Found Comput Math (2017) 17:527–566
DOI 10.1007/s10208-015-9296-2

FOUNDATIONS OF COMPUTATIONAL MATHEMATICS
The Journal of the Society for the Foundations of Computational Mathematics

Random Gradient-Free Minimization of Convex Functions

Yurii Nesterov[1] · **Vladimir Spokoiny**[2]

Received: 15 July 2013 / Revised: 4 May 2015 / Accepted: 24 September 2015 /
Published online: 30 November 2015

Abstract In this paper, we prove new complexity bounds for methods of convex optimization based only on computation of the function value. The search directions of our schemes are normally distributed random Gaussian vectors. It appears that such methods usually need at most n times more iterations than the standard gradient methods, where n is the dimension of the space of variables. This conclusion is true for both nonsmooth and smooth problems. For the latter class, we present also an accelerated scheme with the expected rate of convergence $O\left(\frac{n^2}{k^2}\right)$, where k is the iteration counter. For stochastic optimization, we propose a zero-order scheme and justify its expected rate of convergence $O\left(\frac{n}{k^{1/2}}\right)$. We give also some bounds for the rate of convergence of the random gradient-free methods to stationary points of nonconvex functions, for both smooth and nonsmooth cases. Our theoretical results are supported by preliminary computational experiments.

Keywords Convex optimization · Stochastic optimization · Derivative-free methods · Random methods · Complexity bounds

Communicated by Michael Overton.

Y. Nesterov: This work was done in affiliation with Higher School of Economics, Moscow.

✉ Yurii Nesterov
yurii.nesterov@uclouvain.be

[1] Center for Operations Research and Econometrics (CORE), Catholic University of Louvain (UCL), Leuven, Belgium

[2] Weierstrass Institute for Applied Analysis and Stochastics (WIAS), Humboldt University of Berlin, Berlin, Germany

Mathematics Subject Classification 90C25 · 0C47 · 68Q25

1 Introduction

1.1 Motivation

Derivative-free optimization methods were among the first schemes suggested in the early days of the development of optimization theory [12]. These methods have an evident advantage of a simple preparatory stage (the program of computation of the function value is always much simpler than the program for computing the vector of the gradient). However, very soon it was realized that these methods are much more difficult for theoretical investigation. For example, even for moderate dimension, the famous method by Nelder and Mead [13] has only an empirical justification up to now (justification for low-dimensional problems were given in [10,11]). Moreover, the possible rate of convergence of the derivative-free methods (established usually on an empirical level) is far below the efficiency of the usual optimization schemes.

On the other hand, as it was established in the beginning of 1980s, any function, represented by an explicit sequence of differentiable operations, can be automatically equipped with a program for computing the whole vector of its partial derivatives. Moreover, the complexity of this program is at most four times bigger than the complexity of computation of the initial function (this technique is called *Fast Differentiation*, or a backward mode of *Automatic Differentiation*). It seems that this observation destroyed the last arguments for supporting the idea of derivative-free optimization. During several decades, these methods were almost out of computational practice.

However, in the last years, we can see a restoration of the interest to this topic. The current state of the art in this field was recently updated by a comprehensive monograph [5]. It appears that, despite serious theoretical objections, the derivative-free methods can probably find their place on the software market. For that, there exist at least several reasons.

- In many applied fields, there exist some models, which are represented by an old black-box software for computing only the values of the functional characteristics of the problem. Modification of this software is either too costly or impossible.
- There exist some restrictions for applying the Fast Differentiation technique. In particular, it is necessary to store the results of *all* intermediate computations. Clearly, for some applications, this is impractical by memory limitations.
- In any case, creation of a program for computing partial derivatives requires some (substantial) efforts of a qualified programmer. Very often his/her working time is much more expensive than the computational time. Therefore, in some situations it is reasonable to buy a cheaper software and accept significantly increased computational time.
- Finally, the extension of the notion of the gradient onto nonsmooth case is a nontrivial operation. The generalized gradient *cannot* be formed by partial derivatives. The most popular framework for defining the *set* of local differential characteristics (*Clarke subdifferential* [4]) suffers from an incomplete chain rule. The only known technique for automatic computations of such characteristics (*lexicographic differ-*

entiation [17]) requires an increase in complexity of function evaluation in $O(n)$ times, where n is the number of variables.

Thus, it is interesting to develop the derivative-free optimization methods and obtain the theoretical bounds for their performance. It is interesting that such bounds are almost absent in this field (see, for example, [5]). One of the few exceptions is a derivative-free version of cutting plane method presented in Section 9.2 of [15] and improved by [21].

In this paper, we present several *random* derivative-free methods and provide them with some complexity bounds for different classes of *convex* optimization problems. As we will see, the complexity analysis is crucial for finding the reasonable values of their parameters.

Our approach can be seen as a combination of several popular ideas. First of all, we mention the *random optimization approach* [12], as applied to the problem

$$\min_{x \in R^n} f(x), \tag{1}$$

where f is a differentiable function. It was suggested to sample a point y randomly around the current position x (in accordance with Gaussian distribution) and move to y if $f(y) < f(x)$. The performance of this technique for nonconvex functions was estimated in [6] and criticized by [22] from the numerical point of view.

Different improvements of the random search idea were discussed in Section 3.4 [20]. In particular, it was mentioned that the scheme

$$x_{k+1} = x_k - h_k \tfrac{f(x_k+\mu_k u)-f(x_k)}{\mu_k} u, \tag{2}$$

where u is a random vector distributed uniformly over the unit sphere and converges under assumption $\mu_k \to 0$. However, no explicit rules for choosing the parameters were given, and no particular rate of convergence was established.

The main goal of this paper is the complexity analysis of different variants of method (2) and its accelerated versions. We study these methods for both smooth and nonsmooth optimization problems. It appears that the most powerful version of the scheme (2) corresponds to $\mu_k \to 0$. Then we get the following process:

$$x_{k+1} = x_k - h_k f'(x_k, u)u, \tag{3}$$

where $f'(x, u)$ is a directional derivative of function $f(x)$ along $u \in R^n$. As compared with the gradient, directional derivative is a much simpler object. Its value can be easily computed even for nonconvex nonsmooth functions by a forward differentiation. Or it can be approximated very well by finite differences. Note that in the gradient schemes, the target accuracy ϵ for problem (1) is not very high. Hence, as we will see, the accuracy of the finite differences can be kept on a reasonable level.

For our technique, it is convenient to work with a normally distributed Gaussian vector $u \in R^n$. Then we can define

$$g_0(x) \stackrel{\text{def}}{=} f'(x, u)u.$$

It appears that for convex f, vector $E_u(g_0(x))$ is always a subgradient of f at x.

Thus, we can treat the process (3) as a method with *random oracle*. Usually, these methods are analyzed in the framework of stochastic approximation (see [14] for the state of art of the field). However, our random oracle is very special. The standard assumption in stochastic approximation is the boundedness of the second moment of the random estimate $\nabla_x F(x,u)$ of the gradient for the objective function $f(x) = E_u(F(x,u))$:

$$E_u(\|\nabla_x F(x,u)\|_2^2) \leq M^2, \quad x \in R^n. \tag{4}$$

(see, for example, condition (2.5) in [14]). However, in our case, if f is differentiable at x, then

$$E_u(\|g_0(x)\|_2^2) \leq (n+4)\|\nabla f(x)\|_2^2.$$

This relation makes the analysis of our methods much simpler and leads to the faster schemes. In particular, for the method (3) as applied to Lipschitz-continuous functions, we can prove that the expected rate of convergence of the objective function is of the order $O(\sqrt{\frac{n}{k}})$. If a function has Lipschitz-continuous gradient, then the rate is increased up to $O(\frac{n}{k})$. If in addition, our function is strongly convex, then we have a global linear rate of convergence. Note that in the smooth case, using the technique of estimate sequences (e.g., Section 2.2 in [16]), we can accelerate method (3) up to convergence rate $O(\frac{n^2}{k^2})$.

For justifying the versions of random search methods with $\mu_k > 0$, we use a smoothed version of the objective function

$$f_\mu(x) = E_u(f(x+\mu u)). \tag{5}$$

This object is classical in optimization theory. For the complexity analysis of the random search methods, it was used, for example, in Section 9.3 [15][1] However, in their analysis the authors used the first part of the representation

$$\nabla f_\mu(x) = \frac{1}{\mu} E_u(f(x+\mu u)u) \overset{(!)}{\equiv} \frac{1}{\mu} E_u([f(x+\mu u) - f(x)]u).$$

In our analysis, we use the second part, which is bounded in μ. Hence, our conclusions are more optimistic.

Our results complement a series of developments in the machine learning community, related to randomized algorithms based on zero-order oracles. First algorithms of this type were proposed in [8] under the name of *bandit convex optimization* for a noisy oracle. The obtained complexity results were of the order $\epsilon^{-1/4}$ for Lipschitz-continuous convex functions. Another important contribution is [1], where the authors consider a noisy zero-order oracle and obtain complexity results for different classes

[1] In [15], u was uniformly distributed over a unit ball. In our comparison, we use a direct translation of the constructions in [15] into the language of the normal Gaussian distribution.

of convex functions (e.g., $O(\frac{n^4}{\epsilon^2})$ for Lipschitz-continuous functions). In [2], the model of the oracle admits even more noise. It seems that the methods with the absence of noise were not in the main focus of this line of research.

Randomized optimization algorithms were intensively studied in the theoretical computer science literature in the framework of random walks in convex sets (e.g., [3]). For global optimization, many authors were applying randomization ideas (e.g., [9]; see also [7] for relevant lower bounds). In our approach, we significantly simplify the analysis allowing random displacements in the full neighborhood of the current test point.

This paper is an extended version of preprint [18].

1.2 Contents

In Sect. 2, we introduce the *Gaussian smoothing* (5) and study its properties. In particular, for different functional classes, we estimate the error of approximation of the objective function and the gradient with respect to the smoothing parameter μ. The proofs of all statements of this section can be found in "Appendix".

In Sect. 3, we introduce the *random gradient-free oracles*, which are based either on finite differences or on directional derivatives. The main results of this section are the upper bounds for the expected values of squared norms of these oracles. In Sect. 4, we apply the simple random search method to a nonsmooth convex optimization problem with simple convex constraints. We show that the scheme (3) works at most in $O(n)$ times slower than the usual subgradient method. For the finite-difference version (2), this factor is increased up to $O(n^2)$. Both methods can be naturally modified to be used for stochastic programming problems.

In Sect. 5, we estimate the performance of method (2) on smooth optimization problems. We show that, under proper choice of parameters, it works at most n times slower than the usual gradient method. In Sect. 6, we consider an accelerated version of this scheme with the convergence rate $O(\frac{n^2}{k^2})$. For all methods, we derive the upper bounds for the value of the smoothing parameter μ. It appears that in all situations, their dependence in ϵ and n is quite moderate. For example, for the fast random search presented in Sect. 6, the average size of the trial step μu is of the order $O(n^{-1/2}\epsilon^{3/4})$, where ϵ is the target accuracy for solving (1). For the simple random search, this average size is even better: $O(n^{-1/2}\epsilon^{1/2})$.

In Sect. 7, we estimate a rate of convergence for the random search methods to a stationary point of a nonconvex function (in terms of the norm of the gradient). We consider both smooth and nonsmooth cases. Finally, in Sect. 8, we present the preliminary computational results. In the tested methods, we were checking the validity of our theoretical conclusions on stability and the rate of convergence of the scheme, as compared with the prototype gradient methods.

1.3 Notation

For a finite-dimensional space E, we denote by E^* its dual space. The value of a linear function $s \in E^*$ at point $x \in E$ is denoted by $\langle s, x \rangle$. We endow the spaces E and E^* with Euclidean norms

$$\|x\| = \langle Bx, x\rangle^{1/2}, \ x \in E, \quad \|s\|_* = \langle s, B^{-1}s\rangle^{1/2}, \ s \in E^*,$$

where $B = B^* \succ 0$ is a linear operator from E to E^*. For any $u \in E$, we denote by uu^* a linear operator from E^* to E, which acts as follows:

$$uu^*(s) = u \cdot \langle s, u\rangle, \quad s \in E^*.$$

In this paper, we consider functions with different levels of smoothness. It is indicated by the following notation.

- $f \in C^{0,0}(E)$ if $|f(x) - f(y)| \leq L_0(f)\|x - y\|$, $x, y \in E$.
- $f \in C^{1,1}(E)$ if $\|\nabla f(x) - \nabla f(y)\|_* \leq L_1(f)\|x - y\|$, $x, y \in E$. This condition is equivalent to the following inequality:

$$|f(y) - f(x) - \langle \nabla f(x), y - x\rangle| \leq \tfrac{1}{2}L_1(f)\|x - y\|^2, \quad x, y \in E. \tag{6}$$

- $f \in C^{2,2}(E)$ if $\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L_2(f)\|x - y\|$, $x, y \in E$. This condition is equivalent to the inequality

$$\begin{aligned} &|f(y) - f(x) - \langle \nabla f(x), y - x\rangle - \tfrac{1}{2}\langle \nabla^2 f(x)(y - x), y - x\rangle| \\ &\quad \leq \tfrac{1}{6}L_2(f)\|x - y\|^3, \quad x, y \in E. \end{aligned} \tag{7}$$

We say that $f \in C^{1,1}(E)$ is strongly convex, if for any x and $y \in E$ we have

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \tfrac{\tau(f)}{2}\|y - x\|^2, \tag{8}$$

where $\tau(f) \geq 0$ is the *convexity parameter*.

Let $\epsilon \geq 0$. For convex function f, we denote by $\partial f_\epsilon(x)$ its ϵ-subdifferential at $x \in E$:

$$f(y) \geq f(x) - \epsilon + \langle g, y - x\rangle, \quad g \in \partial f_\epsilon(x), \ y \in E.$$

If $\epsilon = 0$, we simplify this notation to $\partial f(x)$.

2 Gaussian Smoothing

Consider a function $f : E \to R$. We assume that at each point $x \in E$, it is differentiable along any direction. Let us form its *Gaussian approximation*

$$f_\mu(x) = \tfrac{1}{\kappa}\int_E f(x + \mu u)e^{-\frac{1}{2}\|u\|^2}du, \tag{9}$$

where

$$\kappa \stackrel{\text{def}}{=} \int_E e^{-\frac{1}{2}\|u\|^2}du = \tfrac{(2\pi)^{n/2}}{[\det B]^{1/2}}. \tag{10}$$

All results of this section, related to the properties of this function, are rather general. Therefore, we put their proofs in "Appendix".

As we will see later, for $\mu > 0$ function f_μ is always differentiable, and $\mu \geq 0$ plays a role of smoothing parameter. Clearly, $\frac{1}{\kappa} \int\limits_E u e^{-\frac{1}{2}\|u\|^2} du = 0$. Therefore, if f is convex and $g \in \partial f(x)$, then

$$f_\mu(x) \geq \frac{1}{\kappa} \int\limits_E [f(x) + \mu\langle g, u\rangle] e^{-\frac{1}{2}\|u\|^2} du \ = \ f(x). \tag{11}$$

Note that in general, f_μ has better properties than f. At least, all initial characteristics of f are preserved by any f_μ with $\mu \geq 0$.

- If f is convex, then f_μ is also convex.
- If $f \in C^{0,0}$, then $f_\mu \in C^{0,0}$ and $L_0(f_\mu) \leq L_0(f)$. Indeed, for all $x, y \in E$ we have

$$|f_\mu(x) - f_\mu(y)| \leq \frac{1}{\kappa} \int\limits_E |f(x + \mu u) - f(y + \mu u)| e^{-\frac{1}{2}\|u\|^2} du \ \leq \ L_0(f)\|x - y\|.$$

- If $f \in C^{1,1}$, then $f_\mu \in C^{1,1}$ and $L_1(f_\mu) \leq L_1(f)$:

$$\begin{aligned} \|\nabla f_\mu(x) - \nabla f_\mu(y)\|_* &\leq \frac{1}{\kappa} \int\limits_E \|\nabla f(x + \mu u) - \nabla f(y + \mu u)\|_* e^{-\frac{1}{2}\|u\|^2} du \\ &\leq L_1(f)\|x - y\|, \quad x, y \in E. \end{aligned} \tag{12}$$

From definition (10), we get also the identity

$$\ln \int\limits_E e^{-\frac{1}{2}\langle Bu,u\rangle} du \equiv \frac{n}{2} \ln(2\pi) - \frac{1}{2} \ln \det B.$$

Differentiating this identity in B, we get the following representation:

$$\frac{1}{\kappa} \int\limits_E uu^* e^{-\frac{1}{2}\|u\|^2} du \ = \ B^{-1}. \tag{13}$$

Taking a scalar product of this equality with B, we obtain

$$\frac{1}{\kappa} \int\limits_E \|u\|^2 e^{-\frac{1}{2}\|u\|^2} du = n. \tag{14}$$

In what follows, we often need upper bounds for the moments $M_p \stackrel{\text{def}}{=} \frac{1}{\kappa} \int\limits_E \|u\|^p e^{-\frac{1}{2}\|u\|^2} du$. We have exact simple values for two cases:

$$M_0 \stackrel{(10)}{=} 1, \quad M_2 \stackrel{(14)}{=} n. \tag{15}$$

For other cases, we will use the following simple bounds.

Lemma 1 *For $p \in [0, 2]$, we have*

$$M_p \le n^{p/2}. \tag{16}$$

If $p \ge 2$, then we have two-side bounds

$$n^{p/2} \ \le \ M_p \le (p+n)^{p/2}. \tag{17}$$

Now we can prove the following useful result.

Theorem 1 *Let $f \in C^{0,0}(E)$, then*

$$|f_\mu(x) - f(x)| \le \mu L_0(f) n^{1/2}, \quad x \in E. \tag{18}$$

If $f \in C^{1,1}(E)$, then

$$|f_\mu(x) - f(x)| \le \tfrac{\mu^2}{2} L_1(f) n, \quad x \in E. \tag{19}$$

Finally, if $f \in C^{2,2}(E)$, then

$$|f_\mu(x) - f(x) - \tfrac{\mu^2}{2}\langle \nabla^2 f(x), B^{-1}\rangle| \le \tfrac{\mu^3}{3} L_2(f)(n+3)^{3/2}, \quad x \in E. \tag{20}$$

Inequality (20) shows that increasing the level of smoothness of function f beyond $C^{1,1}(E)$ cannot improve the quality of approximation of f by f_μ. If, for example, f is quadratic and $\nabla^2 f(x) \equiv G$, then

$$f_\mu(x) \overset{(20)}{=} f(x) + \tfrac{\mu^2}{2}\langle G, B^{-1}\rangle.$$

The constant term in this identity can reach the right-hand side of inequality (19).

For any positive μ, function f_μ is differentiable. Let us obtain a convenient expression for its gradient. For that, we rewrite definition (9) in another form by introducing a new integration variable $y = x + \mu u$:

$$f_\mu(x) = \tfrac{1}{\mu^n \kappa} \int_E f(y) e^{-\frac{1}{2\mu^2}\|y-x\|^2} dy.$$

Since the value and the partial derivative in x of the argument of this integral are continuous in (x, y), we can apply the standard differentiation rule for finding the gradient:

$$\nabla f_\mu(x) = \frac{1}{\mu^{n+2}\kappa} \int_E f(y) e^{-\frac{1}{2\mu^2}\|y-x\|^2} B(y-x)\, dy$$

$$= \frac{1}{\mu\kappa} \int_E f(x+\mu u) e^{-\frac{1}{2}\|u\|^2} Bu\, du \tag{21}$$

$$= \frac{1}{\kappa} \int_E \frac{f(x+\mu u) - f(x)}{\mu} e^{-\frac{1}{2}\|u\|^2} Bu\, du.$$

It appears that this gradient is Lipschitz-continuous even if the gradient of f is not.

Lemma 2 *Let $f \in C^{0,0}(E)$ and $\mu > 0$. Then $f_\mu \in C^{1,1}(E)$ with*

$$L_1(f_\mu) = \frac{n^{1/2}}{\mu} L_0(f). \tag{22}$$

Denote by $f'(x,u)$ the directional derivative of f at point x along direction u:

$$f'(x,u) = \lim_{\alpha \downarrow 0} \frac{1}{\alpha}[f(x+\alpha u) - f(x)]. \tag{23}$$

Then we can define the limiting vector of the gradients (21):

$$\nabla f_0(x) = \frac{1}{\kappa} \int_E f'(x,u) e^{-\frac{1}{2}\|u\|^2} Bu\, du. \tag{24}$$

Note that at each $x \in E$, the vector (24) is uniquely defined. If f is differentiable at x, then

$$\nabla f_0(x) = \frac{1}{\kappa} \int_E \langle \nabla f(x), u\rangle e^{-\frac{1}{2}\|u\|^2} Bu\, du \overset{(13)}{=} \nabla f(x). \tag{25}$$

Let us prove that in convex case, $\nabla f_\mu(x)$ always belongs to some ϵ-subdifferential of function f.

Theorem 2 *Let f be convex and Lipschitz continuous. Then, for any $x \in E$ and $\mu \geq 0$, we have*

$$\nabla f_\mu(x) \in \partial_\epsilon f(x), \quad \epsilon = \mu L_0(f) n^{1/2}.$$

Note that expression (21) can be rewritten in the following form:

$$\begin{aligned} \nabla f_\mu(x) &= \frac{1}{\kappa} \int_E \frac{f(x) - f(x-\mu u)}{\mu} e^{-\frac{1}{2}\|u\|^2} Bu\, du \\ &\overset{(21)}{=} \frac{1}{\kappa} \int_E \frac{f(x+\mu u) - f(x-\mu u)}{2\mu} e^{-\frac{1}{2}\|u\|^2} Bu\, du. \end{aligned} \tag{26}$$

Lemma 3 *If $f \in C^{1,1}(E)$, then*

$$\|\nabla f_\mu(x) - \nabla f(x)\|_* \le \frac{\mu}{2} L_1(f)(n+3)^{3/2}. \tag{27}$$

For $f \in C^{2,2}(E)$, we can guarantee that

$$\|\nabla f_\mu(x) - \nabla f(x)\|_* \le \frac{\mu^2}{6} L_2(f)(n+4)^2. \tag{28}$$

Finally, we prove one more relation between the gradients of f and f_μ.

Lemma 4 *Let $f \in C^{1,1}(E)$. Then, for any $x \in E$, we have*

$$\|\nabla f(x)\|_*^2 \le 2\|\nabla f_\mu(x)\|_*^2 + \frac{\mu^2}{2} L_1^2(f)(n+6)^3. \tag{29}$$

3 Random Gradient-Free Oracles

Let random vector $u \in E$ have Gaussian distribution with correlation operator B^{-1}. Denote by $E_u(\psi(u))$ the expectation of corresponding random variable. For $\mu \ge 0$, using expressions (21), (26), and (24), we can define the following *random gradient-free oracles*:

$$\begin{aligned} &\mathbf{1}. \text{ Generate random} u \in E \text{ and return } g_\mu(x) = \tfrac{f(x+\mu u)-f(x)}{\mu} \cdot Bu.\\ &\mathbf{2}. \text{ Generate random} u \in E \text{ and return } \hat{g}_\mu(x) = \tfrac{f(x+\mu u)-f(x-\mu u)}{2\mu} \cdot Bu.\\ &\mathbf{3}. \text{ Generate random} u \in E \text{ and return } g_0(x) = f'(x,u) \cdot Bu. \end{aligned} \tag{30}$$

As we will see later, oracles g_μ and $\hat{g}_\mu$ are more suitable for minimizing smooth functions. Oracle g_0 is more universal. It can be also used for minimizing nonsmooth convex functions. Recall that in view of (24) and Theorem 2, we have[2]

$$E_u(g_0(x)) = \nabla f_0(x) \; \in \; \partial f(x). \tag{31}$$

We can establish now several useful upper bounds. First of all, note that for function f differentiable at point x, we have

$$\|g_0(x)\|_*^2 = \langle \nabla f(x), u\rangle^2 \cdot \|u\|^2 \;\le\; \|\nabla f(x)\|_*^2 \cdot \|u\|^4.$$

[2] Presence of this oracle is the main reason why we call our methods *gradient* free (not *derivative free*!). Indeed, directional derivative is a much simpler object as compared with the gradient. It can be easily defined for a very large class of functions. At the same time, definition of the *gradient* (or subgradient) is much more involved. It is well known that in nonsmooth case, collection of partial derivatives *is not* a subgradient of convex function. For nonsmooth nonconvex functions, the possibility of computing a single subgradient needs a serious mathematical justification [17]. On the other hand, if we have an access to a program for computing the value of our function, then the program for computing directional derivatives can be obtained by a trivial automatic *forward* differentiation.

Hence, $E_u(\|g_0(x)\|_*^2) \overset{(17)}{\leq} (n+4)^2 \|\nabla f(x)\|_*^2$. It appears that this bound can be significantly strengthened.

Theorem 3 1. *If f is differentiable at x, then*

$$E_u(\|g_0(x)\|_*^2) \leq (n+4)\|\nabla f(x)\|_*^2. \tag{32}$$

2. *Let f be convex. Denote $D(x) = \operatorname{diam} \partial f(x)$. Then, for any $x \in E$ we have*

$$E_u(\|g_0(x)\|_*^2) \leq (n+4)\left(\|\nabla f_0(x)\|_*^2 + nD^2(x)\right). \tag{33}$$

Proof Indeed, let us fix $\tau \in (0, 1)$. Then,

$$\begin{aligned}
E_u(\|g_0(x)\|_*^2) &\overset{(30)}{=} \frac{1}{\kappa} \int_E \|u\|^2 \mathrm{e}^{-\frac{1}{2}\|u\|^2} f'(x,u)^2 \mathrm{d}u \\
&= \frac{1}{\kappa} \int_E \|u\|^2 \mathrm{e}^{-\frac{\tau}{2}\|u\|^2} f'(x,u)^2 \mathrm{e}^{-\frac{1-\tau}{2}\|u\|^2} \mathrm{d}u \\
&\overset{(80)}{\leq} \frac{2}{\kappa\tau e} \int_E f'(x,u)^2 \mathrm{e}^{-\frac{1-\tau}{2}\|u\|^2} \mathrm{d}u \\
&= \frac{2}{\kappa\tau(1-\tau)^{1+n/2} e} \int_E f'(x,u)^2 \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u.
\end{aligned}$$

The minimum of the right-hand side in τ is attained for $\tau_* = \frac{2}{n+4}$. In this case,

$$\tau_*(1-\tau_*)^{\frac{n+2}{2}} = \frac{2}{n+4}\left(\frac{n+2}{n+4}\right)^{\frac{n+2}{2}} > \frac{2}{(n+4)e}.$$

Therefore,

$$E_u(\|g_0(x)\|_*^2) \leq \frac{n+4}{\kappa} \int_E f'(x,u)^2 \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u.$$

If f is differentiable at x, then $f'(x,u) = \langle \nabla f(x), u \rangle$, and we get (32) from (13).

Suppose that f is convex and not differentiable at x. Denote by $g(u)$ an arbitrary point from the set $\operatorname{Arg}\max_g \{\langle g, u \rangle : \; g \in \partial f(x)\}$. Then

$$f'(x,u)^2 = (\langle \nabla f_0(x), u \rangle + \langle g(u) - \nabla f_0(x), u \rangle)^2.$$

Note that

$$
\begin{aligned}
E_u(\langle \nabla f_0(x), u\rangle \cdot \langle g(u) - \nabla f_0(x), u\rangle) &\overset{(13)}{=} E_u(\langle \nabla f_0(x), u\rangle \cdot f'(x,u)) - \|\nabla f_0(x)\|_*^2 \\
&= \langle \nabla f_0(x), E_u(u \cdot f'(x,u))\rangle - \|\nabla f_0(x)\|_*^2 \\
&\overset{(24)}{=} 0.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
E_u(\|g_0(x)\|_*^2) &\le \tfrac{n+4}{\kappa} \int\limits_E \left(\langle \nabla f_0(x), u\rangle^2 + D^2(x)\|u\|^2\right) \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u \\
&\overset{(13)}{=} (n+4)\left(\|\nabla f_0(x)\|_*^2 + \tfrac{D^2(x)}{\kappa} \int\limits_E \|u\|^2 \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u\right) \\
&\overset{(14)}{=} (n+4)\left(\|\nabla f_0(x)\|_*^2 + nD^2(x)\right).
\end{aligned}
$$

□

Let us prove now the similar bounds for oracles g_μ and $\hat{g}_\mu$.

Theorem 4 *Let function f be convex.*
1. *If $f \in C^{0,0}(E)$, then*

$$
E_u(\|g_\mu(x)\|_*^2) \le L_0^2(f)(n+4)^2. \tag{34}
$$

2. *If $f \in C^{1,1}(E)$, then*

$$
\begin{aligned}
E_u(\|g_\mu(x)\|_*^2) &\le \tfrac{\mu^2}{2} L_1^2(f)(n+6)^3 + 2(n+4)\|\nabla f(x)\|_*^2, \\
E_u(\|\hat{g}_\mu(x)\|_*^2) &\le \tfrac{\mu^2}{8} L_1^2(f)(n+6)^3 + 2(n+4)\|\nabla f(x)\|_*^2.
\end{aligned} \tag{35}
$$

3. *If $f \in C^{2,2}(E)$, then*

$$
E_u(\|\hat{g}_\mu(x)\|_*^2) \le \tfrac{\mu^4}{18} L_2^2(f)(n+8)^4 + 2(n+4)\|\nabla f(x)\|_*^2. \tag{36}
$$

Proof Note that $E_u(\|g_\mu(x)\|_*^2) = \frac{1}{\mu^2} E_u\left([f(x+\mu u) - f(x)]^2]\|u\|^2\right)$. If $f \in C^{0,0}(E)$, then we obtain (34) directly from the definition of the functional class and (17).

Let $f \in C^{1,1}(E)$. Since

$$
\begin{aligned}
[f(x+\mu u) - f(x)]^2 &= [f(x+\mu u) - f(x) - \mu\langle \nabla f(x), u\rangle + \mu\langle \nabla f(x), u\rangle]^2 \\
&\overset{(6)}{\le} 2\left(\tfrac{\mu^2}{2} L_1(f)\|u\|^2\right)^2 + 2\mu^2\langle \nabla f(x), u\rangle^2,
\end{aligned}
$$

we get

$$E_u(\|g_\mu(x)\|_*^2) \quad \leq \quad \frac{\mu^2}{2} L_1^2(f) E_u(\|u\|^6) + 2E_u(\|g_0(x)\|_*^2)$$
$$\stackrel{(17),(32)}{\leq} \frac{\mu^2}{2} L_1^2(f)(n+6)^3 + 2(n+4)\|\nabla f(x)\|_*^2.$$

For the symmetric oracle $\hat{g}_\mu$, since f is convex, we have

$$f(x+\mu u) - f(x-\mu u) = [f(x+\mu u) - f(x)] + [f(x) - f(x-\mu u)]$$
$$\stackrel{(6)}{\leq} \left[\mu\langle\nabla f(x), u\rangle + \frac{\mu^2}{2} L_1(f)\|u\|^2\right] + \mu\langle\nabla f(x), u\rangle.$$

Similarly, we have $f(x+\mu u) - f(x-\mu u) \geq 2\mu\langle\nabla f(x), u\rangle - \frac{\mu^2}{2} L_1(f)\|u\|^2$. Therefore,

$$E_u(\|\hat{g}_\mu(x)\|_*^2) \quad = \quad \frac{1}{4\mu^2} E_u\left([f(x+\mu u) - f(x-\mu u)]^2\|u\|^2\right)$$
$$\leq \quad \frac{1}{2\mu^2}\left[E_u\left(\frac{\mu^4}{4} L_1^2(f)\|u\|^6\right) + E_u\left(4\mu^2\langle\nabla f(x), u\rangle^2\|u\|^2\right)\right]$$
$$\stackrel{(17),(32)}{\leq} \frac{\mu^2}{8} L_1^2(f)(n+6)^3 + 2(n+4)\|\nabla f(x)\|_*^2.$$

Let $f \in C^{2,2}(E)$. We will use notation of Lemma 3. Since

$$[f(x+\mu u) - f(x-\mu u)]^2 = [f(x+\mu u) - f(x-\mu u) - 2\mu\langle\nabla f(x), u\rangle$$
$$+ 2\mu\langle\nabla f(x), u\rangle]^2 \leq 2[a_u(\mu) - a_u(-\mu)]^2$$
$$+ 8\mu^2\langle\nabla f(x), u\rangle^2 \stackrel{(7)}{\leq} \frac{2\mu^6}{9} L_2^2(f)\|u\|^6 + 8\mu^2\langle\nabla f(x), u\rangle^2,$$

we get

$$E_u(\|\hat{g}_\mu(x)\|_*^2) \quad \leq \quad \frac{\mu^4}{18} L_2^2(f) E_u(\|u\|^8) + 2E_u(\|g_0(x)\|_*^2)$$
$$\stackrel{(17),(32)}{\leq} \frac{\mu^4}{18} L_2^2(f)(n+8)^4 + 2(n+4)\|\nabla f(x)\|_*^2.$$

□

Sometimes it is more convenient to have in the right-hand side of inequality (35) the gradient of Gaussian approximation.

Lemma 5 *Let $f \in C^{1,1}(E)$. Then, for any $x \in E$ we have*

$$E_u(\|g_\mu(x)\|_*^2) \leq 4(n+4)\|\nabla f_\mu(x)\|_*^2 + 3\mu^2 L_1^2(f)(n+4)^3. \tag{37}$$

Proof Indeed,

$$(f(x+\mu u)-f(x))^2 = (f(x+\mu u)-f_\mu(x+\mu u)-f(x)+f_\mu(x)+f_\mu(x+\mu u)$$

$$-f_\mu(x))^2 \leq 2(f(x+\mu u)-f_\mu(x+\mu u)-f(x)+f_\mu(x))^2$$

$$+2(f_\mu(x+\mu u)-f_\mu(x))^2.$$

Note that $|f(x+\mu u)-f_\mu(x+\mu u)-f(x)+f_\mu(x)| \overset{(19)}{\leq} \mu^2 L_1(f)n$, and

$$(f_\mu(x+\mu u)-f_\mu(x))^2 \leq 2(f_\mu(x+\mu u)-f_\mu(x)-\mu\langle\nabla f_\mu(x),u\rangle)^2$$

$$+2\mu^2\langle\nabla f_\mu(x),u\rangle^2 \leq \tfrac{\mu^4}{2}L_1^2(f)\|u\|^4+2\mu^2\langle\nabla f_\mu(x),u\rangle^2.$$

Applying (32) to function f_μ, we get $E_u(\langle\nabla f_\mu(x),u\rangle^2\|u\|^2) \leq (n+4)\|\nabla f_\mu(x)\|_*^2$. Hence,

$$\begin{aligned} E_u(\|g_\mu(x)\|_*^2) &\leq \tfrac{1}{\mu^2}E_u((f(x+\mu u)-f(x))^2\|u\|^2) \\ &\leq 2\mu^2 L_1^2(f)n^2 M_2+\mu^2 L_1^2(f)M_6+4(n+4)\|\nabla f_\mu(x)\|_*^2 \\ &\leq \mu^2 L_1^2(f)(2n^3+(n+6)^3)+4(n+4)\|\nabla f_\mu(x)\|_*^2. \end{aligned}$$

It remains to note that $2n^3+(n+6)^3 \leq 3(n+4)^3$. □

Example $f(x)=\|x\|$, $x=0$, shows that the pessimistic bound (34) cannot be significantly improved.

4 Random Search for Nonsmooth and Stochastic Optimization

Unless otherwise noted, we assume that f is convex. Let us show how to use the oracles (30) for solving the following nonsmooth optimization problem:

$$f^* \overset{\text{def}}{=} \min_{x\in Q} f(x), \tag{38}$$

where $Q \subseteq E$ is a closed convex set and f is a nonsmooth convex function on E. Denote by $x^* \in Q$ one of its optimal solutions. Recall that we measure distances in E by the primal Euclidean norm $\|u\| = \langle Bu,u\rangle^{1/2}$, $u \in E$. Distances in E^* are measured by the conjugate norm: $\|g\|_* = \langle g, B^{-1}g\rangle^{1/2}$, $g \in E^*$.

Let us choose a sequence of positive steps $\{h_k\}_{k\geq 0}$. Consider the following method.

$$\boxed{\begin{array}{l} \textbf{Method } \mathcal{RS}_\mu : \text{Choose} x_0 \in Q. \text{If} \mu = 0, \text{we need} D(x_0) = 0. \\ \hline \\ \textbf{Iteration } k \geq 0. \\ \\ \text{a).Generate} u_k \text{and corresponding} g_\mu(x_k). \\ \\ \text{b).Compute} x_{k+1} = \pi_Q \left(x_k - h_k B^{-1} g_\mu(x_k) \right). \end{array}} \tag{39}$$

We use notation $\pi_Q(x)$ for Euclidean projection onto the closed convex set Q. Thus, $\|\pi_Q(x) - y\| \leq \|x - y\|$ for all $y \in Q$.

Method (39) generates random vectors $\{x_k\}_{k\geq 0}$. Denote by $\mathcal{U}_k = (u_0, \ldots, u_k)$ a random vector composed by independent and identically distributed variables $\{u_k\}_{k\geq 0}$ (i.i.d.) attached to each iteration of the scheme. Let $\phi_0 = f(x_0)$, and $\phi_k \stackrel{\text{def}}{=} E_{\mathcal{U}_{k-1}}(f(x_k))$, $k \geq 1$.

Theorem 5 *Let sequence $\{x_k\}_{k\geq 0}$ be generated by $\mathcal{RS}_0$. Then, for any $N \geq 0$ we have*

$$\sum_{k=0}^{N} h_k(\phi_k - f^*) \leq \tfrac{1}{2}\|x_0 - x^*\|^2 + \tfrac{n+4}{2} L_0^2(f) \sum_{k=0}^{N} h_k^2. \tag{40}$$

Proof Let point x_k with $k \geq 1$ be generated after k iterations of the scheme (39). Denote $r_k = \|x_k - x^*\|$. Then

$$r_{k+1}^2 \leq \|x_k - h_k g_0(x_k) - x^*\|^2 = r_k^2 - 2h_k\langle g_0(x_k), x_k - x^*\rangle + h_k^2\|g_0(x_k)\|_*^2.$$

Note that function f is differentiable at x_k with probability one. Therefore, using representation (25) and the estimate (32), we get

$$\begin{aligned} E_{u_k}\left(r_{k+1}^2\right) &\leq r_k^2 - 2h_k\langle \nabla f(x_k), x_k - x^*\rangle + h_k^2(n+4)L_0^2(f) \\ &\leq r_k^2 - 2h_k(f(x_k) - f^*) + h_k^2(n+4)L_0^2(f). \end{aligned}$$

Taking now the expectation in $\mathcal{U}_{k-1}$, we obtain

$$E_{\mathcal{U}_k}\left(r_{k+1}^2\right) \leq E_{\mathcal{U}_{k-1}}\left(r_k^2\right) - 2h_k(\phi_k - f^*) + h_k^2(n+4)L_0^2(f).$$

Using the same reasoning, we get

$$E_{\mathcal{U}_0}\left(r_1^2\right) \leq r_0^2 - 2h_0(f(x_0) - f^*) + h_0^2(n+4)L_0^2(f).$$

Summing up these inequalities, we come to (40). □

Denote $S_N = \sum_{k=0}^{N} h_k$, and define $\hat{x}_N = \arg\min_x [f(x) : x \in \{x_0, \ldots, x_N\}]$. Then

$$E_{\mathcal{U}_{N-1}}\left(f(\hat{x}_N)\right) - f^* \le E_{\mathcal{U}_{N-1}}\left(\frac{1}{S_N}\sum_{k=0}^{N} h_k(f(x_k) - f^*)\right)$$

$$\stackrel{(40)}{\le} \frac{1}{S_N}\left[\frac{1}{2}\|x_0 - x^*\|^2 + \frac{n+4}{2} L_0^2(f) \sum_{k=0}^{N} h_k^2\right].$$

In particular, if the number of steps N is fixed, and $\|x_0 - x^*\| \le R$, we can choose

$$h_k = \frac{R}{(n+4)^{1/2}(N+1)^{1/2}L_0(f)}, \quad k = 0, \ldots, N. \tag{41}$$

Then we obtain the following bound:

$$E_{\mathcal{U}_{N-1}}\left(f(\hat{x}_N)\right) - f^* \le L_0(f) R \left[\frac{n+4}{N+1}\right]^{1/2}. \tag{42}$$

Hence, inequality $E_{\mathcal{U}_{N-1}}\left(f(\hat{x}_N)\right) - f^* \le \epsilon$ can be ensured by $\mathcal{RS}_0$ in

$$\frac{n+4}{\epsilon^2} L_0^2(f) R^2 \tag{43}$$

iterations.

Same as in the standard nonsmooth minimization, instead of fixing the number of steps apriori, we can define

$$h_k = \frac{R}{(n+4)^{1/2}(k+1)^{1/2}L_0(f)}, \quad k \ge 0. \tag{44}$$

This modification results in a multiplication of the right-hand side of the estimate (42) by a factor $O(\ln N)$ (e.g., Section 3.2 in [16]).

Let us consider now the random search method (39) with $\mu > 0$.

Theorem 6 *Let sequence $\{x_k\}_{k\ge 0}$ be generated by $\mathcal{RS}_\mu$ with $\mu > 0$. Then, for any $N \ge 0$ we have*

$$\frac{1}{S_N}\sum_{k=0}^{N} h_k(\phi_k - f^*) \le \mu L_0(f) n^{1/2} + \frac{1}{S_N}\left[\frac{1}{2}\|x_0 - x^*\|^2 + \frac{(n+4)^2}{2} L_0^2(f) \sum_{k=0}^{N} h_k^2\right]. \tag{45}$$

Proof Let point x_k with $k \ge 1$ be generated after k iterations of the scheme (39). Denote $r_k = \|x_k - x^*\|$. Then

$$r_{k+1}^2 \le \|x_k - h_k g_\mu(x_k) - x^*\|^2 = r_k^2 - 2h_k\langle g_\mu(x_k), x_k - x^*\rangle + h_k^2\|g_\mu(x_k)\|_*^2.$$

Using representation (21) and the estimate (34), we get

$$E_{u_k}\left(r_{k+1}^2\right) \le r_k^2 - 2h_k\langle \nabla f_\mu(x_k), x_k - x^*\rangle + h_k^2(n+4)^2 L_0^2(f)$$
$$\stackrel{(11)}{\le} r_k^2 - 2h_k(f(x_k) - f_\mu(x^*)) + h_k^2(n+4)^2 L_0^2(f).$$

Taking now the expectation in $\mathcal{U}_{k-1}$, we obtain

$$E_{\mathcal{U}_k}\left(r_{k+1}^2\right) \le E_{\mathcal{U}_{k-1}}\left(r_k^2\right) - 2h_k(\phi_k - f_\mu(x^*)) + h_k^2(n+4)^2 L_0^2(f).$$

It remains to note that $f_\mu(x^*) \stackrel{(18)}{\le} f^* + \mu L_0(f) n^{1/2}$. □

Thus, in order to guarantee inequality $E_{\mathcal{U}_{N-1}}\left(f(\hat{x}_N)\right) - f^* \le \epsilon$ by method $\mathcal{RS}_\mu$, we can choose

$$\mu \le \frac{\epsilon}{2L_0(f)n^{1/2}}, \quad h_k = \frac{R}{(n+4)(N+1)^{1/2}L_0(f)}, \; k = 0, \ldots, N,$$
$$N = \frac{4(n+4)^2}{\epsilon^2} L_0^2(f) R^2. \tag{46}$$

Note that this complexity bound is in $O(n)$ times worse than the complexity bound (43) of the method $\mathcal{RS}_0$. This can be explained by the different upper bounds provided by inequalities (32) and (34). It is interesting that the smoothing parameter μ is not used in the definition (46) of the step sizes and in the total length of the process generated by method $\mathcal{RS}_\mu$.

Finally, let us compare our results with the following *Random Coordinate Method*:

$$\begin{array}{l} 1. \text{ Generate a uniformly distributed number } i_k \in \{1, \ldots, n\}. \\ 2. \text{ Update } x_{k+1} = \pi_Q\left(x_k - h e_{i_k}\langle g(x_k), e_{i_k}\rangle\right), \end{array} \tag{47}$$

where e_i is a coordinate vector in R^n and $g(x_k) \in \partial f(x_k)$. By the same reasoning as in Theorem 5, we can show that (compare with [19])

$$\frac{1}{N+1}\sum_{k=0}^{N}(\phi_k - f^*) \le \frac{nR^2}{2(N+1)h} + \frac{h}{2}L_0^2(f).$$

Thus, under an appropriate choice of h, method (47) has the same complexity bound (43) as $\mathcal{RS}_0$. However, note that (47) requires computation of the coordinates of the *subgradient* $g(x_k)$. This computation *cannot* be arranged with directional derivatives, or with function values. Therefore, for general convex functions method (47) cannot be transformed in a gradient-free form.

A natural modification of method (39) can be applied to the problems of stochastic optimization. Indeed, assume that the objective function in (38) has the following form:

$$f(x) = E_\xi[F(x,\xi)] \stackrel{\text{def}}{=} \int_\Xi F(x,\xi)\, dP(\xi), \quad x \in Q, \tag{48}$$

where ξ is a random vector with probability distribution $P(\xi), \xi \in \Xi$. We assume that $f \in C^{0,0}(E)$ is convex (this is a relaxation of the standard assumption that $F(x,\xi)$ is convex in x for any $\xi \in \Xi$). Similarly to (30), we can define *random stochastic gradient-free oracles*:

$$
\begin{aligned}
&\mathbf{1}.\ \text{Generaterandom} u \in E, \xi \in \Xi. \text{ Return } s_\mu(x) = \tfrac{F(x+\mu u,\xi)-F(x,\xi)}{\mu} \cdot Bu.\\
&\mathbf{2}.\ \text{Generaterandom} u \in E, \xi \in \Xi. \text{ Return } \hat{s}_\mu(x) = \tfrac{F(x+\mu u,\xi)-F(x-\mu u,\xi)}{2\mu} \cdot Bu. \qquad (49)\\
&\mathbf{3}.\ \text{Generaterandom} u \in E, \xi \in \Xi. \text{ Return } s_0(x) = D_x F(x,\xi)[u] \cdot Bu.
\end{aligned}
$$

Note that the first and the second oracles require computation of two values of random function $F(\cdot,\xi)$ defined by the same value of stochastic parameter ξ. In some application, this is impossible. For example, the random function $F(\cdot,\xi)$ can be observable during a very short period of time, which is sufficient only for measuring some of its instantaneous characteristics. Then, the third oracle must be used.

Consider the following method with smoothing parameter $\mu > 0$.

Method $\mathcal{SS}_\mu$: Choose $x_0 \in Q$.
Iteration $k \geq 0$. a). For $x_k \in Q$, generate independent random vectors $\xi_k \in \Xi$ and u_k. b). Compute $s_\mu(x_k)$, and $x_{k+1} = \pi_Q \left(x_k - h_k B^{-1} s_\mu(x_k)\right)$.

(50)

Its justification is very similar to the proof of Theorem 6.

Theorem 7 *Let $L_0(F(\cdot,\xi)) \leq L$ for all $\xi \in \Xi$. Assume the sequence $\{x_k\}_{k\geq 0}$ be generated by $\mathcal{SS}_\mu$ with $\mu > 0$. Then, for any $N \geq 0$ we have*

$$
\frac{1}{S_N}\sum_{k=0}^{N} h_k(\phi_k - f^*) \leq \mu L n^{1/2} + \frac{1}{S_N}\left[\frac{1}{2}\|x_0 - x^*\|^2 + \frac{(n+4)^2}{2} L^2 \sum_{k=0}^{N} h_k^2\right], \quad (51)
$$

where $\phi_k = E_{\mathcal{U}_{k-1},\mathcal{P}_{k-1}}(f(x_k))$, and $\mathcal{P}_k = \{\xi_0, \ldots, \xi_k\}$.

Proof In the notation of Theorem 6, we have

$$
r_{k+1}^2 \leq r_k^2 - 2h_k \langle s_\mu(x_k), x_k - x^*\rangle + h_k^2 \|s_\mu(x_k)\|_*^2.
$$

In view of our assumptions, $\|s_\mu(x_k)\|_* \le L\|u_k\|^2$. Since $E_\xi\left(s_\mu(x)\right) = g_\mu(x)$, we have

$$\begin{aligned} E_{u_k,\xi_k}(r_{k+1}^2) &\le r_k^2 + E_{u_k}\left(-2h_k\langle g_\mu(x_k), x_k - x^*\rangle + h_k^2 L^2\|u_k\|^4\right) \\ &\overset{(21),(17)}{\le} r_k^2 - 2h_k\langle \nabla f_\mu(x_k), x_k - x^*\rangle + h_k^2(n+4)^2 L^2 \\ &\le r_k^2 - 2h_k(f_\mu(x_k) - f_\mu(x^*)) + h_k^2(n+4)^2 L^2. \end{aligned}$$

Taking now the expectation in $\mathcal{U}_{k-1}$ and $\mathcal{P}_{k-1}$, we get

$$E_{\mathcal{U}_k,\mathcal{P}_k}(r_{k+1}^2) \overset{(11)}{\le} E_{\mathcal{U}_{k-1},\mathcal{P}_{k-1}}(r_k^2) - 2h_k(\phi_k - f_\mu(x^*)) + h_k^2(n+4)^2 L^2.$$

It remains to note that $f_\mu(x^*) \overset{(18)}{\le} f^* + \mu L n^{1/2}$. □

Thus, choosing the parameters of method $\mathcal{SS}_\mu$ in accordance with (46), we can solve the minimization problem (38) with stochastic objective (48) in $O(\frac{n^2}{\epsilon^2})$ iterations. A similar justification can be done also for method $\mathcal{SS}_0$.

Some minimization schemes can be used for justifying adjustment processes in a stochastic environment, where even the data transmission is subject to random errors. Consider, for example, the following optimization procedure, which takes into account the *random implementation errors*.

$$\boxed{\begin{array}{l} \textbf{Method } \mathcal{SD}_\mu. \text{ For} k \ge 0 do: \\ \text{a).At} x_k \in Q, \text{generaterandomindependentvectors} \xi_k \in \Xi,\ u_k' \text{and} u_k''. \\ \text{b).Form} y_k' = x_k + \mu u_k' \text{and} y_k'' = x_k + \mu u_k''.\text{Compute}\delta_k = \frac{F(y_k',\xi_k) - F(y_k'',\xi_k)}{2\mu}. \\ \text{c).Update} x_{k+1} = \pi_Q\left(x_k - h_k\delta_k(y_k' - y_k'')\right). \end{array}} \quad (52)$$

Using the same arguments as for method (50), we can prove the complexity bound for this scheme of the order $O(\frac{n^2}{\epsilon^2})$.

5 Simple Random Search for Smooth Optimization

Consider the following smooth unconstrained optimization problem:

$$f^* \overset{\text{def}}{=} \min_{x \in E} f(x), \quad (53)$$

where f is a smooth convex function on E. Assume that this problem is solvable and denote by x^* one of its optimal solutions. For the sake of notation, we assume that $\dim E \geq 2$.

Consider the following method.

$$\begin{array}{|l|}\hline \textbf{Method } \mathcal{RG}_\mu: \text{ Choose } x_0 \in E. \\ \hline \textbf{Iteration } k \geq 0. \\ \text{a). Generate } u_k \text{ and corresponding } g_\mu(x_k). \\ \text{b). Compute } x_{k+1} = x_k - hB^{-1} g_\mu(x_k). \\ \hline \end{array} \tag{54}$$

This is a random version of the standard primal gradient method. A version of method (54) with oracle $\hat{g}_\mu$ will be called $\widehat{\mathcal{RG}}_\mu$.

Since the bounds (35) and (36) are continuous in μ, we can justify all variants of method $\mathcal{RG}_\mu$, $\mu \geq 0$, by a single statement.

Theorem 8 *Let $f \in C^{1,1}(E)$, and sequence $\{x_k\}_{k\geq 0}$ be generated by $\mathcal{RG}_\mu$ with*

$$h = \frac{1}{4(n+4)L_1(f)}. \tag{55}$$

Then, for any $N \geq 0$, we have

$$\frac{1}{N+1} \sum_{k=0}^{N} (\phi_k - f^*) \leq \frac{4(n+4)L_1(f)\|x_0 - x^*\|^2}{N+1} + \frac{9\mu^2(n+4)^2 L_1(f)}{25}. \tag{56}$$

Let function f be strongly convex. Denote $\delta_\mu = \frac{18\mu^2(n+4)^2}{25\tau(f)} L_1(f)$. Then

$$\phi_N - f^* \leq \frac{1}{2} L_1(f) \left[\delta_\mu + \left(1 - \frac{\tau(f)}{8(n+4)L_1(f)}\right)^N \left(\|x_0 - x^*\|^2 - \delta_\mu\right) \right]. \tag{57}$$

Proof Let point x_k with $k \geq 0$ be generated after k iterations of the scheme (54). Denote $r_k = \|x_k - x^*\|$. Then

$$r_{k+1}^2 = r_k^2 - 2h\langle g_\mu(x_k), x_k - x^* \rangle + h^2 \|g_\mu(x_k)\|_*^2.$$

Using representation (26) and the estimate (35), we get

$$E_{u_k}\left(r_{k+1}^2\right) \le r_k^2 - 2h\langle\nabla f_\mu(x_k), x_k - x^*\rangle + h^2\left[\tfrac{\mu^2(n+6)^3}{2}L_1^2(f) + 2(n+4)\|\nabla f(x)\|_*^2\right]$$

$$\overset{(11)}{\le} r_k^2 - 2h(f(x_k) - f_\mu(x^*)) + h^2\left[\tfrac{\mu^2(n+6)^3}{2}L_1^2(f) + 4(n+4)L_1(f)(f(x_k) - f^*)\right]$$

$$\overset{(19)}{\le} r_k^2 - 2h(1 - 2h(n+4)L_1(f))(f(x_k) - f^*) + \mu^2 nhL_1(f) + \tfrac{\mu^2(n+6)^3}{2}h^2L_1^2(f)$$

$$\overset{(55)}{=} r_k^2 - \tfrac{f(x_k)-f^*}{4(n+4)L_1(f)} + \tfrac{\mu^2}{4}\left[\tfrac{n}{n+4} + \tfrac{(n+6)^3}{8(n+4)^2}\right] \le r_k^2 - \tfrac{f(x_k)-f^*}{4(n+4)L_1(f)} + \tfrac{9\mu^2(n+4)}{100}.$$

Taking now the expectation in $\mathcal{U}_{k-1}$, we obtain

$$\rho_{k+1} \overset{\text{def}}{=} E_{\mathcal{U}_k}\left(r_{k+1}^2\right) \le \rho_k - \tfrac{\phi_k - f^*}{4(n+4)L_1(f)} + \tfrac{9\mu^2(n+4)}{100}.$$

Summing up these inequalities for $k = 0, \ldots, N$, and dividing the result by $N+1$, we get (56).

Assume now that f is strongly convex. As we have seen,

$$E_{u_k}\left(r_{k+1}^2\right) \le r_k^2 - \tfrac{f(x_k)-f^*}{4(n+4)L_1(f)} + \tfrac{9\mu^2(n+4)}{100} \overset{(8)}{\le} \left(1 - \tfrac{\tau(f)}{8(n+4)L_1(f)}\right)r_k^2 + \tfrac{9\mu^2(n+4)}{100}.$$

Taking the expectation in $\mathcal{U}_{k-1}$, we get

$$\rho_{k+1} \le \left(1 - \tfrac{\tau(f)}{8(n+4)L_1(f)}\right)\rho_k + \tfrac{9\mu^2(n+4)}{100}.$$

This inequality is equivalent to the following one:

$$\rho_{k+1} - \delta_\mu \le \left(1 - \tfrac{\tau(f)}{8(n+4)L_1(f)}\right)(\rho_k - \delta_\mu) \le \left(1 - \tfrac{\tau(f)}{8(n+4)L_1(f)}\right)^{k+1}(\rho_0 - \delta_\mu).$$

It remains to note that $\phi_k - f^* \overset{(6)}{\le} \frac{1}{2}L_1(f)\rho_k$. □

Let us discuss the choice of parameter μ in method $\mathcal{RG}_\mu$. Consider first the minimization of functions from $C^{1,1}(E)$. Clearly, the estimate (56) is valid also for $\hat{\phi}_N \overset{\text{def}}{=} E_{\mathcal{U}_{k-1}}(f(\hat{x}_N))$, where $\hat{x}_N = \arg\min_x[f(x) : x \in \{x_0, \ldots, x_N\}]$. In order to get the final accuracy ϵ for the objective function, we need to choose μ sufficiently small:

$$\mu \le \tfrac{5}{3(n+4)}\sqrt{\tfrac{\epsilon}{2L_1(f)}}. \tag{58}$$

Taking into account that $E_u(\|u\|) = O(n^{1/2})$, we can see that the average length of the finite-difference step in computation of the oracle g_μ is of the order $O\left(\sqrt{\frac{\epsilon}{nL_1(f)}}\right)$.

It is interesting that this bound is much more relaxed with respect to ϵ than the bound (46) for nonsmooth version of the random search. However, it depends now on the dimension of the space of variables. At the same time, inequality $\hat{\phi}_N - f^* \leq \epsilon$ is satisfied at most in $O(\frac{n}{\epsilon}L_1(f)R^2)$ iterations.

Consider now the strongly convex case. Then, we choose μ satisfying the equation $\frac{1}{2}L_1(f)\delta_\mu \leq \frac{\epsilon}{2}$. This is

$$\mu \leq \frac{5}{3(n+4)}\sqrt{\frac{\epsilon}{2L_1(f)} \cdot \frac{\tau(f)}{L_1(f)}}. \tag{59}$$

The number iterations of this method is of the order $O\left(\frac{nL_1(f)}{\tau(f)} \ln \frac{L_1(f)R^2}{\epsilon}\right)$. It is natural that a faster scheme needs a higher accuracy of the finite-difference oracle (or a smaller value of μ).

The complexity analysis of the method $\widehat{\mathcal{RG}}_\mu$ can be done in a similar way. In accordance with the estimate (35), the corresponding results will have slightly better dependence in μ. Note that our complexity results are also valid for the limiting version $\mathcal{RG}_0 \equiv \widehat{\mathcal{RG}}_0$.

6 Accelerated Random Search

Let us apply to problem (53) a random variant of the fast gradient method. We assume that function $f \in C^{1,1}(E)$ is strongly convex with convexity parameter $\tau(f) \geq 0$. Denote by $\kappa(f) \stackrel{\mathrm{def}}{=} \frac{\tau(f)}{L_1(f)}$ its *condition number*. And let $\theta_n = \frac{1}{16(n+1)^2 L_1(f)}$, $h_n = \frac{1}{4(n+4)L_1(f)}$.

Method $\mathcal{FG}_\mu$: Choose $x_0 \in E$, $v_0 = x_0$, and a positive $\gamma_0 \geq \tau(f)$.
Iteration $k \geq 0$: a) Compute $\alpha_k > 0$ satisfying $\theta_n^{-1}\alpha_k^2 = (1-\alpha_k)\gamma_k + \alpha_k\tau(f) \equiv \gamma_{k+1}$. b) Set $\lambda_k = \frac{\alpha_k}{\gamma_{k+1}}\tau(f)$, $\beta_k = \frac{\alpha_k\gamma_k}{\gamma_k+\alpha_k\tau(f)}$, and $y_k = (1-\beta_k)x_k + \beta_k v_k$. c). Generate random u_k and compute corresponding $g_\mu(y_k)$. d). Set $x_{k+1} = y_k - h_n B^{-1}g_\mu(y_k)$, $v_{k+1} = (1-\lambda_k)v_k + \lambda_k y_k - \frac{\theta_n}{\alpha_k}B^{-1}g_\mu(y_k)$.

(60)

Note that the parameters of this method satisfy the following relations:

$$1-\lambda_k = (1-\alpha_k)\frac{\gamma_k}{\gamma_{k+1}}, \quad 1-\beta_k = \frac{\gamma_{k+1}}{\gamma_k+\alpha_k\tau(f)}, \quad (1-\lambda_k)\frac{1-\beta_k}{\beta_k} = \frac{1-\alpha_k}{\alpha_k}. \tag{61}$$

Theorem 9 *For all $k \geq 0$, we have*

$$\phi_k - f^* \leq \psi_k \cdot [f(x_0) - f(x^*) + \tfrac{\gamma_0}{2}\|x_0 - x^*\|^2] + \mu^2 L_1(f)\left(n + \tfrac{3(n+8)}{16} C_k\right), \tag{62}$$

where $\psi_k \leq \min\left\{\left(1 - \frac{\kappa^{1/2}(f)}{4(n+4)}\right)^k, \left(1 + \frac{k}{8(n+4)}\sqrt{\frac{\gamma_0}{L_1(f)}}\right)^{-2}\right\}$, *and* $C_k \leq \min\left\{k, \frac{4(n+4)}{\kappa^{1/2}(f)}\right\}$.

Proof Assume that after k iterations, we have generated points x_k and v_k. Then we can compute y_k and generate $g_\mu(y_k)$. Taking a random step from this point, we get

$$f_\mu(x_{k+1}) \overset{(12)}{\leq} f_\mu(y_k) - h_n\langle \nabla f_\mu(y_k), B^{-1} g_\mu(x_k)\rangle + \tfrac{h_n^2}{2} L_1(f)\|g_\mu(y_k)\|_*^2.$$

Therefore,

$$\begin{aligned}
E_{u_k}\left(f_\mu(x_{k+1})\right) &\overset{(26)}{\leq} f_\mu(y_k) - h_n\|\nabla f_\mu(y_k)\|_*^2 + \tfrac{h_n^2}{2} L_1(f) E_{u_k}\left(\|g_\mu(y_k)\|_*^2\right) \\
&\overset{(37)}{\leq} f_\mu(y_k) - \tfrac{h_n}{4(n+4)}\left(E_{u_k}\left(\|g_\mu(y_k)\|_*^2\right) - 3\mu^2 L_1^2(f)(n+5)^3\right) \\
&\quad + \tfrac{h_n^2}{2} L_1(f) E_{u_k}\left(\|g_\mu(y_k)\|_*^2\right) \\
&= f_\mu(y_k) - \tfrac{1}{2}\theta_n E_{u_k}\left(\|g_\mu(y_k)\|_*^2\right) + \xi_\mu,
\end{aligned}$$

where $\xi_\mu \overset{\text{def}}{=} \frac{3(n+5)^3\mu^2}{16(n+4)^2} L_1(f)$. Note that $\frac{(n+5)^3}{(n+4)^2} \leq n+8$ for $n \geq 2$.

Let us fix an arbitrary $x \in E$. Note that

$$\begin{aligned}
\delta_{k+1}(x) &\overset{\text{def}}{=} \tfrac{\gamma_{k+1}}{2}\|v_{k+1} - x\|^2 + f_\mu(x_{k+1}) - f_\mu(x) \\
&= \tfrac{\gamma_{k+1}}{2}\|(1-\lambda_k)v_k + \lambda_k y_k - x\|^2 - \tfrac{\theta_n \gamma_{k+1}}{\alpha_k}\langle g_\mu(y_k), (1-\lambda_k)v_k + \lambda_k y_k - x\rangle \\
&\quad + \tfrac{\theta_n^2 \gamma_{k+1}}{2\alpha_k^2}\|g_\mu(y_k)\|_*^2 + f_\mu(x_{k+1}) - f_\mu(x).
\end{aligned}$$

Taking the expectation in u_k, and using the equation of Step a) in (60), we get

$$\begin{aligned}
E_{u_k}(\delta_{k+1}(x)) &\overset{(21)}{\leq} \tfrac{\gamma_{k+1}}{2}\|(1-\lambda_k)v_k + \lambda_k y_k - x\|^2 - \alpha_k\langle \nabla f_\mu(y_k), (1-\lambda_k)v_k + \lambda_k y_k - x\rangle \\
&\quad + \tfrac{1}{2}\theta_n E_{u_k}\left(\|g_\mu(y_k)\|_*^2\right) + E_{u_k}\left(f_\mu(x_{k+1})\right) - f_\mu(x) \\
&\leq \tfrac{\gamma_{k+1}}{2}\|(1-\lambda_k)v_k + \lambda_k y_k - x\|^2 + \alpha_k\langle \nabla f_\mu(y_k), x - (1-\lambda_k)v_k \\
&\quad - \lambda_k y_k\rangle + f_\mu(y_k) - f_\mu(x) + \xi_\mu.
\end{aligned}$$

Note that $v_k = y_k + \frac{1-\beta_k}{\beta_k}(y_k - x_k)$. Therefore,

$$(1-\lambda_k)v_k + \lambda_k y_k = y_k + (1-\lambda_k)\frac{1-\beta_k}{\beta_k}(y_k - x_k) \overset{(61)}{=} y_k + \frac{1-\alpha_k}{\alpha_k}(y_k - x_k).$$

Hence,

$$\begin{aligned} & f_\mu(y_k) + \alpha_k \langle \nabla f_\mu(y_k), x - (1-\lambda_k)v_k - \lambda_k y_k \rangle - f_\mu(x) \\ = \; & f_\mu(y_k) + \langle \nabla f_\mu(y_k), \alpha_k x + (1-\alpha_k)x_k - y_k \rangle - f_\mu(x) \\ \overset{(8)}{\le} \; & (1-\alpha_k)(f(x_k) - f_\mu(x)) - \tfrac{1}{2}\alpha_k \tau(f)\|x - y_k\|^2, \end{aligned}$$

and we can continue:

$$\begin{aligned} E_{u_k}(\delta_{k+1}(x)) &\le \tfrac{\gamma_{k+1}}{2}\|(1-\lambda_k)v_k + \lambda_k y_k - x\|^2 + \xi_\mu \\ &\quad + (1-\alpha_k)(f(x_k) - f_\mu(x)) - \tfrac{1}{2}\alpha_k\tau(f)\|x - y_k\|^2 \\ &\le \tfrac{\gamma_{k+1}}{2}(1-\lambda_k)\|v_k - x\|^2 + \tfrac{\gamma_{k+1}}{2}\lambda_k\|y_k - x\|^2 + \xi_\mu \\ &\quad + (1-\alpha_k)(f(x_k) - f_\mu(x)) - \tfrac{1}{2}\alpha_k\tau(f)\|x - y_k\|^2 \\ &\overset{(61)}{=} (1-\alpha_k)\delta_k(x) + \xi_\mu. \end{aligned}$$

Denote $\phi_k(\mu) = E_{\mathcal{U}_{k-1}}(f_\mu(x_k))$, $\rho_k = \frac{\gamma_k}{2}E_{\mathcal{U}_{k-1}}(\|v_k - x^*\|^2)$. Then, taking the expectation of the latter inequality in $\mathcal{U}_{k-1}$, we get

$$\begin{aligned} \phi_{k+1}(\mu) - f_\mu(x) + \rho_{k+1} &\le (1-\alpha_k)(\phi_k(\mu) - f_\mu(x^*) + \rho_k) + \xi_\mu \\ &\le \ldots \le \psi_{k+1}\cdot\left(f_\mu(x_0) - f_\mu(x) + \tfrac{\gamma_0}{2}\|x_0 - x\|^2\right) \\ &\quad + \xi_\mu \cdot C_{k+1}, \end{aligned}$$

where $\psi_k = \prod\limits_{i=0}^{k-1}(1-\alpha_i)$, and $C_k = 1 + \sum\limits_{i=1}^{k-1}\prod\limits_{j=k-i}^{k-1}(1-\alpha_j)$, $k \ge 1$. Defining $\psi_0 = 1$ and $C_0 = 0$, we get $C_k \le k$, $k \ge 0$. On the other hand, by induction it is easy to see that $\gamma_k \ge \tau(f)$ for all $k \ge 0$. Therefore,

$$\alpha_k \ge [\tau(f)\theta_n]^{1/2} = \frac{\kappa^{1/2}(f)}{4(n+4)} \overset{\text{def}}{=} \omega_n, \quad k \ge 0.$$

Then, $C_k \le 1 + \sum\limits_{i=1}^{k-1}\prod\limits_{j=k-i}^{k-1}(1-\omega_n)^i = 1 + (1-\omega_n)\frac{(1-(1-\omega_n)^k)}{\omega_n} \le \omega_n^{-1}$. Thus,

$$C_k \le \min\left\{k, \frac{4(n+4)}{\kappa^{1/2}(f)}\right\}, \quad \psi_k \le \left(1 - \frac{\kappa^{1/2}(f)}{4(n+4)}\right)^k, \quad k \ge 0.$$

Further,[3] let us prove that $\gamma_k \geq \gamma_0 \psi_k$. For $k = 0$, this is true. Assume it is true for some $k \geq 0$. Then

$$\gamma_{k+1} \geq (1-\alpha_k)\gamma_k \;\geq\; \gamma_0 \psi_{k+1}.$$

Denote $a_k = \frac{1}{\psi_k^{1/2}}$. Then, in view of the established inequality we have:

$$a_{k+1} - a_k = \frac{\psi_k^{1/2} - \psi_{k+1}^{1/2}}{\psi_k^{1/2}\psi_{k+1}^{1/2}} \;=\; \frac{\psi_k - \psi_{k+1}}{\psi_k^{1/2}\psi_{k+1}^{1/2}(\psi_k^{1/2} + \psi_{k+1}^{1/2})} \;\geq\; \frac{\psi_k - \psi_{k+1}}{2\psi_k \psi_{k+1}^{1/2}}$$

$$= \frac{\psi_k - (1-\alpha_k)\psi_k}{2\psi_k \psi_{k+1}^{1/2}} \;=\; \frac{\alpha_k}{2\psi_{k+1}^{1/2}} \;=\; \frac{\gamma_{k+1}^{1/2}\theta_n^{1/2}}{2\psi_{k+1}^{1/2}} \;\geq\; \frac{1}{8(n+4)}\sqrt{\frac{\gamma_0}{L_1(f)}}.$$

Hence, $\frac{1}{\psi_k 1/2} \geq 1 + \frac{k}{8(n+4)}\sqrt{\frac{\gamma_0}{L_1(f)}}$ for all $k \geq 0$. It remains to note that

$$E_{\mathcal{U}_{k-1}}(f(x_k)) - f(x^*) \overset{(11)}{\leq} \phi_k(\mu) - f(x^*) \overset{(19)}{\leq} \phi_k(\mu) - f_\mu(x^*) + \frac{\mu^2}{2} L_1(f) n$$

$$\leq \psi_k \cdot \left(f_\mu(x_0) - f_\mu(x^*) + \frac{\gamma_0}{2}\|x_0 - x^*\|^2\right) + \xi_\mu \cdot C_k + \frac{\mu^2}{2} L_1(f) n$$

$$\overset{(19)}{\leq} \psi_k \cdot \left(f(x_0) - f(x^*) + \frac{\gamma_0}{2}\|x_0 - x^*\|^2\right) + \xi_\mu \cdot C_k + \mu^2 L_1(f) n.$$

It remains to apply the upper bounds for ψ_k. □

Let us discuss the complexity estimates of the method (60) for $\tau(f) = 0$. In order to get accuracy ϵ for the objective function, it suffices that both terms in the right-hand side of inequality (62) be smaller than $\frac{\epsilon}{2}$. Thus, we need

$$N(\epsilon) = O\left(\frac{n L_1^{1/2}(f) R}{\epsilon^{1/2}}\right) \tag{63}$$

iterations. Similarly to the simple random search method (39), this estimate is n times larger than the estimate of the corresponding scheme with full computation of the gradient. The parameter of the oracle μ must be chosen as

$$\mu \leq O\left(\frac{\epsilon^{1/2}}{L_1^{1/2}(f)(n \cdot N(\epsilon))^{1/2}}\right) \;=\; O\left(\frac{\epsilon^{3/4}}{n L_1^{3/4}(f) R^{1/2}}\right)$$
$$= O\left(\frac{1}{n}\left[\frac{\epsilon}{L_1(f)} \cdot \left[\frac{\epsilon}{L_1(f) R^2}\right]^{1/2}\right]^{1/2}\right). \tag{64}$$

As compared with (58), the average size of the trial step μu is a tighter function of ϵ. This is natural, since the method (54) is much faster. On the other hand, this size is still quite moderate (this is good for numerical stability of the scheme).

[3] The rest of the proof is very similar to the proof of Lemma 2.2.4 in [16]. We present it here just for the reader convenience.

Remark 1 1. Method (60) can be seen as a variant of the constant step scheme (2.2.8) in [16]. Therefore, the sequence $\{v_k\}$ can be expressed in terms of $\{x_k\}$ and $\{y_k\}$ (see Section 2.2.1 in [16] for details).

2. Linear convergence of method (60) for strongly convex functions allows an efficient generation of random approximations to the solution of problem (53) with arbitrary high confidence level. This can be achieved by an appropriate regularization of the initial problem, as suggested in Section 3 of [19].

7 Nonconvex Problems

Consider now the problem

$$\min_{x \in E} \; f(x), \tag{65}$$

where the objective function f is nonconvex. Let us apply to it method (39). Now it has the following form:

$$\boxed{\begin{array}{l} \textbf{Method } \widehat{\mathcal{RS}}_\mu : \text{Choose} x_0 \in E. \\ \hline \textbf{Iteration } k \geq 0. \\ \text{a).Generate} u_k \text{andcorresponding} g_\mu(x_k). \\ \text{b).Compute} x_{k+1} = x_k - h_k B^{-1} g_\mu(x_k). \end{array}} \tag{66}$$

Let us estimate the evolution of the value of function f_μ after one step of this scheme. Since f_μ has Lipschitz-continuous gradient, we have

$$f_\mu(x_{k+1}) \overset{(6)}{\leq} f_\mu(x_k) - h_k \langle \nabla f_\mu(x_k), B^{-1} g_\mu(x_k) \rangle + \tfrac{1}{2} h_k^2 L_1(f_\mu) \| g_\mu(x_k) \|_*^2.$$

Taking now the expectation in u_k, we obtain

$$E_{u_k}(f_\mu(x_{k+1})) \overset{(21)}{\leq} f_\mu(x_k) - h_k \| \nabla f_\mu(x_k) \|_*^2 + \tfrac{1}{2} h_k^2 L_1(f_\mu) E_{u_k} \left(\| g_\mu(x_k) \|_*^2 \right). \tag{67}$$

Consider now two cases.

1. $f \in C^{1,1}(E)$. Then

$$\begin{aligned} E_{u_k}(f_\mu(x_{k+1})) &\overset{(37)}{\leq} f_\mu(x_k) - h_k \| \nabla f_\mu(x_k) \|_*^2 \\ &\quad + \tfrac{1}{2} h_k^2 L_1(f) \left(4(n+4) \| \nabla f_\mu(x_k) \|_*^2 + 3\mu^2 L_1^2(f)(n+4)^3 \right) \end{aligned}$$

Choosing now $h_k = \hat{h} \stackrel{\text{def}}{=} \frac{1}{4(n+4)L_1(f)}$, we obtain

$$E_{u_k}(f_\mu(x_{k+1})) \le f_\mu(x_k) - \tfrac{1}{2}\hat{h}\|\nabla f_\mu(x_k)\|_*^2 + \tfrac{3\mu^2}{32}L_1(f)(n+4).$$

Taking the expectation of this inequality in $\mathcal{U}_k$, we get

$$\phi_{k+1} \le \phi_k - \tfrac{1}{2}\hat{h}\eta_k^2 + \tfrac{3\mu^2(n+4)}{32}L_1(f),$$

where $\eta_k^2 \stackrel{\text{def}}{=} E_{\mathcal{U}_k}\left(\|\nabla f_\mu(x_k)\|_*^2\right)$. Assuming now that $f(x) \ge f^*$ for all $x \in E$, we get

$$\frac{1}{N+1}\sum_{k=0}^{N}\eta_k^2 \le 8(n+4)L_1(f)\left[\frac{f(x_0)-f^*}{N+1} + \frac{3\mu^2(n+4)}{32}L_1(f)\right]. \tag{68}$$

Since $\theta_k^2 \stackrel{\text{def}}{=} E_{\mathcal{U}_k}\left(\|\nabla f(x_k)\|_*^2\right) \stackrel{(29)}{\le} 2\eta_k^2 + \frac{\mu^2(n+6)^3}{2}L_1^2(f)$, the expected rate of decrease in θ_k is of the same order as (68). In order to get $\frac{1}{N+1}\sum_{k=0}^{N}\theta_k^2 \le \epsilon^2$, we need to choose

$$\mu \le O\left(\frac{\epsilon}{n^{3/2}L_1(f)}\right).$$

Then, the upper bound for the expected number of steps is $O(\frac{n}{\epsilon^2})$.

2. $f \in C^{0,0}(E)$. Then,

$$\begin{aligned}E_{u_k}(f_\mu(x_{k+1})) &\stackrel{(34)}{\le} f_\mu(x_k) - h_k\|\nabla f_\mu(x_k)\|_*^2 + \tfrac{1}{2}h_k^2 L_1(f_\mu)\cdot L_0^2(f)(n+4)^2\\ &\stackrel{(22)}{=} f_\mu(x_k) - h_k\|\nabla f_\mu(x_k)\|_*^2 + \tfrac{1}{\mu}h_k^2 n^{1/2}(n+4)^2\cdot L_0^3(f).\end{aligned}$$

Assume $f(x) \ge f^*$, $x \in E$, and denote $S_N \stackrel{\text{def}}{=} \sum_{k=0}^{n} h_k$. Taking the expectation of the latter inequality in $\mathcal{U}_k$, and summing them up, we get

$$\begin{aligned}\frac{1}{S_N}\sum_{k=0}^{N}h_k\eta_k^2 &\le \frac{1}{S_N}\left[(f_\mu(x_0)-f^*) + C(\mu)\sum_{k=0}^{N}h_k^2\right],\\ C(\mu) &\stackrel{\text{def}}{=} \tfrac{1}{\mu}n^{1/2}(n+4)^2\cdot L_0^3(f).\end{aligned} \tag{69}$$

Thus, we can guarantee a convergence of the process (66) to a stationary point of the function f_μ, which is a smooth approximation of f. In order to bound the gap in this approximation by ϵ, we need to choose $\mu \le \bar{\mu} \stackrel{(18)}{=} \frac{\epsilon}{n^{1/2}L_0(f)}$. Let us assume for simplicity that we are using a constant step scheme: $h_k \equiv h$, $k \ge 0$. Then the right-hand side of inequality (69) becomes

$$\frac{f_{\bar{\mu}}(x_0)-f^*}{(N+1)h} + \frac{h}{\epsilon}n(n+4)^2L_0^4(f) \le \frac{L_0(f)R}{(N+1)h} + \frac{h}{\epsilon}N(n+4)^2L_0^4(f) \stackrel{\text{def}}{=} \rho(h).$$

Minimizing this upper bound in h, we get is optimal value:

$$h^* = \left[\frac{\epsilon R}{n(n+4)^2 L_0^3(f)(N+1)}\right]^{1/2}, \quad \rho(h^*) \; = \; 2\left[\frac{n(n+4)^2 L_0^5(f) R}{\epsilon(N+1)}\right]^{1/2}.$$

Thus, in order to guarantee the expected squared norm of the gradient of function $f_{\bar{\mu}}$ of the order δ, we need

$$O\left(\frac{n(n+4)^2 L_0^5(f) R}{\epsilon \delta^2}\right)$$

iterations of the scheme (66). To the best of our knowledge, this is the first complexity bound for the methods for minimizing nonsmooth nonconvex functions. Note that allowing in the method (66) $h_k \to 0$ and $\mu \to 0$, we can ensure convergence of the scheme to a stationary point of the initial function f. But this proof is quite long and technical. Therefore, we omit it.

8 Preliminary Computational Experiments

The main goal of our experiments was the investigation of the impact of the random oracle on the actual convergence of the minimization methods. We compared the performance of the randomized gradient-free methods with the classical gradient schemes. As suggested by our efficiency estimates, it is normal if the former methods need n times more iterations as compared with the classical ones. Let us describe our results.

8.1 Smooth Minimization

We checked the performance of the methods (54) and (60) on the following test function:

$$f_n(x) = \tfrac{1}{2}(x^{(1)})^2 + \tfrac{1}{2}\sum_{i=1}^{n-1}\left(x^{(i+1)} - x^{(i)}\right)^2 + \tfrac{1}{2}\left(x^{(n)}\right)^2 - x^{(1)}, \quad x_0 = 0. \qquad (70)$$

This function was used in Section 2.1 in [16] for proving the lower complexity bound for the gradient methods as applied to functions from $C^{1,1}(R^n)$. It has the following parameters:

$$L_1(f_n) \le 4, \quad R^2 = \|x_0 - x^*\|^2 \; \le \; \tfrac{n+1}{3}, \quad n = 256.$$

These values were used for defining the trial step size μ by (58) and (64). We also tested the versions of corresponding methods with $\mu = 0$. Finally, we compared these results with the usual gradient and fast gradient method.

Our results for the simple gradient schemes are presented in the following table. The first column of the table indicates the current level of relative accuracy with respect to

Table 1 Simple random search $\mathcal{RG}_\mu$

Accuracy	$\mu = 0$			$\mu = 8.9 \times 10^{-6}$			GM
	Min	Max	Mean	Min	Max	Mean	
2.0×10^{-3}	3	4	4.0	3	4	3.9	1
9.8×10^{-4}	20	22	21.3	21	22	21.3	5
4.9×10^{-4}	85	89	86.8	85	89	86.8	22
2.4×10^{-4}	329	343	335.5	327	342	335.4	83
1.2×10^{-4}	1210	1254	1232.8	1204	1246	1231.8	304
6.1×10^{-5}	4129	4242	4190.3	4155	4235	4190.4	1034
3.1×10^{-5}	12440	12611	12536.7	12463	12645	12538.1	3092
1.5×10^{-5}	30883	31178	31054.6	30939	31269	31058.1	7654

the scale $S \stackrel{\mathrm{def}}{=} \frac{1}{2} L_1(f_n) R^2$. The kth row of the table, $k = 2, \ldots, 9$, shows the number of iterations spent for achieving the absolute accuracy $2^{-(k+7)} S$. This table aggregates the results of 20 attempts of the method $\mathcal{RG}_0$ and $\mathcal{RG}_\mu$ to minimize the function (70). The columns 2–4 of the table represent the minimal, maximal and average number of blocks by n iterations, executed by $\mathcal{RG}_0$ in order to reach corresponding level of accuracy. The next three columns represent this information for $\mathcal{RG}_\mu$ with μ computed by (58) with $\epsilon = 2^{-16}$. The last column contains the results for the standard gradient method with constant step $h = \frac{1}{L_1(f_n)}$ (Table 1).

We can see a very small variance of the results presented in each column. Moreover, the finite-difference version with an appropriate value of μ demonstrates practically the same performance as the version based on the directional derivative. Moreover, the number of blocks by n iterations of the random schemes is practically equal to the number of iterations of the standard gradient method multiplied by four. A plausible explanation of this phenomena is related to the choice of the step size $h = \frac{1}{\mathbf{4} \cdot (n+4) L_1(f)}$. However, we prefer to use this value since there is no theoretical justification for a larger step.

Let us present the results of 20 runs of the accelerated schemes. The structure of Table 2 is similar to that of Table 1. Since these methods are faster, we give the results for a more accurate solution, up to $\epsilon = 2^{-30}$.

As we can see, the accelerated schemes are indeed faster than the simple random search. On the other hand, same as in Table 1, the variance of the results in each line is very small. Method with $\mu = 0$ demonstrates almost the same efficiency as the method with μ defined by (64). And again, the number of the blocks by n iterations of the random methods is proportional to the number of iterations of the standard gradient methods multiplied by four.

8.2 Minimization of Piecewise Linear Functions

For nonsmooth problems, we present first the computational results of two variants of method (39) on the following test functions:

Table 2 Fast random search $\mathcal{FG}_\mu$

Accuracy	$\mu = 0$			$\mu = 3.5 \times 10^{-10}$			FGM
	Min	Max	Mean	Min	Max	Mean	
2.0×10^{-3}	7	7	7.0	7	7	7.0	1
9.8×10^{-4}	21	22	21.1	21	22	21.1	4
4.9×10^{-4}	45	47	45.8	46	47	46.2	10
2.4×10^{-4}	93	96	94.1	93	96	94.5	22
1.2×10^{-4}	182	187	184.7	180	188	185.4	44
6.1×10^{-5}	338	350	345.4	342	349	346.6	84
3.1×10^{-5}	597	611	603.2	599	609	604.3	147
1.5×10^{-5}	944	967	953.1	948	964	954.9	233
7.6×10^{-6}	1328	1355	1339.6	1332	1351	1341.5	328
3.8×10^{-6}	1671	1695	1679.4	1671	1688	1680.3	411
1.9×10^{-6}	1915	1934	1922.6	1916	1928	1923.1	471
9.5×10^{-7}	2070	2083	2075.3	2070	2080	2075.7	508
4.8×10^{-7}	2177	2189	2182.1	2177	2187	2182.6	535
2.4×10^{-7}	2270	2281	2274.4	2268	2279	2274.4	557
1.2×10^{-7}	2360	2375	2366.8	2355	2375	2366.3	580
6.0×10^{-8}	4294	4308	4299.9	4291	4308	4300.9	1056
3.0×10^{-8}	4396	4410	4402.4	4392	4411	4403.6	1081
1.5×10^{-8}	4496	4521	4506.9	4495	4518	4508.0	1107
7.5×10^{-9}	6519	6537	6529.0	6517	6540	6529.1	1604
3.7×10^{-9}	6624	6669	6646.2	6623	6672	6644.4	1633
1.9×10^{-9}	8680	8718	8700.3	8682	8712	8699.1	2139
9.3×10^{-10}	10770	10805	10789.9	10779	10808	10791.2	2653

$$
\begin{aligned}
F_1(x) &= |x^{(1)} - 1| + \sum_{i=1}^{n-1} |1 + x^{(i+1)} - 2x^{(i)}|, \\
F_\infty(x) &= \max\left\{ |x^{(1)} - 1|, \max_{1 \le i \le n-1} |1 + x^{(i+1)} - 2x^{(i)}| \right\}.
\end{aligned} \tag{71}
$$

For both functions, $x_0 = 0$, $(x^*)^{(i)} = 1$, $i = 1, \ldots, n$, and $F_1^* = F_\infty^* = 0$. They have the following parameters:

$$
L_0(F_1) \le 3n^{1/2}, \quad L_0(F_\infty) \le 3, \quad R^2 = \|x_0 - x^*\|^2 \le n.
$$

Despite their trivial form, these functions are very badly conditioned. Let us define the condition number of the level set of function f:

$$
\kappa_t(f) \overset{\text{def}}{=} \inf_{x,y} \left\{ \frac{\|x-x^*\|_\infty}{\|y-x^*\|_\infty} : f(x) = f(y) = f^* + t \right\}, \quad t \ge 0,
$$

Such a condition number can be defined with respect to any norm in E. Since all norms on finite-dimensional spaces are compatible, any of these numbers provides us with a useful estimate of the level of degeneracy of corresponding functions.

Lemma 6 *For any* $t \geq 0$, *we have* $\kappa_t(F_1) \leq \frac{2n}{3(2^n-1)}$, *and* $\kappa_t(F_\infty) \leq \frac{1}{2^n-1}$.

Proof Indeed, define $x^{(1)} = 1 + \frac{t}{2}$, and $x^{(i)} = 1$, $i = 2, \ldots, n$. Then,

$$x^{(1)} - 1 = \tfrac{t}{2}, \quad 1 + x^{(2)} - 2x^{(1)} \; = \; -t,$$

$$1 + x^{(i+1)} - 2x^{(i)} = 0, \;\; i = 2, \ldots, n-1.$$

Thus, $F_1(x) = \frac{3}{2}t$, and $F_\infty(x) = t$. Further, define $y^{(i)} = 1 + \gamma t(2^i - 1)$, $i = 1, \ldots, n$. Then,

$$y^{(1)} - 1 = \gamma t,$$

$$1 + y^{(i+1)} - 2y^{(i)} = 1 + \gamma t(2^{i+1} - 1) + 1 - 2[\gamma t(2^i - 1) + 1] \; = \; \gamma t,$$

$$i = 1, \ldots, n-1.$$

Hence, $F_1(y) = n\gamma t$, and $F_\infty(y) = \gamma t$. Note that $\|x - x^*\|_\infty = t$, and $\|y - x^*\|_\infty = \gamma t(2^n - 1)$. Taking now $\gamma = \frac{3}{2n}$ for F_1, and $\gamma = 1$ for F_∞, we get the desired results. □

Using the technique of Section 2.1 in [16] as applied to functions (71), it is possible to prove the lower complexity bound $O(\frac{1}{\epsilon^2})$ for nonsmooth optimization methods.

In Table 3, we compare three methods: method $\mathcal{RS}_0$, method $\mathcal{RS}_\mu$ with μ defined by (46), and the standard subgradient method (e.g., Section 3.2.3 in [16]), as applied to the function F_1. The first column of the table shows the required accuracy with respect to the scale $L_0(F_1)R$. The theoretical upper bound for achieving the corresponding level of accuracy is $\frac{\kappa}{\epsilon^2}$, where κ is an absolute constant. We present the results for three dimensions $n = 16, 64, 256$. For the first two methods, we display the number of blocks of n iterations that were required in order to reach this level of accuracy. If this was impossible after 10^5 iterations, we put in the cell the best value found by the scheme. For the standard subgradient scheme, we show the usual number of iterations. These results correspond only to a single run since the variability in the performance of the random schemes is very small.

As compared with the theoretical upper bounds, all methods perform much better. We observe an unexpectedly good performance of method $\mathcal{RS}_\mu$. It is usually better than its variant with exact directional derivative. Moreover, for a higher accuracy, it is often better than the usual subgradient method. Let us present now the computational results for function F_∞ (Table 4).

We can see that at this test problem, the finite-difference version $\mathcal{RS}_\mu$ is less dominant. Nevertheless, in two cases from three it is a clear winner.

Table 3 Different methods for function F_1, Limit $= 10^5$

Method	$\mathcal{RS}_0$	$\mathcal{RS}_\mu$	$\mathcal{SG}$	$\mathcal{RS}_0$	$\mathcal{RS}_\mu$	$\mathcal{SG}$	$\mathcal{RS}_0$	$\mathcal{RS}_\mu$	$\mathcal{SG}$
$\epsilon \backslash n$	16			64			256		
2.5E−1	4	1	1	2	9	1	4	33	1
1.3E−1	7	18	4	7	58	3	11	221	3
6.3E−2	11	38	12	25	105	4	21	381	4
3.1E−2	27	60	30	59	137	10	74	482	4
1.6E−2	104	88	40	187	161	24	263	546	14
7.8E−3	328	108	94	685	180	48	1045	590	36
3.9E−3	1086	114	248	2749	199	118	3848	624	94
2.0E−3	4080	273	3866	10828	221	368	14773	656	202
9.8E−4	10809	884	17698	41896	698	904	54615	698	392
4.9E−4	39157	3714	46218	6.0E−4	2213	3570	7.5E−4	981	566
2.4E−4	3.0E−4	11156	85778		9506	18354		3759	904
1.2E−4		26608	2.2E−4		37870	1.8E−4		14961	1.7E−4

Table 4 Different methods for function F_∞, Limit $= 10^5$

Method:	$\mathcal{RS}_0$	$\mathcal{RS}_\mu$	$\mathcal{SG}$	$\mathcal{RS}_0$	$\mathcal{RS}_\mu$	$\mathcal{SG}$	$\mathcal{RS}_0$	$\mathcal{RS}_\mu$	$\mathcal{SG}$
$\epsilon \backslash n$	16			64			256		
2.5E−1	1	1	1	1	1	1	1	1	1
1.3E−1	1	1	1	1	1	1	1	1	1
6.3E−2	43	73	19	1	1	1	1	1	1
3.1E−2	63	207	79	245	675	77	1	1	1
1.6E−2	115	321	278	337	3650	343	1301	9123	322
7.8E−3	201	432	1159	546	6098	1265	1921	56604	1340
3.9E−3	1101	471	5058	2579	7503	5060	3335	95699	5058
2.0E−3	1601	504	20228	7637	8322	20233	12328	3.5E−3	20231
9.8E−4	5972	542	80912	27417	8755	80916	42798		80915
4.9E−4	29873	1923	8.8E−4	91102	9008	8.8E−4	6.9E−4		8.8E−4
2.4E−4	93887	5685		4.3E−4	9431				
1.2E−4	1.8E−4	21896			25424				

Let us compare these methods on a more sophisticated test problem. Denote by $\Delta_m \subset R^m$ the standard simplex. Consider the following matrix game:

$$\min_{x \in \Delta_m} \max_{y \in \Delta_m} \langle Ax, y \rangle = \max_{y \in \Delta_m} \min_{x \in \Delta_m} \langle Ax, y \rangle, \tag{72}$$

Table 5 Saddle point problem

Dim	$\mathcal{RS}_0$	$\mathcal{RS}_\mu$	$\mathcal{SG}$
8	1.3E−5	5.3E−6	1.4E−4
16	3.3E−5	8.3E−6	1.3E−4
32	4.80E−5	7.0E−6	1.3E−4
64	2.3E−4	2.2E−4	2.4E−4
128	9.3E−5	3.1E−5	1.6E−4
256	9.3E−5	2.1E−5	1.7E−4

where A is an $m \times m$-matrix. Define the following function:

$$f(x, y) = \max \left\{ \max_{1 \le i,j \le m} \left[\langle A^T e_i, x \rangle - \langle A e_j, y \rangle \right], \ |\langle \bar{e}, x \rangle - 1|, \ |\langle \bar{e}, y \rangle - 1| \right\},$$

where $e_i \in R^m$ are coordinate vectors and $\bar{e} \in R^m$ is the vector of all ones. Clearly, the problem (72) is equivalent to the following minimization problem:

$$\min_{x,y \ge 0} \ f(x, y). \tag{73}$$

The optimal value of this problem is zero. We choose the starting points $x_0 = \frac{\bar{e}}{m}$, $y_0 = \frac{\bar{e}}{m}$, and generate A with random entries uniformly distributed in the interval $[-1, 1]$. Then the parameters of problem (38) are as follows:

$$n = 2m, \quad Q = R^n_+, \quad L_0(f) \le n^{1/2}, \quad R \le 2.$$

In Table 5, we present the computational results for two variants of method $\mathcal{RS}_\mu$ and the subgradient scheme. For problems (73) of dimension $n = 2^p$, $p = 3 \dots 16$, we report the best accuracy achieved by the schemes after 10^5 iterations (as usual, for random methods, we count the blocks of n iterations). The parameter μ of method $\mathcal{RS}_\mu$ was computed by (46) with target accuracy $\epsilon = 9.5\text{E} - 7$.

Clearly, in this competition method $\mathcal{RS}_\mu$ is again a winner. The two other methods demonstrate very similar performance.

8.3 Test Functions Based on Chebyshev Polynomials

Chebyshev polynomials of the first kind are defined by the recurrence relation

$$T_0(t) = 1, \quad T_1(t) = t,$$

$$T_{n+1}(t) = 2t T_n(t) - T_{n-1}(t), \quad n \ge 1.$$

In particular, $T_2(t) = 2t^2 - 1$. The absolute value of such a polynomial achieves its maximum (equal to one) exactly at $n + 1$ points of the segment $[-1, 1]$.

Chebyshev polynomials satisfy the following *nesting property*

$$T_n(T_m(t)) = T_{nm}(t),$$

which allows to create test functions with very high oscillating behavior. Indeed, consider the system of equations

$$x^{(k+1)} = T_2\left(x^{(k)}\right), \quad k = 1, \ldots, n-1. \tag{74}$$

Then the last component of the vector $x \in R^n$ depends on the first one in a very oscillating manner:

$$x^{(n)} = T_{2^{n-1}}\left(x^{(1)}\right).$$

Penalizing the residual in the system of nonlinear equations (74), we can get many interesting objective functions. On one hand, they do not have local minimums, and on the other hand, they exhibit an oscillatory behavior of the level sets. The simplest function of this type is as follows:

$$\begin{aligned} f(x) &= \tfrac{1}{4}\left(x^{(1)} - 1\right)^2 + \sum_{i=1}^{n-1}\left(x^{(i+1)} + 1 - 2\left(x^{(i)}\right)^2\right)^2, \\ x^* &= (1, \ldots, 1)^T, \quad x_0 = (-1, 1, \ldots, 1)^T. \end{aligned} \tag{75}$$

However, function (75) is not convex. In order to see that Chebyshev polynomials can deliver interesting test functions for convex optimization, note that the polynomial $T_2(t)$ is convex.

Consider the following system of nonlinear inequalities:

$$\begin{aligned} x^{(k+1)} &\geq 2\left(x^{(k)}\right)^2 - 1, \quad k = 1, \ldots, n-1, \\ x^{(1)} &\geq 1. \end{aligned} \tag{76}$$

Lemma 7 *Let point $x \in R^n$ be feasible for the system of convex inequalities (76). If for some $\delta \in [0, 1]$ we have $x^{(1)} \geq 1 + \frac{\delta^2}{2(1+\delta)}$, then*

$$x^{(n)} \geq \tfrac{1}{2}\left[(1+\delta)^{2^{n-1}} + (1+\delta)^{-2^{n-1}}\right] \geq 2^{-1+\frac{\delta}{2}2^n}. \tag{77}$$

Proof Indeed, let us prove by induction that $x^{(i)} \geq \frac{1}{2}\left[(1+\delta)^{2^{i-1}} + (1+\delta)^{-2^{i-1}}\right]$. For $i = 1$, this inequality is valid by the assumption. Let it be valid for some $i \geq 1$. Then,

Table 6 Results for problem (79)

Dim	$\mathcal{SG}$	$\mathcal{RS}_0$	$\mathcal{RS}_\mu$
8	833650	415165	11862
16	532123	34471	16939
32	454043	441065	34741
64	428966	15665	130
128	421854	5636	250
256	416582	913	2577
512	416143	707	923
1024	413209	1419	2001

$$
\begin{aligned}
x^{(i+1)} &\geq \tfrac{1}{2}\left[(1+\delta)^{2^{i-1}} + (1+\delta)^{-2^{i-1}}\right]^2 - 1\\
&= \tfrac{1}{2}\left[(1+\delta)^{2^{i}} + (1+\delta)^{-2^{i}}\right].
\end{aligned}
$$

It remains to note that for $\delta \in [0, 1]$ we have $\ln(1+\delta) \geq \delta \ln 2$. □

Thus, the system of inequalities (76) has very poor Slater condition, which makes it difficult for the interior-point methods. This ill-conditioning is inherited by another representation of this set. For example, it can be defined as follows:

$$
\begin{aligned}
\sqrt{\tfrac{1+x^{(k+1)}}{2}} &\geq x^{(k)}, \quad k = 1, \ldots, n-1,\\
x^{(1)} &\geq 1.
\end{aligned}
\tag{78}
$$

Note that the functional components of this representation are Lipschitz continuous on the positive orthant. Let us define now the following function:

$$
\psi(x) = \max\left\{1 - x^{(1)}, x^{(n)} - 1, \max_{1\leq k\leq n-1}\left[x^{(k)} - \sqrt{\tfrac{1+x^{(k+1)}}{2}}\right]\right\}, \quad x \in Q \equiv R^n_+.
$$

This function is nonnegative on its feasible set and $L_0(\psi) = \frac{3}{4}\sqrt{2}$. It attains its minimum at $x^* = (1, \ldots, 1)^T$. Thus, our next test problem is as follows:

$$
\min_x\{\psi(x) : x \in Q\}, \quad x_0 = (1, \ldots, 1, 2)^T. \tag{79}
$$

Thus, we can choose $R = 1$. The results of our experiments are presented in Table 6.

The problem (79) was solved by all methods up to accuracy $\epsilon = 10^{-4}$. As we can see, the standard subgradient method behaves on this problem rather poorly. This may confirm an intuition that on highly degenerate problems, the random search directions have more chances to succeed as compared with regular short-step procedures. On the other hand, we recall the reader that in this table, the results for random search methods are given in blocks of n iterations. Therefore, if we would compare the number of calls of oracle, subgradient method will be almost the best, at least for the problems of high dimension.

8.4 Conclusion

Our experiments confirm the following conclusion. If the computation of the gradient is feasible, then the cost per iteration for random methods and gradient methods is approximately the same. In this situation, the total time spent by the random methods is typically $O(n)$ times bigger than the time required for the gradient schemes to reach the same accuracy. Hence, the random gradient-free methods should be used only if creation of the code for computing the gradient is too costly or just impractical.

In the latter case, for smooth functions, the accelerated scheme (60) demonstrates better performance. This practical observation is confirmed by the theoretical results. For nonsmooth problems, the situation is more delicate. In our experiments, the finite-difference version $\mathcal{RS}_\mu$ was always better than the method $\mathcal{RS}_0$, based on the exact directional derivative. Up to now, we did not manage to find a reasonable explanation for this phenomenon. It remains an interesting topic for the future research.

Acknowledgements The authors would like to thank two anonymous referees for enormously careful and helpful comments. Pavel Dvurechensky proposed a better proof of inequality (37), which we use in this paper. Research activity of the first author for this paper was partially supported by the grant "Action de recherche concertè ARC 04/09-315" from the "Direction de la recherche scientifique - Communautè française de Belgique," and RFBR research projects 13-01-12007 ofi_m. The second author was supported by Laboratory of Structural Methods of Data Analysis in Predictive Modeling, MIPT, through RF government grant, ag.11.G34.31.0073.

Appendix: Proofs of Statements of Sect. 2

Proof of Lemma 1 Denote $\psi(p) = \ln M_p$. This function is convex in p. Let us represent $p = (1-\alpha)\cdot 0 + \alpha \cdot 2$ (thus, $\alpha = \frac{p}{2}$). For $p \in [0, 2]$, we have $\alpha \in [0, 1]$. Therefore,

$$\psi(p) \leq (1-\alpha)\psi(0) + \alpha\psi(2) \overset{(14)}{=} \frac{p}{2}\ln n.$$

This is the upper bound (16). If $p \geq 2$, then $\alpha \geq 1$, and $\alpha\psi(2)$ becomes a lower bound for $\psi(p)$. It remains to prove the upper bound in (17).

Let us fix some $\tau \in (0, 1)$. Note that for any $t \geq 0$ we have

$$t^p \mathrm{e}^{-\frac{\tau}{2}t^2} \leq \left(\frac{p}{\tau e}\right)^{p/2}. \tag{80}$$

Therefore,

$$\begin{aligned} M_p &= \frac{1}{\kappa}\int\limits_E \|u\|^p \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u = \frac{1}{\kappa}\int\limits_E \|u\|^p \mathrm{e}^{-\frac{\tau}{2}\|u\|^2} \mathrm{e}^{-\frac{1-\tau}{2}\|u\|^2} \mathrm{d}u \\ &\overset{(80)}{\leq} \frac{1}{\kappa}\left(\frac{p}{\tau e}\right)^{p/2} \int\limits_E \mathrm{e}^{-\frac{1-\tau}{2}\|u\|^2} \mathrm{d}u = \left(\frac{p}{\tau e}\right)^{p/2} \frac{1}{(1-\tau)^{n/2}}. \end{aligned}$$

The minimum of the right-hand side in $\tau \in (0, 1)$ is attained at $\tau = \frac{p}{p+n}$. Thus,

$$M_p \leq \left(\tfrac{p}{e}\right)^{p/2} \left(1 + \tfrac{n}{p}\right)^{p/2} \left(1 + \tfrac{p}{n}\right)^{n/2} \leq (p+n)^{p/2}.$$

□

Proof of Theorem 1 Indeed, for any $x \in E$ we have $f_\mu(x) - f(x) = \frac{1}{\kappa} \int\limits_E [f(x + \mu u) - f(x)] \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u$. Therefore, if $f \in C^{0,0}(E)$, then

$$\begin{aligned} |f_\mu(x) - f(x)| &\leq \tfrac{1}{\kappa} \int\limits_E |f(x + \mu u) - f(x)| \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u \\ &\leq \tfrac{\mu L_0(f)}{\kappa} \int\limits_E \|u\| \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u \overset{(16)}{\leq} \mu L_0(f) n^{1/2}. \end{aligned}$$

Further, if f is differentiable at x, then

$$f_\mu(x) - f(x) = \tfrac{1}{\kappa} \int\limits_E [f(x + \mu u) - f(x) - \mu \langle \nabla f(x), u \rangle] \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u.$$

Therefore, if $f \in C^{1,1}(E)$, then

$$|f_\mu(x) - f(x)| \overset{(6)}{\leq} \tfrac{\mu^2 L_1(f)}{2\kappa} \int\limits_E \|u\|^2 \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u \overset{(14)}{=} \tfrac{\mu^2 L_1(f)}{2} n.$$

Finally, if f is twice differentiable at x, then

$$\begin{aligned} &\tfrac{1}{\kappa} \int\limits_E [f(x + \mu u) - f(x) - \mu \langle \nabla f(x), u \rangle - \tfrac{\mu^2}{2} \langle \nabla^2 f(x) u, u \rangle] \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u \\ &\quad \overset{(13)}{=} f_\mu(x) - f(x) - \tfrac{\mu^2}{2} \langle \nabla^2 f(x), B^{-1} \rangle. \end{aligned}$$

Therefore, if $f \in C^{2,2}(E)$, then

$$\begin{aligned} |f_\mu(x) - f(x) - \frac{\mu^2}{2} \langle \nabla^2 f(x), B^{-1} \rangle| &\overset{(7)}{\leq} \frac{\mu^3 L_2(f)}{6\kappa} \int\limits_E \|u\|^3 \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u \\ &\overset{(17)}{=} \frac{\mu^3 L_1(f)}{6} (n+3)^{3/2}. \end{aligned}$$

□

Proof of Lemma 2 Indeed, for all x and y in E, we have

$$\begin{aligned}\|\nabla f_\mu(x) - \nabla f_\mu(y)\|_* &\overset{(21)}{\leq} \tfrac{1}{\kappa\mu} \int\limits_E |f(x+\mu u) - f(y+\mu u)|\, \|u\| e^{-\frac{1}{2}\|u\|^2} du \\ &\leq \tfrac{1}{\kappa\mu} L_0(f) \int\limits_E \|u\| e^{-\frac{1}{2}\|u\|^2} du \cdot \|x-y\|.\end{aligned}$$

It remains to apply (16). □

Proof of Theorem 2 Let $\mu > 0$. Since f_μ is convex, for all x and $y \in E$ we have

$$\begin{aligned} f(y) + \mu L_0(f) n^{1/2} &\overset{(18)}{\geq} f_\mu(y) \geq f_\mu(x) + \langle \nabla f_\mu(x), y-x\rangle \\ &\overset{(11)}{\geq} f(x) + \langle \nabla f_\mu(x), y-x\rangle.\end{aligned}$$

Taking now the limit as $\mu \to 0$, we prove the statement for $\mu = 0$. □

Proof of Lemma 3 Indeed, for function $f \in C^{1,1}(E)$, we have

$$\begin{aligned}\|\nabla f_\mu(x) - \nabla f(x)\|_* &\overset{(25)}{=} \left\| \tfrac{1}{\kappa} \int\limits_E \left(\tfrac{f(x+\mu u) - f(x)}{\mu} - \langle \nabla f(x), u\rangle \right) Bu\, e^{-\frac{1}{2}\|u\|^2} du \right\|_* \\ &\leq \tfrac{1}{\kappa\mu} \int\limits_E |f(x+\mu u) - f(x) - \mu\langle \nabla f(x), u\rangle|\, \|u\| e^{-\frac{1}{2}\|u\|^2} du \\ &\overset{(6)}{\leq} \tfrac{\mu L_1(f)}{2\kappa} \int\limits_E \|u\|^3 e^{-\frac{1}{2}\|u\|^2} du \overset{(17)}{\leq} \tfrac{\mu}{2} L_1(f)(n+3)^{3/2}.\end{aligned}$$

Let $f \in C^{2,2}(E)$. Denote $a_u(\tau) = f(x+\tau u) - f(x) - \tau\langle \nabla f(x), u\rangle - \frac{\tau^2}{2}\langle \nabla^2 f(x) u, u\rangle$. Then, $|a_u(\pm\mu)| \overset{(7)}{\leq} \frac{\mu^3}{6} L_2(f)\|u\|^3$. Since

$$\begin{aligned}\nabla f_\mu(x) - \nabla f(x) \overset{(13)}{=} \frac{1}{2\kappa\mu} \int\limits_E &[f(x+\mu u) - f(x-\mu u) \\ &-2\mu\langle \nabla f(x), u\rangle]\, Bu e^{-\frac{1}{2}\|u\|^2} du,\end{aligned}$$

we have

$$\begin{aligned}\|\nabla f_\mu(x) - \nabla f(x)\|_* &\leq \frac{1}{2\kappa\mu} \int\limits_E |f(x+\mu u) - f(x-\mu u) - 2\mu\langle \nabla f(x), u\rangle|\, \|u\| e^{-\frac{1}{2}\|u\|^2} du \\ &= \frac{1}{2\kappa\mu} \int\limits_E |a_u(\mu) - a_u(-\mu)|\, \|u\| e^{-\frac{1}{2}\|u\|^2} du\end{aligned}$$

$$\leq \frac{\mu^2 L_2(f)}{6\kappa} \int\limits_E \|u\|^4 \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u \overset{(17)}{\leq} \frac{\mu^2}{6} L_2(f)(n+4)^2.$$

□

Proof of Lemma 4 Indeed,

$$\|\nabla f(x)\|_*^2 \overset{(13)}{=} \|\tfrac{1}{\kappa} \int\limits_E \langle \nabla f(x), u\rangle Bu\mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u\|_*^2$$

$$= \|\tfrac{1}{\kappa\mu} \int\limits_E ([f(x+\mu u)-f(x)]-[f(x+\mu u)-f(x)-\mu\langle \nabla f(x), u\rangle])\, Bu\mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u\|_*^2$$

$$\overset{(26)}{\leq} 2\|\nabla f_\mu(x)\|_*^2 + \tfrac{2}{\mu^2}\|\tfrac{1}{\kappa} \int\limits_E [f(x+\mu u) - f(x) - \mu\langle \nabla f(x), u\rangle] Bu\mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u\|_*^2$$

$$\leq 2\|\nabla f_\mu(x)\|_*^2 + \tfrac{2}{\mu^2\kappa} \int\limits_E [f(x+\mu u) - f(x) - \mu\langle \nabla f(x), u\rangle]^2 \|u\|^2 \mathrm{e}^{-\frac{1}{2}\|u\|^2} \mathrm{d}u$$

$$\overset{(6)}{\leq} 2\|\nabla f_\mu(x)\|_*^2 + \tfrac{\mu^2}{2} L_1^2(f) M_6.$$

It remains to use inequality (17). □

References

1. A. Agarwal, O. Dekel, and L. Xiao, *Optimal algorithms for online convex optimization with multi-point bandit feedback*, in Proceedings of the 23rd Annual Conference on Learning, 2010, pp. 2840.
2. A. Agarwal, D. Foster, D. Hsu, S. Kakade, and A. Rakhlin, *Stochastic convex optimization with bandit feedback*,. SIAM J. on Optimization, 23 (2013), pp. 213–240.
3. D. Bertsimas and S. Vempala, *Solving convex programs by random walks*, J. of the ACM, 51 (2004), pp. 540–556.
4. F. Clarke, *Optimization and nonsmooth analysis*, Wliley, New York, 1983.
5. A. Conn, K. Scheinberg, and L. Vicente , *Introduction to derivative-free optimization. MPS-SIAM series on optimization*, SIAM, Philadelphia, 2009.
6. C. Dorea, *Expected number of steps of a random optimization method*, JOTA, 39 (1983), pp. 165–171.
7. J. Duchi, M.I. Jordan, M.J. Wainwright, and A. Wibisono, *Finite sample convergence rate of zero-order stochastic optimization methods*, in NIPS, 2012, pp. 1448–1456.
8. A. D. Flaxman, A.T. Kalai, and B.H. McMahan, *Online convex optimization in the bandit setting: gradient descent without a gradient*, in Proceedings of the 16th annual ACM-SIAM symposium on Discrete Algorithms, 2005, pp. 385–394 .
9. R. Kleinberg, A. Slivkins, and E. Upfal, *Multi-armed bandits in metric spaces*, in Proceedings of the 40th annual ACM symposium on Theory of Computing, 2008, pp. 681–690.
10. J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, *Convergence properties of the Nelder-Mead Simplex Algorithm in low dimensions*, SIAM J. Optimization, 9 (1998), pp. 112–147.
11. J. C. Lagarias, B. Poonen, and M. H. Wright, *Convergence of the restricted Nelder-Mead algorithm in two dimensions*, SIAM J. Optimization, 22 (2012), pp. 501–532.
12. J. Matyas, *Random optimization*. Automation and Remote Control, 26 (1965), pp. 246–253.
13. J. A. Nelder and R. Mead, *A simplex method for function minimization*, Computer Journal, 7 (1965), pp. 308–3013

14. A. Nemirovski, A. Juditsky, G. Lan, and A.Shapiro, *Robust Stochastic Approximation approach to Stochastic Programming*, SIAM J. on Optimization, 19 (2009), pp. 1574–1609.
15. A. Nemirovsky and D.Yudin, *Problem complexity and method efficiency in optimization*, John Wiley and Sons, New York, 1983.
16. Yu. Nesterov, *Introductory Lectures on Convex Optimization*, Kluwer, Boston, 2004.
17. Yu. Nesterov, *Lexicographic differentiation of nonsmooth functions'*, Mathematical Programming, 104 (2005), pp. 669–700.
18. Yu. Nesterov, *Random gradient-free minimization of convex functions*, CORE Discussion Paper # 2011/1, (2011).
19. Yu. Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*. SIAM J. on Optimization, 22 (2012), pp. 341–362.
20. B. Polyak, *Introduction to Optimization*. Optimization Software - Inc., Publications Division, New York, 1987.
21. V. Protasov, *Algorithms for approximate calculation of the minimum of a convex function from its values*, Mathematical Notes, 59 (1996), pp. 69–74.
22. M. Sarma, *On the convergence of the Baba and Dorea random optimization methods*, JOTA, 66 (1990), pp. 337–343.