Schedae Informaticae Vol. 23 (2014): 9–20
doi: 10.4467/20838476SI.14.001.3018

A Short Introduction to Stochastic Optimization

Jerzy Ombach
Department of Mathematics
Faculty of Mathematics and Computer Science
Jagiellonian University
ul. Łojasiewicza 6, 30-348 Kraków
e-mail: *ombach@im.uj.edu.pl*

Abstract. We present some typical algorithms used for finding global minimum/maximum of a function defined on a compact finite dimensional set, discuss commonly observed procedures for assessing and comparing the algorithms' performance and quote theoretical results on convergence of a broad class of stochastic algorithms.

Keywords: global optimization, stochastic algorithm, random search, convergence of metaheuristics.

1. Introduction

One of the most common problems in applied mathematics is how to find approximation of an optimal solution of a function defined on some subset of finite dimesnional space. In particular, optimization problems lie at the heart of most machine learning approaches. There exists a lot of numerical optimization procedures. Even fifty years ago most of them were deterministic methods. However, with the spread of computers, stochastic methods have appeared and in recent years we have been witnessing an explosion of heuristic stochastic algorithms. Generally a heuristic is understood to be a rule of thumb learned from experience but not always justified by an underlying theory. Actually, we will consider metaheuristic which designates a computational method that optimizes a problem by iteratively trying to improve a candidate solution. While the performance of some of metaheuristics applied to specific instances looks good and is experimentally confirmed, theoretical background is definitely behind.

The paper is addressed to computer scientists and mathematicians who have been not yet familiar with the stochastic optimization. So our first goal is to demonstrate a few stochastic algorithms and quote some results on their convergence, Then, to mention the problem of experimental comparative study of such algorithms with machine learning perspective. And in Section 5, to present some theoretical results on convergence of a broad class of Markov type algorithms to the optimal solution based mostly on the author's and his coworkers papers: [3], [8], [7], [9], [10], [11], [13], [14], [15]. We complete the paper with a list of R packages designated for stochastic optimization. We recommend books [5] and [17] for further reading on meataheuristics. Many recent information and materials about the subject can be also found at http://coco.gforge.inria.fr, see Section 4 for more details about that page.

Given set $A \subset \mathbb{R}^n$ and continuous function $f : A \longrightarrow \mathbb{R}$ denote $A^\star = \arg\min f = \{a \in A : f(a) \leq f(x) \text{ for all } \in A\}$. If A is compact, then $A^\star$ is nonempty. We want to find points that approximate $A^\star$. As a metaheuristics generate random points, these points are considered as realisations of some random vectors, and then we are interested in convergence of a sequence of n-dimensional random vectors, say X_t, to $A^\star$. If all X_t are defined on the same probability space, say, $X_t : \Omega \longrightarrow A$, we consider here two types od such convergence. Stochastic convergence, i.e.

$$\forall \varepsilon > 0 \;\; \text{Prob}(dist(X_t, A^*) < \varepsilon) \;\longrightarrow 1, \text{ as } t \longrightarrow \infty,$$

and almost sure convergence, i.e.

$$\text{Prob}\left(X_t \longrightarrow A^\star, \text{ as } t \longrightarrow \infty\right) = 1,$$

i.e.

$$\text{Prob}\left(\{\omega \in \Omega\colon \;\text{dist}(X_t(\omega), A^\star) \longrightarrow 0, \text{ as } t \longrightarrow \infty\}\right) = 1,$$

where $\text{dist}(x, K)$ denotes the distance x from K.

2. Simple Random Search Algorithms

In this Section we present a collection of typical and simple stochastic algorithms of global optimizations. Our presentation starts from the simplest algorithm and gradually becomes more advanced.

Pure Random Search

Most natural seems to be Pure Random Search (PRS). We quote it in a standard context. Namely, we assume that that $f : A \longrightarrow \mathbb{R}$ is continuous where A is the unit cube, i.e. $A = [0, 1]^n \subset \mathbb{R}^n$.

Algorithm

0 Set $t = 0$. Generate a point x_0 from the uniform distribution on A.

1 Given x_t, generate y_t from the uniform distribution on A.

2 If $f(y_t) < f(x_t)$, then let $x_{t+1} = y_t$.

3 Increase $t := t + 1$ and go to Step 1.

Let X_t, $t = 0, 1, 2, \ldots$ be random vectors which realizations are generated by PRS. The following rather obvious and well-known result can be proved by using probabilistic arguments like Borel-Cantelli Lemma, see [10] for a detailed proof.

Theorem 1

$$Prob\,(X_t \longrightarrow A^\star,\ as\ t \longrightarrow \infty) = 1.$$

Accelerated Random Search

Accelerated Random Search1 (ARS), see [1], is a variant of PRS: the search is confined to shrinking neighborhoods of a previous record-generating value, with the search neighborhood reinitialized to the entire space when a new record is found. Local minima are avoided by including an automatic restart feature which reinitializes the search neighborhood after some number of shrink steps have been performed.

As above we assume that $f : A \longrightarrow \mathbb{R}$ is continuous and $A = [0, 1]^n \subset \mathbb{R}^n$. Fix $c > 1$ (a shrinking factor) and $\rho > 0$ (a precision threshold).

Algorithm

0 Set $t = 1$ and $r_1 = 1$. Generate x_1 from the uniform distribution on A.

1 Given $x_t \in A$ and $r_t \in (0, 1]$, generate y_t from the uniform distribution on $B(x_t, r_t) \cap A$, where $B(x, r)$ is the ball of radius r centered at x.

2 If $f(y_t) < f(x_t)$, then let $x_{t+1} = y_t$ and $r_{t+1} = 1$.

3 If $f(y_t) \geq f(x_t)$, then:

(a) If $r_t \geq \rho$, put $x_{t+1} = x_t$ and $r_{t+1} = r_t/c$.

(b) If $r_{t+1} < \rho$, put $r_{t+1} = 1$.

4 Increase $t := t + 1$ and go to Step 1.

Let X_t, $t = 0, 1, 2, \ldots$ be random vectors which realizations are generated by ARS. We quote two results on convergence of ARS.

Theorem 2 ([15]) *Assume, that for any $c \in \mathbb{R}$ the level cirve $l_c = \{x \in A : f(x) = c\}$ has its Lebesgue measure 0. Then:*

$$Prob\,(X_t \longrightarrow A^\star,\ as\ t \longrightarrow \infty) = 1.$$

Theorem 3 ([1]) *Assume that f has finitely many global minima. Let $\{M_t\}$ be the record sequence produced by ARS above, i.e. $M_t = \min\{f(X_s) : s = 1 \dots t\}$. and analogously let $\{\tilde{M}_t\}$ be the record sequence produced by PRS. Given a contraction factor $c > 1$ and a precision threshold $\rho \in (0, 1)$, let $m = \frac{|\ln \rho|}{\ln c}$. For each positive integer $C < \frac{c^m}{3m}$ there exists a positive integer t_C, depending only on C, such that for each $t > t_C$:*

$$E(M_t) \leq E(\tilde{M}_{t_C}).$$

The above theorem says, that one can choose the shrinking factor and the precision constance such that eventually ARS will require less steps than PRS to attain an approximation of the solution which is at leat of the sam quality.

More interesting properties of ARS, comparisons with other algorithms as well as some its applications can be found in [1].

Hybrid and Multistart algorithms

Hybrid algorithms being still stochastic take advantage of some deterministic methods used for local optimization and this sometimes speeds up the convergence. Let $\varphi : A \longrightarrow A$ be such a method and assume that $\varphi(f(x)) \leq f(x)$ for all $x \in A$. We assume that $A \subset \mathbb{R}^n$ is a compact set. Consider Borel probabilistic measures μ_0, ν on the set A.

Algorithm

0 Set $t = 0$. Generate a point x_0 from the distribution μ_0 on A.

1 Given point x_t, generate $y_t \in A$ according to the distribution ν.

2 Apply φ to y_t.

3 If $f(\varphi(y_t)) < f(x_t)$, then $x_{t+1} = \varphi(y_t)$.

3 Increase $t := t + 1$ and go to Step 1.

More general is Multistart algorithm.

Let M be the set of all Borel probabilistic measures on A. We consider the weak topology on M.

Let $\mu_0 \in M$ and let k, m be natural numbers. Let Φ denote a set of local methods, let $N \subset M$ be compact and let N_0 be a closed subset of N, such that for any $\nu \in N_0$, $\nu(G) > 0$ for any open neighborhood G of the set $A^\star$.

Algorithm

0 Let $t = 0$. Choose an initial population, i.e. a simple sample of points from A distributed according to μ_0:

$$x = (x^1, \ldots, x^m) \in A^m.$$

1 Given t-th population $x = (x^1, \ldots, x^m) \in A^m$ generate independently k points $y^i \in A$ according to a distribution $\nu^{t_i} \in N$ each, $i = 1, \ldots, k$. Let $y = (y^1, \ldots, y^k) \in A^k$.

2 Apply $\varphi^{t_i} \in \Phi$ to x^i , $i = 1, \ldots, m$.

3 Sort the sequence $(\varphi^{t_i}(x^1), \ldots, \varphi^{t_m}(x^m), y^1, \ldots, y^k)$ using f as a criterion to get

$$(\bar{x}^1, \ldots, \bar{x}^{m+k}) \text{ with } f(\bar{x}^1) \leq \cdots \leq f(\bar{x}^{m+k}).$$

4 Form the next population with the first m points

$$\bar{x} = (\bar{x}^1, \ldots, \bar{x}^m)$$

5 Increase $t := t + 1$, let $x = \bar{x}$ and go to Step 1.

There is a number of local methods available. For example, a classical one is the gradient method. It requires differentiability of the objective function f and still it is quite effective in finding local minima attained at interior points of the set A. If f is not a smooth function or its local minimum point is at the boundary of A, then more sophisticated methods can be used, see [6], [12] and survey paper [18]. The Algorithm above admits application of various methods at the same time or just one method with various parameters (like a step size or a number of steps taken). Obviously, the identity map is a local method.

Let $\widehat{f} : A^m \longrightarrow \mathbb{R}$ be defined as $\widehat{f}(x) = f(x^1)$. Let us note that $\widehat{A}^\star = A^\star \times A^{m-1}$ is the set of global minimums of $\widehat{f}$.

The following theorem gives sufficient conditions for almost sure convergence of the above algorithm to the set of solutions of the global minimization problem.

Theorem 4 ([9]) *Let $\{X_t : t = 1, 2, 3, \ldots\}$ be the sequence generated by the Algorithm, where $X_0 = (X_0^1, \ldots, X_0^m)$ is a random vector with distribution $(\mu_0)^m$. Let for each $t = 1, 2, 3, \ldots$, $Y_t = (Y_t^1, \ldots, Y_t^k)$ be independent random vectors, and independent of X_0, distributed according to $\nu^{t_1} \times \cdots \times \nu^{t_k}$ with $\nu^{t_i} \in N$. Assume that:*

(z1) *for any $c \in \mathbb{R}$ and $\nu \in N$, $\nu(l_c) = 0$.*

(z2) *There exists t_0 such that for any $t \geq 1$ there is $0 \leq s \leq t_0$ and some $1 \leq j \leq k$ with $\nu^{(+s)_j} \in N_0$.*

Then,

$$\text{Prob}\,(X_t \longrightarrow \widehat{A}^\star, \ \text{as} \ \ t \longrightarrow \infty) = 1. \tag{1}$$

3. Simulated Annealing

Simulated annealing originates from the analogy between the physical annealing process and the problem of finding minimal argument for discrete minimization problems. The physical annealing process is known in condensed matter physics as a thermal process for obtaining low energy states of a solid in a heat bath. In optimization Simulated Algorithm (SA) contains a probabilistic mechanism to escape a global minimum.

Let as above $f : A \longrightarrow \mathbb{R}$ be a continuous function, where $A \subset \mathbb{R}^n$ is a compact set. Let $B \subset \mathbb{R}^d$. Let $M > 0$ and $[0, M] \ni \beta_t$ satisfies $\lim\limits_{t \longrightarrow \infty} \beta_t = 0$. We are given Borel measures μ_0 on A and ν B and a measurable operator $Q\colon A \times B \longrightarrow A$.

Algorithm

0 Set $t = 0$. Generate a point x_0 from the distribution μ_0 on A.

1 Given x_t generate point $z \in B$ according to distribution ν.

2 If $f(Q(x_t, z)) \leq f(x_t)$, then $x_{t+1} = f(Q(x_t, z))$.

3 If $f(Q(x_t, z)) > f(x_t)$, then generate point $r \in (0, 1)$ according to the uniform distribution. If

$$r \leq \exp\left(- \frac{f(Q(x_t, z)) - f(x_t)}{\beta_t} \right),$$

$x_{t+1} = f(Q(x_t, z))$.

4 Increase $t := t + 1$ and go to step 2.

The essence of the Algorithm is to create an opportunity to substitute the approximation with the next approximation even if the new one is worse, the chance of such action decreases with time, but can be zoomed, where the approximation is only slightly better than the the new one.

Let X_t be random vectors which realizations are generated by SA. The following theorem on convergence might be found in [15], see also [10]. For a similar result with a different proof we refer to [4] and [19]

Theorem 5 *Assume that for all $x \in A$, $\nu(D_{f\circ Q}(x)) = 0$, where $D_{f\circ Q}(x)$ consists of $z \in B$ such, that $f \circ Q$ is not continuous at point (x, z). Assume also, that for all $x \in A \setminus A^\star$,*

$$\nu(\{z \in B\colon f(Q(x, z)) < f(x)\}) > 0. \tag{2}$$

Then,

$$\forall \varepsilon > 0 \ Prob(dist(X_t, A^*) < \varepsilon) \overset{t \longrightarrow \infty}{\longrightarrow} 1 \quad and \quad E(f(X_t)) \overset{t \longrightarrow \infty}{\longrightarrow} \min_A f.$$

4. Evaluating stochastic algorithms

In the previous Sections we have presented just few stochastic optimization algorithms. Our choice depended on two criteria, simplicity of presentation and existing theoretical results on the convergence. Yet, there exist hundreds of another stochastic algorithms:

Evolutionary Algorithms (EA) including
Genetic Algorithms (GA),
Particle Swarm Optimization (PSO),
Ant Colony Optimization (ACO),
Artificial Bee Colony (ABC),
Grenade Explosion Method (GEM),
Covariance Matrix Adaptation (CMA),
Markov Chain Monte Carlo (MCMC),
Differential Evolution (DE),

and more. They are still being improved and the new algorithms are still being invented. Any of them has a number of particular versions. Also, the majority of algorithms depends on some parameters, finite dimensional (like ARS above did on shrinking and precision constants), and infinite dimensional (like Multistart above did on the choice of a measure and a local method), and the suitable choice of them may essentially results in good or poor performance of the algorithm. And what is important, the concept of good or pure performance is not always clear but depends on the specific situation in which the algorithm is used. In fact, in online optimization encountered, for example, in robot localization, load balancing, services composition for business processes or updating information we would prefer short time criterion than accuracy. For example, in [3] a problem of fast short time interval prediction during aircraft landing is discussed, when optimization process has to be as quick as possible. On the other hand, in design optimization, creating long-term schedules or data mining, when optimization processes would usually be carried out only once in a long time, the accuracy and certainty of the result is then crucial.

According to [2] desirable properties of an optimization algorithm from the Machine Learning perspective are: good generalization, scalability to large problems, good performance in practice in terms of execution times and memory requirements, simple and easy implementation of algorithm, exploitation of problem structure fast convergence to an approximate solution of model, robustness and numerical stability for class of machine learning models attempted, theoretically known convergence and complexity.

Therefore there is need to make it possible to asses quality of a given algorithm and to compare it with the others according to some of the above factors. Hence, numerous empirical studies have attempted to show the effectiveness of some particular optimization algorithms. A common practise is to run the algorithms on some already known test (benchmark) functions or on a collections of such functions known as suites or testbeds and compare the results. A typical paper presenting a new, just developed, algorithm contains comparative experimental result taking

into accounts some already known algorithms and suitable chosen test suite. No need to add that such comparison in many cases could favor our algorithm. Generally, empirical and experimental approaches to comparing algorithms have many disadvantages, especially when the algorithms are designed to be robust, general purpose optimization tools. One obvious danger with empirically evaluating algorithms is that the resulting conclusions depend as much on what problems are used for testing as they do on the algorithms that are being compared. This can have the side effect that algorithms are designed and tuned to perform well on a particular test suite; the resulting specialization may or may not translate into improved performance on other problems or applications.

So, there are attempts to workout the methodology to specify evaluation goals, comparison criteria and to construct test suites and to examine the role of test suites as they have been used to evaluate optimization algorithms, see for example [16].

COCO (COmparing Continuous Optimisers) is a platform for systematic and sound comparisons of real-parameter global optimisers. COCO provides benchmark function testbeds and tools for processing and visualizing data generated by one or several optimizers. The COCO platform has been used for the Black-Box-Optimization-Benchmarking (BBOB) workshops that took place during the GECCO conference in 2009, 2010, 2012, and 2013. The next edition is going to take place as a special session in May 2015 during the next IEEE Congress on Evolutionary Computation (CEC'2015) in Sendai, Japan. The COCO source code is available at the downloads page at http://coco.gforge.inria.fr.

On the other hand, mathematical theory concerning stochastic optimization algorithms are quite limited. As we have seen in the above two Sections there are some results on convergence, see also Section 5, and almost no results on the convergence rate. There are also results that partially justify a particular algorithm, but most of them are far from mathematical accuracy. However, we have to admit, that from practical point of view the problem of convergence may be not a crucial matter. It seems that more important would be to know how fast a particular algorithm converges to the optimal solution. Unfortunately, this aspect from the theoretical point of view is even worse examined. Still, we believe that the tools used for proving convergence, mentioned roughly in the Section following, could be also used in estimation of convergence rate.

5. Markov type algorithms

The algorithms presented in Section 2 and Section 3, and in fact, many more algorithms are instances of a general stochastic algorithm, which mathematical model is described by the following Markov type recurrent formula:

$$X_t = T_t(X_{t-1}, Y_t), \text{ for } t = 1, 2, 3 \ldots. \tag{3}$$

Here X_t, for $t \geq 0$ denote random variables corresponding to successive outcomes of the algorithm and Y_t are vectors responsible for randomness in steps $1, 2, 3, \ldots$.

T_t define the mechanism of the algorithm itself.

More formally, we are given two sets $A \subset \mathbb{R}^n$, $B \subset \mathbb{R}^d$, measurable operators $T_t : A \times B \longrightarrow A$, for $t = 1, 2, 3 \ldots$, a probability space $(\Omega, \Sigma, \mathrm{Prob})$, random vector $X_0 : \Omega \longrightarrow A$ distributed according to some measure μ_0, and sequence of random vectors $Y_t : \Omega \longrightarrow B$ distributed according to some measures ν_t. We assume that $X_0, Y_1, Y_2, Y_3, \ldots$ are independent. Then equation (3) determines random vectors $X_1, X_2, X_3, \ldots$.

For example, if we put $A = B = [0,1]^n$, take μ_0, ν_t for all t as the Lebesgue measure and:

$$T_t(x,y) = \begin{cases} x, & \text{if } f(y) \geq f(x) \\ y, & \text{if } f(y) < f(x), \end{cases}$$

then we get PRS.

To get the hybrid algorithm in the form stated in Section 2 we have $\nu_t = \nu$ and specify T as:

$$T_t(x,y) = \begin{cases} x, & \text{if } f(\varphi(y)) \geq f(x) \\ \varphi(y), & \text{if } f(\varphi(y)) < f(x), \end{cases}$$

For the multistart algorithm define $T_t : A^m \times A^k \longrightarrow A^m$ as:

$$T_t(x,y) = \bar{x},$$

where $\bar{x}$ was defined in step 4 of the Algorithm. Instead of measures μ_0 and ν_t we use the product measures μ_0^m and ν^k respectively. Let us note, that the construction of $\bar{x}$ may depend on t as in any step we can choose different local methods. Hence, in this case T_t essentially depends on t.

In the SA case we encounter two random mechanism. Define T_t as:

$$T_t(x,z,r) = \begin{cases} Q(x,z), \text{ if } f(Q(x,z)) \leq f(x), \\ Q(x,z), \text{ if } f(Q(x,z)) > f(x) \wedge r \leq \exp\left(-\frac{f(Q(x_t,z))-f(x_t)}{\beta_t}\right), \\ x, \text{ otherwise.} \end{cases}$$

So, $T_t : A \times (B \times [01,1]) \longrightarrow A$. $\nu_t = \nu \times \lambda$ is the product measure, where λ is the Lebesgue measure on the the unit interval.

Tarłowski in [15] proved that ARS and evolutionary strategy $ES(\mu/\varrho + \lambda)$ are instances of (3), see also. [10]. Similar result for PSO can be found in [11] and for GEM in [13]. Actually, the vast part of modern stochastic algorithm seems to be of the form (3). We can then await some general results concerning system (3) which imply particular results for specific algorithms. We present one of such results and it implies convergence of PRS, Theorem 1 and multistart, Theorem 4. We refer to [15] for more, even stronger, results concerning system (3) and corresponding proofs of convergence for particular instances including ARS, GEM and $ES(\mu/\varrho + \lambda)$, see also [10].

Denote $M(A)$ and $M(B)$ the sets of all probability Borel measures on A and B respectively. They are topological spaces with the weak topology. By $\mathcal{T}$ we denote the space of the all measurble operators $T : A \times B \longrightarrow A$ equipped with the topology of uniform convergence.

Theorem 6 ([9]) *Assume that A is a compact set. Let $U \subset \mathcal{T} \times M(B)$ be a compact set. Assume that for any $u = (T, \nu) \in U$:*

(A) *For any $x_0 \in A$, there is a Borel set $D_T(x_0) \subset B$ with $\nu(D_T(x_0)) = 0$, such that T is continuous in (x_0, y), for any $y \notin D_T(x_0)$.*

(B) *For any $x \in A^\star$ and $y \in B$, $T(x, y) \in A^\star$.*

(C1) *For any $x \in A \setminus A^\star$:*

$$\int_B f(T(x,y))\,\nu(dy) \leq f(x). \tag{4}$$

(C2) *There is a closed set $U_0 \subset U$ such that for any $(T, \nu) \in U_0$ and $x \in A \setminus A^\star$:*

$$\int_B f(T(x,y))\,\nu(dy) < f(x). \tag{5}$$

Let $\{u_t = (T_t, \nu_t) : t \geq 1\} \subset U$ satisfy the following:

(U0) *There is $t_0 \geq 1$ such that for any $t \geq 1$ there is $s \leq t_0$ with $u_{t+s} \in U_0$.*

Then, for every $\varepsilon > 0$:

$$\lim_{t\to\infty} \mathrm{Prob}\left(\mathrm{dist}(X_t, A^\star) < \varepsilon\right) = 1. \tag{6}$$

Assume additionally

(D) *For any $t \geq 1$, $x \in A$ and $y \in B$: $f(T_t(x, y)) \leq f(x)$.*

Then,

$$\mathrm{Prob}\left(X_t \longrightarrow A^\star, \ \text{as} \ \ t \longrightarrow \infty\right) = 1. \tag{7}$$

One can release the assumption of compactness of the set A assuming **(D)** and:

(E) There exists $r > \min f$ such that set $A_r := \{x \in A : f(x) \leq r\}$ is compact and $\mathrm{supp}\,\mu_0 \subset A_r$.

In fact, by **(D)** $T_t(A_r \times B) \subset A_r$. Clearly, μ_0 is a probability measure on A_r and $A^\star \subset A_r$. Hence, we may apply Theorem 6 to set A_r.

The proof of the above Theorem is done in [9] and its main idea is to consider a nonautonomous dynamical system on the space of measures $M(A)$ given by the Foias operators identified by pairs (T_t, ν_t) from U. The orbit of μ_0 coincides with the sequence of successive distributions μ_0^t of X_t. Our assumptions guarantee existence of a suitable Liapunov function, which by a modification of the Liapunov Theorem implies attractiveness of the set consisting of the all probability measures supported on $A^\star$, and then stochastic convergence (6). Almost sure convergence (7) follows from a simple observation that stochastic convergence together with monotocinity imply almost sure convergence.

6. Stochastic algorithms in R

There are a lot of computer packages designated to run optimization algorithms, stochastic global optimization algorithm including. In particular, environment R makes it possible to access a variety of packages designated for stochastic optimization. Among them are:

- GenSA – Generalized Simulated Annealing,
- DEoptim – Differential Evolutionary Optimization,
- soma – Self-Organising Migrating Algorithm,
- rgenoud – GENetic Optimization Using Derivatives,
- cmaes – Covariance Matrix Adapting Evolutionary Strategy,
- mco – Multi Criteria Optimisation,
- mcga – Machine Coded Genetic Algorithm,
- emoa – Evolutionary Multiobjective Optimisation Algorithms,
- soobench – Single Objective Optimization Benchmark Functions.

7. References

[1] Appel M.J., Labarre R., Radulovic D., *On Accelerated Random Search*, SIAM J. Optim. 14(3), 2003, pp. 708–731.

[2] Bennett K.P., Parrado-Hern E., *The Interplay of Optimization and Machine Learning Research*, Journal of Machine Learning Research 7, 2006, pp. 1265–1281.

[3] Bialy J., Ciecko A., Cwiklak J., Grzegorzewski M., Koscielniak P., Ombach J., Oszczak S., *Aircraft Landing System Utilizing a GPS Receiver with Position Prediction Functionality*, Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011), Portland, OR, September 2011, pp. 457–467.

[4] Locatelli M., *Convergence of a Simulated Annealing Algorithm for Continuous Global Optimization*, J. Global Optim. 18, 2000, pp. 219–233.

[5] Luke S., *Essentials of Metaheuristics*, Lulu.com, 2011.

[6] Nocedal J., Wright S. J., *Numerical optimization*, Springer Series in Operations Research, Springer-Verlag, New York, 1999.

[7] Ombach J., *A Proof of Convergence of General Stochastic Search for Global Minimum*, Journal of Difference Equations and Applications 13, 2007, pp. 795–802.

[8] Ombach J., *Stability of evolutionary algorithms*, Journal Math Anal Appl. 342, 2008, pp. 326–333.

[9] Ombach J., Tarłowski D., *Nonautonomous Stochastic Search in Global Optimization*, Journal in Nonlinear Sci. 22, 2012, pp. 169–185.

[10] Ombach J., Tarłowski D., *Stochastyczne algorytmy optymalizacji z perspektywy układów dynamicznych*, in Polish, preprint.

[11] Radwański M., *Convergence of nonautonomous evolutionary algorithm*, Universitatis Iagellonicae Acta Mathematica 45, 2007, pp. 197–206.

[12] Robert Ch., Casella G., *Monte Carlo Statistical Methods*. Springer Heidelberg 2004.

[13] Tarłowski D., *Sufficient conditions for the convergence of non-autonomous stochastic search for a global minimum*, UIAM, 2011, pp. 73–83.

[14] Tarłowski D., *Nonautonomous stochastic search for global minimum in continuous optimization*, Journal Math Anal Appl. 412, 2014, pp. 631–645.

[15] Tarłowski D., *Nonautonomous Dynamical Systems in Stochastic Global Optimization*, Ph.D. thesis, Department of Mathematics, Jagiellonian University 2014.

[16] Whitley D., Mathias K., Rana S., Dzubera J., *Evaluating Evolutionary Algorithms*, preprint.

[17] Weise T., *Global Optimization Algorithms – Theory and Application*, http://www.it-weise.de/

[18] Wright M., *The interior-point revolution in optimization: History, recent developments, and lasting consequences*, Bull. Amer. Math. Soc. 42, 2005, pp. 39–56.

[19] Yang R.L., *Convergence of the Simulated Annealing Algorithm for Continuous Global Optimization*, Journal of Optimization Theory and Applications 104, 2000, pp. 691–716.