1_9

In DE/rand/all variant, the vectors of the population are selected one by one, and scaled differences are created with the pairs. Once a vector has been selected, it can not be picked up any more. Taking into account that the base vector is the first selection, for a population with a even number of vectors $\frac{N}{2} - 1$ scaled differences can be formed. The initial attempts in which the mutation factor F acts over the $\frac{N}{2} - 1$ scaled differences leaded to a poor performance. For this reason, in DE/rand/all the mutation factor is halved for consecutive scaled differences, so that for the first scaled difference, the mutation factor is F, for the second one is $F/2$, and so on.

For the second DE-variant proposed in this work, DE/rand/alldiff, the scaled-differences and their addition to the base vector are replaced by a sum of the products of the mutation factor divided by the population size, with a binomial distribution of the sign of F, by vectors randomly selected. Each vector can be selected only once. In DE/rand/1 variant, the scaled difference is applied on a base vector. In DE/rand/alldiff, the addition to the base vector is replaced by adding a factor equal to the inverse of the population size, $\frac{1}{pop\,size}$, to all the factors of the product.

In order to ease the handling of the new proposed variants, a matrix notation for DE has been created (Sect. 2.2). This matrix notation allows representing the classical DE-variants as well as the variants proposed in this work. Furthermore, this alternative representation can inspire new DE-variants.

Finally the proposed DE-variants are confronted to the classic DE-variants (Sect. 2.1) as well as to jitter and dither ones for a wide set of benchmark functions (Sect. 2.4).

To the author's knowledge, no similar works have been proposed.

The rest of the paper is organized as follows: Section 2 summarizes the most relevant points in the state-of-the-art and in-detail describes the proposed variants. Results are presented and analysed in Sect. 3. Finally, Sect. 4 contains the conclusions of this work.

2 Methodology

2.1 Differential Evolution and Its Variants

DE is based on altering the population members (vectors) of each generation with other population members, randomly selected, in its turn is modified by the scaled difference of a pair of population members, being all these members distinct. Two operators compose the DE optimizer: the Mutation operator (Eq. 1) and the Crossover operator. The two most popular implementations of the Crossover operator are the Binomial one (Eq. 2) and the Exponential one (Eq. 3). On the other hand, numerous schemas for the Mutation operator have been proposed.

Mutation operator is governed by a parameter, termed mutation factor F, which quantifies the amount of alteration supplied to the base vector. F weights the addition of the scaled difference to the base vector. The vectors produced by the Mutation operator are termed mutant vectors.

On the other hand, the Crossover operator is characterized by a parameter, termed crossover rate C_r. C_r governs how many components from the mutant vector are inherited by the trial vector.

Finally, a selection process is undertaken. It consists of the selection of the most suitable vector between the target vector, vector selected from population in the current generation, and the trial vector.

$$\boldsymbol{v_i} \; = \; \boldsymbol{x_1} + F \cdot (\boldsymbol{x_2} - \boldsymbol{x_3}), \quad \boldsymbol{x_1} \neq \boldsymbol{x_2} \neq \boldsymbol{x_3} \tag{1}$$

$$u_i(j) \; = \; \begin{cases} v_i(j) \text{ if } rand \; \leq C_r; \\ x_i(j) \text{ otherwise.} \end{cases} \tag{2}$$

$$u_i(j) \; = \; \begin{cases} v_i(j), \;\; \text{for } \; j = \langle n \rangle_D \; \langle n+1 \rangle_D, \ldots, \langle n+L-1 \rangle_D \\ x_i(j), \; \forall j \in [1, D] \end{cases} \tag{3}$$

From the initial publication [1], numerous variants have been proposed to improve the efficiency of the initial implementation. The original authors of DE even proposed some variants [9]. They try to balance the exploration of the search space and the exploitation of the most suitable candidate solutions. For this purpose diverse combination of vectors are selected for the base vectors as well as for the scaled differences. A detailed review of these variants and other ground-breaking modifications for DE can be found in [6,7,15].

For mentioning the DE-variants, a nomenclature based on the pattern DE/X/Y/Z is frequently used. The two initial letters (DE) correspond to the name of the algorithm, whereas the following ones consecutively correspond to the mechanism to select the base vector (X), the number of vectors involved in the scaled difference (Y), and the crossover operator (Z). Following this nomenclature, the classical DE is DE/rand/1/bin (Eqs. 1, 2). When using the Exponential Crossover operator instead of Binomial one, then the nomenclature is DE/rand/1/exp (Eqs. 1, 3).

In this work the following DE-variants have been analysed[2]. In all the tests, the Binomial Crossover operator is used, and therefore omitted in the nomenclature.

DE/rand/1 Original proposal of DE.
$\boldsymbol{v_i} \; = \; \boldsymbol{x_1} + F \cdot (\boldsymbol{x_2} - \boldsymbol{x_3})$

DE/rand/2 In this case, the scaled differences is generated with four vectors.
$\boldsymbol{v_i} \; = \; \boldsymbol{x_1} + F \cdot (\boldsymbol{x_2} + \boldsymbol{x_3} - \boldsymbol{x_4} - \boldsymbol{x_5})$

DE/best/1 In this variant, the base vector is not randomly selected, but the best vector of the current generation is used, $\boldsymbol{x_{best}}$.
$\boldsymbol{v_i} \; = \; \boldsymbol{x_{best}} + F \cdot (\boldsymbol{x_1} - \boldsymbol{x_2})$

DE/best/2 Similar to DE/best/1, but the number of vectors involved in the scaled difference is doubled.
$\boldsymbol{v_i} \; = \; \boldsymbol{x_{best}} + F \cdot (\boldsymbol{x_1} + \boldsymbol{x_2} - \boldsymbol{x_3} - \boldsymbol{x_4})$

[2] For the sake of brevity, the symbol corresponding to the crossover operator has been omitted.

DE/current-to-best/1 In this variant, a difference including the best vector in the current generation and the target vector is included. Furthermore, the target vector is used as base vector.

$\boldsymbol{v_i} \;=\; \boldsymbol{x_i} + F \cdot (\boldsymbol{x_{best}} - \boldsymbol{x_i}) + F \cdot (\boldsymbol{x_1} - \boldsymbol{x_2})$

This variant and the previous ones were proposed by Price and Storn in [9].

DE/rand-to-best/1 In the scaled differences, a difference including the best vector in the current generation and the base vector is included.

$\boldsymbol{v_i} \;=\; \boldsymbol{x_1} + F \cdot (\boldsymbol{x_{best}} - \boldsymbol{x_1}) + F \cdot (\boldsymbol{x_2} - \boldsymbol{x_3})$

DE/rand/2/dir This variant incorporates information of the fitness function. The objective is to guide the evolution towards favourable regions [17].

$\boldsymbol{v_i} \;=\; \boldsymbol{x_1} + \frac{\mu}{2} \cdot (\boldsymbol{x_1} - \boldsymbol{x_2} + \boldsymbol{x_3} - \boldsymbol{x_4})$

where $f(\boldsymbol{x_1}) < f(\boldsymbol{x_2})$ and $f(\boldsymbol{x_3}) < f(\boldsymbol{x_4})$.

All the previous variants have steady values of parameters F and C_r, and therefore they can be considered as static variants. However, these parameter can vary along the cycles. In [9] two variants, termed jitter and dither, are proposed with random values for the scaling factor, F. Dither is used when generating a new random F value for each difference vector; whereas if the new random F value is generated for each dimension of each difference vector, then the variant is termed jitter. In [11] a dither schema with F randomly varying between 0.5 and 1 for each vector is proposed.

For comparison purposed, two dither variants are implemented: the first one for which the scaling parameter is generated from a uniform statistical distribution in the range $(0.5, 1.0)$, and the second one for which F is generated from a Gaussian distribution with parameters $N(0.5, 0.25)$ [11]. Also in [11], a schema where F is reduced from 1.0 to 0.5 is proposed. This schema, termed DETVSF (DE with time varying scale factor), aims at promoting the exploration in the initial cycles, and reinforcing the exploitation in the final ones.

Apart from randomly generated scaling factors, they can be modified taking into account the best suited vectors from the previous generations, self-adapting variants [2,12,18,19]. Similarly to randomly generated, this mechanism aims at improving the overall performance.

The efforts in self-adapting variants arise from the initial studies about the importance of the control parameters for the final performance of the variants [20], and specially due to the dispersion of the values for these parameters. In [1] a value for the mutation factor of $F = 0.5$ is proposed. In [8], authors proposed mutation factor in the range $0.5 < F < 0.95$ with $F = 0.9$ as initial choice.

Some works have explored the DE performance when combining several trial-vector generation strategies, namely multi-strategy variants. Examples can be found in CoDE [21] and in EPSDE [22]. Comparisons with self-adapting or multi-strategy variants are not considered in this work, and are proposed as Future Work.

2.2 Matrix Notation for DE

In the matrix notation for DE/rand/1, the population of N vectors of d dimensions is represented by a matrix $\mathcal{P}$ (Eq. 4). $\mathcal{P}$ is formed by arranging the vectors in rows. x_k^i is the i dimension of the k vector, and it corresponds to the element (k, i) of the matrix $\mathcal{P}$.

$$\mathcal{P} = \begin{bmatrix} x_1^1 & \dots & x_1^i & \dots & x_1^d \\ \vdots & \ddots & & \ddots & \vdots \\ x_k^1 & \dots & x_k^i & \dots & x_k^d \\ \vdots & \ddots & & \ddots & \vdots \\ x_N^1 & \dots & x_N^i & \dots & x_N^d \end{bmatrix} \tag{4}$$

The first point to be addressed is the selection of a vector k from $\mathcal{P}$ (Eq. 5). This can be done by the product of a vector S_k and $\mathcal{P}$. Vector S_k is composed of null elements except for one position with a unitary value. By varying the position of the unitary value, distinct vectors can be selected.

$$\boldsymbol{x_k} = S_k \times \mathcal{P} = \begin{bmatrix} 0 \dots 1 \dots 0 \end{bmatrix}_k \times \begin{bmatrix} x_1^1 & \dots & x_1^i & \dots & x_1^d \\ \vdots & \ddots & & \ddots & \vdots \\ x_k^1 & \dots & x_k^i & \dots & x_k^d \\ \vdots & \ddots & & \ddots & \vdots \\ x_N^1 & \dots & x_N^i & \dots & x_N^d \end{bmatrix} \tag{5}$$

Thus, the difference between two vectors, $\boldsymbol{x_i} - \boldsymbol{x_j}$, can be rewritten as Eq. 6.

$$\boldsymbol{x_i} - \boldsymbol{x_j} = S_i \times \mathcal{P} \; - \; S_j \times \mathcal{P} = S_{ij} \times \mathcal{P} \tag{6}$$

where S_{ij} is a vector with all elements null, except for two elements which take positive and negative unitary values:

$$\begin{bmatrix} \dots 0 \; 1 \; 0 \dots 0 \; {-1} \; 0 \dots \end{bmatrix}_{ij} \tag{7}$$

Taking into account this notation and incorporating the mutation factor, Eq. 1 can be rewritten as Eq. 8.

$$\boldsymbol{x_k} + F \cdot (\boldsymbol{x_i} - \boldsymbol{x_j}) = S_k \times \mathcal{P} + F \cdot S_{ij} \times \mathcal{P} = F^*_{kij} \times \mathcal{P} \tag{8}$$

where F^*_{kij} is a vector with dimensionality $1 \times \mathit{Pop.size}$ (Eq. 9). The omitted elements in F^*_{kij} are all zeros.

$$F^*_{kij} = \begin{bmatrix} \dots F \dots 1 \dots -F \dots \end{bmatrix} \tag{9}$$

The F^*_{kij} can be vertically stacked, so that a matrix of $\mathit{Popsize} \times \mathit{Popsize}$ is established, $\mathcal{F}^*$. Each row of $\mathcal{F}^*$ is created similarly to Eq. 9, with random positions for the values: 1, F and $-F$.

In this way, the next generation of mutant vectors, $\mathcal{P}_{mutant}$ is produced by the product of the $\mathcal{F}^*$ matrix and the current population $\mathcal{P}_G$ (Eq. 10).

$$\mathcal{P}_{mutant} = \mathcal{F}^* \times \mathcal{P}_G \tag{10}$$

The DE/rand/2 variant can be easily expressed with the $\mathcal{F}^*$ notation (Eq. 9). For this variant, two positions per row of zero values have to be replaced by a F value and a $-F$ value. For other variants involving a larger number of vectors, DE/rand/Y, can also expressed by modifying the number of positions with F and $-F$ values. The DE/best/Y variants are transposed to matrix notation by replacing the S_k vector selector, by the appropriate selector for the best vector of the current generation, S_{best}.

In all these implementations, $\mathcal{F}^*$ keeps the following property $\sum_i \mathcal{F}^*_{ij} = 1$. This property is used as common ground for new DE-variants.

2.3 New Variants for Differential Evolution

The $\mathcal{F}^*$ notation paves the way for new variants of DE. For example, the $\mathcal{F}^*$ notation eases the implementation of a variant involving all the vectors. If the population size is odd, in DE/rand/all all the vectors of the population contribute as base vector or as part of the scaled differences. In this case, $\mathcal{F}^*$ has not zero elements. Conversely, if the population size is even, then one element of the matrix is null. The sum of any row of $\mathcal{F}^*$ is the unit $\sum_i \mathcal{F}^*_{ij} = 1$.

The first attempt for implementing DE/rand/all simply creates as scaled differences as possible with the population. However, this increment in the number of scaled differences carries out a degradation of the performance. For this reason, an alternative implementation, DE/rand/all$\frac{F}{2^n}$, where the mutation factor for the scaled differences are progressively halved is proposed of DE/rand/all (Eq. 11).

$$F^* = \begin{bmatrix} -F \ldots & F/2 & \ldots & 1 & \ldots -F/2 \ldots & F \\ \vdots \quad \ddots & & \ddots & & \ddots & \ldots \vdots \\ 1 \quad \ldots & -F/2 & \ldots F/2 \ldots & & -F & \ldots F \end{bmatrix} \tag{11}$$

In DE/rand/alldiff the elements of matrix $\mathcal{F}^*$ are generated following the schema $(-1)^{B(0.5)} \cdot \frac{0.1*U(0,1)}{pop\,size} + \frac{1}{pop\,size}$, where $U(0,1)$ is a uniform probability distribution and $B(0.5)$ is a binomial distribution of probability 0.5. In this schema the base vector is completely removed, whereas the scaled differences are not longer created by subtracting of pairs of vectors. Conversely, each vector is weighted through the product of fixed and variable factors. The contribution of the base vector is equally distributed among all the vector of the population through the factor $\frac{1}{pop\,size}$. This schema keeps the property $\sum_i \mathcal{F}^*_{ij} = 1$.

2.4 Benchmark Functions

In order to evaluate the new DE-variants, a set of fitness functions are used (Table 1). This set includes both multimodal and monomodal, separable and non-separable functions. These fitness functions have been used in CEC contests [13,14] and also in works where presenting cutting-edge DE-variants [2,3,5,10,16].

In all the tests performed in this work, a dimensionality of 30 and a population size of 10 vector have been used. Two configurations are used for the number of cycles: 100 and 1000, and the mean and the standard deviation of 25 independent runs per case are shown. Generally, a larger number of cycles leads to higher-quality solutions. However, it is specially interesting when these high-quality solutions can be produced with few cycles, thus CPU-time is saved. As pseudorandom number generator, a subroutine based on Mersenne Twister [4] has been used.

3 Results and Discussion

In this section the performance of the proposed variants are confronted to the a set of classic variants of DE for a set of benchmark functions. Tests with jitter and dither variants are also undertaken.

In Table 2, the numerical results obtained with DE/rand/alldiff variant are compared with the previous best results of the classical DE-variants[3]. Two configurations for the number of cycles are used: 100 and 1000 cycles. This intends to evaluate the DE-variants when trying to produce high-quality solutions with short number of cycles and when enough number of cycles to maximize the performance of the variants are supplied.

As can be appreciated, the DE/rand/alldiff variant outperforms the other static DE-variants in 26 of the 32 cases. They exclude the two configurations of the functions: Rastrigin, Schwefel, and Styblinski-Tang. Two of these functions: Schwefel and Styblinski-Tang, have optimal solution not in $\mathbf{0}$, whereas the third one, Rastrigin function, has the optimal solution in $\mathbf{0}$. For the other 13 functions, including Whitley function which has the optimal solution in $\mathbf{1}$, the DE/rand/alldiff produces better solutions than the other DE-variants included in the test. Concerning the cases where DE/rand/alldiff is outperformed, the DE-variants producing the best results vary from DE/rand/1 and DE/best/2 with 2 cases, and DE/rand/2/dir and DE/rand-to-best/1 with 1 case.

With regard to the performance of DE/rand/all$\frac{F}{2^n}$ (comparing with static DE-variants and DE/rand/alldiff), it only outperforms a single case. For Styblinski-Tang and 1000 cycles, the best mean result obtained is -1111 ± 26, whereas for DE/rand/all$\frac{F}{2^n}$ is -1127 ± 26. Although this is a single case, it corresponds to a function for which the results of DE/rand/alldiff systematically is outperformed by other variants. So that, it is a good candidate to hybridize with other DE-variant, or for studying its features in the exploration-exploitation phases.

Beyond the static DE-variants, further comparisons can be made with dither and jitter variants (Table 3). When comparing with these variants, the

[3] The best DE variant for each configuration and fitness function appears in boldface type.

Table 1. Benchmark functions used in this work.

Name	Function	Search range	Optimum
Ackley	$f(\boldsymbol{x}) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)) + 20 + \exp$	$[-32.0, 32.0]$	$0\, at\, \mathbf{0}$
Griewank	$f(\boldsymbol{x}) = 1 + \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} cos(\frac{x_i}{\sqrt{i}})$	$[-600.0, 600.0]$	$0\, at\, \mathbf{0}$
Hyperellipsoid	$f(\boldsymbol{x}) = \sum_{i=1}^{D} i \cdot x_i^2$	$[-5.12, 5.12]$	$0\, at\, \mathbf{0}$
Rana	$f(\boldsymbol{x}) = \sum_{i=1}^{D-1}(x_{i+1} + 1.0) \cdot cos(t_2) \cdot sin(t_1) + cos(t_1) \cdot sin(t_2) \cdot x_i$	$[-100.0, 100.0]$	$0\, at\, \mathbf{0}$
	$t_1 = \sqrt{\lvert x_{i+1} + x_i + 1.0\rvert},\ t_2 = \sqrt{\lvert x_{i+1} - x_i + 1.0\rvert}$		
Rastrigin	$f(\boldsymbol{x}) = 10 \cdot D + \sum_{i=1}^{D}(x_i^2 - 10 \cdot cos(2\pi x_i))$	$[-5.12, 5.12]$	$0\, at\, \mathbf{0}$
Rosenbrock	$f(\boldsymbol{x}) = \sum_{i=1}^{D-1}(x_i - 1)^2 + 100 \cdot (x_{i+1} - x_i^2)^2$	$[-30, 30]$	$0\, at\, \mathbf{0}$
Schaffer's F6	$f(\boldsymbol{x}) = 0.5 + \frac{sin^2(\sqrt{\sum_{j=1}^{D} x_i^2}) - 0.5}{[1+0.001 \cdot (\sum_{j=1}^{D} x_i^2)]^2}$	$[-100.0, 100.0]$	$0\, at\, \mathbf{0}$
Schaffer's F7	$f(\boldsymbol{x}) = \frac{1}{n-1}\sum_{i=1}^{D-1}[\sqrt{s_i} \cdot (sin^2(50.0 s_i^{\frac{1}{5}}) + 1)]^2,$	$[-100.0, 100.0]$	$0\, at\, \mathbf{0}$
	$s_i = \sqrt{x_i^2 + x_{i+1}^2}$		
Schwefel	$f(\boldsymbol{x}) = \sum_{i=1}^{D}(-x_i \cdot sin\sqrt{\lvert x_i\rvert}) + 418.982887 \cdot D$	$[-500.0, 500.0]$	$0\, at$ $\mathbf{420.968746}$
Schwefel Problem 1.2	$f(\boldsymbol{x}) = \sum_{i=1}^{D}(\sum_{j=1}^{i} x_j)^2$	$[-100.0, 100.0]$	$0\, at\, \mathbf{0}$
Schwefel Problem 2.22	$f(\boldsymbol{x}) = \sum_{i=0}^{D} \lvert x_i\rvert + \prod_{i=0}^{D} \lvert x_i\rvert$	$[-10.0, 10.0]$	$0\, at\, \mathbf{0}$
Schwefel Problem 2.21	$f(\boldsymbol{x}) = \max \lvert x_i\rvert, 1 \leq i \leq D$	$[-100.0, 100.0]$	$0\, at\, \mathbf{0}$
Sphere	$f(\boldsymbol{x}) = \sum_{i=1}^{D} x_i^2$	$[-5.12, 5.12]$	$0\, at\, \mathbf{0}$
Step	$f(\boldsymbol{x}) = \sum_{j=1}^{D}(\lfloor x_i + 0.5\rfloor)^2$	$[-1000, 1000]$	$0\, at\, \mathbf{0}$
Styblinski-Tang	$f(\boldsymbol{x}) = 0.5 \cdot \sum_{i=1}^{D} x_i^4 - 16x_i^2 + 5x_i$	$[-5, 5]$	$-39.16599 \cdot D$ $at\, \mathbf{-2.903534}$
Whitley	$f(\boldsymbol{x}) = \sum_{i=1}^{D}\sum_{j=1}^{D}(\frac{(s_{ij})^2}{4000} - cos(s_{ij}) + 1)$	$[-10.24, 10.24]$	$0\, at\, \mathbf{1}$
	$s_{ij} = 100(x_i^2 - x_j)^2 + (1 - x_j)^2$		
Zakharov	$f(\boldsymbol{x}) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5 \cdot i \cdot x_i)^2 + (\sum_{i=1}^{D} 0.5 \cdot i \cdot x_i)^4$	$[-5, 10]$	$0\, at\, \mathbf{0}$

Table 2. Comparison of DE/rand/alldiff variant with best results of the previous DE-variants. 25 independent runs per case with $F = Cr = 0.5$, for two configurations for the number of cycles: 100 and 1000, and Binomial Crossover operator.

Function	Cycles	Best previous		DE/rand/alldiff
Ackley	100	DE/best/2	6.83 ± 1.46	**0.36 ± 0.08**
	1000	DE/best/2	$(5.1 \pm 9.7)10^{-5}$	**$(6.5 \pm 0.7)10^{-15}$**
Griewank	100	DE/best/2	17.19 ± 9.88	**0.030 ± 0.011**
	1000	DE/best/2	$(0.7 \pm 1.3)10^{-9}$	**$(7.5 \pm 5.2)10^{-17}$**
Hyperellipsoid	100	DE/best/2	48.88 ± 22.25	**0.10 ± 0.05**
	1000	DE/best/2	$(0.8 \pm 3.1)10^{-8}$	**$(1.5 \pm 2.5)10^{-34}$**
Rastrigin	100	DE/best/1	**115.47 ± 35.91**	149.97 ± 27.59
	1000	DE/rand/2/dir	**52.29 ± 19.77**	86.03 ± 16.04
Rosenbrock	100	DE/best/2	$(48 \pm 20)10^{3}$	**68.95 ± 18.02**
	1000	DE/best/2	$(1.3 \pm 0.8)10^{3}$	**28.80 ± 0.16**
Schaffer's F6	100	DE/rand-to-best/1	$(4925 \pm 58)10^{-4}$	**$(3715 \pm 444)10^{-4}$**
	1000	DE/rand/1	0.27 ± 0.11	**0.03724 ± 0.00007**
Schaffer's F7	100	DE/rand-to-best/1	25.61 ± 0.80	**4.15 ± 0.96**
	1000	DE/rand/1	1.22 ± 1.46	**0.0012 ± 0.0006**
Schwefel	100	DE/best/1	**5130 ± 734**	9948 ± 398
	1000	DE/rand/1	**1282 ± 977**	8891 ± 512
Schwefel P. 1.2	100	DE/current-to-best/1	12766 ± 4593	**99 ± 47**
	1000	DE/best/2	4621 ± 1931	**$(1.0311 \pm 3.6)10^{-12}$**
Schwefel P. 2.21	100	DE/current-to-best/1	34.07 ± 6.51	**3.80 ± 1.07**
	1000	DE/rand/2	22.94 ± 6.96	**$(6.8 \pm 3.5)10^{-8}$**
Schwefel P. 2.22	100	DE/rand-to-best/1	18.07 ± 7.00	**0.67 ± 0.18**
	1000	DE/best/2	$(0.3 \pm 1.8)10-3$	**$(1.2 \pm 0.8)10^{-17}$**
Sphere	100	DE/rand/2/dir	1.99 ± 2.28	**0.008 ± 0.004**
	1000	DE/best/2	$(1.7 \pm 3.5)10^{-10}$	**$(4.3 \pm 5.9)10^{-36}$**
Step	100	DE/best/2	$(0.16 \pm 0.07)10^{6}$	**317.16 ± 181.21**
	1000	DE/rand/2	1.12 ± 1.63	**0.0 ± 0.0**
Styblinski -Tang	100	DE/rand-to-best/1	**-947 ± 41**	-670 ± 62
	1000	DE/rand/1	**-1111 ± 26**	-838 ± 59
Whitley	100	DE/best/2	$(5 \pm 4)10^{6}$	**0.48 ± 0.06**
	1000	DE/rand/2	1.25 ± 1.78	**0.3590 ± 0.0006**
Zakharov	100	DE/rand-to-best/1	93.7 ± 38.3	**1.5 ± 1.3**
	1000	DE/best/2	0.41 ± 1.14	**$(1.84 \pm 9.03)10^{-10}$**

DE/rand/alldiff variant outperforms the dither and jitter variants in 27 of the 32 cases. With regard to the cases where DE/rand/alldiff is outperformed, the two configuration of Schwefel and Styblinski-Tang functions can be mentioned, as well as the Rastrigin function when using 1000 cycles.

In the previous comparisons, it is appreciated a slight bias toward a better performance when the fitness function has the optimal solution at **0**. Probably

Table 3. Comparison of DE/rand/alldiff variant with best results of the DE dither and jitter variants. 25 independent runs per case with $F = Cr = 0.5$, for two configurations for the number of cycles: 100 and 1000, and Binomial Crossover operator.

Function	Cycles	Dither Uniform	Dither Gaussian	Jitter	DE/rand/alldiff
Ackley	100	16.07 ± 1.08	8.49 ± 1.62	8.27 ± 1.22	$\mathbf{0.36 \pm 0.08}$
	1000	0.015 ± 0.009	4.65 ± 3.18	2.80 ± 2.28	$\mathbf{(6.5 \pm 0.7)10^{-15}}$
Griewank	100	190.70 ± 57.76	24.65 ± 10.45	25.27 ± 9.79	$\mathbf{0.030 \pm 0.011}$
	1000	$(4.3 \pm 3.3)10^{-5}$	3.06 ± 2.60	2.55 ± 6.057	$\mathbf{(7.5 \pm 5.2)10^{-17}}$
Hyperellipsoid	100	559.09 ± 209.25	108.23 ± 41.46	89.37 ± 35.55	$\mathbf{0.10 \pm 0.05}$
	1000	$(14 \pm 10)10^{-5}$	22.65 ± 26.50	3.87 ± 7.84	$\mathbf{(1.5 \pm 2.5)10^{-34}}$
Rastrigin	100	314.7 ± 26.7	207.3 ± 22.2	235.5 ± 33.7	$\mathbf{149.97 \pm 27.59}$
	1000	165.8 ± 20.0	$\mathbf{30.0 \pm 8.2}$	103.1 ± 23.0	86.03 ± 16.04
Rosenbrock	100	$(441.0 \pm 112.7)10^3$	$(80.9 \pm 39.9)10^3$	$(145.2 \pm 61.9)10^3$	$\mathbf{68.95 \pm 18.02}$
	1000	8621 ± 5030	8013 ± 7775	3438 ± 2235	$\mathbf{28.80 \pm 0.16}$
Schaffer's F6	100	$0.49987 \pm 8 \cdot 10^{-5}$	0.4994 ± 0.0003	0.4996 ± 0.0003	$\mathbf{(3715 \pm 444)10^{-4}}$
	1000	0.38 ± 0.05	0.36 ± 0.10	0.26 ± 0.10	$\mathbf{0.03724 \pm 0.00007}$
Schaffer's F7	100	100.98 ± 10.47	43.62 ± 10.91	49.09 ± 9.36	$\mathbf{4.15 \pm 0.96}$
	1000	4.79 ± 2.01	4.25 ± 2.62	0.79 ± 0.59	$\mathbf{0.0012 \pm 0.0006}$
Schwefel	100	10184 ± 492	$\mathbf{7949 \pm 670}$	9383 ± 495	9948 ± 398
	1000	7862 ± 453	$\mathbf{2459 \pm 586}$	4517 ± 1915	8891 ± 512
Schwefel Problem 1.2	100	126024 ± 34941	64622 ± 12833	76447 ± 20686	$\mathbf{99 \pm 47}$
	1000	67049 ± 18226	10405 ± 5685	22894 ± 10063	$\mathbf{(1.0311 \pm 3.6)10^{-12}}$
Schwefel Problem 2.21	100	81.11 ± 5.18	56.90 ± 8.84	68.72 ± 9.59	$\mathbf{3.80 \pm 1.07}$
	1000	23.41 ± 8.879	45.18 ± 9.13	37.63 ± 7.82	$\mathbf{(6.8 \pm 3.5)10^{-8}}$
Schwefel Problem 2.22	100	848.70 ± 1776.78	16.22 ± 3.58	29.26 ± 8.34	$\mathbf{0.67 \pm 0.18}$
	1000	0.015 ± 0.007	1.73 ± 2.35	0.36 ± 0.80	$\mathbf{(1.2 \pm 0.8)10^{-17}}$
Sphere	100	52.46 ± 18.05	7.80 ± 4.74	6.38 ± 2.04	$\mathbf{0.008 \pm 0.004}$
	1000	$(0.11e \pm 9.14)10^{-6}$	0.98 ± 1.60	0.56 ± 1.01	$\mathbf{(4.3 \pm 5.9)10^{-36}}$
Step	100	$(20.3 \pm 5.8)10^5$	$(2.6 \pm 1.2)10^5$	$(2.4 \pm 0.9)10^5$	$\mathbf{317 \pm 181}$
	1000	1.36 ± 2.33	83705 ± 128632	16079 ± 25020	$\mathbf{0.0 \pm 0.0}$
Styblinski -Tang	100	-570 ± 48	$\mathbf{-905 \pm 48}$	-744.9 ± 60.618	-670 ± 62
	1000	-1101 ± 73	-1072 ± 25	$\mathbf{-1104 \pm 25}$	-838 ± 59
Whitley	100	$(4.0 \pm 3.0)10^9$	$(36.7 \pm 67.9)10^6$	$(28.8 \pm 22.7)10^6$	$\mathbf{0.48 \pm 0.06}$
	1000	2.07 ± 3.43	$(8.5 \pm 20.2)10^6$	$0.24 \pm 0.50)10^6$	$\mathbf{0.3590 \pm 0.0006}$
Zakharov	100	683.8 ± 126.4	437.4 ± 148.3	427.3 ± 93.6	$\mathbf{1.5 \pm 1.3}$
	1000	290.00 ± 113.28	4.53 ± 7.60	5.14 ± 6.45	$\mathbf{(1.84 \pm 9.03)10^{-10}}$

this is due to a greedy behaviour of the DE/rand/alldiff: it is intensive in the exploitation of promising-candidate solutions, but its efficiency for exploring the search space decreases. In order to improve the efficient of this variant in future tests, this greedy behaviour should be corrected by improving the exploratory capacity.

4 Conclusions

In this paper, two new variants of DE are proposed and evaluated: DE/rand/all$\frac{F}{2^n}$ and DE/rand/alldiff. Differently to the variants proposed in the past, these new variants include scaled differences involving all the vectors. In DE/rand/all$\frac{F}{2^n}$ progressively halves the mutation factor, so that the importance of the scaled difference is also reduced. And in DE/rand/alldiff, the base vector and the scaled differences are replaced by weighted contributions of all the vectors in the population. The results state that this last variant outperforms a large set of other DE-variants, including jitter and dither ones, over a wide set of fitness functions.

Furthermore, a matrix notation for generating the mutant vectors population in DE is introduced. This notation helps visualizing the process, at the same time it might inspire new DE-variants.

This work opens diverse lines in relation of Future Work. They include the adaptation of the DE/randalldiff and the DE/rand/all$\frac{F}{2^n}$ to adaptive variation of the mutation factor, statistical studies about the values of the scaled differences for these and other variants, and the generation of new variants based on the matrix notation.

Acknowledgement. The research leading to these results has received funding by the Spanish Ministry of Economy and Competitiveness (MINECO) for funding support through the grant FPA2013-47804-C2-1-R, FPA2016-80994-C2-1-R, and "Unidad de Excelencia María de Maeztu": CIEMAT - FÍSICA DE PARTÍCULAS through the grant MDM-2015-0509.

References

1. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1997)
2. Zamuda, A., Brest, J.: Self-adaptive control parameters' randomization frequency and propagations in differential evolution. Swarm Evol. Comput. **25**, 72–99 (2015)
3. Peñuñuri-Anguiano, F.R., Cab-Cauich, C.A., Carvente-Muñoz, O., Zambrano-Arjona, M.A., Tapia-González, J.A.: A study of the classical differential evolution control parameters. Swarm Evol. Comput. **26**, 86–96 (2016)
4. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator. ACM Trans. Model. Comput. Simul. **8**(1), 3–30 (1999)
5. Mezura-Montes, E., Velazquez-Reyes, J., Coello, C.A.C.: A comparative study of differential evolution variants for global optimization. In: GECCO, pp. 485–492 (2006)

6. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evol. Comput. **15**(1), 4–31 (2011)
7. Neri, F., Tirronen, V.: Recent advances in differential evolution: a survey and experimental analysis. Artif. Intell. Rev. **33**(1), 61–106 (2010)
8. Rönkkönen, J., Kukkonen, S., Price, K.V.: Real-parameter optimization with differential evolution. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, Edinburgh, UK, 2–4 , pp. 506–513. IEEE (2005)., September 2005
9. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution a Practical Approach to Global Optimization. Natural Computing Series. Springer, Berlin (2005)
10. Lu, X., Tang, K., Sendhoff, B., Yao, X.: A new self-adaptation scheme for differential evolution. Neurocomput. 146(C), 2–16 (2014)
11. Das, S., Konar, A., Chakraborty, U.K.: Two improved differential evolution schemes for faster global search. In: Genetic and Evolutionary Computation Conference, GECCO 2005, Proceedings, Washington DC, USA, 25–29 , pp. 991–998. ACM (2005)., June 2005
12. Lu, X., Tang, K., Sendhoff, B., Yao, X.: A new self-adaptation scheme for differential evolution. Neurocomputing **146**, 2–16 (2014)
13. Tang, K., Li, X., Suganthan, P.N., Yang, Z., Weise, T.: Benchmark functions for the cec'2010 special session and competition on large-scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory (NICAL), School of Computer Science and Technology, University of Science and Technology of China (USTC) (2009)
14. Tang, K., Yao, X., Suganthan, P.N., MacNish, C., Chen, Y.P., Chen, C.M., Yang, Z.: Benchmark functions for the CEC 2008 special session and competition on large scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China (2007)
15. Das, S., Mullick, S.S., Suganthan, P.: Recent advances in differential evolution-an updated survey. Swarm Evol. Comput. **27**, 1–30 (2016)
16. Chen, Y., Xie, W., Zou, X.: A binary differential evolution algorithm learning from explored solutions. Neurocomputing **149**, 1038–1047 (2015)
17. Feoktistov, V., Janaqi, S.: Generalization of the strategies in differential evolution. In: 18th International Parallel and Distributed Processing Symposium (IPDPS 2004), CD-ROM/Abstracts Proceedings, Santa Fe, New Mexico, USA, 26–30. IEEE Computer Society (2004)., April 2004
18. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Evol. Comput. **13**(2), 398–417 (2009)
19. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, Edinburgh, UK, 2–4 , pp. 1785–1791. IEEE (2005)., September 2005
20. Gämperle, R., Müller, S.D., Koumoutsakos, P.: A parameter study for differential evolution. In: WSEAS International Conference on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, Press, pp. 293–298 (2002)
21. Wang, Y., Cai, Z., Zhang, Q.: Differential evolution with composite trial vector generation strategies and control parameters. Trans. Evol. Comp. **15**(1), 55–66 (2011)
22. Mallipeddi, R., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F.: Differential evolution algorithm with ensemble of parameters and mutation strategies. Appl. Soft Comput. **11**(2), 1679–1696 (2011)