Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 171–181

ISSN 1896-7094
© 2007 PIPS

A Comparative Analysis of Crossover Variants in Differential Evolution

Daniela Zaharie

Faculty of Mathematics and Computer Science
West University of Timişoara
bv. Vasile Pârvan, nr. 4, 300223 Timişoara, Romania
dzaharie@info.uvt.ro

Abstract. This paper presents a comparative analysis of binomial and exponential crossover in differential evolution. Some theoretical results concerning the probabilities of mutating an arbitrary component and that of mutating a given number of components are obtained for both crossover variants. The differences between binomial and exponential crossover are identified and the impact of these results on the choice of control parameters and on the adaptive variants is analyzed.

Keywords: differential evolution, crossover operators, parameter control

1 Introduction

Differential evolution (DE) [9] is a population based stochastic heuristic for global optimization on continuous domains which is characterized by simplicity, effectiveness and robustness.

Its main idea is to construct, at each generation, for each element of the population a 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 values selected from $\{1, \ldots, m\}$ and $F > 0$ is a scaling factor. This difference based mutation operator is the distinctive element of DE algorithms allowing a gradual exploration of the search space. 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 $CR \in [0, 1]$. 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. In the following we shall consider objective functions, $f : D \subset \mathbb{R}^n \to \mathbb{R}$, to be minimized.

The general structure of a DE (see algorithm 1) is typical for evolutionary algorithms, the particularities of the algorithm being related with the mutation and crossover operators. By combining different mutation and crossover operators various schemes have been designed. In the DE literature these schemes are denoted by using the convention $DE/a/b/c$ where a denotes the manner of constructing the mutant vector, b denotes the number of differences involved in the construction of the mutant vector and c denotes the crossover type.

Algorithm 1 The general structure of a generational DE

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

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 [4, 8, 7] and a significant number of adaptive and self-adaptive variants have been proposed [2, 6, 10, 12]. Most of these results were obtained based on empirical studies. Despite some theoretical analysis of the DE behavior [1, 3, 11] 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. [5, 1]). The crossover operator attracted much less attention, just two variants being currently used, the so-called binomial and exponential crossover. If the exponential crossover is that proposed in the original work of Storn and Price [9], the binomial variant was much more used in applications. Besides statements like ”*The crossover method is not so important although Ken Price claims that binomial is never worse than exponential*” [13] or some recent experimental studies involving both binomial and exponential variants [7] no systematic comparison between these two crossover types was conducted.

The aim of this paper is to analyze the similarities and differences between binomial and exponential crossover emphasizing on their theoretical properties and on their influence on the choice of appropriate control parameters. By such an analysis we can hope to find insights into the behavior of DE and to find explanations for statements like: "*if you choose binomial crossover like, CR is usually higher than in the exponential crossover variant*" [13].

The rest of the paper is structured as follows. Section 2 presents the implementation details of binomial and exponential crossover variants. In section 3 some theoretical results concerning the probability of selecting a component from the mutant vector and concerning the average number of mutated components are derived. Based on these results, in section 4 is analyzed the influence of crossover variants on the choice of control parameters. Section 5 concludes the paper.

2 Crossover variants in differential evolution

The crossover operator aims to construct an offspring by mixing components of the current element and of that generated by mutation. There are two main crossover variants for DE: binomial (see Algorithm 2) and exponential (see Algorithm 3). In the description of both algorithms $irand$ denotes a generator of random values uniformly distributed on a finite set, while $rand$ simulates a uniform random value on a continuous domain. For both crossover variants the mixing process is controlled by a so-called crossover probability usually denoted by CR.

In the case of binomial crossover a component of the offspring is taken with probability CR from the mutant vector, y, and with probability $1-CR$ from the current element of the population, x. The condition "$rand(0,1) < CR$ or $j = k$" of the **if** statement in Algorithm 2 ensures the fact that at least one component is taken from the mutant vector. This type of crossover is very similar with the so-called uniform crossover used in evolutionary algorithms. On the other hand, the exponential crossover 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 [9], 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, by following a power law. 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 it is 1 when $j = n$.

Besides the fact that the exponential crossover allows to mutate just consecutive, in a circular manner, elements while binomial crossover allows any configuration of mutated and non-mutated components there is also another difference between these strategies. In the binomial case the parameter CR determines

Algorithm 2 Binomial crossover

```
crossoverBin (x,y)
$k \leftarrow irand(\{1,\ldots,n\})$
for $j = \overline{1,n}$ do
    if $rand(0,1) < CR$ or $j = k$ then
        $z^j \leftarrow y^j$
    else
        $z^j \leftarrow x^j$
    end if
end for
return $z$
```

Algorithm 3 Exponential crossover

```
crossoverExp (x,y)
$z \leftarrow x$; $k \leftarrow irand(\{1,\ldots,n\})$; $j \leftarrow k$; $L \leftarrow 0$
repeat
    $z^j \leftarrow y^j$; $j \leftarrow \langle j+1 \rangle_n$; $L \leftarrow L+1$
until $rand(0,1) > CR$ or $L = n$
return $z$
```

explicitly the probability for a component to be replaced with a mutated one. In the implementation of the exponential crossover, CR is used to decide how many components will be mutated. However, in both situations CR influences the probability for a component to be selected from the mutant vector. This probability, denoted in the following by p_m, is in fact similar with the mutation probability in genetic algorithms and it is expected that its value has an influence on the DE behavior.

Because of these differences between the two crossover variants one can have different mutation probabilities and different distributions of the number of the mutated components for the same value of CR. These aspects are analyzed in more details in the next section.

3 A theoretical analysis

From a statistical point of view, the binomial crossover is achieved by a set of n independent Bernoulli trials, the result of each trial being used in selecting a component of the offspring from the mutant vector. 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 (event "$rand(0,1) < CR$") and one of probability $1/n$ (event "$j = irand(\{1,\ldots,n\})$"). Thus the probability that a component is mutated is $p_m = CR(1-1/n) + 1/n$.

The number, L, of components selected from the mutated vector has a binomial distribution with parameters n and p_m. Thus the probability that h components are mutated is $Prob(L = h) = C_n^h p_m^h (1-p_m)^{n-h}$. Based on the properties of the binomial distribution it follows that the average of the number of mutated components is $E(L) = np_m$.

If the stopping condition of the **repeat** loop in the Algorithm 3 would be just $rand(0,1) > CR$ then L would take values according to the geometric 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 $Prob(L = h) = CR^{h-1}(1 - CR)$. However in the exponential crossover the number of mutated components is bounded by L, thus we are dealing with a truncated geometric distribution.

Thus the probability distribution of L is given by:

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

Using eq. 1 it follows that the average of L is $E(L) = (1 - CR^n)/(1 - CR)$. 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. 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:

$$Prob(z^j = y^j) = \frac{1}{n}\sum_{k=1}^{n} Prob(d(j,k) < L) = \frac{1}{n}\sum_{d=0}^{n-1} Prob(L > d) \tag{2}$$

Since $Prob(L > d) = CR^d$ it follows that

$$Prob(z^j = y^j) = \frac{1}{n}\sum_{d=0}^{n-1} CR^d = \frac{1 - CR^n}{n(1 - CR)} \tag{3}$$

A summary of all these values corresponding to binomial and exponential crossover is presented in Table 1.

Table 1. Summary of theoretical results

Crossover type	p_m	$Prob(L = h)$	$E(L)$
Binomial	$CR(1 - 1/n) + 1/n$	$C_n^h p_m^h (1 - p_m)^{n-h}$	$CR(n - 1) + 1$
Exponential	$\frac{1 - CR^n}{n(1 - CR)}$	$\begin{cases} (1 - CR)CR^{h-1} & 1 \leq h < n \\ CR^{n-1} & h = n \end{cases}$	$\frac{1 - CR^n}{1 - CR}$

Thus both the probability that an arbitrary component is mutated and the probability distribution of the number of mutated components are different between binomial and exponential crossover. More specifically, the dependence between p_m and CR is linear in the case of binomial crossover and nonlinear in the exponential case. 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.

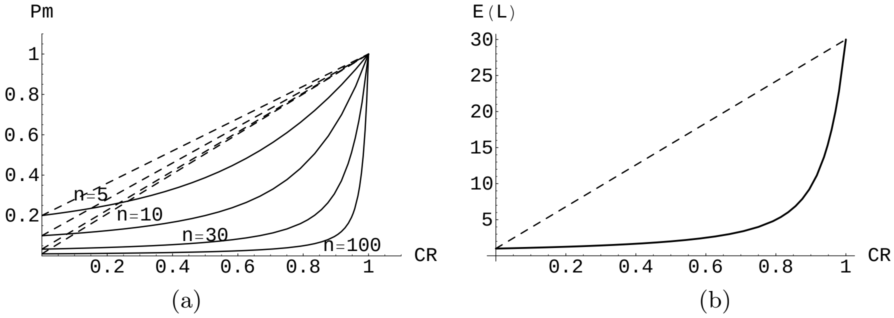

Fig. 1. Influence of CR on the mutation probability (a) and on the average of the number of mutated components for $n = 30$ (b) in the case of binomial crossover (dashed line) and exponential crossover (continuous line)

4 Influence of the crossover variant on the choice of control parameters

Since for the same value of CR the probability of mutating a component and the average number of mutated components are different for binomial and exponential crossover it follows that the results of a parameter study conducted for one crossover variant are not necessary true for the other one. The correspondence between values of CR and mutation probability in binomial and exponential crossover is presented in Table 2 for two dimensions of the problem ($n = 30$, $n = 100$). As figure 1 also suggests, in the case of exponential crossover there are two ranges of values for CR with different impact on the effect of crossover. The first range, $[0, CR_1]$, is characterized by a low sensitivity of the algorithm behavior to the value of CR while the second one $[CR_1, 1]$ characterized by a high sensitivity. The threshold value, CR_1, is higher for higher values of n. Thus, in the case of exponential crossover, as n is higher the sensitive range of CR is smaller (being included in $[0.9, 1]$). For instance when $n = 100$ for $CR \in [0, 0.9]$ the mutation probability, p_m, varies only between 0.01 and 0.09. This means

Table 2. Correspondence between CR and the mutation probability, p_m, for binomial and exponential crossover

n	CR	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.92	0.95	0.97	0.99	1
30 (bin)	p_m	0.03	0.13	0.23	0.32	0.42	0.52	0.61	0.71	0.81	0.90	0.92	0.95	0.97	0.99	1
30 (exp)	p_m	0.03	0.03	0.04	0.04	0.05	0.06	0.08	0.11	0.16	0.31	0.38	0.52	0.66	0.86	1
100 (bin)	p_m	0.01	0.11	0.21	0.31	0.41	0.51	0.60	0.70	0.80	0.90	0.92	0.95	0.97	0.99	1
100 (exp)	p_m	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.03	0.05	0.09	0.12	0.19	0.31	0.63	1

that parameters studies concerning CR should be differently conducted in the case of binomial and exponential crossover.

In order to illustrate the influence of the crossover variant on the sensitivity of DE to different values of CR some empirical tests were conducted. Tables 3 and 4 present the dependence between the number of function evaluations (nfe*1000) until the global optimum is approached with an accuracy of $\epsilon = 10^{-6}$ and values of CR in the case of two test functions: a multimodal separable one (e.g. Rastrigin [6]) and a multimodal nonseparable one (e.g. Griewank [6]) both of dimension $n = 30$ and having a global minimum in 0. In both cases we used a $DE/rand/1/c$ variant, a rather small population size ($m = 50$) and the same value for the scaling factor, $F = 0.5$. The maximal number of evaluations was set to 250000 (nfe=250). The absence of the nfe value means that the algorithm did not approximated the global minimum with the desired accuracy. All results are averages obtained for 30 independent runs. They confirm the fact that the behavior of the algorithm depends on the value of the mutation probability, p_m, meaning that for the same value of p_m (which corresponds to different values of CR for binomial and exponential crossover) similar behavior is observed. In the case of Rastrigin function, which is a separable one, good behavior is obtained for small values of p_m [8] which in the case of the binomial crossover corresponds to small values of CR. In the case of exponential crossover, since for a large range of CR the mutation probability is small, the set of CR values for which the algorithm is able to identify the global optimum with the desired accuracy is significantly larger than in the case of binomial crossover.

For the Griewank function the best behavior is obtained for values of CR in the range $[0.1, 0.5]$ in the case of binomial crossover and in the range $[0.7, 0.95]$ in the case of exponential crossover. Both ranges of CR values corresponds to similar ranges of p_m: $[0.13, 0.52]$ (binomial crossover) and $[0.11, 0.52]$ (exponential crossover). The different results obtained by the two crossover variants for similar values of p_m can be explained by the fact that in the case of nonseparable functions mutating a sequence of components (as in exponential crossover) or arbitrary components (as in binomial crossover) generates different exploration patterns.

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. Since DE is prone to premature convergence

Table 3. Number of function evaluations (nfe*1000) needed to approximate the optimum with the accuracy $\epsilon = 10^{-6}$. Test function: Rastrigin, $n = 30$; Algorithm: DE/rand/1, $m = 50$, $F = 0.5$.

CR	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.92	0.95	0.97	0.99	1
p_m (bin)	0.03	0.13	0.23	0.32	0.42	0.52	0.61	0.71	0.80	0.90	0.92	0.95	0.97	0.99	1
nfe (bin)	45.4	74.6	182.8	-	-	-	-	-	-	-	-	-	-	-	-
p_m (exp)	0.03	0.04	0.04	0.05	0.06	0.07	0.08	0.11	0.16	0.31	0.38	0.52	0.66	0.86	1
nfe (exp)	45.5	46.0	46.8	47.8	49.2	51.3	53.8	59.0	68.4	99.2	115.4	162.2	240	-	-

Table 4. Number of function evaluations (nfe*1000) needed to approximate the optimum with the accuracy $\epsilon = 10^{-6}$. Test function: Griewank, $n = 30$; Algorithm: DE/rand/1, $m = 50$, $F = 0.5$.

CR	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.92	0.95	0.97	0.99	1
p_m (bin)	0.03	0.13	0.23	0.32	0.42	0.52	0.61	0.71	0.80	0.90	0.92	0.95	0.97	0.99	1
nfe (bin)	62.6	39.0	36.2	35.1	36.6	39.5	41.4	45.7	59.3	72.4	72.9	72.9	75.9	96.5	91.9
p_m (exp)	0.03	0.04	0.04	0.05	0.06	0.07	0.08	0.11	0.16	0.31	0.38	0.52	0.66	0.86	1
nfe (exp)	60.9	58.6	58.7	56.2	51.6	49.6	46.6	43.5	40.2	42.2	41.2	44.7	48.8	65.1	94.4

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 [11] is derived a theoretical relationship between the control parameters and the population variance after and before applying the variation operators. More specifically, if $\mathrm{Var}(z)$ and $\mathrm{Var}(x)$ denote the averaged variance of the trial and current populations respectively then the following relationship is true:

$$\mathrm{Var}(z) = (2p_m F^2 - \frac{2p_m}{m} + \frac{p_m^2}{m} + 1)\mathrm{Var}(x) \tag{4}$$

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

$$2p_m F^2 - \frac{2p_m}{m} + \frac{p_m^2}{m} + 1 = c \tag{5}$$

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 [11] was obtained only in the case of binomial crossover considering that $p_m = CR$. By replacing in eq. 5 p_m with $CR(1-1/n)+1/n$ or with $(1-CR^n)/(n(1-CR))$, one obtains 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 exponential crossover 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.

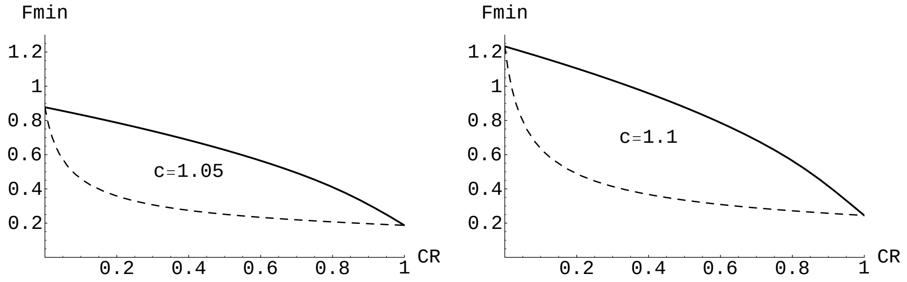

Fig. 2. Lower bound for F vs. CR for binomial crossover (dashed line) and exponential crossover (continuous line). Parameters: $m = 50$, $n = 30$

These differences should be taken into account when conducting parameter studies on DE variants involving binomial and exponential crossover(like in [7]). For instance when tuning the value of CR the use of a uniform discretization of $[0, 1]$ as in [7] 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 $(0, 0.9]$). 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 [12] the adaptation rules for F and CR should be modified according to eq. 4 and to the relationship between p_m and CR.

On the other hand for self-adaptive variants which use random selection for the control parameters values (like in [2]) one have to take into account the fact that if for binomial crossover a uniform distribution of CR values is appropriate (leading to a uniform distribution of p_m) a different situation appears in the case of exponential crossover. In this case uniformly distributed values of CR do not necessarily lead to uniformly distributed values of p_m, meaning that the adaptation strategy should be changed.

5 Conclusions

The comparative analysis of binomial and exponential crossover variants offered us some information about the influence of parameter CR on the behavior of DE. The dependence between the mutation probability, p_m and the crossover parameter, CR, was derived both in the case of binomial and of exponential crossover applied to differential evolution. This dependence is linear in the binomial case and nonlinear in the exponential one. 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. This means that in order to reach a similar effect by the mutation step a DE algorithm with exponential crossover should use a larger value for CR.

Moreover, in the case of exponential crossover one have to be aware of the fact that 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: *"use values of CR in the range $[0.9, 1]$"*. On the other hand, for the same value of CR the exponential variant needs a larger value for the scaling parameter, F, in order to avoid premature convergence.

Acknowledgment

This work is supported Romanian grants
99-II CEEX 03-INFOSOC 4091/31.07.2006 and CNCSIS-MindSoft.

References

1. Ali M. M. and Fatti L. P.: *A Differential Free Point Generation Scheme in the Differential Evolution Algorithm,* Journal of Global Optimization, **35**, pp. 551-572, 2006.
2. Brest J., Boškovič B., Greiner S., Žurner V. and Maučec M. S.: *Performance comparison of self-adaptive and adaptive differential evolution algorithms,* Soft Computing, **11**(7), pp. 617–629, 2007.
3. Ter Braak J. F. C.: *A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces,* Stat Comput, **16**, pp. 239–249, 2006.
4. Gämperle R., Müller S. D. and Koumoutsakos P.: *A Parameter Study for Differential Evolution,* in A. Grmela, N. E. Mastorakis (eds.), Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, WSEAS Press, pp. 293–298, 2002.
5. Fan H. Y. and Lampinen J.: *A Trigonometric Mutation Operation to Differential Evolution,* Journal of Global Optimization, **27**, pp. 107–129, 2003.
6. Liu J. and Lampinen J.: *A fuzzy adaptive Differential Evolution,* Soft Computing, **9**, pp. 448–462, 2005.
7. Mezura-Montes E., Velásquez-Reyes J. and Coello Coello C. A.: *A Comparative Study of Differential Evolution Variants for Global Optimization,* in Maarten Keijzer et al. (editors), 2006 Genetic and Evolutionary Computation Conference (GECCO'2006), Vol. 1, ACM Press, Seattle, Washington, USA, July, pp. 485–492, 2006.
8. Rönkkönen J., Kukkonen S. and Price K. V.: *Real-parameter optimization with differential evolution,* Proceedings of CEC 2005, 2-3 September, Edinburgh, **1**, pp. 567–574, 2005.
9. Storn R. and Price K.: *Differential Evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces,* International Computer Science Institute, Berkeley, TR-95-012, 1995.
10. Tvrdik J.: *Differential Evolution with Competitive Setting of Control Parameters,* Task Quarterly, **11**(1-2), pp. 169–179, 2007.

11. Zaharie D.: *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, pp. 62–67, 2002.
12. Zaharie D.: *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, pp. 41–46, 2003.
13. `http://www.icsi.berkeley.edu/~storn/code.html` Differential Evolution web page [last access: july, 2007].