

# Randomization in Robustness, Estimation, and Optimization



B. Polyak and P. Shcherbakov

**Abstract** This is an attempt to discuss the following question: When is a random choice better than a deterministic one? That is, if we have an original deterministic setup, is it wise to exploit randomization methods for its solution? There exist numerous situations where the positive answer is obvious; e.g., stochastic strategies in games, randomization in experiment design, randomization of inputs in identification. Another type of problems where such approach works successfully relates to treating uncertainty, see Tempo R., Calafiore G., Dabbene F., “Randomized algorithms for analysis and control of uncertain systems,” Springer, New York, 2013. We will try to focus on several research directions including optimization problems with no uncertainty and compare known deterministic methods with their stochastic counterparts such as random descent, various versions of Monte Carlo etc., for convex and global optimization. We survey some recent results in the field and ascertain that the situation can be very different.

## 1 Introduction

The use of a random mechanism to solve a problem in a deterministic setup is very common not only in mathematics but much beyond formal framework. One can remember that random decisions were performed in ancient times, and the procedure of drawing lots was very common. Moreover, political events such as election of governing officers in Athens were randomized. Nowadays, elements of randomization are often exploited in sport competitions to equalize the chances of the participants. Very important role of random mixing in medical and biological experiments is of no doubt.

---

B. Polyak (✉) · P. Shcherbakov  
Institute of Control Science, Russian Academy of Sciences, Moscow, Russia  
e-mail: [polyak@ipu.ru](mailto:polyak@ipu.ru)

P. Shcherbakov  
e-mail: [cavour118@mail.ru](mailto:cavour118@mail.ru)

© Springer Nature Switzerland AG 2018  
T. Başar (ed.), *Uncertainty in Complex Networked Systems*,  
Systems & Control: Foundations & Applications,  
[https://doi.org/10.1007/978-3-030-04630-9\\_5](https://doi.org/10.1007/978-3-030-04630-9_5)

Probably one of the first application of stochastic approach in mathematics is the theory of mixed strategies in zero-sum games by John von Neumann. Here the role of randomization is to make secret a strategy of the player against the competitor. Approximately at the same time Fisher proposed to apply mixed strategies in experiment design; here their role was different. A real breakthrough was the invention of the Monte Carlo methods by Ulam, Metropolis, von Neumann and Teller [43], and the ideas of random sampling became very popular in modeling and numerical analysis. Thus randomization methods found numerous applications in various fields of research; to survey all of them does not seem to be realistic. In this chapter we restrict ourselves to some problems related to estimation, robustness, and continuous optimization. The typical question to be analysed is as follows. Given a deterministic problem (say, unconstrained smooth optimization), how randomization ideas can be exploited for its solution and are randomized methods true competitors with deterministic ones? We will see that situation differs in various domains of interest.

The role of Roberto Tempo in progress of this approach can not be overestimated. His research since 2000 was mostly dedicated to randomization methods in control, robustness, and optimization, see the monograph [77]. In the present chapter, we continue this line of research, but also we address the directions which have little intersections with [77] as well with other monographs and surveys on randomization [14, 28, 29].

Due to the wide spectrum of the problems under consideration, we are forced to provide really brief presentation of the problems; the references do not pretend to be complete. However we have tried to emphasize the pioneering works and surveying publications.

## 2 Uncertainty and Robustness

Mathematical models for systems and control are often unsatisfactory due to the incompleteness of the parameter data. For instance, the ideas of off-line optimal control can only be applied to real systems if all the parameters, exogenous perturbations, state equations, etc., are known precisely. Moreover, feedback control also requires a detailed information which is not available in most cases. For example, to drive a car with four-wheel control, the controller should be aware of the total weight, location of the center of gravity, weather conditions and highway properties as well as many other data which may not be known. In that respect, even such a relatively simple real-life system can be considered a *complex* one; in such circumstances, control under uncertainty is a highly important issue.

In this section we consider the *parametric uncertainty*; other types of uncertainty can be treated within more general models of robustness.

There are numerous tools to check robustness under parametric uncertainty; below we focus on randomized methods. This line of research goes back to pioneering papers by Stengel and Ray [74]. Within this approach, the uncertain parameters are assumed to have random rather than deterministic nature; for instance, they are

assumed to be uniformly distributed over the respective intervals of uncertainty. Next, an acceptable tolerance  $\varepsilon$ , say  $\varepsilon = 0.01$  is specified, and a check is performed if the resulting random family (of polynomials, matrices, transfer functions) is stable with probability no less than  $(1 - \varepsilon)$ ; see [77] for a comprehensive exposition of such a *randomized approach to robustness*.

In many of the NP-hard robustness problems, such a reformulation often leads to exact or approximate solutions. Moreover, the randomized approach has several attractive properties even in the situations where the deterministic solution is available. Indeed, the deterministic statements of robustness problems are minimax, hence, the answer is dictated by the “worst” element in the family, whereas these critical values of the uncertain parameters are rather unlikely to occur. Therefore, by neglecting a small risk of violation of the desired property (say, stability), the admissible domains of variation of the parameters may be considerably extended. This effect is known as the *probabilistic enhancement of robustness margins*; it is particularly tangible for the large number of the parameters. Another attractive property of the randomized approach is its low computational complexity which only slowly grows with increase in the number of uncertain parameters.

We illustrate some of these concepts and effects.

## 2.1 Volume of Violation and Approximate Feasibility

We consider robustness problems for systems described in terms of a design vector  $x \in X \subseteq \mathbb{R}^n$  and a real uncertain parameter vector  $q \in Q \subset \mathbb{R}^\ell$ , where  $Q$  is a box. For such systems, the objective is to select  $x \in X$  such that a given continuous performance specification

$$f(x, q) \leq 0 \tag{1}$$

is satisfied for all  $q \in Q$ . When such a design vector  $x$  exists, the triple  $(f, X, Q)$  is said to be *robustly feasible*.

In a number of situations, robust feasibility of  $f(x, q) \leq 0$  is guaranteed if and only if  $f(x, q^i) \leq 0$  for each of the vertices  $q^i$  of the  $\ell$ -dimensional box  $Q$ , and we use the term *vertexization*. A typical example of a vertexization is the quadratic stability problem for the system with state space matrix  $A(q) = A_0 + \sum_{i=1}^{\ell} A_i q_i$ , where  $A_i \in \mathbb{R}^{n \times n}$  are fixed and known, and the uncertainty parameter vector  $q \in Q$ . The goal is to find a symmetric candidate Lyapunov matrix  $P = P(x)$  with entries  $x_i \in \mathbb{R}$  viewed as the design variables, such that  $P(x) \succ 0$  and the linear matrix inequality (LMI)  $A^\top(q)P(x) + P(x)A(q) \prec 0$  holds for all  $q \in Q$  (throughout the text, the signs  $\succ$  and  $\prec$  denote the positive and negative sign-definiteness of a symmetric matrix). Hence, with

$$f(x, q) = \lambda_{\max}\left(A^\top(q)P(x) + P(x)A(q)\right),$$

this strict feasibility design problem in  $x$  is reducible to the vertices  $q^i$  of  $Q$ . That is, the satisfaction of the Lyapunov inequality above for all  $q \in Q$  is equivalent to  $A^\top(q^i)P(x) + P(x)A(q^i) < 0$  for  $i = 1, 2, \dots, N$ . However, since  $N = 2^\ell$ , we see that the computational task can easily get out of hand. For example, with five states and ten uncertain parameters, the resulting LMI is of size greater than  $5000 \times 5000$ .

As an alternative to the computational burden associated with vertexization, it is often possible to introduce an *overbounding function* in such a way as to enable *convex programming* in order to test for robust feasibility; also, see Sect. 4 for a different approach to solving the feasibility problem for LMIs. Note also that a reduction to checking the vertices is rather an exception and is considered here for illustrative purposes, while the *overbounding techniques* may be applied to much broader classes of systems.

Specifically, given  $x$ , introduce the associated *violation set*

$$Q_{bad}(x) \doteq \{q \in Q : f(x, q) > 0\}$$

and estimate from above its volume. Equivalently, assuming that the uncertainty vector  $q$  is random, uniformly distributed over  $Q$ , we estimate from above the probability of violation for the performance specification.

In [4, 5], a computationally modest method for finding such overbounding functions is proposed and numerical examples are presented.

More delicate constructions are also described in [4, 5], where the notion of approximate feasibility is introduced. Namely, the triple  $(f, X, Q)$  is said to be *approximately feasible* if the following condition holds: Given any  $\varepsilon > 0$ , there exists some  $x^\varepsilon \in X$  such that

$$\mathbf{Vol}(\{q \in Q : f(x^\varepsilon, q) > 0\}) < \varepsilon,$$

where  $\mathbf{Vol}(\cdot)$  stands for the volume of a set. For such  $\varepsilon$ ,  $x^\varepsilon$  is called an  $\varepsilon$ -*approximate solver*. So, instead of guaranteeing satisfaction of  $f(x, q) \leq 0$  for all  $q \in Q$ , we seek solution vectors  $x$  with associated violation set having volume less than any arbitrarily small prespecified level  $\varepsilon > 0$ .

We present a formal result on approximate feasibility in general terms; the details can be found in [4, 5]. First, we consider so-called *homogenizable* in  $x$  functions  $f(x, q)$  and use their homogenized versions denoted by  $f^+(x_0, x, q)$ . In [4, 5] this requirement was shown to be not very much restrictive, covering quite a large class of functions. Next, the notion of *approximate feasibility indicator* (AFI) is introduced; in a sense, it is a convex generalization of the classical indicator function. For instance, a “natural” type of AFI is the exponential one,  $\phi(\zeta) = e^\zeta$ .

In the theorem to follow, the approximate feasibility indicator  $\phi(\zeta)$  is used with argument  $\zeta = f^+(x_0, x, q)$  in the determination of approximate feasibility.

**Theorem 1** ([4, 5]) *Given the continuous homogenizable performance specification function  $f(x, q)$ ,  $X = \mathbb{R}^n$  and an approximate feasibility indicator  $\phi(\cdot)$ , define*

$$\Phi(x_0, x) \doteq \int_Q \phi(f^+(x_0, x, q))dq$$

and

$$\Phi^* \doteq \inf_{x_0 > 0, x} \Phi(x_0, x).$$

Then the following holds:

- (i)  $\Phi^* = 0$  implies approximate feasibility of  $(f, X, Q)$ ;
- (ii) For any  $x_0 > 0$  and  $x \in \mathbb{R}^n$ ,

$$\mathbf{Vol} \left( Q_{bad} \left( \frac{x}{x_0} \right) \right) \leq \Phi(x_0, x).$$

A similar idea of overbounding was presented in [6]. Multivariable polynomials  $f(x)$  with parameter vector  $x$  restricted to a hypercube  $X \in \mathbb{R}^n$  were considered, and the objective was to check the robust positivity of  $f(x)$ , i.e., to determine if  $f(x) > 0$  for all  $x \in X$ . Again, instead of solving the original NP-hard problem, the authors proposed a straightforward procedure for the computation of an upper bound on the volume of violation by computing a respective *dilation integral* that depends on the degree  $k$  of a certain auxiliary polynomial, followed by a convex minimization in one scalar parameter. By increasing the degree  $k$ , the authors obtain a sequence of upper bounds  $\varepsilon_k$  which are shown to be “sharp” in the sense that they converge to zero whenever the positivity requirement is satisfied. Notably, that this dilation integral method applies to a general *polynomial dependence* on the variables.

## 2.2 Probabilistic Predictor

In the discussion above, the stochastic nature of the uncertain parameters was somewhat hidden; we just evaluated the bad portion of the uncertainty box. Assume now that the originally deterministic parameters are randomly distributed over the given uncertainty set  $Q$ . Then it seems natural to sample the uncertainty set  $Q$  and arrive at conclusions on the probability of robustness. In the control-related literature, these ideas have been first formulated in [74]; also see [3].

Together with numerous advantages of this approach, it also suffers serious drawbacks. First, it is usually desired to have any closed-form estimates of the robustness margin, rather than to rely on the results of simulations; moreover, in practical applications, such a simulation is often prohibited. Next, the sample size that guarantees high reliability of the result may happen to be rather large [76, 77], hence, simulations may be very time-consuming. On top of that, sampling in accordance with one or another distribution over a given set may be highly nontrivial [30, 57]. Finally, the results of Monte Carlo simulation heavily depend on the probabilistic distribution

adopted and may lead to overly optimistic estimates of the robustness margin; the correct choice of the distribution is a nontrivial problem [2].

In this section, assuming the uniform distribution of the uncertain parameters over  $q \in \gamma Q$ , where  $Q \subset \mathbb{R}^\ell$  is the uncertainty set and  $\gamma \in \mathbb{R}$  is the scaling factor, we characterize the probability of stability of a system and evaluate the *probabilistic stability margin*

$$\gamma_{\max} := \sup\{\gamma : \text{Prob}\{\text{the system is stable for } q \in \gamma Q\} \geq p\}, \quad p \in (0, 1),$$

where  $\text{Prob}(\cdot)$  denotes the probability of an event.

Without getting deep into the details, we describe the idea of the probabilistic approach to robustness as applied to polynomial families.

Since the early 1990s, numerous *graphical tests* for robust stability proved themselves to be efficient; these are based on the famous zero exclusion principle, which is formulated next. Consider the family of polynomials  $p(s, q)$  which depend on the vector  $q$  of uncertain parameters confined to the connected set  $Q \subset \mathbb{R}^\ell$ . For a fixed  $s = j\omega$ , the set

$$V(\omega) \doteq \{p(j\omega, q) : q \in Q\}$$

is referred to as the *value set* of the family  $p(s, q)$ ; it is the 2D image of  $Q$  under the mapping  $p(j\omega, \cdot)$ . Let the polynomial  $p(s, q^0)$  be stable for some  $q^0 \in Q$ ; then, for robust stability, the following condition is necessary and sufficient:

$$0 \notin V(\omega) \quad \text{for all } \omega \in [0, \infty). \quad (2)$$

To exploit this result, one has to efficiently construct the set  $V(\omega)$  and check condition (2). This is doable in a number of simple cases; however, for more or less involved dependence of  $p(s, q)$  on  $q$ , this approach cannot be applied, since no closed-form description of the boundary of the value set is available, and checking condition (2) is complicated by the nonconvexity of  $V(\omega)$ .

Taking the probabilistic point of view and letting  $q$  be random, uniformly distributed over  $Q$ , we consider the two-dimensional random variable

$$z_\omega = [\text{Re}p(j\omega, q); \text{Im}p(j\omega, q)]$$

and construct its confidence domain

$$V_{1-\varepsilon}(\omega) : \text{Prob}\{z_\omega \in V_{1-\varepsilon}(\omega)\} \geq 1 - \varepsilon, \quad \varepsilon > 0 \text{ is small.}$$

This set is referred to as a  $100(1 - \varepsilon)\%$  *probabilistic predictor* of the value set  $V(\omega)$ . The condition (2) now has to be checked for the predictor, rather than for the value set, hence, evaluating the probability of stability of the uncertain polynomial family.

Often, the construction of the predictor can be accomplished via using the central limiting behavior of the random vector  $z_\omega$ . Indeed, if  $p(s, q)$  depends affinely on  $q$ , and the  $q_i$ s are mutually independent, the random vector  $z_\omega$  is represented by the

sum of independent random vectors, and if the number  $\ell$  of the parameters is large enough, then, under the general assumptions on  $p_i(s)$  it is well described by the two-dimensional Gaussian random vector with mean  $\bar{z}_\omega = \mathbf{E}z_\omega$  and the covariance matrix  $S = \mathbf{Cov} z_\omega$ . Therefore,  $V(\omega)$  may be approximated by the confidence ellipse

$$\mathcal{E}_\nu(\omega) \doteq \{z \in \mathbb{R}^2: (z - \bar{z}_\omega)^\top S^{-1}(z - \bar{z}_\omega) \leq \nu^2\},$$

where  $\nu$  specifies the confidence level. In other words, if  $\mathbf{p}_\nu$  is the associated confidence probability, then for a given  $\omega$  we have

$$\text{Prob}\{p(j\omega, q) \in \mathcal{E}_\nu(\omega)\} \approx \mathbf{p}_\nu = 1 - e^{-\nu^2/2}.$$

In a number of situations, it is possible to obtain a precise nonasymptotic distribution of the random vector  $z_\omega$  and, respectively, a precise description of the probabilistic predictor.

We illustrate these ideas via the problem of robust stability of uncertain delay systems; i.e., those described by uncertain quasipolynomials, see [58]. In this case, the generic value set has a very complicated geometry; application of the zero exclusion principle is hardly possible, and we lean on the probabilistic approach.

Consider the delay system specified by the characteristic quasipolynomial

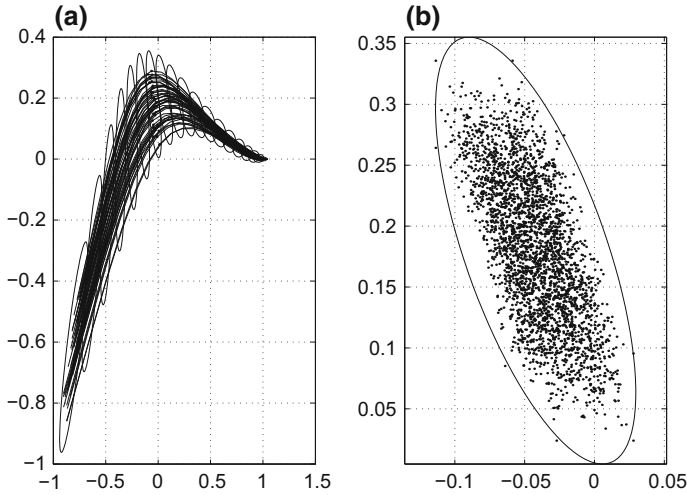
$$h(s, a, \tau) = a_0 + a_1 s + s^2 + 2s e^{-\tau_1 s} + e^{-\tau_2 s}, \quad (3)$$

$$|a_0| \leq \gamma, \quad |a_1| \leq \gamma, \quad |1 - \tau_1| \leq \gamma, \quad |2 - \tau_2| \leq \gamma.$$

Here, both the coefficients and the delays are subject to interval uncertainty. The nominal system  $h(s) = s^2 + 2s e^{-s} + e^{-2s}$  is stable,  $\max_k \text{Res}_k = -0.3181$ , where  $s_k$  are the roots of the quasipolynomial  $h(s)$  (the roots of  $h(s)$  are the values of the Lambert function  $W(x)e^{W(x)} = x$  at the point  $x = -1$ ). For this system, the value of the radius of robustness cannot be found exactly, but the estimate  $0.01 < \gamma_{\max} < 0.05$  is known from the literature. For the confidence level  $\nu = 3$ , the probabilistic approach gives  $\gamma_\nu = 0.0275$ , so that it fits well the deterministic estimate.

To illustrate, for a set of frequencies in  $0 \leq \omega \leq 2$ , Fig. 1a depicts the confidence ellipses  $\mathcal{E}_\nu(\omega)$ ,  $\nu = 3$ , for the uncertainty range  $\gamma = 0.0275$ . Also, presented are the frequency responses  $h(j\omega, q)$  for a number of sampled values of the uncertainty  $q = (a_0, a_1, \delta\tau_1, \delta\tau_2)$  in the box  $|q_i| \leq \gamma$ . The curves are seen to remain inside the ‘‘corridor’’ defined by the confidence ellipses. Figure 1b depicts the confidence ellipse  $\mathcal{E}_\nu(\omega)$  for a ‘‘typical’’  $\omega = 1.3113$  together with sampled points  $h(j\omega, q)$ ; the predictor is seen to approximate nicely the value set.

Probabilistic robustness techniques can be effectively exploited for robust control design [12, 39, 53, 54, 61, 77, 78].



**Fig. 1** **a** The plot of  $h(j\omega, q)$  and confidence ellipses  $\mathcal{E}_\nu(\omega)$ ,  $\nu = 3$  for system (3). **b** Probabilistic predictor of the value set for  $\omega = 1.3113$

### 2.3 Probabilistic Enhancement of Robustness Margins

It is important to note that, even for the values of  $\mathbf{p}_\nu = \mathbf{p}$  close to unity, the ellipse  $\mathcal{E}_\nu(\omega)$  is often considerably smaller than the value set  $\mathbf{Vol}(\omega)$ . Let us make use of the probabilistic counterpart of the zero exclusion principle (the origin does not belong to  $\mathcal{E}_\nu(\omega)$  for all  $\omega$ ) and evaluate the *probabilistic stability margin* defined as

$$\gamma_p \doteq \sup\{\gamma : 0 \notin \mathcal{E}_\nu(\omega) \text{ for all } \omega \in [0, \infty)\}.$$

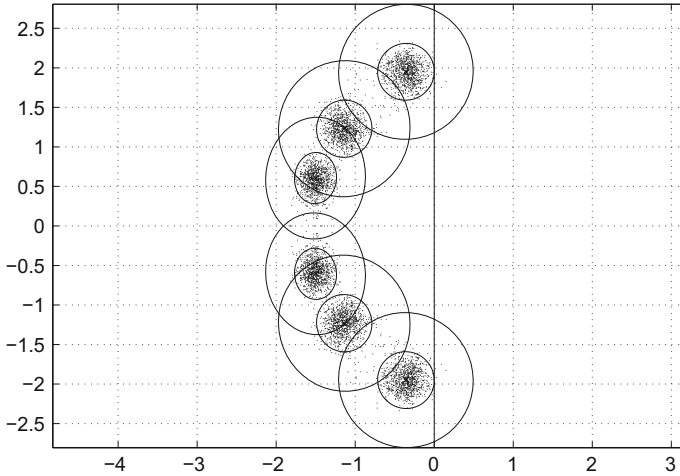
It then usually happens that  $\gamma_p \gg \gamma_{\max}$ , where  $\gamma_{\max}$  is the deterministic stability margin. Hence, the uncertainty range may be considerably enlarged at the expense of neglecting low-probability events. This phenomenon is referred to as *probabilistic enhancement of classical robustness margins* [40]. Moreover, in accordance with the central limit theorem, this enlargement gets bigger as the number of uncertainties grow, and it is this case which is most problematic for deterministic methods. At the same time, the computational burden of probabilistic methods does not depend on the dimension of the vector of uncertain parameters. Indeed, putting the precise description of the value set aside, we make use of an approximation of it, which is defined by the two-dimensional covariance matrix.

We illustrate use of the probabilistic approach to the assessment of such an enhancement via the case of matrix uncertainty. Specifically, let us consider the uncertain matrix family

$$A = A_0 + \Delta, \quad \Delta \in \gamma Q, \quad (4)$$

where  $A_0 \in \mathbb{R}^{n \times n}$  is a known, Hurwitz stable matrix and  $\Delta$  is its bounded perturbation confined to the ball in the Frobenius norm  $\gamma Q = \{\Delta \in \mathbb{R}^{n \times n} : \|\Delta\|_F \leq \gamma\}$ ; the goal





**Fig. 2** The pseudospectrum of  $A_0$ , its linear approximation, and the probabilistic predictor

is to estimate the robust stability margin of  $A_0$ . To this end, we provide an approximate description of the *pseudospectrum* of  $A$  (4), the set of the eigenvalues of  $A$  for all admissible values of the uncertainty  $\Delta$ .

For a generic case of simple complex eigenvalues  $\lambda = \lambda(A_0) \in \mathbb{C}$ , the perturbed eigenvalue  $\lambda(A_0 + \Delta)$  is well described by the linear approximation

$$\tilde{\lambda} = \lambda + Rq, \quad R \in \mathbb{R}^{2 \times \ell}, \quad \ell = n^2,$$

provided that  $\gamma$  is small enough. Here,  $q \in \mathbb{R}^\ell$  is the vectorization of  $\Delta$ , and the matrix  $R$  is defined by the left and right eigenvectors of  $\lambda$ .

It can be shown that, as  $q$  sweeps the ball  $\gamma Q$ , the 2D-vector  $[\text{Re } \tilde{\lambda}, \text{Im } \tilde{\lambda}]$  sweeps the ellipse

$$\mathcal{E} := \{x \in \mathbb{R}^2 : (S^{-1}(x - \lambda), x - \lambda) \leq \gamma^2\}, \quad S := RR^\top.$$

Now, assuming that the uncertainty  $q$  is random, uniformly distributed over the ball  $\gamma Q$ , and specifying a confidence probability  $\mathfrak{p}$ , we make use of Lemma 2 (see Sect. 5.1) to shape an ellipsoidal probabilistic predictor  $\mathcal{E}_\mathfrak{p}$  of the ellipse  $\mathcal{E}$ .

A schematic illustration of the ideas above is given next. For a  $6 \times 6$  stable matrix having  $\ell = 36$  uncertain entries, quite an accurate upper bound  $\bar{\gamma} = 0.3947$  of the stability margin can be found.

Let us specify  $\mathfrak{p} = 0.99$ ; then the constructions above yield  $\hat{\gamma}_\mathfrak{p} = 0.7352$  as an estimate of the value of the probabilistic margin. In other words, the uncertainty radius is almost doubled, at the expense of admitting the 1%-probability of instability. To confirm these conclusions, we performed straightforward Monte Carlo simulations for  $\gamma = \hat{\gamma}_\mathfrak{p}$ , which resulted in the sampled probability of stability  $p_{MC} = 0.9989$  (from a sample of 40,000 points  $q$ ). Figure 2 depicts the linear approximation of

the pseudospectrum of  $A$  (larger ellipses) and its ellipsoidal probabilistic predictors (smaller ellipses, rightmost of them touch the imaginary axis), along with sampled values of the pseudospectrum.

Other examples relate to the probability of a polynomial with coefficients in a cube to be stable [46] and to the generation of random stable polynomials [69].

### 3 Randomization in Estimation

Usual assumptions on the noise in linear regression problems are that it is a sequence of independent zero-mean random variables (vectors). However in practical situations these assumptions are often violated which may strongly affect the performance of standard estimators. Therefore it is important to examine the possibility to estimate the regression parameters under minimal assumptions on the noise. It may appear surprising that the regression parameters can be consistently estimated in the case of biased, correlated and even nonrandom noise. However, it can be done under certain conditions when the inputs (regressors) are random. We consider a linear regression model

$$y_n = x_n^\top \theta + \xi_n \quad (5)$$

with the parameter vector  $\theta \in \mathbb{R}^N$  to be estimated from the observations  $y_n, x_n, n = 1, 2, \dots$ . It is assumed that the inputs  $x_n$  are zero-mean random vectors independent of the noise  $\xi_k$ . This assumption ensures “good” properties of estimators under extremely mild restrictions on the noise. The idea of using random inputs to eliminate bias was put forward by Fisher [22] as the randomization principle in the design of experiments. Besides settings of design type where regressors are randomized by the experimenter, random inputs arise in many applications of identification, filtering, recognition, etc. Having these applications in mind, we use the terms “inputs,” “outputs,” etc., rather than those traditional to the regression analysis (say, “regressors”).

We follow the results in [25], see also [27]. Let us formulate the rigorous assumptions on the data for the regression problem (5).

(A) the inputs  $x_n$  are represented by a sequence of independent, identically distributed random vectors with symmetric distribution function, zero mean value  $\mathbf{E}x_n = 0$ , positive-definite covariance matrix  $\mathbf{E}x_n x_n^\top = B > 0$ , and a finite fourth moment  $\mathbf{E}\|x_n\|^4 < \infty$ ; moreover,  $x_n$  is independent of  $\{\xi_0, \xi_1, \dots, \xi_n\}$ .

(B) the noise  $\xi_n$  is mean-square bounded:  $\mathbf{E}|\xi_n|^2 \leq \sigma^2$ .

**Theorem 2** *Under the assumptions above, the least square estimate  $\theta_n$  of the true parameter  $\theta$  is mean-square consistent, and the rate of convergence is given by*

$$\mathbf{E}(\theta_n - \theta)(\theta_n - \theta)^\top = \frac{\sigma^2}{n} B^{-1} + o\left(\frac{1}{n}\right). \quad (6)$$

If the inputs are deterministic and  $B = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^{\infty} x_i x_i^\top$ , one can obtain a similar estimate for the least squares algorithm under the standard assumption that the noise is zero mean,  $\mathbf{E}\xi_n = 0$ . The principal contribution of Theorem 2 is the removal of this restrictive assumption.

A result similar to Theorem 2 holds true for the Polyak–Ruppert online averaging algorithm [64]:

$$\theta_n = \theta_{n-1} + \gamma_n x_n (y_n - \theta_{n-1}^\top x_n) \quad (7)$$

$$\hat{\theta}_n = (1 - n^{-1})\hat{\theta}_{n-1} + n^{-1}\theta_n, \quad (8)$$

where

$$\gamma_n/\gamma_{n+1} = 1 + o(\gamma_n); \quad (9)$$

for instance,  $\gamma_n = 1/n^r$  for some  $0 < r < 1$ . It is proved in [25] that estimate (6) is true under assumptions (A), (B) for no-zero-mean noise.

The fruitful idea of randomizing the inputs is exploited in numerous problems of identification, control, optimization in the monographs [28, 29]. These results confirm the general conclusion: Randomization enables for a considerable relaxation of the standard assumptions on the noise. In Sect. 5, we focus on such approaches to stochastic optimization problems.

## 4 Feasibility

The problem of solving convex inequalities (also known as convex feasibility problem) is one of the basic problems of numerical analysis. It arises in numerous applications, including statistics, parameter estimation, pattern recognition, image restoration, tomography and many others, see, e.g., monographs and surveys [7, 15, 17] and references therein. Particular cases of the problem relate to solving of linear inequalities and to finding a common point of convex sets. The specific feature of some applications is a huge number of inequalities to be solved, while the dimensionality of the variables is moderate, see, e.g., the examples of applied problems below. Under these circumstances many known numerical methods are inappropriate. For instance, finding the most violated inequality may be a hard task; dual methods also cannot be applied due to large number of dual variables.

In this survey we follow mainly the paper [56] and focus on simple iterative methods which are applicable to the case of very large (and even infinite) number of inequalities. They are based on projection-like algorithms, originated in the works [1, 31, 36, 44]. There are many versions of such algorithms; they can be either parallel or non-parallel (row-action); in the latter case the order of projections is usually chosen as cyclical one or the-most-violated one, see [7, 15, 17]. All these methods are well suited for the finite (and not too large) number of constraints. The novelty of the method under consideration is its random nature, which allows to treat

large-dimensional- and infinite-dimensional cases. Although the idea of exploiting stochastic algorithms for optimization problems with continua of constraints has been known in the literature [34, 51, 80], it led to much more complicated calculations than the proposed method. Another feature of the method is its finite termination property—under the strong feasibility assumption a solution is found after a finite number of steps with probability one. The version of a projection method for linear inequalities with this property has been proposed first by V.A. Yakubovich [81]. Below we survey the main results from [56]. Related contributions can be found in [13, 61].

Consider the general convex feasibility problem: find a point  $x$  in the set

$$C = \{x \in X : f(x, q) \leq 0 \quad \forall q \in Q\}. \quad (10)$$

Here  $X \subset \mathbb{R}^n$  is a convex closed set,  $f(x, q)$  is convex in  $x$  for all  $q \in Q$ , where  $Q$  is an arbitrary set of indices (finite or infinite). Note that this formulation is similar to the robust feasibility problem (1) considered above. However, instead of finding its approximate solution or evaluating the volume of violation, we are aimed at finding a solution satisfying *all* inequalities, but using *randomized methods*.

Particular cases of problem (10) are:

1. *Finite number of inequalities*:  $Q = \{1, \dots, m\}$ .
2. *Semi-infinite problem*:  $Q = [0, T] \subset \mathbb{R}^1$ .
3. *Finding a common point of convex sets*:  $f(x, q) = \text{dist}(x, C_q) = \min_{y \in C_q} \|x - y\|$ , where the sets  $C_q := \{x \in X : f(x, q) \leq 0 \text{ for a } q \in Q\} \subset \mathbb{R}^n$  are closed and convex and  $C = \bigcap_{q \in Q} C_q$ . Here,  $\|x\|$  denotes the Euclidean norm of a vector.
4. *Linear inequalities*:  $f(x, q) = a(q)^\top x - b(q)$ .

We assume that a subgradient  $\partial_x f(x, q)$  is available at any point  $x \in X$  for all  $q \in Q$  (we mean an arbitrary subgradient if the set of them is not a singleton).

The algorithm has the following structure. At the  $k$ th iteration, we generate randomly  $q_k \in Q$ ; we assume that the  $q_k$ 's are independent and identically distributed (i.i.d.) samples from some probabilistic distribution  $p_q$  on  $Q$ . Two key assumptions are adopted.

**Assumption 1** (*strong feasibility*). The set  $C$  is nonempty and contains an interior point

$$\exists x^* \in C : \|x - x^*\| \leq r \implies x \in C.$$

Here,  $r > 0$  is a constant which is assumed to be known.

**Assumption 2** (*distinguishability of feasible and infeasible points*). For  $x \in X \setminus C$ , the probability of generating a violated inequality is not vanishing:

$$\text{Prob}\{f(x, q) > 0\} > 0.$$

This is the only assumption on the probability distribution  $p_q$ . For instance, if  $Q$  is a finite set and each element in  $Q$  is generated with nonzero probability, then Assumption 2 holds. The *feasibility algorithm* is then formulated as follows:

**Algorithm 1:** Given an initial point  $x_0 \in X$ , proceed as follows:

$$x_{k+1} = \pi_X(x_k - \lambda_k \partial_x f(x_k, q_k)), \quad (11)$$

$$\lambda_k = \begin{cases} \frac{f(x_k, q_k) + r \|\partial_x f(x_k, q_k)\|}{\|\partial_x f(x_k, q_k)\|^2} & \text{if } f(x_k, q_k) > 0; \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

Here,  $\pi_X$  is a projection operator onto  $X$ ; that is,  $\|x - \pi_X(x)\| = \text{dist}(x, X)$ . Hence, at every step, the calculation of a subgradient is performed just for one inequality, which is randomly chosen among all inequalities  $\mathcal{Q}$ . Note that the value of  $r$  (the radius of a ball in the feasible set) is used in the algorithm; its modification for  $r$  unknown will be presented later. To explain the choice of the step-size  $\lambda_k$  in the algorithm, we consider the two particular cases.

1. *Linear inequalities:*  $f(x, q) = a(q)^\top x - b(q)$ ,  $X = \mathbb{R}^n$ .

Then we have  $\partial_x f(x_k, q_k) = a_k$ , where  $f(x_k, q_k) = a_k^\top x_k - b_k$  and  $a_k = a(q_k)$ ,  $b_k = b(q_k)$ , so that the algorithm takes the form

$$x_{k+1} = x_k - \frac{(a_k^\top x_k - b_k)_+ + r \|a_k\|}{\|a_k\|^2} a_k$$

for  $(a_k^\top x_k - b_k)_+ \neq 0$ , otherwise  $x_{k+1} = x_k$ ; here,  $c_+ = \max\{0, c\}$ . For  $r = 0$ , the method coincides with the projection method for solving linear inequalities by Agmon–Motzkin–Shoenberg [1, 44].

2. *Common point of convex sets:*  $f(x, q) = \text{dist}(x, C_q)$ ,  $C = \bigcap_{q \in \mathcal{Q}} C_q$ ,  $X = \mathbb{R}^n$ .

Then we have  $\partial_x f(x_k, q_k) = (x_k - \pi_k(x_k))/\rho_k$ , where  $\pi_k$  denotes the projection onto the set  $C_k = C_{q_k}$  and  $\rho_k = \|x_k - \pi_k(x_k)\|$ . The algorithm takes the form

$$x_{k+1} = \pi_k(x_k) + \frac{r}{\rho_k} (\pi_k(x_k) - x_k),$$

provided that  $x_k \notin C_k$ ; otherwise  $x_{k+1} = x_k$ . We conclude that, for  $r = 0$ , each iteration of the algorithm is the same as for the projection method for finding the intersection of convex sets [7, 31].

Having this in mind, the rule for selecting the step-size  $\lambda_k$  has a very natural explanation. Denote by  $y_{k+1}$  the point which is generated via the same formula as  $x_{k+1}$ , but with  $r = 0$ ; assume also  $X = \mathbb{R}^n$ . Then, for the case of linear inequalities,  $y_{k+1}$  is the projection of  $x_k$  onto the half-space  $\{x: a_k^\top x - b_k \leq 0\}$ . Similarly, if we deal with finding a common point of convex sets,  $y_{k+1}$  is the projection of  $x_k$  onto the set  $C_k$ . It is easy to show that  $\|x_{k+1} - y_{k+1}\| = r$ . Thus the step in the algorithm is an (additively) over-relaxed projection; we perform an extra step (of length  $r$ ) inside the current feasible set.

The idea of additive over-relaxation is due to V.A. Yakubovich who applied such a method to linear inequalities [81]. In the papers mentioned above, the order of sorting out the inequalities was either cyclic or the-most-violated one was taken, in contrast with the random order in the proposed algorithm.

Now we formulate the main result on the convergence of the algorithm.

**Theorem 3** *Under Assumptions 1, 2, Algorithm 1 finds a feasible point in a finite number of iterations with probability one, i.e., with probability one there exists  $N$  such that  $x_N \in C$  and  $x_k = x_N$  for all  $k \geq N$ .*

We now illustrate how the general algorithm can be adapted to two particular important cases.

**1. Linear Matrix Inequalities** are one of the most powerful tools for model formulation in various fields of systems and control, see [10]. There exist well-developed techniques for solving such inequalities as well as for solving optimization problems subject to such inequalities (Semidefinite Programming, SDP). However in a number of applications (for instance, in robust stabilization and control), the number of LMIs is extremely large or even infinite, and such problems are beyond the applicability of the standard LMI tools. Let us cast these problems in the framework of the approach proposed above.

The space  $\mathbb{S}_m$  of  $m \times m$  symmetric real matrices equipped with the scalar product  $\langle A, B \rangle = \text{tr} AB$  and the Frobenius norm, becomes a Hilbert space ( $\text{tr}(\cdot)$  denotes the trace of a matrix). Then we can define the projection  $A_+$  of a matrix  $A$  onto the cone of positive semidefinite matrices. This projection can be found in explicit form. Indeed, if  $A = RDR^\top$ ,  $R^{-1} = R^\top$ , is the eigenvector–eigenvalue decomposition of  $A$  and  $D = \text{diag}(d_1, \dots, d_m)$ , then

$$A_+ = RD_+R^\top, \quad (13)$$

where  $D_+ = \text{diag}(d_1^+, \dots, d_m^+)$  and  $d_i^+ = \max\{0, d_i\}$ .

Linear matrix inequality is the expression of the form

$$A(x) = A_0 + \sum_{i=1}^n x_i A_i \preceq 0,$$

where  $A_i \in \mathbb{S}_m$ ,  $i = 0, 1, \dots, n$ , are given matrices and  $x = (x_1, \dots, x_n) \in \mathbb{R}^n$  is the vector variable. Another form of LMI was mentioned in Sect. 2; it is reducible to the *canonical form* above.

The general system of LMIs can be written as

$$A(x, q) = A_0(q) + \sum_{i=1}^n x_i A_i(q) \preceq 0 \quad \forall q \in Q. \quad (14)$$

Here,  $Q$  is the set of indices which can be finite or infinite. The problem under consideration is to find an  $x \in \mathbb{R}^n$  which satisfies LMIs (14). Our first goal is to

convert these LMIs into a system of convex inequalities. For this purpose, introduce the scalar function

$$f(x, q) = \|A_+(x, q)\| \quad (15)$$

where  $A(x, q)$  is given by (14) and  $A_+$  is defined in (13).

**Lemma 1** *The matrix inequalities (14) are equivalent to the scalar inequalities*

$$f(x, q) \leq 0 \quad \forall q \in \mathcal{Q}.$$

*The function  $f(x, q)$  is convex in  $x$  and its subgradient is given by*

$$\partial_x f(x, q) = \frac{1}{f(x, q)} \begin{pmatrix} \text{tr } A_1 A_+(x, q) \\ \vdots \\ \text{tr } A_n A_+(x, q) \end{pmatrix}$$

*if  $f(x, q) > 0$ ; otherwise  $\partial_x f(x, q) = 0$ .*

Hence, solving linear matrix inequalities can be converted into solving a convex feasibility problem.

**2. Solving linear equations.** This case has some peculiarities—the solution set is either a single point or a linear subspace, so that it never contains an interior point and Algorithm 1 with  $r > 0$  does not converge. However it can be applied with  $r = 0$ ; for a deterministic choice of the alternating directions it is precisely the Kaczmarz algorithm [36]. Its randomized version with equal probabilities for all equations has been proposed in [56]; it converges with linear rate. More recently, Strohmer and Vershynin [75] studied this method with the probabilities for choosing the equation  $(a_i, x) = b_i$  being proportional to  $\|a_i\|^2$ . They proved that the rate of convergence depends on the condition number of the matrix  $A$ , but not on the number of equations. This result stimulated further research in [15, 16, 20, 26, 41].

## 5 Optimization

After the invention of the Monte Carlo (MC) paradigm by N. Metropolis and S. Ulam in the late 1940s [43], it has become extremely popular in numerous application areas such as physics, biology, economics, social sciences, and other areas. As far as mathematics is concerned, Monte Carlo methods proved to be exceptionally efficient in the simulation of various probability distributions, numerical integration, estimation of the mean values of the parameters, etc. [37, 67, 77]. More recent version of the approach, *Markov Chain Monte Carlo*, is often referred to as *MCMC revolution* [23]. The salient feature of MC approach to solution of various problems of this sort is that “often,” it is dimension-free in the sense that, given  $N$  samples, the accuracy of the result does not depend on the dimension of the problem.

On the other hand, applications of the MC paradigm in the area of optimization are not that successful. In this regard, problems of global optimization deserve special attention. As explained in [82] (see beginning of Chapter 1.2), “*In global optimization, randomness can appear in several ways. The main three are: (i) the evaluations of the objective function are corrupted by random errors; (ii) the points  $x_i$  are chosen on the base of random rules, and (iii) the assumptions about the objective function are probabilistic.*” Pertinent to the exposition of this paper is only case (ii). Monte Carlo is the simplest, brute force example of randomness-based methods (in [82] it is referred to as “Pure Random Search”). With this method, one samples points uniformly in the feasible domain, computes the values of the objective function, and picks the record value as the output.

Of course, there are dozens of more sophisticated stochastic methods such as multistart, simulated annealing, genetic algorithms, evolutionary algorithms, etc.; e.g., see [24, 35, 52, 70, 72, 82] for an incomplete list of relevant references. However, most of these methods are heuristic in nature; often, they lack rigorous justification, and the computational efficiency is questionable. Moreover, there exist pessimistic results on “insolvability of global optimization problems.” This phenomenon has first been observed as early as in the monograph [47] by A. Nemirovskii and D. Yudin, both in the deterministic and stochastic optimization setups (see Theorem, Section 1.6 in [47]). Specifically, the authors of [47] considered the minimax approach to the minimization of the class of Lipschitz functions and proved that, no matter what the optimization method is, it is possible to construct a problem which will require exponential (in the dimension) number of function evaluations. The “same” number of samples is required for the simplest MC method. Similar results can be found in [48], Theorem 1.1.2, where the construction of “bad” problems is exhibited. Below we present another example of such problems (with very simple objective functions, close to linear ones) which are very hard to optimize. Concluding this brief survey, we see that any advanced method of global optimization cannot outperform Monte Carlo when optimizing “bad” functions.

This explains our interest in the MC approach as applied to the optimization setup. In spite of the pessimistic results above, there might be a belief that, if Monte Carlo is applied to a “good” optimization problem (e.g., a convex one), the results would not be so disastrous. Our goal in this section is to blow up these optimistic expectations. We examine the “best” optimization problems (the minimization of a linear function on a ball) and estimate the accuracy of the Monte Carlo method. Unfortunately, the dependence on the dimension remains exponential, and practical solution of these simplest problems via such an approach is impossible for high dimensions.

The second part of the section is devoted to randomized algorithms for convex optimization. The efficiency of such an approach has been discovered recently; it became clear that advanced randomized coordinate descent and similar approaches for distributed optimization are strong competitors to deterministic versions of the methods.



## 5.1 Direct Monte Carlo in Optimization

In this subsection we show that *straightforward* use of Monte Carlo in optimization, both global and convex is highly inefficient in problems of high dimensions. The material is based on the results in [60].

**Global optimization: A pessimistic example.** We first present a simple example showing failure of stochastic global optimization methods in high-dimensional spaces. This example is constructed along the lines suggested in [47] (also, see [48], Theorem 1.1.2) and is closely related to one of the central problems discussed below, the minimization of a linear function over a ball in  $\mathbb{R}^n$ .

Consider an unknown vector  $c \in \mathbb{R}^n$ ,  $\|c\| = 1$ , and the function

$$f(x) = \min \left\{ 99 - c^\top x, (c^\top x - 99)/398 \right\}$$

to be minimized over the Euclidean ball  $Q \subset \mathbb{R}^n$  of radius  $r = 100$  and centered at the origin. Obviously, the function has one local minimum  $x_1 = -100c$ , with the function value  $f_1 = -0.5$ , and one global minimum  $x^* = 100c$ , with the function value  $f^* = -1$ . The objective function is Lipschitz with Lipschitz constant equal to 1, and  $\max f(x) - \min f(x) = 1$ .

Any standard (not problem-oriented) version of stochastic global search (such as multistart, simulated annealing, etc.) will miss the domain of attraction of the global minimum with probability  $1 - V^1/V^0$ , where  $V^0$  is the volume of the ball  $Q$ , and  $V^1$  is the volume of the set  $C = \{x \in Q: c^\top x \geq 99\}$ . In other words, the probability of success is equal to

$$\text{Prob} = \frac{V^1}{V^0} = \frac{1}{2} I \left( \frac{2rh - h^2}{r^2}; \frac{n+1}{2}, \frac{1}{2} \right),$$

where  $I(x; a, b)$  is the regularized incomplete beta function with parameters  $a$  and  $b$ , and  $h$  is the height of the spherical cap  $C$ ; in this example,  $h = 1$ . This probability quickly goes to zero as the dimension of the problem grows; say, for  $n = 15$ , it is of the order of  $10^{-15}$ . Hence, any “advanced” method of global optimization will find the minimum with relative error not less than 50%; moreover, such methods are clearly seen to be no better than a straightforward Monte Carlo sampling. The same is true if our goal is to estimate the minimal value of the function  $f^*$  (not the minimum point  $x^*$ ). Various methods based on ordered statistics of sample values (see Section 2.3 in [82]) fail to reach the set  $C$  with high probability, so that the prediction will be close to  $f_1 = -0.5$  instead of  $f^* = -1$ .

**Scalar convex optimization: Pessimistic results.** Let  $Q$  denote the unit Euclidean ball in  $\mathbb{R}^n$  and let  $\xi^{(i)}|_1^N = \{\xi^{(1)}, \dots, \xi^{(N)}\}$  be a multisample of size  $N$  from the uniform distribution  $\xi \sim \mathcal{U}(Q)$ .

Given the scalar-valued linear function

$$g(x) = c^\top x, \quad c \in \mathbb{R}^n, \tag{16}$$

defined on  $Q$ , estimate its maximum value from the multisample.

More specifically, let  $\eta^*$  be the true maximum of  $g(x)$  on  $Q$  and let

$$\eta = \max\{g^{(1)}, \dots, g^{(N)}\}, \quad g^{(i)} = g(\xi^{(i)}), \quad i = 1, \dots, N, \tag{17}$$

be the empirical maximum; we say that  $\eta$  approximates  $\eta^*$  with accuracy at least  $\delta$  if

$$\frac{\eta^* - \eta}{\eta^*} \leq \delta.$$

Then the problem is: *Given a probability level  $p \in ]0, 1[$  and accuracy  $\delta \in ]0, 1[$ , determine the minimal length  $N_{\min}$  of the multisample such that, with probability at least  $p$ , the accuracy of approximation is at least  $\delta$  (i.e., with high probability, the empirical maximum nicely evaluates the true one).*

The results presented below are based on the following fact established in [59]; it relates to the probability distribution of a specific quadratic function of the random vector uniformly distributed on the Euclidean ball.

**Lemma 2** ([59]) *Let the random vector  $\xi \in \mathbb{R}^n$  be uniformly distributed on the unit Euclidean ball  $Q \subset \mathbb{R}^n$ . Assume that a matrix  $A \in \mathbb{R}^{m \times n}$  has rank  $m \leq n$ . Then the random variable*

$$\rho = \left( (AA^\top)^{-1} A\xi, A\xi \right)$$

has the beta distribution  $\mathcal{B}(\frac{m}{2}, \frac{n-m}{2} + 1)$  with probability density function

$$f_\rho(x) = \begin{cases} \frac{\Gamma(\frac{n}{2} + 1)}{\Gamma(\frac{m}{2})\Gamma(\frac{n-m}{2} + 1)} x^{\frac{m}{2}-1} (1-x)^{\frac{n-m}{2}} & \text{for } 0 \leq x \leq 1, \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where  $\Gamma(\cdot)$  is the Euler gamma function.

Alternatively, the numerical coefficient in (18) writes

$$\frac{\Gamma(\frac{n}{2} + 1)}{\Gamma(\frac{m}{2})\Gamma(\frac{n-m}{2} + 1)} = 1/B\left(\frac{m}{2}, \frac{n-m}{2} + 1\right),$$

where  $B(\cdot, \cdot)$  is the beta function.

We consider the scalar case (16) and discuss first a qualitative result that follows immediately from Lemma 2. Without loss of generality, let  $c = (1, 0, \dots, 0)^\top$ , so that the function  $g(x) = x_1$  takes its values on the segment  $[-1, 1]$ , and the true maximum of  $g(x)$  on  $Q$  is equal to 1 (respectively,  $-1$  for the minimum) and is attained with  $x = c$  (respectively,  $x = -c$ ). Let us compose the random variable

$$\rho = g^2(\xi),$$

which is the squared first component  $\xi_1$  of  $\xi$ . By Lemma 2 with  $m = 1$  (i.e.,  $A = c^\top$ ), for the probability density function (pdf) of  $\rho$  we have

$$f_\rho(x) = \frac{\Gamma(\frac{n}{2} + 1)}{\Gamma(\frac{1}{2})\Gamma(\frac{n+1}{2})} x^{-\frac{1}{2}}(1-x)^{\frac{n-1}{2}} := \beta_n x^{-\frac{1}{2}}(1-x)^{\frac{n-1}{2}}.$$

Straightforward analysis of this function shows that, as dimension grows, the mass of the distribution tends to concentrate closer to the origin, meaning that the random variable (r.v.)  $\rho$  is likely to take values which are far from the maximum, equal to unity.

We next state the following rigorous result [60].

**Theorem 4** *Let  $\xi$  be a random vector uniformly distributed over the unit Euclidean ball  $Q \subset \mathbb{R}^n$  and let  $g(x) = x_1$ ,  $x \in Q$ . Given  $p \in ]0, 1[$  and  $\delta \in ]0, 1[$ , the minimal sample size  $N_{\min}$  that guarantees, with probability at least  $p$ , for the empirical maximum of  $g(x)$  to be at least a  $\delta$ -accurate estimate of the true maximum, is given by*

$$N_{\min} = \frac{\ln(1-p)}{\ln\left[\frac{1}{2} + \frac{1}{2}I\left((1-\delta)^2; \frac{1}{2}, \frac{n+1}{2}\right)\right]}, \quad (19)$$

where  $I(x; a, b)$  is the regularized incomplete beta function with parameters  $a$  and  $b$ .

Clearly, a correct notation should be  $N_{\min} = \lceil \cdot \rceil$ , i.e., rounding toward the next integer; we omit it, but it is implied everywhere in the sequel.

Numerical values of the function  $I(x; a, b)$  can be computed via use of the MATLAB routine `betainc`. For example, with the modest values  $n = 10$ ,  $\delta = 0.05$ , and  $p = 0.95$ , formula (19) gives  $N_{\min} \approx 8.9 \cdot 10^6$ , and this quantity grows quickly as the dimension  $n$  increases.

Since we are interested in small values of  $\delta$ , i.e., in  $x$  close to unity, a ‘‘closed-form’’ lower bound for  $N_{\min}$  can be computed as stated below.

**Corollary 1** *In the conditions of Theorem 4*

$$N_{\min} > N_{\text{appr}} = \frac{\ln(1-p)}{\ln\left[1 - \frac{\beta_n}{n+1} \frac{1}{1-\delta} (2\delta - \delta^2)^{(n+1)/2}\right]},$$

where  $\beta_n = \frac{\Gamma(\frac{n}{2}+1)}{\Gamma(\frac{1}{2})\Gamma(\frac{n+1}{2})} = 1/B(\frac{1}{2}, \frac{n+1}{2})$ .

Further simplification of the lower bound can be obtained

$$N_{\text{appr}} > \tilde{N}_{\text{appr}} = -\frac{\ln(1-p)}{\sqrt{2\pi(n+1)} \frac{1}{1-\delta} (2\delta - \delta^2)^{(n+1)/2}}.$$

The lower bounds obtained above are quite accurate; for instance, with  $n = 10$ ,  $\delta = 0.05$ , and  $p = 0.95$ , we have  $N_{\min} \approx 8.8694 \cdot 10^6$ , while  $N_{\text{appr}} \approx 8.7972 \cdot 10^6$  and  $\tilde{N}_{\text{appr}} = 8.5998 \cdot 10^6$ .

The moral of this subsection is that, for high dimensions, a straightforward use of Monte Carlo sampling cannot be considered as a tool for finding extreme values of a function, even in the convex case.

## 5.2 Randomized Methods

On the other hand, exploiting randomized methods in different forms can be highly efficient; in many cases they are strong competitors of deterministic algorithms.

**Unconstrained minimization.** We start with *random search* methods for unconstrained minimization

$$\min f(x), \quad x \in R^n.$$

Probably the first publication relates to the 1960s [42, 65]. The idea was to choose a random direction in the current point and make a step resulting in decrease of the objective function. Rigorous results on convergence of some random search algorithms were obtained in [19]. Nevertheless the practical experiments with similar methods were mostly disappointing, and they did not attract much attention (excluding global optimization, see above). For convex problems the situation has changed recently, when the dimension of problems under consideration became very large ( $n$  is of the order  $10^6$ ) or when distributed optimization problems arose ( $f(x) = \sum_{i=1}^N f_i(x_i)$ ,  $x = (x_1, \dots, x_N)$ ,  $N$  is large). We survey some results in this direction first.

The basic algorithm of random search can be written as

$$x_{k+1} = x_k - \gamma_k \frac{\hat{f}(x_k + \mu_k u_k) - \hat{f}(x_k)}{\mu_k} u_k, \quad (20)$$

where  $x_k$  is a  $k$ -th approximation to the solution  $x^*$ ,  $u_k$  is a random vector,  $\gamma_k$ ,  $\mu_k$  are step-sizes, and  $\hat{f}(x_k)$  is a measured value of  $f(x_k)$ ; either  $\hat{f}(x_k) = f(x_k)$  (deterministic setup) or  $\hat{f}(x_k) = f(x_k) + \xi_k$ ,  $\xi_k$  being a random noise (stochastic optimization). Algorithm (20) requires one calculation of the objective function per iteration, its symmetric version

$$x_{k+1} = x_k - \gamma_k \frac{\hat{f}(x_k + \mu_k u_k) - \hat{f}(x_k - \mu_k u_k)}{2\mu_k} u_k, \quad (21)$$

uses two calculations. The strategy of choosing step-sizes depends on smoothness of  $f(x)$  and on the presence of errors  $\xi_k$  in function evaluation. The following result is adaptation of more general theorems in [62, 63] for  $C^2$  functions.

**Theorem 5** Consider the problem of unconstrained minimization of  $f(x)$ , where  $f(x)$  is strongly convex, twice differentiable, with gradient satisfying the Lipschitz condition. Suppose  $u_k$  are random i.i.d. uniformly distributed in the cube  $\|u\|_\infty \leq 1$ . Noises  $\xi_k$  are independent of  $u_1, \dots, u_k$  and have bounded second moment  $\mathbf{E}|\xi_i|^2 \leq \sigma^2$ . The step-size satisfies the following conditions:  $\gamma_k = a/k$ ,  $\mu_k = \mu/k^4$ ,  $a$  is large enough. Then the iterations  $x_k$  in algorithms (20), (21) converge to the minimum point  $x^*$  in mean-square and

$$\mathbf{E}\|x_k - x^*\|^2 = O(1/\sqrt{k}).$$

It is worth mentioning that randomization of directions  $u_k$  allows to remove the assumption  $\mathbf{E}x_k = 0$ , which is standard in stochastic optimization methods [38]; a similar effect for estimation is exhibited in Theorem 2. If compared with the classical Kiefer–Wolfowitz (KW) method, algorithms (20), (21) are less laborious: they require just one or two function evaluations per iteration vs  $n$  or  $2n$  in the KW-method. On the other hand, asymptotic rate of convergence is the same:  $O(1/\sqrt{n})$ . More details about convergence, various forms, computational experience of such algorithms can be found in the publications of J. Spall (e.g., [73]); he names the algorithms SPSA (Simultaneous Perturbation Stochastic Approximation). The pioneering research on the algorithms are due to Yu. Ermoliev [21] and H. Kushner [38].

Now we focus on purely deterministic version of problem (5), where measurements of the objective function do not contain errors:  $\hat{f}(x_k) = f(x_k)$ . As we mentioned above, the interest to such methods grew enormously when very high-dimensional problems became appealing due to such applications as machine learning and neural networks. The interest has been triggered with Yu. Nesterov’s paper [49]. Roughly speaking, the approach of [49] is as follows. It is assumed that the Lipschitz constants  $L_i$  for partial derivatives  $\partial f/\partial x_i$  are known (and they can be easily estimated for quadratic functions). Then, at the  $k$ th iteration, the index  $i = \alpha$  is chosen with probability proportional to  $L_i$ , and new iteration is obtained by changing coordinate  $i\alpha$  with step-size  $(1/L_\alpha)\partial f/\partial x_\alpha$ . Yu. Nesterov provides sharp estimates on the rate of convergence and also presents the accelerated version of the algorithm. These theoretical results supported with intensive numerical experiments for huge-scale problems confirm advantages of the random coordinate descent. This line of research found numerous applications in distributed optimization [9, 45, 66]. The titles of many publications (e.g., recent one [33]) confirm advantages of randomized algorithms.

Randomization techniques are also helpful for minimization of nonsmooth convex functions, when the only data available are the values of the function  $f(x)$  at an arbitrary point. The idea of the following algorithm is due to A. Gupal [32], also see [55], Section 6.5.2. In contrast with algorithm (21), we generate a random point  $\tilde{x}_k$  in the neighborhood of the current iteration point  $x_k$  and then make a step similar to (21) from this point. Thus the algorithm is written as

$$x_{k+1} = x_k - \gamma_k \frac{f(\tilde{x}_k + \mu_k u_k) - f(\tilde{x}_k - \mu_k u_k)}{2\mu_k} u_k, \quad (22)$$

$$\tilde{x}_k = x_k + \alpha_k h_k \quad (23)$$

where  $u_k, h_k$  are independent random vectors uniformly distributed in the cube  $\|u\|_\infty \leq 1$ , while  $\alpha_k, \gamma_k, \mu_k$  are scalar step-sizes. It can be seen that randomization step with  $h_k$  is equivalent to smoothing of the original function, thus algorithm similar to (21) is applied to the smoothed function. By adjusting the parameters  $\alpha_k, \gamma_k, \mu_k$ , we arrive at the convergence result.

**Theorem 6** *Let  $f(x)$  be convex, and let a unique minimum point  $x^*$  exist. Let the step-sizes satisfy the conditions*

$$\sum_{k=1}^{\infty} \gamma_k = \infty, \quad \sum_{k=1}^{\infty} \gamma_k^2 < \infty, \quad \gamma_k/\alpha_k \rightarrow 0, \quad \mu_k/\alpha_k \rightarrow 0, \quad \alpha_k \rightarrow 0, \quad |\alpha_k - \alpha_{k+1}|/\gamma_k \rightarrow 0.$$

*Then  $x_k \rightarrow x^*$  with probability one.*

This result guarantees convergence of the algorithm to the minimum point. However it does not provide effective strategies for choosing parameters, neither it estimates the rate of convergence. Above-mentioned problems are deeply investigated in [50]. The authors apply Gaussian smoothing technique (i.e., the vectors  $u_k$  are Gaussian) and present randomized methods for various classes of functions (smooth and nonsmooth) for different situations (gradient or gradient-free oracles). The versions of the algorithms with the best rate of convergence are indicated.

To conclude, we remind that there exist no-zero-order deterministic methods for minimization of nondifferentiable convex functions, so that randomized methods provide the only option.

**Constrained minimization.** There are various problem formulations related to randomized methods for optimization in the presence of constraints.

One of them is closely related to feasibility problem (10), but now we are looking to the feasible point which minimizes an objective function

$$\min(c, x) \quad f(x, q) \leq 0 \quad \forall q \in Q. \quad (24)$$

Here we have taken the objective function to be linear without loss of generality. Constraint functions  $f(x, q)$  are convex in the variable  $x \in \mathbb{R}^n$  for all values of the parameters  $q$ . Numerous examples of constraints of this form were discussed in Sect. 4. Such problems are closely related to *robust optimization*, see [8] and Sect. 2. A randomized approach to the problem consists of a random choice of  $N$  parameters  $q_1, \dots, q_N$  from the set  $Q$  and solving the convex optimization problem with a finite number of constraints

$$\min_{x \in C}(c, x) \quad f(x, q_i) \leq 0 \quad i = 1, \dots, N. \quad (25)$$

We suppose that this problem can be solved with high accuracy (e.g., if  $f(x, q)$  are linear in  $x$ , then (25) is LP), and denote the solution by  $x_N$ . Such an approach has been proposed in [11]; the authors answer the following question: *How many samples ( $N$ ) need to be drawn in order to guarantee that the resulting randomized solution violates only a small portion of the constraints?* They assume that there is some probability measure on  $Q$  which defines the probability of violation of constraints  $V(x)$  for arbitrary  $x$ . The main result in [11] states

**Theorem 7**  $\mathbb{E} V(x_N) \leq \frac{n}{N+1}$ .

Of course this result says nothing about the accuracy of the randomized solution (i.e., how close  $x_N$  is to the true solution  $x^*$  or how small  $(c, x_N - x^*)$  is. However, it provides much useful information. Some related results can be found in Sect. 2 above.

Another type of constrained optimization problems reads as

$$\min(c, x), \quad x \in Q, \tag{26}$$

where  $Q \subset \mathbb{R}^n$  is a closed bounded set (convex or nonconvex) such that it is hard to solve explicitly the problem above, and projection on  $Q$  is also unavailable. Then a possible option is to sample random points in  $Q$  and take the best point having the minimal value of the objective function. It is exactly the “direct Monte-Carlo” we have considered in Sect. 2 and found it to be inefficient. However, another approach, based on cutting plane ideas, might be more promising. We assume that a so-called *boundary oracle* is available, that is for an  $x \in Q$  and  $y \in \mathbb{R}^n$ , the quantities

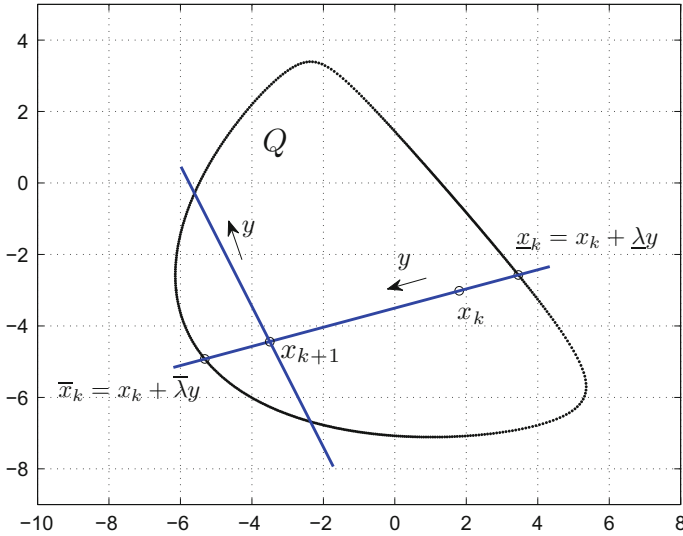
$$\underline{\lambda} = \arg \max\{\lambda \geq 0: x - \lambda y \in Q\}, \quad \bar{\lambda} = \arg \max\{\lambda \geq 0: x + \lambda y \in Q\},$$

can be found efficiently. Numerous examples of sets with known boundary oracles can be found in [30, 68, 71]. Then, starting with some known  $x_0 \in Q$ , we proceed sampling in  $Q$  by using the technique described below.

*Hit-and-Run algorithm (HR).* For  $x_k \in Q$ , take a direction vector  $y$  uniformly distributed on the unit sphere; the oracle returns  $\underline{x}_k = x_k - \underline{\lambda}y$  and  $\bar{x}_k = x_k + \bar{\lambda}y$ . Then, draw  $x_{k+1}$  uniformly distributed on  $[\underline{x}_k, \bar{x}_k]$ . Repeat. Schematically, this algorithm is illustrated in Fig. 3.

This technique was proposed in [71, 79]; under mild assumptions on  $Q$ , the distribution of the random point  $x_k$  was proved to approach the uniform distribution on  $Q$ . Instead of using the “direct Monte-Carlo,” we now apply the randomized cutting plane algorithm, following the ideas of [18, 57].

*A cutting plane algorithm.* Start with  $X_0 = Q$ . For  $X_k$ , generate  $3N$  points  $x_k, \underline{x}_k, \bar{x}_k, k = 1, \dots, N$ , by the HR algorithm and find  $f_k = \min(c, x)$ , where the minimum is taken over these  $3N$  points. Proceed to the new set  $X_{k+1} = X_k \cap \{x: (c, x) \leq f_k\}$  and the initial point  $x_0 = \arg \min(c, x)$ , where the minimum is also taken over the  $3N$  points mentioned above.



**Fig. 3** The idea of the HR algorithm

Rigorous results on the rate of convergence of such an algorithm are lacking. For the idealized analog of it (with the points  $x$  “truly” uniformly distributed in  $X_k$ ), the results on convergence can be found in [18, 57]. Moreover, the algorithm presented above includes the boundary points  $\underline{x}_k, \bar{x}_k$ ; this essentially improves the convergence, since the minimum in the original problem (26) is attained at a boundary point. Numerical experiments in [18, 57] confirm a nice convergence if the set  $Q$  is not too “flat.”

## 6 Conclusions

We have covered in this chapter several topics—in robustness, estimation, control, feasibility, constrained and unconstrained optimization—where the ideas of randomization can be applied and moreover can provide better results than deterministic methods. We could see that the situation with regard to effectiveness of randomized methods is not completely clarified; e.g., some straightforward attempts to apply Monte Carlo for optimization do not work for high dimensions. On the other hand, the only approach to minimization of nonsmooth convex functions with zero-order oracle (i.e., only function values are available) is based on randomization. We hope that the survey will stimulate further interest toward this exciting field of research.

**Acknowledgements** Financial support for this work was provided by the Russian Science Foundation through project no. 16-11-10015.



## References

1. Agmon, S.: The relaxation method for linear inequalities. *Canad. J. Math.* **6**, 382–393 (1954)
2. Barmish, B.R., Lagoa, C.M.: The uniform distribution: A rigorous justification for its use in robustness analysis. *Math. Control Sign. Syst.* **10**(3), 203–222 (1997)
3. Barmish, B., Polyak, B.: A new approach to open robustness problems based on probabilistic prediction formulae. In: Proc. 13th World Congress of IFAC. San Francisco, **H**, 1–6 (1996)
4. Barmish, B.R., Shcherbakov, P.S.: On avoiding vertexization of robustness problems: The approximate feasibility concept. In: Proc. 39th Conference on Decision and Control, Sydney, Australia (2000)
5. Barmish, B.R., Shcherbakov, P.S.: On avoiding vertexization of robustness problems: The approximate feasibility concept. *IEEE Transa Autom. Control* **47**(5), 819–824 (2002)
6. Barmish, B.R., Shcherbakov, P.S., Ross, S.R., Dabbene, F.: On positivity of polynomials: The dilation integral method. *IEEE Transa Autom. Control* **54**(5), 965–978 (2009)
7. Bauschke, H.H., Borwein, J.M.: On projection algorithms for solving convex feasibility problems. *SIAM Review* **38**(3):367–426 (1996)
8. Ben-Tal, A., Nemirovski, A.: Robust convex optimization. *Math. Oper. Res.* **23**(4), 769–805 (1998)
9. Bertsekas, D.P., Tsitsiklis, J.N.: *Parallel and Distributed Computation*. Prentice Hall Inc. (1989)
10. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: *Linear Matrix Inequalities in Systems and Control Theory*. SIAM Publ., Philadelphia (1994)
11. Calafiore, G., Campi, M.C.: Uncertain convex programs: Randomized solutions and confidence levels. *Math. Prog.* **201**(1), 25–46 (2005)
12. Calafiore, G., Campi, M.: The scenario approach to robust control design. *IEEE Trans. Autom. Control* **45**(5), 742–753 (2006)
13. Calafiore G., Polyak, B.: Stochastic algorithms for exact and approximate feasibility of robust LMIs. *IEEE Trans. Autom. Control.* **46**(11), 1755–1759 (2001)
14. Campi, M.: Why is resorting to fate wise? A critical look at randomized algorithms in systems and control. *Eur. J. Control* **16**(5), 419430 (2010)
15. Censor, Y., Cegielski, A.: Projection methods: An annotated bibliography of books and reviews. *Optimization: A Journal of Math. Progr. Oper. Res.* **64**(11), 2343–2358 (2015)
16. Censor, Y., Herman, G.T., Jiang, M.: A note on the behavior of the randomized Kaczmarz algorithm of Strohmer and Vershynin. *J. Fourier Anal. Appl.* **15**(4), 431–436 (2009)
17. Censor, Y., Zenios, S.A.: *Parallel Optimization: Theory, Algorithms, and Applications*. New York, NY, USA: Oxford University Press; (1997)
18. Dabbene, F., Scherbakov, P.S., Polyak, B.T.: A randomized cutting plane method with probabilistic geometric convergence. *SIAM J. Optimiz.* **20**(6), 3185–3207 (2010)
19. Dorea, C.: Expected number of steps of a random optimization method. *J. Optimiz. Th. Appl.* **39**(2), 165–171 (1983)
20. Eldar, E., Needell, D.: Acceleration of randomized Kaczmarz method via the Johnson Lindenstrauss Lemma *Numerical Algorithms* **58**(2), 163–177 (2011)
21. Ermoliev, Yu., Wets, R. (eds.): *Numerical Techniques for Stochastic Optimization*. Springer (1988)
22. Fisher, R.A.: *The Design of Experiments*. Oliver and Boyd, Edinburgh (1935)
23. Gilks, W.R., Richardson, S., Spiegelhalter, D.J.: *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London (1996)
24. Goldberg, D.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA (1989)
25. Goldensluger, A., Polyak, B.: Estimation of regression parameters with arbitrary noise. *Math. Meth. Stat.* **2**(1), 18–29 (1993)
26. Gower, M., Richtarik, P.: Randomized iterative methods for linear systems. *SIAM J. Matr. Anal. Appl.* **36**(4)1660–1690 (2015)
27. Granichin, O.: Estimating the parameters of linear regression in an arbitrary noise. *Autom. Remote Control* **63**(1), 25–35 (2002)

28. Granichin, O., Polyak, B.: Randomized Algorithms for Estimation and Optimization under Almost Arbitrary Noises. Nauka, Moscow (2003) (in Russian)
29. Granichin, O., Volkovich, Z., Toledano-Kitai, D.: Randomized Algorithms in Automatic Control and Data Mining. Springer, Berlin-Heidelberg (2015)
30. Gryazina, E.N., Polyak, B.: Random sampling: Billiard Walk algorithm. *Eur. J. Oper. Res.* **238**(2), 497–504 (2014)
31. Gubin, L., Polyak, B., Raik, E.: The method of projections for finding the common point of convex sets. *USSR Comput. Math. Math. Phys.* **7**(6), 1–24 (1967)
32. Gupal, A.: A method for the minimization of almost-differentiable functions. *Cybernetics.* (1), 115–117 (1977)
33. Gürbüzbalaban, M. Ozdaglar, A., Parrilo, P.: Why random reshuffling beats stochastic gradient descent. [arXiv:1510.08560v2](https://arxiv.org/abs/1510.08560v2) [math.OC], May 1, 2018.
34. Heunis, A.J.: Use of Monte Carlo method in an algorithm which solves a set of functional inequalities. *J. Optim. Theory Appl.* **45**(1), 89–99 (1984)
35. Horst, R., Pardalos, Panos M. (eds.): Handbook of Global Optimization, vol. **1**. Kluwer, Dordrecht (1995)
36. Kaczmarz, S.: Angenäherte Auflösung von Systemen linearer Gleichungen. *Bull. Intern. Acad. Polon. Sci., Lett. A.* 355–357 (1937). English translation: Approximate solution of systems of linear equations. *Int. J. Control* **57**(6), 1269–1271 (1993)
37. Kroese, D.P., Taimre, T., Botev, Z.I.: Handbook of Monte Carlo Methods. John Wiley and Sons, New York (2011)
38. Kushner, H.J., Clark, D.S.: Stochastic Approximation Methods for Constrained and Unconstrained Systems. Vol. 26 of Applied Mathematical Sciences. Springer, New York (1978)
39. Lagoa, C.M., Li, X., Sznaier, M.: Probabilistically constrained linear programs and risk-adjusted controller design. *SIAM J. Optimiz.* **15**(3), 938–951 (2005)
40. Lagoa, C.M., Shcherbakov, P.S., Barmish, B.R.: Probabilistic enhancement of classical robustness margins: The unirectangularity concept. *Syst. Control Lett.* **35**(1), 31–43 (1998)
41. Leventhal, D., Lewis, A.S.: Randomized methods for linear constraints: Convergence rates and conditioning. *Math. Oper. Res.* **35**(3) 641–654 (2010)
42. Matyas, J.: Random optimization. *Autom. Remote Control* **26**(2), 246–253 (1965)
43. Metropolis, N., Ulam S.: The Monte Carlo method. *J. Amer. Stat. Assoc.* **44**(247), 335–341 (1949)
44. Motzkin, T.S., Shoenberg, I.J.: The relaxation method for linear inequalities. *Canad. J. Math.* **6**, 393–404 (1954)
45. Nedic, A.: Random algorithms for convex minimization problems. *Math. Progr.* **129**(2), 225–253 (2011)
46. Nemirovskii, A.S, Polyak, B.T.: Necessary conditions for the stability of polynomials and their use. *Autom. Remote Control* **55**(11), 1644–1649 (1994)
47. Nemirovski, A., Yudin, D.B.: Problem Complexity and Method Efficiency in Optimization. John Wiley and Sons, New York (1983)
48. Nesterov, Yu.: Introductory Lectures on Convex Optimization: A Basic Course. Klüwer (2004)
49. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optimiz.* **22**(2), 341–362 (2012)
50. Nesterov, Yu., Spokoiny, V.: Random gradient-free minimization of convex functions. *Foundations Comput. Math.* **17**(2), 527–566 (2017)
51. Novikova, N.M.: Stochastic quasi-gradient method for minimax seeking. *USSR Comp. Math. Math. Phys.* **17**, 91–99 (1977)
52. Pardalos, Panos M., Romeijn, H. Edwin (eds.): Handbook of Global Optimization, vol. **2**. Kluwer, Dordrecht (2002)
53. Petrikevich, Ya. I.: Randomized methods of stabilization of the discrete linear systems. *Autom. Remote Control* **69**(11), 1911–1921 (2008)
54. Petrikevich, Ya.I., Polyak, B.T., Shcherbakov, P.S.: Fixed-order controller design for SISO systems using Monte Carlo technique. In: Proc. 9th IFAC Workshop “Adaptation and Learning in Control and Signal Processing” (ALCOSP’07) St.Petersburg, Russia (2007)

55. Polyak, B.T.: Introduction to Optimization. Optimization Software, New York (1987)
56. Polyak, B.: Random algorithms for solving convex inequalities. In: Butnariu, D., Censor, Y., Reich, S. (eds.) *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, pp. 409–422. Elsevier (2001)
57. Polyak, B.T., Gryazina, E.N.: Randomized methods based on new Monte Carlo schemes for control and optimization. *Ann. Oper. Res.* **189**(1), 342–356 (2011)
58. Polyak, B.T., Shcherbakov, P.S.: A probabilistic approach to robust stability of time delay systems. *Autom. Remote Control* **57**(12), 1770–1779 (1996)
59. Polyak, B.T., Shcherbakov, P.S.: Random spherical uncertainty in estimation and robustness. *IEEE Trans. Autom. Control* **45**(11), 2145–2150 (2000)
60. Polyak, B., Shcherbakov, P.: Why does Monte Carlo Fail to Work Properly in High-Dimensional Optimization Problems? *J. Optim. Th. Appl.* **173**(2), 612–627 (2017)
61. Polyak, B.T., Tempo, R.: Probabilistic robust design with linear quadratic regulators. *Syst. Control Lett.* **43**(5), 343–353 (2001)
62. Polyak, B.T., Tsybakov, A.B.: Optimal order of accuracy for search algorithms in stochastic optimization. *Problems Inform. Transmiss.* **26**(2), 126–133 (1990)
63. Polyak, B.T., Tsybakov, A.B.: On stochastic approximation with arbitrary noise (the KW case). In: Khas'minskii, R.Z. (ed.) *Topics in Nonparametric Estimation. Advances in Soviet Math.* **12**, 107–113 (1992)
64. Polyak, B., Yuditskij A.: Acceleration of stochastic approximation procedures by averaging. *SIAM J. on Control Optimiz.* **30**(4), 838–855 (1992)
65. Rastrigin, L.A.: *Statistical Search Method*. Nauka, Moscow (1968) in Russian
66. Richtárik, P., Tacáč, M.: Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Math. Progr.* 2014, **144**(1–2), 1–38 (2014)
67. Robert, C.P., Casella, G.: *Monte Carlo Statistical Methods*. Springer-Verlag, New York (1999)
68. Shcherbakov, P.: Boundary oracles for control-related matrix sets. In: Proc. 19th Int. Symp. “Mathematical Theory of Networks and Systems” (MTNS-2010), Budapest, Hungary, Jul 5–9, 2010, pp. 665–670.
69. Shcherbakov, P., Dabbene, F.: On the generation of random stable polynomials. *Eur. J. Control* **17**(2), 145–159 (2011)
70. Simon, D.: *Evolutionary Optimization Algorithms*. Wiley, New York (2013)
71. Smith, R.L.: Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Oper. Res.* **32**(6), 1296–1308 (1984)
72. Solis, F.J., Wets, R.J-B.: (1981). Minimization by random search techniques. *Math. Oper. Res.* **6**(1), 19–13 (1981)
73. Spall, J.C.: *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Vol. 64 of Wiley-Interscience series in discrete mathematics and optimization. John Wiley and Sons, Hoboken, NJ (2003)
74. Stengel, R.F., Ray L.R.: Stochastic robustness of linear time-invariant control systems. *IEEE Trans. Autom. Control* **36**(1), 82–87 (1991)
75. Strohmer, T., Vershynin, R.: A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.* **15**(2), 262–278 (2009)
76. Tempo, R., Bai, Er-Wei, Dabbene, F.: Probabilistic robustness analysis: Explicit bounds for the minimum number of samples *Syst. Control Lett.* **30**(5), 237–242 (1997)
77. Tempo, R., Calafiore, G., Dabbene, F.: *Randomized Algorithms for Analysis and Control of Uncertain Systems, with Applications*. Springer, London (2013)
78. Tremba, A., Calafiore, G., Dabbene, F., Gryazina, E., Polyak, B., Shcherbakov, P., Tempo, R.: RACT: Randomized algorithms control toolbox for MATLAB. In: Proc. 17th World Congress of IFAC, Seoul, pp. 390–395 (2008)
79. Turchin, V.F.: On the computation of multidimensional integrals by the Monte-Carlo method. *Theory of Probability and its Applications*, **16**(4), 720–724 (1972)
80. Volkov, Y.V., Zavriev, S.K.: A general stochastic outer approximation method. *SIAM J. Control Optimiz.* **35**(4), 1387–1421 (1997)

81. Yakubovich, V.A.: Finite terminating algorithms for solving countable systems of inequalities and their application in problems of adaptive systems *Doklady AN SSSR* **189**, 495–498 (1969) (in Russian)
82. Zhigljavsky, A., Žilinskas, A.: *Stochastic Global Optimization*. Springer Science+Business Media, New York (2008)