Influence of Crossover on the Behavior of Differential Evolution Algorithms

Daniela Zaharie

Faculty of Mathematics and Computer Science
West University of Timişoara
bv. Vasile Pârvan, nr. 4, 300223 Timişoara, Romania

Abstract

In Differential Evolution algorithms the crossover operator allows the construction of a new trial element starting from the current and mutant elements. Thus it controls which and how many components are mutated in each element of the current population. This work aims to analyze the impact the crossover operator and its parameter, the crossover rate, has on the behavior of Differential Evolution. The influence of the crossover rate on the distribution of the number of mutated components and on the probability for a component to be taken from the mutant vector (mutation probability) is theoretically analyzed for several variants of crossover, including classical binomial and exponential strategies. For each crossover variant the relationship between the crossover rate and the mutation probability is identified and its impact on the choice and adaptation of control parameters is analyzed theoretically and numerically. The numerical experiments illustrate the fact that the difference between binomial and exponential crossover variants is mainly due to different distributions of the number of mutated components. On the other hand, the behavior of exponential crossover variants was found to be more sensitive to the problem size than the behavior of variants based on binomial crossover.

Key words: differential evolution, binomial crossover, exponential crossover, parameter control, self-adaptation

1. Introduction

Differential evolution (DE) [20] is a population based stochastic meta-heuristic for global optimization on continuous domains related both with simplex methods (e.g. Nelder Mead) and evolutionary algorithms. Due to its simplicity, effectiveness and robustness, DE has been successfully applied in solving optimization problems arising in different practical applications (e.g. parameter

Email address: dzaharie@info.uvt.ro (Daniela Zaharie)

identification [27], image processing [6],[13], data clustering [16], optimal design [2], scheduling [12] etc).

The main particularity of DE is that it constructs, at each generation, for each element of the population a so-called *mutant vector*. This mutant vector is constructed through a specific mutation operation based on adding differences between randomly selected elements of the population to another element. For instance one of the simplest and most used variant to construct a mutant vector, y, starting from a current population $\{x_1, \ldots, x_m\}$ is based on the following rule: $y = x_{r_1} + F \cdot (x_{r_2} - x_{r_3})$ where r_1, r_2 and r_3 are distinct random indices selected from $\{1, \ldots, m\}$ and $F > 0$ is a *scaling factor*. This difference based mutation operator which is more related to a recombination than to a classical mutation operator is the distinctive element of DE algorithms. Its main property is the fact that it acts as a self-referential mutation allowing a gradual exploration of the search space [17].

Based on the mutant vector, a *trial vector* is constructed through a crossover operation which combines components from the current element and from the mutant vector, according to a control parameter $CR \in [0, 1]$, called crossover rate. This trial vector competes with the corresponding element of the current population and the best one, with respect to the objective function, is transferred into the next generation.

The behavior of DE is influenced both by the mutation and crossover operators and by the values of the involved parameters (e.g. F and CR). During the last decade a lot of papers addressed the problem of finding insights concerning the behavior of DE algorithms. Thus, parameter studies involving different sets of test functions were conducted [7], [11], [15] and a significant number of adaptive and self-adaptive variants have been proposed [3],[10],[19],[25],[31]. Most of these results were obtained based on empirical studies. Despite some theoretical analysis of the DE behavior [1],[4],[17],[29],[30] the theory of DE is still behind the empirical studies. Thus theoretical insights concerning the behavior of DE are highly desirable.

On the other hand, most of DE variants and studies are related with the mutation operator. The larger emphasis on mutation is illustrated by the large number of mutation variants, some of them being significantly different from the first versions of DE (e.g. [1], [9]). The crossover operator attracted much less attention, just two variants being currently used, the so-called binomial and exponential crossover. The exponential crossover is that proposed in the original work of Storn and Price [20], but in applications the binomial variant presented in [21] is more frequently used.

The current knowledge concerning the influence of the crossover strategy on the behavior of DE is limited to some experimental remarks. Besides statements like "*The crossover method is not so important ... and binomial is never worse than exponential*" and "*if you choose binomial crossover like, CR is usually higher than in the exponential crossover variant*" [34] or some experimental studies involving both binomial and exponential variants [11] no systematic analysis of crossover strategies in DE was conducted. Thus there still are questions to be answered, e.g.: (i) is binomial crossover better than exponential crossover

or viceversa? (ii) does the crossover variant has an impact on the choice of DE control parameters ? (iii) is the behavior of crossover influenced by the problem size ?

The aim of this paper is to analyze both from a theoretical and a numerical point of view the influence the crossover variant has on the behavior of DE and to try to find answers to above mentioned questions. Some preliminary results concerning the comparison between the binomial and exponential crossover were presented in [32] where relationships between the crossover rate (CR) and the mutation probability (the probability that a component of the trial vector comes from the mutant vector) were obtained for both strategies. Based on these results one can explain why adequate values of the crossover rates are different for different crossover strategies.

In this paper we extend the analysis started in [32] by studying the properties of other related crossover strategies and the impact of crossover on the choice of adequate values for CR and on the behavior of self-adaptive variants based on uniformly random selection of the crossover rate values (as jDE proposed by Brest [3]). Section 2 contains an overview of the main DE strategies and of their particularities. Section 3 presents the properties of the classical binomial and exponential crossover operators and some new implementation versions based on the idea of exponential crossover. For each variant the distribution of the number of mutated elements is derived and the relationship between the crossover rate and the mutation probability is analyzed. In Section 4 is presented an experimental parameter study aiming to emphasize the influence the crossover type has on the behavior of DE for some test functions. An analysis of the impact of crossover type on the choice of appropriate values for the control parameters and on some self-adaptive variants is presented in Section 5. Some conclusions are presented in Section 6.

2. Overview of Differential Evolution

In the following we shall consider objective functions, $f : D \subset \mathbb{R}^n \to \mathbb{R}$, to be minimized, thus we are dealing with minimization problems of size n. The overall structure of differential evolution is typical for evolutionary algorithms consisting of two main steps: initialization and an iterative transformation of a population of vectors from $D \subset \mathbb{R}^n$ which are candidate solutions. Each iteration corresponds to an evolutionary generation and consists of constructing new elements by mutation and crossover, evaluating the new elements and selecting those which are included in the next generation.

In the following we will denote by $\{x_1(g), x_2(g), \ldots, x_m(g)\}$ the population corresponding to the generation g. For each element x_i in the current population is constructed a mutant element denoted by y_i and from x_i and y_i is constructed a trial element denoted by z_i. By using these notations the general structure of the differential evolution is described in Algorithm 1, the particularities of a given DE instance being related with the particularities of the mutation and crossover operators. The selection operator is the same for almost all implementations of a DE algorithm: the trial element is compared with the current

element and the best of them is transferred in the new population. The choice of the stopping condition depends, as in the case of most evolutionary algorithms, on the knowledge we have on the problem. In the case of numerical studies on test functions, the evolution is usually stopped when the optimal value was approximated within a given accuracy or the number of function evaluations reached an upper bound. In the case of real world problems when neither the optimal value nor the length of the evolution is known in advance information on the progress of evolution could be used. In [33] is presented a comprehensive study on different stopping criteria. The most promising seem to be those related with the population diversity.

Algorithm 1 The general structure of a generational DE

```
Population initialization X(0) ← {x_1(0), ..., x_m(0)}
g ← 0
Compute {f(x_1(g)), ..., f(x_m(g))}
while the stopping condition is false do
   for i = 1,m do
      y_i ← generateMutant(X(g))
      z_i ←crossover(x_i(g),y_i)
      if f(z_i) < f(x_i(g)) then
         x_i(g+1) ← z_i
      else
         x_i(g+1) ← x_i(g)
      end if
   end for
   g ← g+1
   Compute {f(x_1(g)), ..., f(x_m(g))}
end while
```

By combining different mutation and crossover operators various schemes have been designed. In the DE literature these schemes are specified by using the convention $DE/a/b/c$ where a and b specify the manner of constructing the mutant vector and c denotes the crossover type. Several of these variants are resumed in the following.

2.1. Differential mutation

The construction of a mutant vector, y_i, consists of selecting from the current population a so-called base vector, x_{r_1}, and of adding to it a scaled difference between two other vectors, x_{r_2} and x_{r_3}:

$$y_i = x_{r_1} + F \cdot (x_{r_2} - x_{r_3}), \qquad F \in (0, 2) \tag{1}$$

The constant F in eq. (1) is a scale factor which influences the diversity of the set of mutant vectors. The indices r_1, r_2, r_3 are usually randomly selected from $\{1, \ldots, m\}$ but such they satisfy the restriction to be distinct and different of i. The influence of this restriction on the DE behavior is analyzed in [17]

where is stated that it enables DE to achieve a good convergence. Choosing randomly the base vector leads to the class of DE/rand/1/* algorithms. If instead of using one difference as perturbation of the base vector one uses two differences, e.g. $y_i = x_{r_1} + F \cdot (x_{r_2} - x_{r_3}) + F \cdot (x_{r_4} - x_{r_5})$, one obtains the class of DE/rand/2/* algorithms. Using more than one difference did not prove to bring a significant difference in the DE behavior, therefore this variant is much less used as the classical DE/rand/1/*. The distribution of trial points generated by the classical DE/rand/1 mutation is derived in [1] where is also proposed a differential free trial point generation based on approximating the derived distribution.

Besides the random choice of the base vector there are other variants which involves the best element or the current element of the population. Thus if the base vector is the best one in the current population, x_*, then one obtains the DE/best/*/* class of algorithms. Other variants are: DE/current-to-best/*/*, which uses $\lambda x_* + (1-\lambda)x_i$ instead x_{r_1} and DE/current-to-rand/*/*, which uses $\lambda x_{r_1} + (1-\lambda)x_i$. In both cases $\lambda \in (0,1)$ controls the influence of the best or the random element on the base vector. Another variant to construct the base vector is to compute the center of the triangle formed by three randomly selected elements as in trigonometric DE and to use three difference terms corresponding to all pairs of elements [9].

Recently, were developed variants which limits the choice of the vectors involved in the mutant element to a neighborhood, defined by a ring topology, of the current element [5],[14]. Another variant of choosing the parents is that based on ranking the population elements [23]. This approach is based on the idea that by increasing the parents selection pressure one can improve the ability of DE to deal with nonseparable functions.

Since the aim of this work is to analyze the impact of crossover on the DE behavior all numerical tests were made using the classical DE/rand/1/* variant.

2.2. Crossover variants

In Evolutionary Algorithms the crossover operator usually combines features from different parents. In the case of DE algorithms, since the mutation operator is already based on a recombination of individuals, the role of crossover is somewhat different. It just allows the construction of an offspring (called trial vector), z_i, by mixing components of the current element, x_i, and of that generated by mutation, y_i. There are two main crossover variants for DE: binomial and exponential.

The binomial crossover constructs the trial vector by taking, in a random manner, elements either from the mutant vector or from the current element, as is described in eq. (2).

$$z_i^j = \begin{cases} y_i^j & \text{if } U_j < CR \text{ or } j = k \\ x_i^j & \text{otherwise} \end{cases}, j = \overline{1,n} \tag{2}$$

In eq. (2) U_j denotes a random value generated for each j in accordance with a uniform distribution over $[0,1]$, $CR \in [0,1]$ is the crossover rate and k is a

randomly selected index from $\{1, \ldots, n\}$. The use of k ensures that at least one component is taken from the mutant vector. The name of this crossover type comes from the property that the number of components taken from the mutant vector has a binomial distribution [17]. This binomial crossover is very similar to the so-called uniform crossover used in evolutionary algorithms.

The exponential crossover was designed to be similar to one point and two points crossover variants used in genetic algorithms. Thus the trial vector contains a sequence of consecutive components (in a circular manner) taken from the mutant vector. The structure of the trial vector can be described as in eq. (3) where $k \in \{1, 2, \ldots, n\}$ is a random index, L is a random value in $\{1, 2, \ldots, n\}$ and $\langle j \rangle_n$ is j if $j \leq n$ and $j - n$ if $j > n$.

$$z_i^j = \begin{cases} y_i^j & \text{if } j \in \{k, \langle k+1 \rangle_n, \ldots \langle k+L-1 \rangle_n\} \\ x_i^j & \text{otherwise} \end{cases}, j = \overline{1, n} \tag{3}$$

In [17] is mentioned that $Prob(L = h) = (1 - CR)CR^{h-1}$ which corresponds to the geometric distribution, the discrete counterpart of the continuous exponential distribution. This motivates the name of this crossover type. Exponential crossover was used in the first version of DE algorithm [20] but currently most of DE implementations use the binomial crossover.

The main difference between binomial and exponential crossover is the fact that while in the binomial case the components inherited from the mutant vector are arbitrarily selected in the case of exponential crossover they form one or two compact subsequences. The impact of this difference on the effectiveness of differential evolution is not yet fully understood. The choice of a crossover variant is difficult as long as there are no results establishing the superiority of one variant for a given class of problems.

Both binomial and exponential crossover have as main disadvantage the fact that they are not rotationally invariant processes making differential evolution less effective for rotated functions. On the other hand completely eliminating the crossover leads to a poor behavior of DE for multimodal problems. In order to solve this problem, K. Price proposed to replace the crossover operator with another way of recombining the population elements, similar to arithmetic recombination, obtaining the so-called DE/rand/either-or variant [17]. More recently, the same author proposed another rotationally invariant DE variant which has the particularity of eliminating the drift bias from its trial vector generating function [18]. The analysis conducted in this paper focuses on binomial, exponential and related crossover variants, thus the properties of crossover-free DE variants are not addressed.

2.3. Controlling the DE behavior

The behavior of the DE algorithms is influenced both by the type of mutation and crossover operators and by their control parameters: scale factor, F, and crossover rate, CR. The scale factor influences the size of perturbation applied to the base vector and has an important role in ensuring the population diversity. Small values of F can lead to premature convergence, i.e. the population loses

its diversity even in the absence of the selection pressure. In [30] is proved that if $F < \sqrt{(1-p_m/2)/m}$, with p_m the probability of a component in the trial element to be inherited from the mutant element, then the DE algorithm will prematurely converge. The range of values for F which proved to be effective in practice is $[0.3, 1]$ [17].

The behavior of both binomial and exponential crossover is influenced by the parameter CR. Since the parameter CR controls the number of components inherited from the mutant vector it can be interpreted as a mutation probability. Most parameter studies concludes that CR influences the convergence speed and the adequate value depends on the problem to be solved [7]. The first work which establishes a clear relationship between the values of CR and the properties of the objective function is [15] where is stated the fact that for separable functions small values of CR ($CR \leq 0.2$) are adequate while for nonseparable ones values larger than 0.9 should be used. The explanation is intuitive as long as for small values of CR the average number of components inherited from the mutant vector is small and, therefore, the evolutionary process takes place almost separately on each component, favoring the separable functions.

All these parameter studies were conducted in the case of binomial crossover and a natural question is if their conclusions remain valid in the case of exponential crossover. In [17] is mentioned that "*the average number of parameters mutated for a given CR depends on the crossover model (e.g. exponential or binomial) but in each a low CR corresponds to a low mutation rate*". However in the current DE literature there are no results concerning a quantitative relationship between the type of crossover and the mutation probability and/or the average number of mutated components. One of the main aims of this paper is to try to fill this gap by providing quantitative relationships between the mutation probability and CR for both types of crossover.

In order to simplify the design of DE algorithms several adaptive and self-adaptive variants have been proposed in the last decade. The adaptive variants, e.g. FADE [10], change the values of F and CR according to the behavior of the algorithm. In FADE some fuzzy rules are used to establish new values of the parameters based on the changes in the population. The self-adaptive variants assign a pair of parameters to each element in the population. The most known self-adaptive variants are SaDE [19] and jDE [3]. In both cases the individual parameters are changed during the evolution by using a probability distribution, normal distribution in the case of SaDE and uniform distribution in the case of jDE. The self-adaptive variants were designed for DE with binomial crossover thus a natural question is if they lead to a similar effect when they are applied in the case of exponential crossover.

The adaptation process can involve, besides the parameters F and CR, also the population size, as in [24], or even the strategy, as in SaDE or in the competitive variant proposed in [26] (where both the parameters and the crossover type are randomly chosen at each generation based on probabilities according to the success rate of candidates). The results presented in [26] illustrate the fact that by combining binomial and exponential crossover one can obtain a more robust DE.

3. Properties of binomial and exponential crossover

Several authors remarked the fact that the adequate value for CR depends on the crossover type [8],[11],[17]. However the current knowledge concerning this issue is limited to the distribution of the random variable L, denoting the number of mutated components. Thus the probabilities that the trial element inherit exactly h components from the mutant vector derived in [17] are: $P(L = h) = C_{n-1}^{h-1} CR^{h-1}(1 - CR)^{n-h}$, in the case of binomial crossover and $P(L = h) = (1 - CR)CR^{h-1}$ for exponential crossover.

Both in binomial and exponential crossover the parameter CR controls the number of components taken from the mutant vector influencing the probability that a component is mutated. Such a probability is usually called mutation probability (p_m). It is expected that the impact of each crossover type is influenced by p_m. Therefore, in the following we shall analyze the relationship between CR and p_m for binomial, exponential and other related crossover variants.

3.1. Binomial crossover

The way the trial element is constructed from the mutant and the current element of the population is illustrated in Algorithm 2. In the description of the algorithm, *irand* denotes a generator of random values uniformly distributed on a finite set, while $rand_j$ simulates a uniformly distributed random variable on a continuous domain for each component j. The condition "$rand_j(0,1) < CR$ or $j = k$" of the **if** statement in Algorithm 2 ensures the fact that at least one component is taken from the mutant vector.

Algorithm 2 Binomial crossover

crossoverBin (x, y)

1: $k \leftarrow irand(\{1, \ldots, n\})$
2: **for** $j = \overline{1,n}$ **do**
3: **if** $rand_j(0,1) \leq CR$ **or** $j = k$ **then**
4: $z^j \leftarrow y^j$
5: **else**
6: $z^j \leftarrow x^j$
7: **end if**
8: **end for**
9: **return** z

The implementation of the binomial crossover is based on the simulation of n independent Bernoulli trials, the result of each trial being used in selecting a component of the offspring from the mutant vector or from the current element. If the constraint of having at least one mutated component is applied, the successful event in each Bernoulli trial is the union of two independent events, one of probability CR (corresponding to the event "$rand_j(0,1) < CR$") and one of probability $1/n$ (corresponding to the event "$j = irand(\{1, \ldots, n\})$").

Since, for two independent events A and B the probability of their union is $Prob(A \cup B) = Prob(A) + Prob(B) - Prob(A)Prob(B)$, it follows that the probability that a component is mutated is $p_m = CR(1 - 1/n) + 1/n$.

The number, L, of components selected from the mutant vector is a random variable with values in $\{1, \ldots, n\}$ having the property that $L = L' + 1$ where L' has a binomial distribution of parameters $n - 1$ and CR. Thus $Prob(L = h) = Prob(L' = h - 1) = C_{n-1}^{h-1} CR^{h-1}(1 - CR)^{n-h}$ which is the distribution derived also in [17]. The average number of mutated components is $E(L) = E(L') + 1 = (n - 1)CR + 1$. These properties of the binomial crossover can be summarized as:

Proposition 1. *In the case of binomial crossover the mutation probability, p_m, and the number of mutated components, L, satisfy the following properties:*

$$p_m = CR(1 - 1/n) + 1/n \tag{4}$$

$$Prob(L = h) = C_{n-1}^{h-1} CR^{h-1}(1 - CR)^{n-h}, \quad h \in \{1, \ldots, n\} \tag{5}$$

$$E(L) = np_m = (n - 1)CR + 1 \tag{6}$$

3.2. Classical exponential crossover

The exponential crossover has been proposed in the first version of Differential Evolution [20]. It is similar with the two-point crossover where the first cut point is randomly selected from $\{1, \ldots, n\}$ and the second point is determined such that L consecutive components (counted in a circular manner) are taken from the mutant vector. In their original paper [20], Storn and Price suggested to choose $L \in \{1, \ldots, n\}$ such that $Prob(L = h) = CR^h$. It is easy to check that this is not a probability distribution on $\{1, \ldots, n\}$ but just a relationship which suggest that the probability of mutating h components increases with the parameter CR and decreases with the value of h. Such a behavior can be obtained by different implementations. The most frequent implementation is that described in Algorithm 3 where $\langle j + 1 \rangle_n$ is just $j + 1$ if $j < n$ and is 1 when $j = n$.

Algorithm 3 Exponential crossover

crossoverExp (x, y)

1: $z \leftarrow x$; $k \leftarrow irand(\{1, \ldots, n\})$; $j \leftarrow k$; $L \leftarrow 0$
2: **repeat**
3: $\quad z^j \leftarrow y^j$; $j \leftarrow \langle j + 1 \rangle_n$; $L \leftarrow L + 1$
4: **until** $rand_j(0, 1) \geq CR$ or $L = n$
5: **return** z

If the stopping condition of the **repeat** loop in the Algorithm 3 would be just $rand_j(0, 1) \geq CR$ then L would take values according to the geometric distribution (which is the discrete counterpart of the continuous exponential distribution) on $\{1, 2, \ldots\}$ corresponding to the parameter $1 - CR$ (CR being interpreted as the success probability). In such a situation the probability that the

number of mutated components is h would be $Prob(L = h) = (1 - CR)CR^{h-1}$, which is the probability derived in [17]. However in the exponential crossover the number of mutated components is bounded by n, thus we are dealing with a truncated geometric distribution.

Taking into consideration the fact that the **repeat** loop stops whenever L becomes n it follows that the probability distribution of L is given by:

$$Prob(L = h) = \begin{cases} (1 - CR)CR^{h-1} & \text{if } 1 \leq h < n \\ CR^{n-1} & \text{if } h = n \end{cases} \tag{7}$$

Using eq. (7), the average of the random variable L can be computed as follows:

$$\begin{aligned} E(L) &= (1 - CR) \sum_{h=1}^{n-1} hCR^{h-1} + nCR^{n-1} \\ &= (1 - CR)(1 + 2CR + \ldots + (n-1)CR^{n-2}) + nCR^{n-1} \\ &= (1 - CR) \left(\frac{1 - CR^{n-1}}{1 - CR} + CR \frac{1 - CR^{n-2}}{1 - CR} + \ldots + CR^{n-2} \frac{1 - CR}{1 - CR} \right) \\ &\quad + nCR^{n-1} \\ &= 1 + CR + CR^2 + \ldots + CR^{n-2} - (n-1)CR^{n-1} + nCR^{n-1} \\ &= \frac{1 - CR^n}{1 - CR} \end{aligned}$$

It remains now to find the value of p_m in the case of exponential crossover. There are two random variables simulated in the implementation of exponential crossover: the index, k, of the first mutated component and the number of mutated components, L. An arbitrary component, j, will be mutated if $d(j,k) < L$, where $d(j,k) = j - k$ if $j \geq k$ and $d(j,k) = n + j - k$ if $j < k$. Since k can take any value from $\{1, \ldots, n\}$ with probability $1/n$, the probability that an arbitrary component, j, is replaced with a component from the mutant vector is:

$$Prob(z^j = y^j) = \frac{1}{n} \sum_{k=1}^{n} Prob(d(j,k) < L) = \frac{1}{n} \sum_{d=0}^{n-1} Prob(L > d)$$

Since

$$Prob(L > d) = \sum_{h=d+1}^{n} Prob(L = h) = \sum_{h=d+1}^{n-1} (1 - CR)CR^{h-1} + CR^{n-1} = CR^d$$

it follows that

$$Prob(z^j = y^j) = \frac{1}{n} \sum_{d=0}^{n-1} CR^d = \frac{1 - CR^n}{n(1 - CR)}$$

Thus the properties satisfied by the exponential crossover can be summarized as follows.

Proposition 2. *In the case of exponential crossover the mutation probability, p_m, and the number of mutated components, L, satisfy the following properties:*

$$p_m = \frac{1 - CR^n}{n(1 - CR)} \tag{8}$$

$$Prob(L = h) = \begin{cases} (1 - CR)CR^{h-1} & \text{if } 1 \leq h < n \\ CR^{n-1} & \text{if } h = n \end{cases} \tag{9}$$

$$E(L) = \frac{1 - CR^n}{1 - CR} \tag{10}$$

Thus unlike the case of binomial crossover where the mutation probability depends linearly on CR, in the case of exponential crossover there is a nonlinear dependence.

3.3. Related crossover variants

Let us go back to the particularities of the crossover presented in the first DE variant [20]. The main properties of these crossover operator are: (i) the number of mutated components, L, is a random variable with the property that $Prob(L = h)$ is proportional to CR^h; (ii) the trial vector contains L consecutive mutated components starting from a random position. Starting from these particularities of the classical exponential crossover other implementation variants can be developed.

A first variant is based on the idea of simulating a random variable L taking values in $\{1, \ldots, n\}$ based on the probability distribution $Prob(L = h) = c \cdot CR^h$ where c is a normalization constant. It is easy to find that $c = (1-CR)/(CR(1-CR^n))$ thus the corresponding distribution probability is:

$$Prob(L = h) = \frac{1 - CR}{1 - CR^n} CR^{h-1} \tag{11}$$

The implementation of such a variant involves three main steps: (i) generate L according to the probability distribution (11); (ii) randomly select a starting position from $\{1, \ldots, n\}$; (iii) mutate L positions starting with the selected one. This crossover strategy is described in Algorithm 4.

Based on eq. (11) one can easily compute the average number of mutated components:

$$\begin{aligned} E(L) &= \frac{1 - CR}{1 - CR^n} \sum_{h=1}^{n} hCR^{h-1} \\ &= \frac{1 - CR}{1 - CR^n}(1 + 2CR + 3CR^2 + \ldots + (n-1)CR^{n-2} + nCR^{n-1}) \\ &= \frac{1 - CR}{1 - CR^n}((1 + \ldots + CR^{n-1}) + CR(1 + \ldots + CR^{n-2}) + \ldots \\ &\quad \ldots + CR^{n-2}(1 + CR) + CR^{n-1}) \end{aligned}$$

Algorithm 4 Variant of exponential crossover based on the simulation of the random variable L

crossoverVariantA (x, y)

1: $u \leftarrow rand(0,1)$; $p \leftarrow (1-CR)/(1-CR^n)$;$s \leftarrow p$; $L \leftarrow 1$;
2: **while** $u < s$ **do**
3: $\quad L \leftarrow L+1$; $p \leftarrow p \cdot CR$; $s \leftarrow s+p$
4: **end while**
5: $z \leftarrow x$; $k \leftarrow irand(\{1,\ldots,n\})$; $h \leftarrow 1$; $j \leftarrow k$;
6: **repeat**
7: $\quad z^j \leftarrow y^j$; $j \leftarrow \langle j+1 \rangle_n$; $h \leftarrow h+1$
8: **until** $h = L$
9: **return** z

$$
\begin{aligned}
&= \frac{1-CR}{1-CR^n}\left(\frac{1-CR^n}{1-CR} + CR\frac{1-CR^{n-1}}{1-CR} + \ldots + CR^{n-1}\frac{1-CR}{1-CR}\right)\\
&= \frac{1}{1-CR^n}(1 + CR + \ldots CR^{n-1} - nCR^n)\\
&= \frac{1}{1-CR^n}\left(\frac{1-CR^n}{1-CR} - nCR^n\right) = \frac{1}{1-CR} - \frac{nCR^n}{1-CR^n}.
\end{aligned}
$$

As in the case of classical exponential crossover the mutation probability satisfies $p_m = \frac{1}{n}\sum_{d=0}^{n-1} Prob(L > d)$. Using the probability distribution (11) one obtains:

$$
\begin{aligned}
Prob(L > d) &= \sum_{h=d+1}^{n} \frac{1-CR}{1-CR^n} CR^{h-1} = \frac{1-CR}{1-CR^n}\sum_{h=d}^{n-1} CR^h\\
&= \frac{1-CR}{1-CR^n}\left(\sum_{h=0}^{n-1} CR^h - \sum_{h=0}^{d-1} CR^h\right)\\
&= \frac{1-CR}{1-CR^n}\left(\frac{1-CR^n}{1-CR} - \frac{1-CR^d}{1-CR}\right) = \frac{CR^d - CR^n}{1-CR^n}.
\end{aligned}
$$

Thus

$$p_m = \frac{1}{n}\sum_{d=0}^{n-1}\frac{CR^d - CR^n}{1-CR^n} = \frac{1}{n(1-CR)} - \frac{CR^n}{1-CR^n}$$

and the properties of this variant of the exponential crossover can be summarized as follows:

Proposition 3. *In the case of exponential crossover variant implemented as in Algorithm 4 the mutation probability, p_m, and the number of mutated components, L, satisfy the following properties:*

$$p_m = \frac{1}{n(1-CR)} - \frac{CR^n}{1-CR^n} \tag{12}$$

$$Prob(L = h) = \frac{1 - CR}{1 - CR^n} CR^{h-1} \tag{13}$$

$$E(L) = \frac{1}{1 - CR} - \frac{nCR^n}{1 - CR^n} \tag{14}$$

An undesirable property of this variant is the fact that when CR goes to 1 the corresponding mutation probability goes to $(n + 1)/(2n)$ not to 1 as we would desire. In order to correct this we can use instead of the value L generated in Algorithm 4 the adjusted value: $L' = \min\{n, L + \lfloor L \cdot CR(n - 1)/(n + 1) \rfloor\}$ where $\lfloor \cdot \rfloor$ denotes the lower integer part. The corresponding adjusted mutation probability is obtained by multiplying the expression of p_m in eq. 12 by $CR(n - 1)/(n + 1) + 1$. In all following references of Algorithm 4 it is supposed that L and p_m are adjusted as is mentioned above.

Both the classical exponential crossover and the above described variant differ from the binomial crossover not only by the fact that they take consecutive components from the mutant vector but also by a different number of mutated components for the same value of the crossover rate, CR. In order to analyze just the influence of taking consecutive components from the mutant vector we introduce a simpler, less stochastic, strategy described in Algorithm 5.

Algorithm 5 Crossover variant based on taking $L = \lfloor CR(n - 1) + 1 \rfloor$ consecutive elements from the mutant vector

crossoverVariantB (x, y)

1: $z \leftarrow x$; $k \leftarrow irand(\{1, \ldots, n\})$; $j \leftarrow k$;
2: $L \leftarrow [CR(n - 1) + 1]$; $h \leftarrow 1$;
3: **repeat**
4: $\quad z^j \leftarrow y^j$; $j \leftarrow \langle j + 1 \rangle_n$; $h \leftarrow h + 1$
5: **until** $h = L$
6: **return** z

This variant does not use a random, but a fixed number of mutated components ($L = \lfloor CR(n - 1) + 1 \rfloor$) and has similar properties with the binomial crossover with respect to the mutation probability which is $p_m = \lfloor CR(n - 1) + 1 \rfloor / n$.

3.4. Implementation issues

In the implementation of random algorithms an important issue is that of the number of *rand* functions calls. The four crossover variants described above differ not only by the probabilities of the number of components taken from the mutant vector but also by the number of *rand* functions calls. Thus the binomial variant needs $n+1$ calls while the number of calls in the case of classical exponential crossover is given by the random variable L whose mean value is $np_m \leq n$ plus one call of *irand* which generates the index of the first replaced component. On the other hand the variant described in Algorithm 4 uses just two calls disregarding the value of n (one is used in the simulation of L and one to select the first mutated component). The classical implementation of

the exponential crossover can be easily transformed such that only two random calls are enough (as is illustrated in Algorithm 6). The difference between the Algorithms 4 and 6) is related only to the number of components taken from the mutant vector.

Algorithm 6 Variant of exponential crossover based on the simulation of the random variable L

crossoverExpModified (x, y)

```
1:  u ← rand(0,1); p ← 1 − CR;s ← p; L ← 1;
2:  while u < s do
3:     L ← L + 1;
4:     if L < n then
5:        p ← p · CR;s ← s + p
6:     else
7:        s ← 1
8:     end if
9:  end while
10: z ← x; k ← irand({1,...,n}); h ← 1; j ← k;
11: repeat
12:    z^j ← y^j; j ← ⟨j + 1⟩_n; h ← h + 1
13: until h = L
14: return z
```

The variants based on a deterministic value for L (as in Algorithm 5), use just one call of the *irand* function. For high-dimensional problems implementation versions which use a small number of *rand* functions calls could lead to important time savings with respect to the classical binomial and exponential crossover implementations. However the possibility of replacing the classical crossover implementations with variants as those described in Algorithms 4 and 5 is still to be investigated with respect to their impact on the behavior of DE.

3.5. Crossover rate and mutation probability

For all crossover variants described in the previous subsections the crossover rate influences the number of components taken from the mutant vector. However, as Propositions 1-3 state the relationship between the crossover rate, CR, and the mutation probability, p_m is different for different crossover variants.

More specifically, the dependence between p_m and CR is linear in the case of binomial crossover and nonlinear in the case of exponential crossover variants. Figure 1 illustrates the fact that for the same value of $CR \in [0, 1]$ the mutation probability is smaller in the case of exponential crossover than in the case of binomial one, the difference being more significant if n is larger. On the other hand the mutation probability corresponding to the variant described in Algorithm 4 is slightly larger than that corresponding to classical exponential crossover. However this difference becomes smaller as the problem size increases.

The impact of the problem size on the values of p_m is also illustrated by the values in Table 1 where corresponding values for CR and p_m are tabulated for

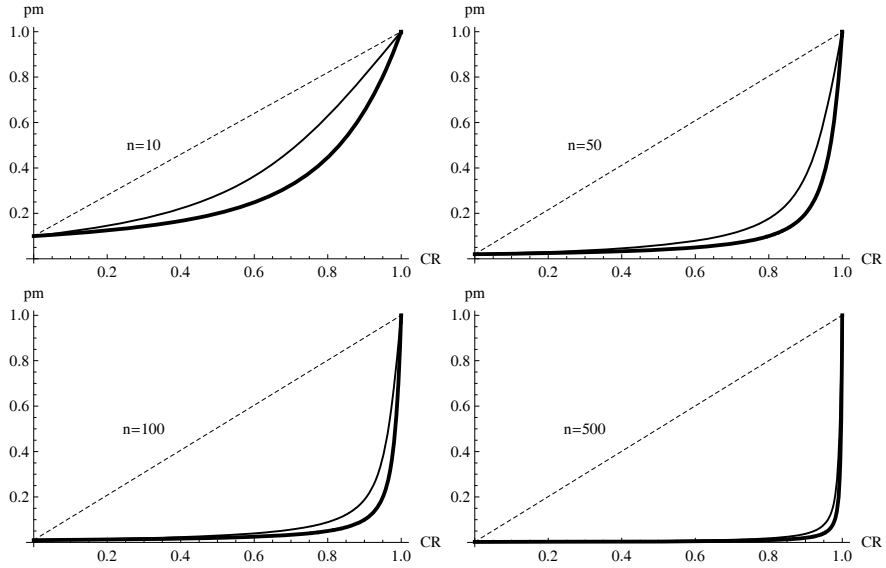

Figure 1: Influence of CR on the mutation probability in the case of binomial crossover (dashed line), classical exponential crossover (thick continuous line) and crossover described in Algorithm 4 (normal continuous line)

CR	$p_m(n=50)$			$p_m(n=100)$			$p_m(n=500)$		
	Bin.	Exp.	Alg. 4	Bin.	Exp.	Alg. 4	Bin.	Exp.	Alg. 4
0	0.02	0.02	0.02	0.01	0.01	0.01	0.002	0.002	0.002
0.1	0.118	0.022	0.024	0.109	0.011	0.012	0.102	0.002	0.002
0.2	0.216	0.025	0.029	0.208	0.012	0.014	0.202	0.003	0.003
0.3	0.314	0.028	0.036	0.307	0.014	0.018	0.301	0.003	0.004
0.4	0.412	0.033	0.046	0.406	0.016	0.023	0.401	0.003	0.005
0.5	0.51	0.04	0.059	0.505	0.02	0.029	0.501	0.004	0.006
0.6	0.608	0.05	0.078	0.604	0.025	0.039	0.601	0.005	0.008
0.7	0.706	0.066	0.111	0.703	0.033	0.056	0.701	0.007	0.011
0.8	0.804	0.099	0.176	0.802	0.05	0.089	0.8	0.01	0.017
0.9	0.902	0.198	0.363	0.901	0.099	0.188	0.9	0.02	0.037
0.92	0.921	0.246	0.441	0.920	0.124	0.237	0.92	0.025	0.047
0.95	0.951	0.369	0.605	0.950	0.198	0.374	0.95	0.04	0.077
0.97	0.970	0.521	0.749	0.970	0.317	0.552	0.97	0.067	0.131
0.99	0.990	0.789	0.913	0.990	0.633	0.832	0.99	0.198	0.384
1	1	1	1	1	1	1	1	1	1

Table 1: Correspondence between CR and the mutation probability, p_m, for binomial and exponential crossover variants.

the crossover variants given in Algorithms 2, 3 and 4 and three problem sizes ($n = 50$, $n = 100$, $n = 500$).

For large values of n the nonlinear dependence between p_m and CR is characterized by the presence of two regimes: a first one where large changes in the CR values lead to small changes in p_m and a second one where small changes in CR lead to large changes in p_m. For each value of n one can identify a critical value of CR which separates these two regimes. As n is larger this value becomes closer to 1. For $n = 100$ such a critical value for CR is near 0.9 (when p_m approaches 0.1) while for $n = 500$ the critical value is around 0.98. These properties suggest that for high-dimensional problems for most values of CR the exponential crossover leads to very small mutation probabilities while the range of CR values to which p_m is sensitive is very narrow. Thus, in the case of problems asking for high mutation probability, exponential crossover would not work except for values of CR in a small range near 1. In fact for exponential crossover variants when n goes to infinity p_m tends toward 0 if $CR < 1$ and to 1 only if $CR = 1$.

4. Experimental parameter study

As the results in the previous sections suggest there are two main differences between binomial and exponential crossover: (i) the same value of the crossover rate lead to different values of the mutation probability; (ii) in the exponential crossover the mutated components are consecutive while in the case of binomial crossover the components to be mutated are randomly selected. The main aim we followed in this experimental study was to see which of these two differences has a larger impact on the DE behavior.

4.1. Previous work

Most of the experimental studies on the role of crossover in differential evolution focus on the impact of the crossover rate on the algorithm behavior. First results state that the most important DE control parameter is the scaling factor, F, while the crossover rate, CR, is useful only for fine tuning. Starting from the suggestions of Storn and Price many of the first DE implementations used values of CR near 0.9. The parameter study conducted in [7] illustrated the fact that for some functions (e.g. Rastrigin) values of CR less than 0.9 can lead to a better behavior. Later, in [15] is stated an important relationship between the properties of the objective function and the adequate values of CR: in the case of separable functions small values of CR are adequate while for nonseparable functions the best choice is to use larger values of CR.

Most of parameter studies use the binomial crossover variant. The studies of exponential crossover are significantly fewer. In [28] the variant DE/rand/1/exp is compared with particle swarm optimization and a simple evolutionary algorithm for a large set of test functions. Despite the fact that the parameters of DE have not been tuned (the used values were $F = 0.5$, $CR = 0.9$ and the population size was 100) the DE/rand/1/exp variant proved to be

very competitive. On the other hand the results of a comparative study presented in [11] suggest that the binomial crossover is better than the exponential one. The study is thoroughly conducted by tuning the crossover rate for each pair (DE variant,problem). However the set of analyzed CR values was $\{0, 0.1, 0.2, ..., 0.8, 0.9, 1\}$. If in the case of the binomial crossover these CR values lead to values of p_m which are uniformly distributed in $[1/n, 1]$, in the case of exponential crossover this is not true (for instance in the case when $n = 30$ the corresponding values of p_m are only from $[0.03, 0.31] \cup \{1\}$). Thus the worse behavior of exponential crossover is not necessarily caused by its particularity of mutating consecutive elements but it can be caused by just an inappropriate choice of CR.

For a fair comparative study between binomial and exponential crossover one should use either a set of CR values which leads to uniformly distributed values of p_m or the variant proposed in Algorithm 5, which is similar with the binomial crossover with respect to the relationship between CR and p_m but mutates consecutive components as in exponential crossover.

4.2. Test functions and parameter setup

The comparative study we conducted aimed to analyze the sensitivity of DE to crossover rate and mutation probability values. It also aimed to comparatively analyze the impact of mutating a sequence of consecutive components and of arbitrary components.

In order to conduct this analysis we selected two multi-modal functions: a separable (Rastrigin function) and a non-separable one (Griewank function). These functions are defined as follows:

Rastrigin:

$$f : [-5.12, 5.12]^n \to \mathbb{R},\ f(x_1, \ldots, x_n) = 10n + \sum_{j=1}^{n} (x_j^2 - 10\cos(2\pi x_j)))$$

Griewank:

$$f : [-600, 600]^n \to \mathbb{R},\ f(x_1, \ldots, x_n) = \frac{1}{4000} \sum_{j=1}^{n} x_j^2 - \prod_{j=1}^{n} \cos(x_j/\sqrt{j}) + 1$$

For each function two variants were used: a shifted (for $n = 100$) and a rotated one (for $n = 50$). For the shifted variant the minimum was randomly generated in the decision variables domain while the rotated variants are taken from the CEC 2005 test suite [22]. In all cases the optimal value is 0 ($f(x^*) = 0$).

The comparative study involved the four crossover variants presented in Section 3. In all cases the scaling factor is $F = 0.5$ and the population size is set to the problem size ($m = n$). The maximum number of function evaluations is set to 500000 and the success threshold to $\epsilon = 10^{-8}$. When the best element in the population has a value less than ϵ then we consider that the algorithm succeeded in approximating the optimum. The success ratio (SR) is defined as the ratio between the number of successful runs and the total number of runs.

4.3. Comparative results

Tables 2-7 present values averaged over 30 independent runs for the best value (f^*), the number of function evaluations (nfe) and the mutation probability (p_m). At each generation the mutation probability is estimated as the

ratio of the number of mutated components and the number of components (n). The value $\langle p_m \rangle$ is averaged over all generations and all runs.

Results in Tables 2 and 3 illustrate the importance of mutation probability on the success of DE. For the separable Rastrigin function the best results are obtained when at each step a small number of components are mutated. As we can see if $p_m < 0.03$ all algorithms succeeded in finding the optimum, by using almost the same number of generations. In the case of binomial crossover and of the variant given by Algorithm 5 such values of p_m are obtained if CR is also less than 0.03. On the other hand in the case of exponential crossover variants given by Algorithms 3 and 4 (with p_m multiplied by $CR(n-1)/(n+1)+1$ in order to extend its range to $[0,1)$) the range of CR values for which $p_m < 0.03$ is significantly larger (approximately $[0, 0.6]$). Since the success is obtained only for a small number of mutated components (at most 3 in average) a significant difference between the case when arbitrary placed components are mutated and the case when a similar number of consecutive components are mutated cannot be identified (see Table 2). On the other hand, the results in Tables 2 and 3 clearly illustrates the importance of p_m over CR and implicitly the fact that the difference between binomial and exponential crossover is mainly determined by the different relationship between the mutation probability and the crossover rate.

CR	Binomial crossover				Algorithm 5			
	$\langle f^* \rangle$	$\langle nfe \rangle$	SR	$\langle p_m \rangle$	$\langle f^* \rangle$	$\langle nfe \rangle$	SR	$\langle p_m \rangle$
0	$< \epsilon$	361676	30/30	0.01	$< \epsilon$	362463	30/30	0.01
0.01	$< \epsilon$	406603	30/30	0.019	$< \epsilon$	362453	30/30	0.01
0.03	$9 \cdot 10^{-6}$	500000	0/30	0.039	10^{-6}	500000	0/30	0.03
0.1	291.44	500000	0/30	0.109	388.34	500000	0/30	0.1
0.3	634.64	500000	0/30	0.306	684.31	500000	0/30	0.3
0.5	799.69	500000	0/30	0.505	769.03	500000	0/30	0.5
0.7	843.63	500000	0/30	0.703	793.41	500000	0/30	0.7
0.9	550.93	500000	0/30	0.9	438.37	500000	0/30	0.9
0.95	71.30	500000	0/30	0.95	71.11	500000	0/30	0.95
0.99	72.19	500000	0/30	0.99	74.05	500000	0/30	0.99

Table 2: Shifted Rastrigin ($n = 100$, $m = 100$, $F = 0.5$, $\epsilon = 10^{-8}$).

In the case of shifted Griewank function (Tables 4 and 5) there is a small difference between the DE behavior in the case of binomial and exponential crossover variants. The binomial crossover leads to best results for either small or large values of p_m ($p_m \in [0, 0.3] \cup [0.8, 0.95]$) which correspond with a similar range of CR values. The exponential variants are successful for $p_m < 0.8$ which correspond to values of CR less than 0.99. Thus as in the case of shifted Rastrigin function the DE with exponential crossover (Algorithms 3 and 4) has a larger range of CR values for which it is successful.

The rotated versions are more difficult to solve. DE/rand/1 strategies are not able to find the global optimum of the rotated Rastrigin function within the specified number of function evaluations and the success ratio in the case

CR	Exponential crossover				Algorithm 4			
	$\langle f^* \rangle$	$\langle nfe \rangle$	SR	$\langle p_m \rangle$	$\langle f^* \rangle$	$\langle nfe \rangle$	SR	$\langle p_m \rangle$
0	$< \epsilon$	361763	30/30	0.01	$< \epsilon$	362206	30/30	0.01
0.1	$< \epsilon$	380616	30/30	0.011	$< \epsilon$	365220	30/30	0.0111
0.3	$< \epsilon$	380616	30/30	0.014	$< \epsilon$	381010	30/30	0.014
0.5	$< \epsilon$	402756	30/30	0.02	$< \epsilon$	410766	30/30	0.023
0.7	$< \epsilon$	454013	30/30	0.033	$3 \cdot 10^{-8}$	500000	0/30	0.051
0.9	24.27	500000	0/30	0.099	100.53	500000	0/30	0.182
0.95	145.26	500000	0/30	0.198	231.57	500000	0/30	0.369
0.99	436.75	500000	0/30	0.634	456.18	500000	0/30	0.827

Table 3: Shifted Rastrigin ($n = 100$, $m = 100$, $F = 0.5$, $\epsilon = 10^{-8}$).

CR	Binomial crossover				Algorithm 5			
	$\langle f^* \rangle$	$\langle nfe \rangle$	SR	$\langle p_m \rangle$	$\langle f^* \rangle$	$\langle nfe \rangle$	SR	$\langle p_m \rangle$
0	$< \epsilon$	380416	30/30	0.01	10^{-8}	419923	28/30	0.01
0.1	$< \epsilon$	280086	30/30	0.108	$< \epsilon$	295083	30/30	0.1
0.3	$< \epsilon$	415083	30/30	0.306	$< \epsilon$	410300	30/30	0.3
0.4	10^{-7}	500000	0/30	0.406	$< \epsilon$	438850	30/30	0.4
0.5	10^{-4}	500000	0/30	0.504	$< \epsilon$	463726	30/30	0.5
0.6	0.0016	500000	0/30	0.603	$< \epsilon$	466200	30/30	0.6
0.7	10^{-5}	500000	0/30	0.702	$< \epsilon$	393703	30/30	0.7
0.8	$< \epsilon$	456566	30/30	0.802	$3 \cdot 10^{-4}$	395626	29/30	0.8
0.9	$< \epsilon$	314824	30/30	0.901	0.0012	350616	26/30	0.9
0.95	$< \epsilon$	318521	30/30	0.95	0.0069	386793	22/30	0.95
0.99	0.0286	500000	0/30	0.99	0.0518	500000	0/30	0.99

Table 4: Shifted Griewank ($n = 100$, $m = 100$, $F = 0.5$, $\epsilon = 10^{-8}$).

CR	Exponential crossover				Algorithm 4			
	$\langle f^* \rangle$	$\langle nfe \rangle$	SR	$\langle p_m \rangle$	$\langle f^* \rangle$	$\langle nfe \rangle$	SR	$\langle p_m \rangle$
0.1	$< \epsilon$	360933	30/30	0.011	$< \epsilon$	358393	30/30	0.011
0.3	$< \epsilon$	341956	30/30	0.012	$< \epsilon$	346550	30/30	0.014
0.5	$< \epsilon$	336633	30/30	0.019	$< \epsilon$	329446	30/30	0.023
0.7	$< \epsilon$	315196	30/30	0.033	$< \epsilon$	315146	30/30	0.051
0.9	$< \epsilon$	316530	30/30	0.1	$< \epsilon$	359050	30/30	0.182
0.95	$< \epsilon$	359996	30/30	0.198	$< \epsilon$	438693	30/30	0.369
0.99	$< \epsilon$	408796	30/30	0.634	$< \epsilon$	392753	30/30	0.827
0.994	$7 \cdot 10^{-4}$	399250	27/30	0.752	10^{-4}	355948	29/30	0.89
0.997	0.0025	374210	24/30	0.865	10^{-4}	341916	29/30	0.943
0.999	0.017	500000	0/30	0.95	10^{-4}	324357	28/30	0.977

Table 5: Shifted Griewank ($n = 100$, $m = 100$, $F = 0.5$, $\epsilon = 10^{-8}$.

of rotated Griewank (Tables 6-7) is less than that corresponding to the shifted version of the function (Tables 4-5). Because of the nonseparable character of the function the success ratio is higher for higher values of p_m. Thus, unlike in the case of separable Rastrigin function, the range of appropriate CR values is smaller in the case of exponential crossover (Table 7) than in the case of binomial crossover. On the other hand results in Table 6 suggest that, in this case, the mutation of consecutive components (as in Algorithm 5) lead to slightly better results than the mutation of arbitrary placed components (as in binomial crossover).

CR	Binomial crossover				Algorithm 5			
	$\langle f^* \rangle$	$\langle nfe \rangle$	SR	$\langle p_m \rangle$	$\langle f^* \rangle$	$\langle nfe \rangle$	SR	$\langle p_m \rangle$
0	0.01	500000	0/30	0.02	0.007	500000	0/30	0.02
0.1	0.011	500000	0/30	0.11	0.0075	500000	0/30	0.099
0.2	0.002	500000	0/30	0.215	0.004	496220	3/30	0.2
0.3	$2 \cdot 10^{-7}$	497065	6/30	0.31	$5 \cdot 10^{-4}$	453552	22/30	0.3
0.5	10^{-8}	483345	22/30	0.509	$4 \cdot 10^{-4}$	359978	27/30	0.5
0.7	$9 \cdot 10^{-4}$	317970	27/30	0.705	0.0012	398508	26/30	0.7
0.9	0.0086	366858	17/30	0.901	0.0077	366955	20/30	0.9
0.95	0.01	492686	8/30	0.95	0.01	460185	14/30	0.95
0.99	2.61	500000	0/30	0.99	0.111	500000	0/30	0.99

Table 6: Rotated Griewank ($n = 50$, $m = 50$, $F = 0.5$, $\epsilon = 10^{-8}$).

CR	Exponential crossover				Algorithm 4			
	$\langle f^* \rangle$	$\langle nfe \rangle$	SR	$\langle p_m \rangle$	$\langle f^* \rangle$	$\langle nfe \rangle$	SR	$\langle p_m \rangle$
0.1	0.0054	500000	0/30	0.022	0.0074	500000	0/30	0.022
0.3	0.0053	500000	0/30	0.028	0.0058	500000	0/30	0.029
0.5	0.0052	500000	0/30	0.04	0.0055	500000	0/30	0.046
0.7	0.0082	500000	0/30	0.066	0.0082	500000	0/30	0.102
0.8	0.0094	500000	0/30	0.099	$3 \cdot 10^{-4}$	489055	16/30	0.166
0.9	0.001	478735	25/30	0.198	$7 \cdot 10^{-4}$	346293	28/30	0.352
0.95	$9 \cdot 10^{-4}$	329315	27/30	0.369	0.0031	322546	23/30	0.594
0.99	0.0087	458396	16/30	0.789	0.0096	414528	14/30	0.902
0.999	8.74	500000	0/30	0.975	0.0073	425000	15/30	0.98

Table 7: Rotated Griewank ($n = 50$, $m = 50$, $F = 0.5$, $\epsilon = 10^{-8}$).

5. Impact of the crossover type on the choice and adaptation of control parameters

The previous experiments were based on the same value of the scaling parameter F. Since the control parameters in DE are interrelated one would expect to have different appropriate values of F for the same value of CR when different types of crossover are used. The aim of this section is to analyze which is the impact of the crossover type on the choice of F and on the behavior of (self)adaptive strategies.

5.1. Control parameters and population diversity

Since DE is prone to premature convergence one first issue in choosing the control parameters of DE is to try to avoid such a situation. Starting from the ideas that premature convergence is related with loss of diversity and that the diversity is related with the population variance, in [30] is derived a theoretical relationship between the population variance, computed separately for each component, and the control parameters after and before applying the variation operators. More specifically, if $\mathrm{Var}(z)$ and $\mathrm{Var}(x)$ denotes the averaged variance of the trial and current populations respectively then the following relationship holds:

$$\mathrm{Var}(z) = (2p_m F^2 - \frac{2p_m}{m} + \frac{p_m^2}{m} + 1)\mathrm{Var}(x) \tag{15}$$

Based on this linear dependence between these two variances one can control the impact of the mutation and crossover steps on the variance modification by imposing that

$$2p_m F^2 - \frac{2p_m}{m} + \frac{p_m^2}{m} + 1 = c \tag{16}$$

where c should be 1 if we are interested to keep the same value of the variance or slightly larger than 1 in order to stimulate an increase of the diversity (for instance $c = 1.05$ means an increase of the population variance with 5% and $c = 1.1$ means an increase with 10%). The result in [30] was obtained only in the case of binomial crossover considering that $p_m = CR$. By replacing in eq. (16) the value of p_m with the expression corresponding to different crossover variants (as in Propositions 1-3), one obtains corresponding equations involving F,CR,m and n. By solving these equations with respect to F one can obtain lower bounds for F which allow avoiding premature convergence. The dependence of such lower bounds of F on the values of CR for two values of the constant c is illustrated in Figure 2. The differences between the value of F_{min} in the case of binomial and the exponential crossover variants suggest that for the same value of $CR \in (0, 1)$ the exponential crossover needs a larger value of F in order to induce the same effect on the population variance.

Attention should be also paid when using in combination with exponential crossover an adaptive or self-adaptive variant of DE which was initially designed for binomial crossover. For instance in the adaptive variant designed to avoid premature convergence [31] the adaptation rules for F and CR should be modified according to eq. (15) and to the relationships between p_m and CR.

5.2. On adaptation of control parameters

The adaptive and self-adaptive variants of DE are based on exploring the control parameters space using deterministic or random strategies. Since the parameters which are currently adapted are F and CR and the behavior of DE is in fact influenced by F and p_m it follows that in the case when exponential crossover is used the strategy used to explore the CR space should take into account the nonlinear relationship between p_m and CR. Let us analyze the simple case of a dynamic CR as it is used for instance in [12]: $CR(t + 1) =$

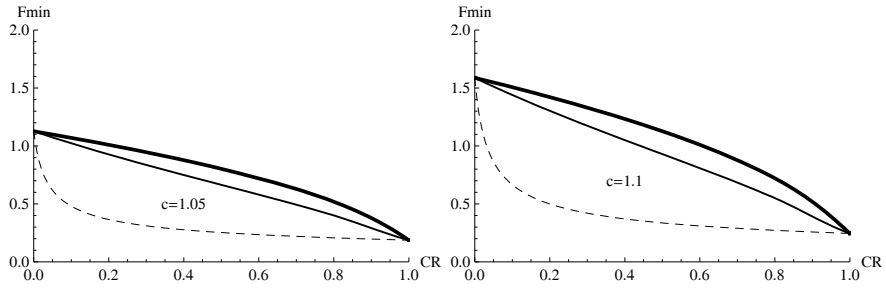

Figure 2: Lower bound for F vs. CR for binomial crossover (dashed line), exponential crossover (thick line) and crossover described in Algorithm 4 (normal line). Parameters: $m = 50$, $n = 50$

.

$0.95^t CR(t)$ (where t denotes the moment when the value of CR is changed). For different crossover variants the impact of these changes on the value of p_m is different. This is illustrated in Figure 3 where one can see that the decrease of p_m is quicker in the case of exponential crossover variants than in the case of binomial crossover. The difference becomes more significant when the problem size becomes larger. These remarks lead to the conclusion that such a decrease scheme of CR could be inappropriate in the case of exponential crossover since it leads to a too fast decrease of the mutation probability.

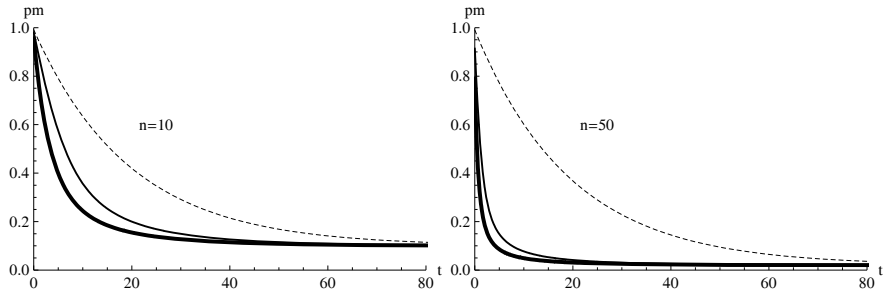

Figure 3: Evolution of p_m when CR is changed according to $CR(t+1) = 0.95^t CR(t)$. Variants: binomial crossover (dashed line), classical exponential crossover (thick continuous line) and the variant described in Algorithm 4 (normal continuous line). Parameters: $CR(0) = 0.99$, $n = 10$ (left) and $n = 50$ (right).

Let us analyze now the case of a DE variant with parameter adaptation based on generating values for CR uniformly distributed in $[0, 1]$. Such an approach is similar with that used in the jDE self-adaptive algorithm proposed in [3]. As in all self-adaptive variants, in jDE each individual is extended with some components corresponding to the control parameters F and CR. Thus CR_i and F_i are the parameters corresponding to element i. At each generation the control

parameters can be randomly perturbed with a small probability (e.g. 0.1). For instance each CR_i is replaced, with probability 0.1 with a value generated by a uniform distribution on $[0,1]$ and remains unchanged (or is replaced with CR_*, the crossover rate corresponding to the best element in the current population) with probability 0.9. Using a uniform distribution for new values of CR ensures a good exploration of the domain of p_m values ($[1/n,1]$) in the case of binomial crossover but in the case of exponential crossover this is not necessarily true. In this last case in order to have a uniform distribution of p_m on $[1/n,1]$ the crossover rate should have a non-uniform distribution.

Let us denote by f_{CR} and f_{p_m} the distribution probabilities of the random variables corresponding to the crossover rate and mutation probability, respectively. If we denote with g the function which expresses the relationship between CR and p_m ($p_m = g(CR)$) then

$$f_{p_m}(y) = f_{CR}(g^{-1}(y))/g'(g^{-1}(y)) \tag{17}$$

If we want a uniform distribution of p_m on $[1/n,1]$ then $f_{p_m}(y) = n/(n-1)$. Thus the distribution probability of CR should be $f_{CR}(x) = g'(x)n/(n-1)$. In the case of classical exponential crossover $g(x) = (1-x^n)/(n(1-x))$ and the distribution probability of CR should be

$$f_{CR}(x) = \frac{n}{n-1}\left(\frac{1-x^n}{n(1-x)^2} - \frac{x^{n-1}}{1-x}\right) \tag{18}$$

with the corresponding cumulative distribution function

$$F_{CR}(x) = \frac{x(1-x^{n-1})}{(n-1)(1-x)} \tag{19}$$

The random variable CR having the cumulative distribution function given by eq. (19) can be simulated by using the inverse cumulative distribution function method which needs a table of values of the inverse of F_{CR}. In our simulations we tabulated F_{CR}^{-1} with the step $h = 0.01$ and we used these values to simulate the non-uniform distribution of CR.

Let us analyze now the impact of using a uniform and a transformed non-uniform distribution on the behavior of a self-adaptive variant which uses a strategy inspired from [3] for changing the values of individual crossover rates corresponding to generation g:

$$CR_i(g) = \begin{cases} u_i & \text{with probability } 0.1 \\ CR_*(g-1) & \text{with probability } 0.9 \end{cases} \tag{20}$$

where u_i is a random value generated at each generation and for each population element by using an uniform distribution on $[0,1]$ or the distribution described in eq. (19). $CR_*(g-1)$ is the crossover rate used in obtaining the best element of the previous generation. Two sets of experiments were conducted: one using a fixed value for F (Tables 9 and 8) and one based on adapting the scaling parameters F_i^j such that the variance of the population is kept at almost the

same level (Table 10). The parameter $F_i^j(g)$, used to generate the component j of the mutant vector corresponding to the element i at generation g is computed, after $CR_i(g)$ was adjusted, by using the following equation, derived by solving eq. (16):

$$F_i^j(g) = \sqrt{\frac{c_j - 1}{2p_m(i,g)} + \frac{1}{m} - \frac{p_m(i,g)}{2m}} \tag{21}$$

where $c_j = \frac{Var(x^j(g-2))}{Var(x^j(g-1))}$ and $p_m(i,g)$ is the mutation probability corresponding to the current value of the crossover rate (given by eq. (4) in the case of binomial crossover and by eq. (7) in the case of exponential crossover). The choice of c_j aims to ensure that the mutation and crossover operators applied at generation g compensate the effect of DE operators had on the population variance at generation $g-1$.

The simulation results are presented in Tables 8 - 10 where for each test function and each crossover variant are presented the average and standard deviation values for the best value in the last generation (f^*) and the number of function evaluations (nfe). The number of successful runs out of 30 (S) and the mutation probability (p_m) estimated based on the last generation are also presented. The analyzed crossover variants are: the classical binomial crossover and the classical exponential crossover combined with a uniform distribution of CR and with the non-uniform distribution given by eq. (18).

The results in Table 8 suggests that for functions like shifted Griewank the behavior of DE is not very sensitive to the choice of the crossover type or distributions involved in the perturbation of the crossover rate. On the other hand for the rotated variant of the Griewank function using an uniform distribution for CR seems to be less effective, with respect to the number of successful runs, than using a non-uniform distribution (as in eq. (18)). On the other hand the results obtained in the case of the uniform distribution of CR are less dispersed than in the case of the non-uniform distribution. The estimated values of the mutation probability suggest that in the case of the rotated variant higher values of the p_m (and implicitly of CR) are favored.

Crossover type	Shifted Griewank fct.($n = 100$)				Rotated Griewank fct. ($n = 50$)			
	$\langle f^* \rangle$ (stdev)	$\langle nfe \rangle$ (stdev)	S	$\langle p_m \rangle$	$\langle f^* \rangle$ (stdev)	$\langle nfe \rangle$ (stdev)	S	$\langle p_m \rangle$
Binomial	$< \epsilon$ (10^{-10})	342920 (22845)	30	0.209	$6 \cdot 10^{-4}$ (0.0024)	397688 (48241)	27	0.287
Exponential (uniform)	$< \epsilon$ (10^{-10})	331510 (7327)	30	0.056	10^{-5} (10^{-5})	500000 (0)	0	0.095
Exponential (non-uniform)	$< \epsilon$ (10^{-10})	363593 (16998)	30	0.367	0.0043 (0.0065)	393525 (77597)	20	0.495

Table 8: Influence of the crossover type and of the distribution of CR on the behavior of an adaptive DE similar with the jDE algorithm [3] ($m = n$, $F = 0.5$, $\epsilon = 10^{-8}$). Test functions: shifted and rotated Griewank.

A different situation arises in the case of the Rastrigin function (Table 9). For the separable variant, the exponential crossover with uniform distribution of CR behaves better because it is biased toward small values of p_m. This is no more true in the rotated case when the function is not separable. None of the variants were able to approach the global optimum of the rotated Rastrigin function. This is in accordance with the results presented in [23] where is also stated that the classical DE/rand/1/* algorithms cannot deal with high nonseparable functions. The results in Tables 8 and 9 were obtained for a fixed value of the scaling factor ($F = 0.5$). A slight improvement, except for the exponential crossover with uniformly distributed CR, is obtained by using individual scaling factors and adapting them according to eq. (21) (see Table 10). None of the variants proved to be consistently better than the other ones.

Crossover type	Shifted Rastrigin fct. ($n = 100$)				Rotated Rastrigin fct. ($n = 50$)			
	$\langle f^* \rangle$ (stdev)	$\langle nfe \rangle$ (stdev)	S	$\langle p_m \rangle$	$\langle f^* \rangle$ (stdev)	$\langle nfe \rangle$ (stdev)	S	$\langle p_m \rangle$
Binomial	59.84 (93.54)	489750 (30023)	5	0.148	177.72 (70.12)	500000 (0)	0	0.098
Exponential (uniform)	$< \epsilon$ (10^{-10})	401066 (21536)	30	0.046	269.65 (53.33)	500000 (0)	0	0.040
Exponential (non-uniform)	41.47 (63.39)	497426 (8145)	3	0.332	196.29 (56.35)	500000 (0)	0	0.203

Table 9: Influence of the crossover type and of the distribution of CR on the behavior of an adaptive DE similar with the jDE algorithm [3] ($m = n$, $F = 0.5$, $\epsilon = 10^{-8}$). Test functions: shifted and rotated Rastrigin.

Crossover type	Shifted Rastrigin fct. ($n = 100$)				Rotated Rastrigin fct. ($n = 50$)			
	$\langle f^* \rangle$ (stdev)	$\langle nfe \rangle$ (stdev)	S	$\langle p_m \rangle$	$\langle f^* \rangle$ (stdev)	$\langle nfe \rangle$ (stdev)	S	$\langle p_m \rangle$
Binomial	1.028 (1.77)	405470 (91634)	16	0.550	99.044 (51.06)	500000 (0)	0	0.254
Exponential (uniform)	0.165 (0.451)	276510 (87813)	26	0.157	209.86 (6899)	500000 (0)	0	0.038
Exponential (non-uniform)	1.79 (3.72)	497090 (15670)	1	0.654	130.73 (27.87)	500000 (0)	0	0.705

Table 10: Influence of the crossover type and of the distribution of CR on the behavior of an adaptive DE with randomly perturbed CR and F chosen according to eq. 21($m = n$, $\epsilon = 10^{-8}$). Test functions: shifted and rotated Rastrigin.

6. Conclusions

Since the DE crossover operator constructs the trial vector from the current and mutant vectors it determines how many and which components are mutated. The number of components to be mutated, and implicitly the mutation

probability, is influenced by the crossover rate, CR. The selection of components to be mutated is based on a distribution probability which is the binomial one (in the case of binomial crossover) and the truncated geometric one (in the case of exponential crossover).

These different distribution probabilities lead to a different dependence between the mutation probability, p_m and the crossover rate, CR. In the case of binomial crossover this dependence is linear while in the case of exponential crossover it is nonlinear. Thus for the same value of CR the percent of mutated components is different for the two crossover variants.

As numerical results suggest the difference between the behavior of binomial and exponential crossover is more influenced by the different impact of CR on p_m than by the different strategies of selecting components from the mutant vector.

For the same value of CR the mutation probability is larger in the case of binomial crossover than in the case of exponential one, the difference being larger as the problem size, n, is larger. In the case of exponential crossover there is a small range of CR values (usually $[0.9, 1]$) to which the DE is sensitive. This could explain the rule of thumb derived for the original variant of DE [20] which was based on the exponential crossover: "*use values of* CR *in the range* $[0.9, 1]$". It also agrees with experimentally derived remarks as: "*larger* CR *is required for exponential crossover to generate crossover segment at a suitable size and lower* CR *is useful for binomial crossover to perform robust convergence*" [8].

Moreover, in the case of exponential crossover, when n is large the range of CR values for which p_m is significant becomes very narrow. In fact when n goes to infinity p_m tends to 0 for all values of CR except for the value 1 and tends to 1 just for $CR = 1$. Thus for high-dimensional problems the use of exponential crossover can generate difficulties in choosing proper values for CR.

Besides the classical implementation of exponential crossover other implementation variants based on similar distribution probabilities but using fewer calls of random functions were proposed and numerically analyzed. For the analyzed test functions the proposed crossover variants (Algorithms 4 and 5) behaves similarly (slightly better in the case of Griewank functions) with the classical exponential crossover.

Since the parameters CR and F are interrelated it is expected that for the same value of CR the adequate value of F can be different for different crossover variants. Indeed, as results in Section 5.1. suggest, for the same value of CR the exponential variant needs a larger value for the scaling parameter, F, in order to avoid premature convergence.

On the other hand, when tuning the value of CR the use of a uniform discretization of $[0, 1]$ as in [11] is appropriate for binomial crossover but not necessarily appropriate for exponential crossover (since exponential DE is more sensitive to values of CR between $(0.9, 1]$ than to values between $(0, 0.9]$). Similarly using a uniformly random perturbation of CR as in self-adaptive jDE algorithm proposed in [3] can be adequate when binomial crossover is used but not necessarily also in the case of exponential crossover.

The aim of this paper was rather to analyze the impact of the crossover

type on the DE behavior and on the choice of adequate control parameters than to prove the superiority of one crossover variant over the other one. Further analysis should be conducted in order to identify a class of problems where one of the crossover types is leading to the best results. The main conclusion is that when using exponential crossover one have to take into account the particular dependence between CR and p_m and its impact on parameter tuning or adaptation.

Acknowledgment. This work is supported by the Romanian grant PN-II 11028/ 14.09.2007.

References

[1] M.M. Ali, L.P. Fatti, A Differential Free Point Generation Scheme in the Differential Evolution Algorithm, Journal of Global Optimization, 35(2006) 551-572.

[2] B.V. Babu, S.A. Munawar, Differential evolution strategies for optimal design of shell-and-tube heat exchangers, Chemical Engineering Science, 62, issue 14 (2007) 3720-2739.

[3] J. Brest, B. Bošković, S. Greiner, V. Žurner, M.S. Maučec, Performance comparison of self-adaptive and adaptive differential evolution algorithms, Soft Computing, 11, issue 7 (2007) 617 - 629.

[4] J.F.C. Ter Braak, A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces, Stat. Comput. 16(2006) 239-249.

[5] U. K. Chakraborty, S. Das, A. Konar, Differential evolution with local neighborhood, in: Proc. of IEEE Congress on Evolutionary Computation (CEC-2006), IEEE Press, 73957402.

[6] I. De Falco, A. Della Cioppa, A. Tarantino, Automatic Classification of Handsegmented Image Parts with Differential Evolution, in: Rothlauf et al. (Eds.), EvoWorkshops 2006, LNCS 3907 (2006) 403-414.

[7] R. Gämperle, S.D. Müller, P. Koumoutsakos, A Parameter Study for Differential Evolution, in: A. Grmela, N. E. Mastorakis (Eds.), Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, WSEAS Press,2002, 293-298.

[8] T. Gong, A. Tuson, Differential Evolution for Binary Encoding, Soft Computing in Industrial Applications, Springer-Verlag, 39 (2007) 251-262.

[9] H.Y. Fan, J. Lampinen, A Trigonometric Mutation Operation to Differential Evolution, Journal of Global Optimization, 27 (2003) 107-129.

[10] J. Liu, J. Lampinen, A fuzzy adaptive Differential Evolution, Soft Computing, 9 (2005) 448-462.

[11] E. Mezura-Montes, J. Velásquez-Reyes, C.A. Coello Coello, A Comparative Study of Differential Evolution Variants for Global Optimization, in: Maarten Keijzer et al., eds., 2006 Genetic and Evolutionary Computation Conference (GECCO'2006), Vol. 1, (ACM Press, 2006) 485-492.

[12] A.C. Nearchou, S.L. Omirou, Differential evolution for sequencing and scheduling optimization, Journal of Heuristics, 12 (2006), 395-411.

[13] M.G.H. Omran, A.P. Engelbrecht, A. Salman, Differential Evolution Methods for Unsupervised Image Classification, in: Proceedings of IEEE Congress on Evolutionary Computation, (IEEE Computer Press, 2005) vol. 2, 966 - 973.

[14] M.G.H. Omran, A.P. Engelbrecht, A. Salman, Using the Ring Neighborhood Topology with Self-adaptive Differential Evolution, in: Lecture Notes of Computer Science (Springer Verlag, 2006), vol. 4221, 976-979.

[15] J. Rönkkönen, S. Kukkonen, K.V. Price, Real-parameter optimization with differential evolution, in: Proceedings of CEC 2005, (IEEE Computer Press, 2005) vol. 1, 567-574.

[16] S. Paterlini and T. Krink, Differential Evolution and Particle Swarm Optimization in Partitional Clustering, Computational statistics & data analysis, 50(5),2005, 1220-1247.

[17] K.V. Price, R. Storn, J. Lampinen, Differential Evolution. A Practical Approach to Global Optimization, Springer, 2005.

[18] K.V. Price, Eliminating drift bias from the Differential Evolution Algorithm, in: U.K. Chakraborty (Ed.), Advances in Differential Evolution, Studies in Computational Intelligence 143, 2008, 33-88.

[19] A. Qin, P. Suganthan, Self-adaptive Differential Evolution for Numerical Optimization, in: Proceedings of CEC 2005, (IEEE Computer Press, 2005) vol. 1, 630-636.

[20] R. Storn, K. Price, Differential Evolution a simple and efficient adaptive scheme for global optimization over continuous spaces. International Computer Science Institute, Berkeley, TR- 95-012, 1995.

[21] R. Storn, K. Price, Differential Evolution A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization, 11(1997) 341359.

[22] 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, Technical Report, Nanyang Technological University, Singapore, AND KanGAL Report # 2005005, IIT Kanpur, India, 2005.

[23] A.M. Sutton, M. Lunacek and L.D. Whitley, Differential Evolution and Non-separability: Using selective pressure to focus search, in: Proceedings of GECCO'07, (ACM Press, 2007) 1428 - 1435.

[24] J. Teo, Exploring dynamic self-adaptive populations in differential evolution, Soft Computing - a fusion of foundations, methodologies and applications, 10(8), 2006, 673-686.

[25] J. Tvrdik, Differential Evolution with Competitive Setting of Control Parameters, in: Task Quarterly, vol. 11 (2007), issues 1-2, 169-179.

[26] J. Tvrdik, Adaptive differential evolution and exponential crossover,In: Proceedings of IMCSIT 2008, vol. 3, 927-931.

[27] K. Ursem, P. Vadstrup, Parameter identification of induction motors using differential evolution, in: Proceedings of CEC 2003, (IEEE Computer Press, 2003) vol. 2, 790- 796.

[28] J. Vesterstrom, R. Thomsen, A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems, in: Proceedings of CEC 2004, (IEEE Computer Press, 2004), vol. 2, 1980-1987.

[29] F. Xue, A.C. Sanderson, R.J. Graves, Multi-objective differential evolution - algorithm, convergence analysis and applications, in: Proceedings of CEC 2005, (IEEE Computer Press, 2005) 743-750.

[30] D. Zaharie, Critical values for the control parameters of differential evolution algorithms, in: R. Matoušek and P. Ošmera (Eds.), Proceedings of 8th International Conference on Soft Computing, Mendel 2002, 62-67.

[31] D. Zaharie, Control of population diversity and adaptation in differential evolution algorithms, in: R. Matoušek and P. Ošmera (Eds.), Proceedings of 9th International Conference on Soft Computing, Mendel 2003, 41-46.

[32] D. Zaharie, A Comparative Analysis of Crossover Variants in Differential Evolution, in: Proceedings of the IMCSIT,2nd International Symposium Advances in Artificial Intelligence and Applications, 2007, 171-181.

[33] K. Zielinski, R. Laur, Stopping Criteria for Differential Evolution in Constrained Single-Objective Optimization, in: U.K. Chakraborty (Ed.): Advances in Differential Evolution, Studies in Computational Intelligence 143, 2008, 111138.

[34] http://www.icsi.berkeley.edu/~storn/code.html: Differential Evolution web page [last access: december, 2007].