ROBUST'2000, 198 – 209

ON CONVERGENCE OF A CLASS OF STOCHASTIC ALGORITHMS

LADISLAV MIŠÍK, JOSEF TVRDÍK AND IVAN KŘIVÝ

Abstract. V tomto príspevku sa skúmajú otázky spojené s konvergenciou obecnej triedy algoritmov pre globálnu optimalizáciu. Za istých predpokladov sú dokázané nutné a postačujúce podmienky konvergencie takýchto algoritmov. Ďalej je ukázaná jednoduchá metóda modifikácie ľubovoľného algoritmu uvažovanej triedy na konvergentný algoritmus tejto triedy. Táto modifikácia môže byť pritom ľubovoľne malá, aby podstatne neovplyvnila praktický beh pôvodného algoritmu. V experimentálnej časti sú aplikované dosiahnuté výsledky na niektoré konkrétne algoritmy uvažovaných tried. Výsledky týchto pokusov potvrdzujú, že z praktického hľadiska nemá splnenie požiadavky konvergencie významný vplyv na chod algoritmu.

Резюме: В этой работе рассматриваются вопросы касающиеся конвергенции общего класса алгоритмов для глобальной оптимизации. Ишодя из весьма общих предположений здесь доказываются необходимые и достаточные условия конвергенции этих алгоритмов. Дальше предлагается простой метод приведения любого алгоритма рассматриваемых классов к конвергентному алгоритму того же самого класса. Эта модификация может быть произвольно маленькая, чтобы на практике не влиять на действие оригинального алгоритма. В экспериментальной части работы полученные результаты применяются к некоторым конкретным алгоритмам обоих классов. Результаты этих экспериментов показывают, что выполнение условий конвергенции не имеет на практике существенное влияние на действие алгоритма.

1. Introduction

Let $D \subset \mathcal{R}^d$. Denote by $\mathcal{L}$ the system of all Lebesgue measurable subsets of D and λ the Lebesgue measure on $\mathcal{L}$. Let f be a real Lebesgue measurable function defined on D. The number $m = inf\{t\,;\ \lambda(f^{-1}(-\infty, t)) > 0\}$, where $f^{-1}(A) = \{x \in D\,;\ f(x) \in A\}$, is called the *essential minimum* of f. The task is to find an arbitrary good approximation of m.

1.1. **Evolutionary algorithm.** Let p_0 be a probability measure on $(D, \mathcal{L})$ which is positive on each open subset of D. Let N be a positive integer called the *size of population* and D^N the set of all populations. Let π be a mapping defined on D^N assigning to each population $\mathcal{P}$ the probability measure $\pi(\mathcal{P})$ on $(D, \mathcal{L})$. Finally, let $\{m_n\}$ be a sequence of numbers from interval $\langle 0, 1\rangle$ and let $\mathcal{C}$ be a rule according to which some points in the old population are replaced by new ones. We will consider a class of *evolutionary algorithms* described as follows:

2000 *Mathematics Subject Classification.* Primary 65C20; Secondary 68U20.
Key words and phrases. Global optimization, convergence of algorithms.
This research was supported by both the grant no. 402/00/1165 of the Czech Grant Agency and the institutional research grant no. CEZ: J09/98:179000002.

- I: Generate an initial population $\mathcal{P}_0 = \{x_1, x_2, \cdots, x_N\}$ chosen as an independent identically distributed sample according to the probability measure p_0 and set $n = 0$.
- C_n: Copy a portion of M best points of $\mathcal{P}_n$ directly into the new population. Here, the best points means the points at which the function f has its lowest values and M is an integer from the set $\{0, 1, 2, \cdots, N-1\}$.
- S_n: Select a new point at random according to the probability measure $p_{n+1} = \pi(\mathcal{P}_n)$ and include it to the new population when fulfilling the condition $\mathcal{C}$. Repeat this procedure until the new population is complete.
- M_n: With the probability m_{n+1} replace a randomly chosen point by its mutation.
- R: Set $n = n + 1$ and go back to C_n.

Mutation of a given point is a point in D generated according to the specified rules. Algorithms using only steps I, C_n, S_n, R we will call *CRS algorithms* (Controlled Random Search). In other words, CRS is an evolutionary algorithm, for which all m_n are equal to zero.

1.2. **Convergence of an algorithm.** By convergence of an algorithm we mean the convergence with probability 1, i.e. an algorithm is convergent if for any measurable function f

$$P(\lim_{n\to\infty} min\{f(x);\ x \in \mathcal{P}_n\} = m) = 1.$$

2. Convergence of CRS algorithm

2.1. **General conditions for the convergence of CRS algorithm.**
Definition. We will say that an CRS algorithm saves the best point if

- the best point in the old population is copied into the new population directly at step C_n, i.e. $M > 0$.
- if a point selected at step S_n has lower f value than any point in the old population then it is included into the new population by rule $\mathcal{C}$.

The following theorem provides the necessary and sufficient condition for the convergence of an CRS algorithm and it is similar to Global Search Convergence Theorem by Solis and Wets in [5].

Theorem 1. Suppose that an CRS algorithm saves the best point. Then it is convergent if and only if for each set S of positive Lebesgue measure, we have

$$\Pi_{n=1}^{\infty} \left(1 - p_n(S)\right) = 0. \tag{1}$$

Proof: First suppose that the equation (1) holds. The infinite product on the left side expresses the probability of repeatedly missing the set S when selecting new points of population. Let f be a measurable function and denote by m its essential minimum. Choose $\epsilon > 0$. Let us consider the probability that a point y in the set $F_\epsilon = f^{-1}(-\infty, m + \epsilon)$ of a positive Lebesgue measure (by definition of essential minimum) will be selected at latest at the step S_n. This probability is greater than or equal to $1 - \Pi_{i=1}^{n} \left(1 - p_i(F_\epsilon)\right)^{N-M}$. According to the assumption of Theorem 1 point y can be removed from the population only if a better point will later be selected. Denote by x_n the best point in population $\mathcal{P}_n$. Then, by equation (1)

$$\lim_{n\to\infty} P(x_n \in F_\epsilon) \geq 1 - \Pi_{n=1}^{\infty} \left(1 - p_n(S)\right)^{N-M} =$$

$$= 1 - \left(\Pi_{n=1}^{\infty} \left(1 - p_n(S)\right)\right)^{N-M} = 1$$

and, as $\epsilon > 0$ was arbitrary, the algorithm is convergent.
Now suppose that the equation (1) does not hold for some set S of positive Lebesgue measure. Define the function f by

$$f(x) = \begin{cases} 0, & x \in S \\ 1, & x \in D - S. \end{cases}$$

Then f is a measurable function and the algorithm does not converge to the essential minimum of f.

2.2. **Necessary condition for the convergence of CRS.** In the case when $M = N - 1$ and the rule $\mathcal{C}$ is as follows

If $f(y_{n+1}) < max\{f(x);\ x \in \mathcal{P}_n\} = f(x_i)$ replace x_i by y_{n+1} in $\mathcal{P}_n$ to produce $\mathcal{P}_{n+1}$, otherwise set $\mathcal{P}_{n+1} = \mathcal{P}_n$,
where a new trial point y_{n+1} is generated from $\mathcal{P}_n$ by a heuristic,

the following theorem provides the necessary condition for the convergence of CRS.

Theorem 2. If an algorithm is convergent then for any collection $\{U_1, U_2, \cdots, U_N\}$ of open subsets of D and for every measurable set $S \subset D$ of positive Lebesgue measure there is a selection $\mathcal{P} = \{x_1 \in U_1, x_2 \in U_2, \cdots x_N \in U_N\}$ such that $\pi(\mathcal{P})(S) > 0$.

Proof: Suppose the contrary and, without loss of generality, let $\left(\bigcup_{i=1}^{N} \overline{U_i}\right) \cap S = \emptyset$. Consider the measurable function

$$f(x) = \begin{cases} -1, & x \in S \\ 0, & x \in \bigcup_{i=1}^{N} U_i \\ 1, & x \in D - \left(S \cup \left(\bigcup_{i=1}^{N} U_i\right)\right) \end{cases}$$

As p_0 is positive on open sets, the probability α of the choice $\mathcal{P}_0 = \{x_1 \in U_1, x_2 \in U_2, \cdots x_N \in U_N\}$ is positive. For such a choice we have $\mathcal{P}_n = \mathcal{P}_0$ for all integers n and the algorithm will never reach the essential minimum in S. Thus

$$P(\lim_{n\to\infty} min\{f(x);\ x \in \mathcal{P}_n\} = m) < 1 - \alpha$$

and the algorithm does not converge with probability 1.

2.3. **Modification of a nonconvergent algorithm.** The following theorem provides a simple way how to modify an arbitrary CRS algorithm to a convergent one.

Theorem 3. Let two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ work with the same size of population and let they use the mappings π_1 and π_2, respectively, to determine the corresponding probability measures. Further, let for every $S \subset D$ of positive Lebesgue measure there exists a positive constant δ_S such that $\pi_1(\mathcal{P})(S) \geq \delta_S$ holds for all $\mathcal{P} \in D^N$. Finally, let $\sum_{n=1}^{\infty} c_n$ be an arbitrary divergent series of real numbers from the interval $\langle 0, 1\rangle$. Then the algorithm $\mathcal{A}$ using the mapping

$$\pi = c_n\pi_1 + (1 - c_n)\pi_2$$

is convergent.

Remark 1. It is evident that the algorithm $\mathcal{A}_1$ itself is convergent. (If $c_n = 1$ for all n then $\mathcal{A}$ is identical with $\mathcal{A}_1$.)

Remark 2. The expression $c_n\pi_1 + (1-c_n)\pi_2$ should be interpreted in the way that the new probability measure p_{n+1} is generated with the probability c_n by the function π_1 and with the probability $(1 - c_n)$ by the function π_2.

In the proof of Theorem 3 the following simple Lemma is used.

Lemma 1. Let $\alpha_n \in \langle 0,1\rangle$ for all integers n and let the infinite series $\sum_{n=1}^{\infty} \alpha_n$ be divergent. Then $\prod_{n=1}^{\infty}(1-\alpha_n)=0$.

Proof of Lemma 1: Note that the condition $\prod_{n=1}^{\infty}(1-\alpha_n)=0$ is equivalent to the condition $\prod_{n=1}^{\infty}\frac{1}{(1-\alpha_n)}=\infty$. An easy calculation

$$\prod_{n=1}^{\infty}\frac{1}{(1-\alpha_n)}=\prod_{n=1}^{\infty}\left(\sum_{k=0}^{\infty}\alpha_n^k\right)>1+\sum_{n=1}^{\infty}\alpha_n=\infty$$

completes the proof.

Proof of Theorem 3: Let $S\subset D$ be of positive Lebesgue measure. According to the definition of the algorithm $\mathcal{A}$ and the properties of π_1

$$\alpha_n=\pi(\mathcal{P}_n)(S)\geq c_n\delta_S$$

holds. The series $\sum_{n=1}^{\infty}\alpha_n$ is divergent and, by Lemma 1, $\prod_{n=1}^{\infty}(1-\alpha_n)=0$. An application of Theorem 1 completes the proof.

3. Convergence of evolutionary algorithms

3.1. **Conditions for the convergence of evolutionary algorithms.** The introduction of mutation into an algorithm can influence its convergence in both the positive and the negative direction. It may adapt the nonconvergent CRS algorithm to the convergent evolutionary algorithm and vice versa.

The positive role of mutation may be described as follows. Suppose that the corresponding CRS algorithm is not convergent and that the sequence (of best points of populations) generated is far from the essential minimum. Since the new point created by mutation need not depend on the history of populations, there can be a positive probability that its value belongs to the small neighbourhood of the essential minimum. Of course, it depends on the way how the mutation does work.

Now suppose that the value of the best point of population is close to the essential minimum. There is a positive probability that the best point will be replaced via mutation by much worse point. Thus it is possible that the mutation can break the convergence of corresponding CRS algorithm. The following definition can be motivated by the above discussion.

Definition. We will say that the mutation excludes the best point if the best point of population must not be replaced by its mutation.

Note that the condition of exclusion of the best point is not too restrictive in the case when the size of population is relatively high. Now we can state the sufficient condition for the convergence of evolutionary algorithms.

Theorem 4. Let for each set $S\in D$ of positive Lebesgue measure, the probability that the new point created by mutation belongs to S be positive. Further suppose that the mutation excludes the best point and that the series $\sum_{n=1}^{\infty} m_n$ is divergent. Then the evolutionary algorithm is convergent provided the corresponding CRS algorithm saves the best point.

Proof: Let $\epsilon>0$ and $F_\epsilon=f^{-1}(-\infty,m+\epsilon)$. It suffices to show that

$$\lim_{k\to\infty}P(x_k\in F_\epsilon)=1.$$

Denote by s the positive probability that the new point created by mutation belongs to F_ϵ. Then the probability that a point from F_ϵ is included to the population $\mathcal{P}_n$ via mutation is equal to $s\ m_n$ and the probability that a point from F_ϵ will be included to the population $\mathcal{P}_n$ either via mutation or via selection at step $[S_n]$ is greater than or equal to $s\ m_n$. Now, because of saving the best point and excluding it from mutation, we have

$$P(x_k \in F_\epsilon) \geq 1 - \Pi_{n=1}^{k}\,(1 - s\ m_n)\,.$$

Lemma 1 and the Theorem 1 show that the algorithm is convergent.

3.2. **Examples.** The assumptions of Theorem 4 provide only a sufficient condition for the convergence of evolutionary algorithms. Thus, in connection with this theorem, some natural questions arise. We will touch two of them here.

Question 1. What can be said about the convergence, when the assumption of excluding the best point is removed in Theorem 2?

Question 2. What can be said about the convergence, when the positive probability of selection of a mutant is restricted only to a part of the domain?

We conjecture that in no one of the above questions the positive answer about the convergence can be stated. Let us give a short informal explanation of the reasons why.

Concerning *Question 1*: For a general evolutionary algorithm, the probability of creation a new point in a sufficiently small neighbourhood of global minimum can be smaller than the probability of its removing from the population via mutation when the point is not protected. We illustrate this possibility by the following example.

Let $a \in (0,1)$, $\epsilon \in (0,1)$ and let $\phi_{a,\epsilon}$ be a measurable function defined in the interval $\langle 0, K\ \rangle$ such that $\phi_{a,\epsilon}(0) = 0$, $\phi_{a,\epsilon}(1) = 1$, $\phi_{a,\epsilon}(K) = \epsilon$, let $\phi_{a,\epsilon}$ be increasing in the interval $(0,a)$ and $\phi_{a,\epsilon}$ decreasing in the interval (a,K). Further, let $D \subset \{X \in \mathrm{R}^d; |X| \leq K\ \}$ and $f_{a,\epsilon}(X) = \phi_{a,\epsilon}(|X|)$ for all $X \in D$. Let $d = 2$, $N = 10$, $M = 5$, $D = \{[x,y];\ \sqrt{x^2+y^2} \leq 100\}$, $m_n = 0.1$, f*or* $n = 1, 2, \ldots$ and consider the function $f_{a,\epsilon}$. For simplicity suppose that all new points and mutants are chosen by the uniform probability distribution. Then the probability that a point in a sufficiently small neighbourhood of the global minimum will appear when forming new population is less than $1 - \left[1 - \pi a^2/\left(\pi 100^2\right)\right]^{N-M} = 1-(1-a^2 10^{-4})^5$. On the other hand, the probability that such a point will be mutated to a point outside the neighbourhood $\{[x,y];\ \sqrt{x^2+y^2} < a\}$ of the global minimum is equal to $m_n \cdot \frac{1}{N}\left[1 - \pi a^2/\left(\pi 100^2\right)\right] = \frac{1}{100}(1-a^2 10^{-4})$. One can show that for sufficiently small values of a, the probability of disappearance of "good" points is greater than the probability of their appearance. Consequently the algorithm can not be convergent.

Concerning *Question 2*: Consider the same function $f_{a,\epsilon}$ as in the previous example and suppose, for example, that mutation changes only a one randomly chosen coordinate of a point. Thus the new mutant can belong to a sufficiently small neighbourhood of global minimum only in the case when all but one coordinates of the original point are very close to 0. On the other hand, the probability of existence of such a point in population is not very high by the nature of the function f and one can demonstrate the non-convergence of algorithm in a similar way as in the previous example.

4. Experimental comparison of algorithms

4.1. **Algorithms selected for tests.** *Modified controlled random search algorithm* (MCRS) is described in [2, 3, 8]. It starts from a population P of N uniformly distributed points in D. A new trial point $\mathbf{x}$ is generated from a simplex S $(d+1$

points chosen at random from P) by the relation

$$\mathbf{x} = \mathbf{g} - Y(\mathbf{z} - \mathbf{g}), \tag{2}$$

where $\mathbf{z}$ is one (randomly taken) pole of the simplex S, $\mathbf{g}$ the centroid of the remaining d poles of the simplex and Y a random multiplication factor. Several distributions of Y were tested in [3] and it was found that good optimization results were obtained with Y distributed uniformly in $\langle 0, \alpha)$ with α ranging from 4 to 8. Point $\mathbf{x}$ may be considered as resulting from the reflection of point $\mathbf{z}$ with respect to centroid $\mathbf{g}$. Using the reflection procedure, initial population P is iteratively contracted by replacing the worst point with a better point (with respect to f-values). Input parameters of MCRS are: population size N and parameter α of uniform distribution in randomized reflection.

Differential evolution (DE) algorithms [7, 6] work with two populations: P (old) and Q (new). Basically, for each vector (point) $\mathbf{x}_{i,P}$, $i = 1, 2, \ldots, N$, of old population P a mutant vector $\mathbf{v}_{i,Q}$ is generated by adding the weighted difference of vectors from P to another vector from P. In order to increase the diversity of the new vectors, crossover is introduced. Therefore, trial vector $\mathbf{u}_{i,Q}$ of new population Q is created by replacing some elements of vector $\mathbf{x}_{i,P}$ by corresponding elements of mutant vector $\mathbf{v}_{i,Q}$.

Two different strategies are used to generate mutant vectors [7, 6]:

- strategy using three randomly taken vectors of P (DERAND algorithm),
- strategy using the best vector and four randomly taken vectors of P (DEBEST algorithm).

The strategies as well as the crossover procedure are described in [6] in detail. The input parameters of DE algorithms are: population size N, dilatation coefficient of vector difference $F > 0$ and parameter C influencing the number of elements L to be exchanged by crossover, L being drawn from $\langle 1, d \rangle$ with the probability $P(L \geq t) = C^{t-1}$ for $t = 1, 2, \ldots, d$, $C \in (0, 1)$. No proof of convergence of differential evolution has been yet known [7].

Evolutionary search (ES) algorithms [2, 8] also work with two populations: old population P and new population Q. The new population inherits the properties of the old one in two ways: directly by surviving a number M of the best points (with respect to f-values) and indirectly by applying the reflection to the old population. Moreover, a point with new properties (even with a larger f-value) is allowed to arise with a small probability p (mutation probability). The convergence of evolutionary search under specific circumstances is proved in Theorem 4. An example of the evolutionary search is the ES5 procedure.

procedure ES5
generate P (an old population of N points taken at random from D)
repeat
    copy M best points of P into new population Q
    find the worst point in P, $\mathbf{x}_{worst}$
    if condition for the mutation is true
        then *mutate* any point of P except the best point
    repeat
        repeat *Reflection* applied to a simplex from P
        until the next trial point is better than $\mathbf{x}_{worst}$
        insert the next trial point into Q

until Q is completed to N points
replace P by Q
until stopping condition is true

The two kinds of mutation were used in the experiments:

- Uniform, when a mutant point is taken at random from the uniform distribution on $D = \prod_{i=1}^{d} < a_i, b_i >, \quad a_i < b_i, \quad i = 1, 2, \ldots, d,$
- Normal, when point $\mathbf{x}$ selected to mutation is changed into point $\mathbf{y} \sim N\left(\mathbf{x}, \sigma^{\mathbf{2}}\right)$, where $\sigma^{\mathbf{2}} = \operatorname{diag}\left({\sigma_1}^2, {\sigma_2}^2, \ldots, {\sigma_d}^2\right), \;\; \sigma_i = (b_i - a_i)/6.$

In the next text the algorithms are called ES5UNI and ES5NOR, resp. Both the ES5UNI and the ES5NOR fulfill the conditions of convergence given in Theorem 4.

The ES5 algorithm without any mutation (when condition for mutation never occurs and $M < N - 1$) can be considered as a kind of modified controlled random search working with two populations and we call it MCRS2.

When compared with MCRS algorithms, ES algorithms have two additional tuning parameters: probability of mutation p and number M of the best points surviving from old population P to new population Q or its upper limit.

Mixtures of the heuristics mentioned above were also tested in this research, namely:

- DExRF12, where in each generation the heuristic DERAND was followed by $2N$ reflections according to (2),
- DExRF11, where in each generation the heuristic DERAND was followed by N reflections,
- DExRF21, where in each generation the heuristic DERAND was followed by $N/2$ reflections,
- DExES5N, where the heuristic DERAND was combined with the heuristic ES5NOR with one mutation in each generation.

4.2. **Test functions.** For the testing of the algorithms we used such problems where at least one of the nonconvergent heuristics, i.e. MCRS or DE, sometimes failed in searching for the global minimum. From this point of view the following functions [7] were chosen:

Shekel's Foxholes

$$f(\mathbf{x}) = \left(0.002 + \sum_{i=0}^{24} \frac{1}{i + 1 + \sum_{j=1}^{2} (x_j - a_{ij})^6}\right)^{-1}$$

where the elements of $\mathbf{A}$ matrix are defined as follows:

$$a_{i1} = -32, -16, 0, 16, 32 \quad \text{for} \quad 0, 1, 2, 3, 4, \quad \text{resp.},$$
$$a_{i1} = a_{i(mod\ 5),1} \quad \text{for} \quad i \geq 5,$$
$$a_{i2} = -32, -16, 0, 16, 32 \quad \text{for} \quad i = 0, 5, 10, 15, 20, \quad \text{resp.},$$
$$a_{i2} = a_{i+k,2} \quad \text{for} \quad k = 1, 2, 3, 4.$$

The global minimum is $\mathbf{x}^\star = (-32, -32)$ and $f(-32, -32) \simeq 0.998004$.

Corana's parabola

$$f(\mathbf{x}) = \sum_{i=1}^{4} \begin{cases} 0.15 a_i \left[z_i - 0.05 \operatorname{sgn}(z_i)\right]^2 & \text{if} \quad |x_i - z_i| < 0.05 \\ a_i {x_i}^2 & \text{otherwise} \end{cases}$$

where $z_i = 0.2 \operatorname{sgn}(x_i) \lfloor |5 x_i| + 0.49999 \rfloor$ and $\mathbf{a} = (1, 1000, 10, 100)$.

The global minimum is $\mathbf{x}^\star = (\langle -0.05, 0.05\rangle, \ldots, \langle -0.05, 0.05\rangle)$ and $f(\mathbf{x}^\star) = 0$.

Griewangk's function

$$f(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{d} {x_i}^2 - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

The global minimum is $\mathbf{x}^\star = (0, 0, \ldots, 0)$ and $f(\mathbf{x}^\star) = 0$.

4.3. **Experimental conditions.** The stopping condition for all the tests were defined as it follows. Suppose that $f_1, f_2, \ldots, f_N$ is a nondecreasing sequence of the function values in a given iteration of the algorithm. The optimization process was stopped when $f_5 - f_1 \leq 10^{-7}$.

For each optimization task 100 independent runs were carried out. The common tuning parameter α for MCRS and ES5 was adjusted to the same value, $\alpha = 8$. In each generation the additional tuning parameter M for ES5 algorithms was set at random from $\langle 1, \lfloor N/2 \rfloor\rangle$, where $\lfloor N/2 \rfloor$ is the largest integer less than or equal to $N/2$. Concerning DE algorithms the tunning parameters were set $F = 0.9$ and $C = 0.9$.

Three basic characteristics of the algorithms were considered in the experiments:

- reliability R defined as (100 − percentage of failures), the failure being considered as the stopping at a local minimum;
- convergence rate measured by the average number of objective function evaluations *NE* for stopping at global minimum;
- number of objective function evaluations *NE*1 needed for reaching a state very close to the global minimum, specially when $f(\mathbf{x}_{best}) - f(\mathbf{x}_{opt}) < 1 \times 10^{-3}$.

4.4. **Results for CRS.** Using Theorem 2, it can be shown that the modified controlled random search heuristic (MCRS) fails to be convergent. In agreement with Theorem 3 we tested its modification based on a *combination* of the MCRS and the simple random search with new trial points uniformly distributed on D, $c_n = c$ for all integers n, $c \in \langle 0, 1\rangle$. This *combined* algorithm was tested on well-known fifth De Jong's function (Shekel's Foxholes), for which MCRS stopped at a local minimum in about 60 % of runs [8]. The relative frequency of the simple random search is an input parameter c of the algorithm. Value $c = 0$ reduces the algorithm to MCRS, value $c = 1$ gives the simple random search. The switching between the MCRS and the simple random search is made at random in each step with the probability c for the simple random search and $(1 - c)$ for the MCRS. Searching space D was constrained to $\prod_{j=1}^{2}[-40, 60]$ in these experiments.

From Fig. 1 it is evident that the reliability of the combined MCRS increases almost linearly with increasing c. Fig. 2 shows the dependence of number of the function evaluations (NE) on the reliability for the combined MCRS (Fig. 1 – left) and the original (nonconvergent) MCRS (Fig. 1 – right). In both cases the growth of NE is approximately exponential but in the case of the nonconvergent MCRS the growth of NE is much slower. Moreover, the original nonconvergent MCRS enables to achieve the same level of reliability even at smaller values of NE by increasing the population size. By increasing the population size from 10 to 35 the reliability of 100 per cent was achieved, see Fig. 2 – right .

4.5. **Results for evolutionary algorithms.** Population size N and the size of searching space $D = \prod_{i=1}^{d} < a_i, b_i >, \quad a_i < b_i, i = 1, 2, \ldots, d$ were set to be the same for all the algorithms depending on the tested function, see Table 1.
The experimental results are summarized in Tables 2, 3 and 4. As we can see in the e 2, any general conclusion cannot be made on nonconvergent heuristics. We obtained

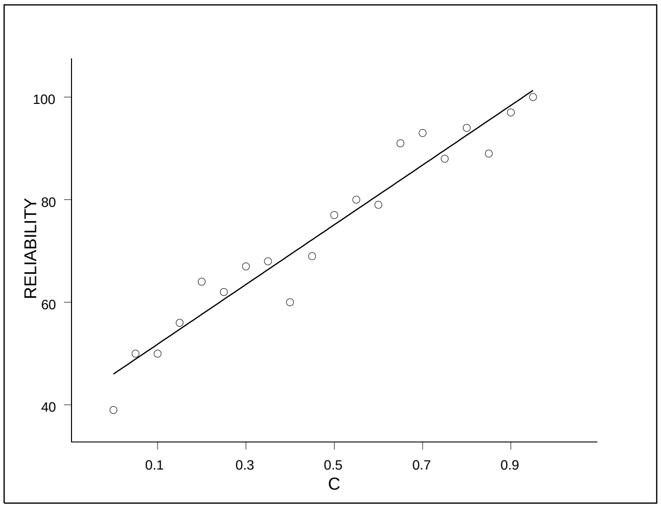

FIGURE 1. Empirical dependence of reliability on c

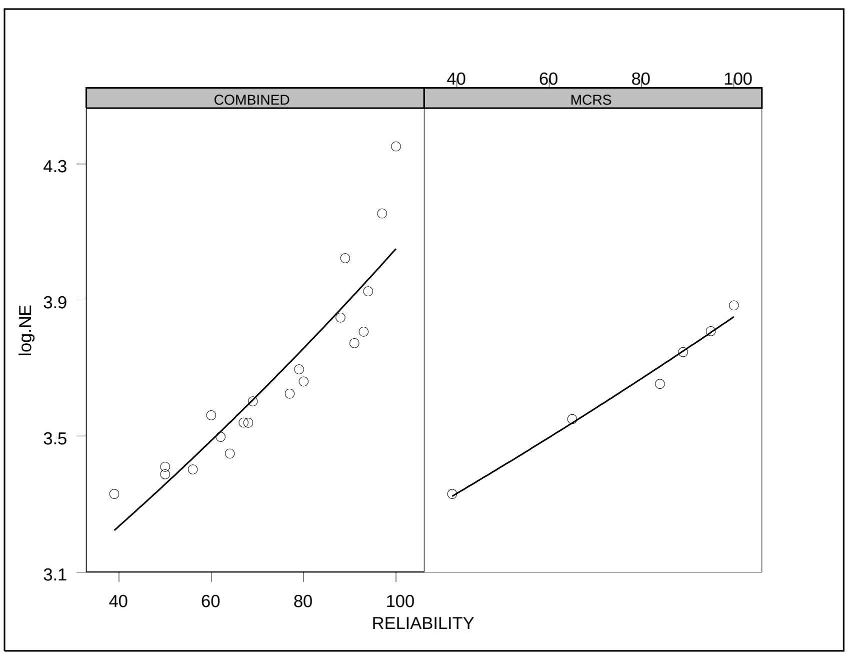

FIGURE 2. Empirical dependence of log NE on reliability

TABLE 1. Population size and searching space

Function	d	N	$\langle a_i, b_i \rangle$
Shekel	2	10	$\langle -65.536, \quad 65.536 \rangle$
Corana	4	20	$\langle -1000, \quad 1000 \rangle$
Griewangk	10	100	$\langle -400, \quad 400 \rangle$

TABLE 2. Nonconvergent simple heuristics

	Corana		Shekel		Griewangk	
Algorithm	R	NE	R	NE	R	NE
DEBEST	97	6855	69	845	100	327810
DERAND	96	4503	37	700	100	199015
MCRS	100	6577	39	2136	23	63256
MCRS2	97	11124	71	2921	100	219342

very different values of R and NE for the three functions under considerations. For Corana's function the MCRS was found to be the only reliable optimization algorithm. On the contrary, the MCRS is the only unreliable algorithm for the optimization of Griewangk's function but it is significantly faster than the others. In the case of Shekel's function all the heuristics are not reliable, both DERAND and MCRS being heavily unreliable.

TABLE 3. ES5 – dependence R and NE on probability of mutation p for Shekel's function

	Uniform			Normal		
p	R	$NE1$	NE	R	$NE1$	NE
0.2	71	2157	3072	72	2272	3173
0.4	69	2269	3189	69	2272	3020
0.6	67	2221	3116	80	2272	3240
0.8	75	2380	3272	79	2417	3395
1.0	82	2273	3252	71	2471	3609

As regards the ES5 algorithm, we can see in Table 3 that for Shekel's function the reliability increases very slowly with increasing probability of mutation and NE values are also slightly increasing. The overall results are not significantly better than the results of simple nonconvergent heuristics (compare with Table 2). For Griewangk's function the full reliability was achieved by the MCRS2 algorithm and additional mutation in ES5 only slightly increased NE values, i.e. it did not bring any improvement of the results.
The results of the mixtures of different heuristics are summarized in Table 4. Interesting results were achieved by using the DExES5N algorithm in the case of Griewangk's function. The DExES5N is the combination of DERAND and ES5NOR with one mutated point in each generation. The results are shown in Fig. 3. We can see that the DExES5N is much better with respect to NE than the both parental algorithms.

Table 4. Mixtures

	Corana		Shekel		Griewangk	
Algorithm	R	NE	R	NE	R	NE
DExRF21	98	3722	51	776	100	162098
DExRF11	93	3380	58	815	100	180115
DExRF12	91	3240	58	949	96	169796
DExES5N	95	9044	80	2467	100	127476

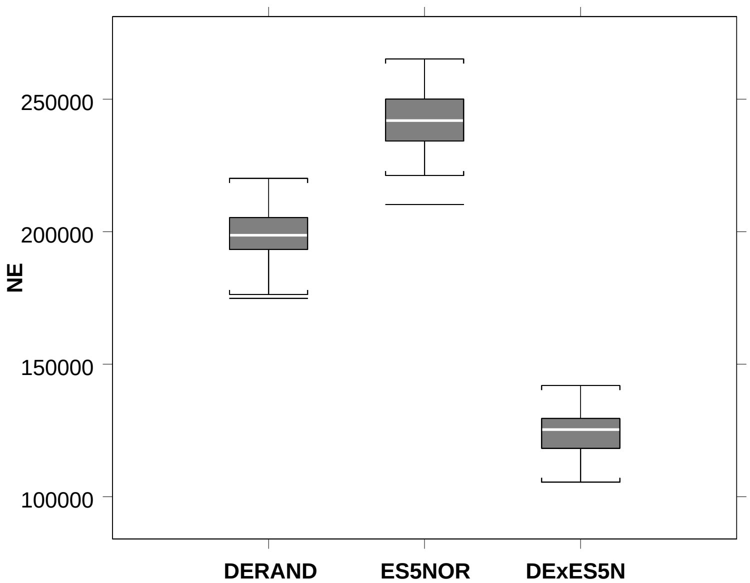

Figure 3. Griewangk's function - comparison of NE for a mixture of DERAND and ES5NOR

5. Conclusions

The proof of convergence for a stochastic algorithm may bring some useful ideas for its implementation. However, the proof of convergence with probability 1 is very weak result from practical point of view. A nonconvergent heuristic can give more reliable results than a related theoretically convergent algorithm even at a smaller number of objective function evaluations. There is no guarantee that the evolutionary algorithm with proved convergence is more reliable than a nonconvergent heuristic in solving a concrete problem of the global optimization of a multimodal function.

Sometimes a mixture of different heuristics can bring a substantial decrease in the number of objective function evaluations and/or an increase in reliability, but any commonly used guideline for such a mixing is not yet available.

References

[1] Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice.* New York: Oxford University Press

[2] Křivý, I. and Tvrdík, J. (1996). Stochastic Algorithms in Estimating Regression Models. In: A. Prat (Ed.), *COMPSTAT'96. Proceedings in Computational Statistics*, Physica Verlag, Heidelberg, 325–330

[3] Křivý, I. and Tvrdík, J. (1995). The Controlled Random Search Algorithm in Optimizing Regression Models. *Comput. Statist. and Data Anal.*, **20**, 229–234

[4] Mišík, L. (2000). On Convergence of a Class of Evolutionary Algorithms. In: *MENDEL'2000. 6th International Conference on Soft Computing* (Tech. Univ. of Brno, June 2000). Brno, 97–100

[5] Solis, F. J. and Wets, R. J-B. (1981). Minimization by Random Search Techniques. *Mathematics of Operations Research* **6**, 19–30

[6] Storn, R. and Price, K. (1995). *Differential evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces.* Technical Report TR-95-012, ISCI, http://http.isci.berkeley.edu/∼storn/litera.html

[7] Storn, R. and Price, K. (1997). Differential Evolution – a Simple and Efficient Heuristic for Global Optimization. *J. Global Optimization* **11**, 341–359

[8] Tvrdík, J. and Křivý, I. (1999). Simple Evolutionary Heuristics for Global Optimization. *Comput. Statist. and Data Anal.* **30**, 345–352

University of Ostrava, 30. dubna 22, 701 03 Ostrava, Czech Republic
E-mail address: `Ladislav.Misik; krivy@osu.cz; tvrdik@osu.cz`