Randomized gradient-free methods in convex optimization

Alexander Gasnikov1,2,3, Darina Dvinskikh1,4,6, Pavel Dvurechensky7, Eduard Gorbunov5, Aleksandr Beznosikov1, Aleksandr Lobanov1,8

1. Moscow Institute of Physics and Technology, Moscow, Russia
2. Caucasus Mathematical Center, Adyghe State University, Maikop, Russia
3. Innopolis University, Innopolis, Russia
4. IITP RAS, Moscow, Russia
5. MBZUAI, Abu Dhabi, UAE
6. WIAS, Berlin, Germany
7. HSE University, Moscow, Russia
8. ISP RAS Research Center for Trusted Artificial Intelligence, Moscow, Russia

MSC Codes: 90C25, 90C30, 90C56, 68Q25, 68W20, 65Y20

Keywords: gradient-free methods, derivative-free methods, zeroth-order methods, stochastic optimization, accelerated methods, batch-parallelization, inexact oracle

1 Introduction

Consider convex optimization problem

\[ \min_{x \in \mathbb{R}^d} f(x) \]  

with convex feasible set Q, convex objective f possessing the zeroth-order (gradient/derivative-free) oracle [83]. The latter means that one has an access only to the values of the objective f(x) rather than to its gradient \( \nabla f(x) \) that is more popular for numerical methods [22,32]. One can find motivating examples in [45, 90, 19]. This paper focuses on numerical zeroth-order methods based on first-order methods with different approximation models of the gradient.

The first such methods (often referred to as Kiefer–Wolfowitz methods) were proposed about 70 years ago (see, e.g., [95] for historical remarks) and further developed in the following 20–30 years [29,70,77,90].

The interest in gradient-free methods has grown considerably since mid-2000s [11,24,25,66,73,87,92]. Particularly, in modern papers [34,26,3], the authors try to develop the best known gradient-free algorithms at least for one of three main criteria: oracle complexity (number of zeroth-oracle calls to guarantee certain accuracy), iteration complexity (number of iterations to guarantee certain accuracy), maximum level of admissible noise still allowing to guarantee certain accuracy. This survey focuses on the same aspects and presents the algorithms through the prism of optimality according to the listed criteria.

2 Full gradient approximation. Smooth case

Assume that one can query zeroth-order oracle which returns a noisy value of f(x): f_{\delta}(x) = f(x) + \delta(x), where the noise is bounded by some \( \Delta \geq 0 \), i.e., \(|\delta(x)| \leq \Delta \). In some applications, (see e.g. [14]) the larger is \( \Delta \) the cheaper is the call of inexact oracle f_{\delta}. If f in (1) has L-Lipschitz gradient (e.g., w.r.t. the Euclidean 2-norm), then one can approximate partial derivatives using the forward differences of inexact values of the objective as follows:

\[
\left| \frac{\partial f(x)}{\partial x_i} - \frac{f_{\delta}(x + he_i) - f_{\delta}(x)}{h} \right| \leq \frac{Lh}{2} + \frac{2\Delta}{h} \leq 2 \sqrt{\Delta L},
\]

where \( e_i \) is the i-th basis vector and \( h = 2 \sqrt{\Delta L} \).

Furthermore, if f in (1) has L-Lipschitz Hessian (e.g., w.r.t. the 2-norm) then one can approximate partial derivatives using symmetric differences as follows:

\[
\left| \frac{\partial f(x)}{\partial x_i} - \frac{f_{\delta}(x + he_i) - f_{\delta}(x - he_i)}{2h} \right| \leq \frac{Lh^2}{6} + \frac{\Delta}{h} \leq 2L^{1/3} \Delta^{2/3}.
\]

where \( h = (3\Delta/L)^{1/3} \). Similarly, if f has Lipschitz n-th order derivatives, one can approximate partial derivatives by using n-th order finite-difference scheme with accuracy \( \sim \Delta^{n/2} \) [93]. This means that one can build a finite-difference inexact gradient \( \nabla f_{\delta}(x) \) with accuracy

\[ \|\nabla f(x) - \nabla f_{\delta}(x)\|_2 \leq \sqrt{n} \Delta^{n/2}. \]

Here and below \( \leq \) is used to denote inequality up to possible dimension-dependent constants.

Most of first-order, i.e., gradient, algorithms under a proper stopping rule are «robust», i.e., they do not accumulate additive error in the gradient [52, 41, 93]. Particularly, if the goal is to solve (1) with a sufficiently small accuracy \( \epsilon \) in terms of the objective function value (i.e., s.t. the point \( \hat{x}^N \) generated after N steps satisfies \( f(\hat{x}^N) - f(x^*) \leq \epsilon \)), then the additive error should be proportional to \( \epsilon \), i.e., \( \sqrt{\Delta L} \approx \epsilon / R \), where \( R = \|x^0 - x^*\|_2 \) with \( x^0 \) being the starting point and \( x^* \) being the solution to (1) closest to \( x^0 \). Note that it is easy to show that \( R \) can be bounded by the diameter of Q. Showing that \( R \) is determined as a distance between the starting point \( x^0 \) and the solution \( x^* \) requires a more accurate analysis. This remark is important when Q is not a compact set or if the goal is to generalize the results for minimizing strongly convex functions by using the restart-technique [26]. Thus, the maximum admissible level of noise is \( \Delta \leq \frac{\epsilon}{\sqrt{d} \sqrt{\log(1/\delta)}} \). If \( n \to \infty \), this bound has a form \( \Delta \leq \frac{\epsilon}{\sqrt{d} \sqrt{\log(1/\delta)}} \), which is a lower bound [88,81]. However, as it is explained below (see Section 4.1), for finite values of \( n \), the bound for the admissible level of noise can be improved.

In the case when f is an analytical function and \( \Delta = 0 \), one can use a complex representation of the partial derivatives [91,51]

\[
\left| \frac{\partial f(x)}{\partial x_i} - \Im \left( \frac{f(x + ih)}{h} \right) \right| \leq h^2.
\]
Typically in practice, when the floating point arithmetic is used, one has a double precision $\Delta \approx 1.1 \cdot 10^{-16}$ (so $h$ should be smaller than $10^{-15}$), and with this precision, when calculating $\partial f(x)/\partial x_i$, one has $\sqrt{\Delta}$ error in the gradient, meaning that one can solve (1) with accuracy $\epsilon$ proportional to this error. A close result for large $n$ can be obtained by using the finite-difference approach with the change $\Delta \rightarrow \Delta$. This example is described to emphasise the difference between the inexactness of the zeroth-order oracle and the finite-precision inexactness. The last one is less restrictive and opens up opportunities to use more unconventional approaches for gradient approximation.

Now, everything it ready for a description of the most simple way to build optimal gradient-free methods. The basic observation here is as follows: optimal Nesterov’s accelerated gradient from [72] is robust [41, 93]. Starting from this gradient-based algorithm one can build an optimal gradient-free algorithm [70] by replacing the exact gradient by its finite-difference approximation. Such gradient-free algorithm, in general, is optimal in terms of the number of oracle calls $O\left(d \sqrt{\frac{\|f\|_2^2}{\epsilon}}\right)$ and iteration complexity $O\left(\sqrt{\frac{\|\gamma\|_e}{\epsilon}}\right)$, and close to optimal algorithms (but is still not optimal) in terms of the maximum level of admissible noise with the bound $\Delta \leq e^{-d}d^{-\frac{3}{2}}$. The lower bound is not known, but the best known result $\Delta \leq e^{-d}d^{-\frac{3}{2}}$ requires more iterations or oracle calls, see Section 4.1 for $n = 1$.

3 Randomized gradient approximation. Smooth case

At the end of Section 2 a rather simple approach that is based on the gradient approximation is described. This approach was announced to be optimal in terms of the number of oracle calls. In this section, alternative randomized approximations that can improve oracle complexity under some additional assumptions are considered.

3.1 Coordinate-wise randomization

Coordinate methods have a long history going back to the Gauss–Seidel method [77]. A contemporary view on these methods was proposed by Yu. Nesterov [71], followed by a large number of works making this area of optimization well-developed [80, 96, 33, 74, 46]. The core idea of these methods is to replace exact gradient by its unbiased approximation

$$\tilde{\nabla} f(x, i) = d \cdot \frac{\partial f(x)}{\partial x_i} \cdot e,$$

where $i$ is chosen from $1, ..., d$ randomly with equal probabilities. If the oracle returns the exact value of $\partial f(x)/\partial x_i$, then $\mathbb{E}[\tilde{\nabla} f(x, i)] = \nabla f(x)$ and $\mathbb{E}[\|\tilde{\nabla} f(x, i)\|_2^2] = d\|
abla f(x)\|_2^2$. This leads to the «strong growth condition» [94] and the same oracle calls complexity bound as that of for the methods with full gradient approximation discussed at the end of Section 2 i.e.,

$$O\left(d \sqrt{\frac{\|f\|_2^2}{\epsilon}}\right).$$

Here and below, when considering randomized algorithms, $\epsilon$ is a desired accuracy in terms of expectation of the objective function value. A more accurate analysis allows to improve this bound. Namely, let $L_{i}$ satisfy

$$\left|\frac{\partial f(x + he_i)}{\partial x_i} - \frac{\partial f(x)}{\partial x_i}\right| \leq L_{i}h,$$

and $L = \frac{1}{d} \sum_{i=1}^{d} L_{i}$. For example, if $f(x) = \frac{1}{d} \langle Ax, x \rangle - \langle b, x \rangle$, $L = \lambda_{\max}(A)$, $L = \frac{1}{d}(\text{tr}(A)) = \frac{1}{d} \sum_{i=1}^{d} L_{i}$. Hence, $\frac{1}{d} L \leq \bar{L} \leq L$ [74]. A proper version of accelerated (catalyst-based) Nesterov’s coordinate descent method [49, 48] requires $O\left(d \sqrt{\frac{\|f\|_2^2}{\epsilon}}\right)$ oracle calls that could be better in $O\left(\sqrt{d}\right)$ times than the previous bound. Further discussions of the gradient-free modifications of coordinate descents are given in Section 3.3.

3.2 Random search randomization

The core idea of random-search methods (see, e.g., [28, 29]) is to replace the exact gradient by its unbiased random approximation

$$\hat{\nabla} f(x, e) = d \cdot \langle \nabla f(x), e \rangle \cdot e,$$

where $e$ is chosen uniformly at random from the unit Euclidean sphere $S^d(1)$. If the oracle returns the exact value of $\langle \nabla f(x), e \rangle$, then $\mathbb{E}[\tilde{\nabla} f(x, e)] = \langle \nabla f(x), e \rangle$ and $\mathbb{E}[\|\tilde{\nabla} f(x, e)\|_2^2] = d\|\nabla f(x)\|_2^2$. Thus, the «strong growth condition» [94] still holds and one can obtain the same oracle calls complexity as that of for the methods with full gradient approximation discussed at the end of Section 2 i.e., $O\left(d \sqrt{\frac{\|f\|_2^2}{\epsilon}}\right)$. A more accurate analysis of proper accelerated (linear-coupling) random-search method [28] allows us to improve this bound to $O\left(d \sqrt{\frac{\|f\|_2^2}{\epsilon}}\right)$.

where $L$ (as it was above) is a Lipschitz gradient constant of $f$ in the 2-norm. On the one hand, $\|\tilde{\nabla} f(x)\|_2 = \|\nabla f(x)\|_2$, and close to optimal algorithms. The upper bound is not known, but the best known result $\Delta \leq e^{-d}d^{-\frac{3}{2}}$ requires more iterations or oracle calls, see Section 4.1 for $n = 1$.

3.3 Gradient-free randomized methods. Smooth case

In Sections 3.1 and 3.2, the ways that make it possible to improve oracle complexity (but worsen the iteration complexity at the same time) are described. But, these approaches make use of the partial derivatives oracle. These results can be naturally generalized to the setting of the zeroth-oracle inexact oracle by the following modification of example (3):

$$\hat{\nabla} f(x, e) = \frac{f(x + he) - f(x - he)}{2h}.$$

If $f$ has Lipschitz $n$-th order derivatives, the «kernel-based» approximation from [28, 7, 17, 7, 48, 65, 66] can be used to avoid $O(n)$-order finite-difference schemes:

$$\hat{\nabla} f(x, e) = \frac{\hat{f}(x + he) - \hat{f}(x - he)}{2h}$$

where $\kappa$ is uniformly distributed in $[-1, 1]$, $e$ is uniformly distributed in $S^d(1)$, and $\epsilon$ are independent, $K_{\kappa}(\epsilon)$ is a kernel function that satisfies $\mathbb{E}[K_{\kappa}(\epsilon)] = 0$, $\mathbb{E}[(x K_{\kappa}(\epsilon)] = 1$, $\mathbb{E}[\epsilon^2 K_{\kappa}(\epsilon)] = 0$, $j = 2, ..., n$, $\mathbb{E}[\|\epsilon^2 K_{\kappa}(\epsilon)] < \infty$. For example,

$$K_{\kappa}(\epsilon) = 3\kappa, n = 1, 2; K_{\kappa}(\epsilon) = \frac{12\epsilon^2}{(5 - 7\kappa^2)}, n = 3, 4 ...$$
A more accurate analysis allows one to replace the gradient approximation condition \( \text{(3)} \) by the following \( \text{[5, 34, 75]} \): for all \( r \in \mathbb{R}^2 \),
\[
\mathbb{E}_r \left( \nabla f(x, e) - \nabla f(x), r \right) \leq \sqrt{d} \left( h^2 + \frac{\Delta}{h} \right) \|r\|_2 \leq \sqrt{d} \Delta^{-\frac{1}{2}} \|r\|_2.
\]
(6)

For coordinate-wise randomization, one can further improve this bound replacing \( \sqrt{d} \|r\|_2 \) by \( \|r\|_1 \). If one wants to solve \( \text{(1)} \) with a sufficiently small accuracy \( \epsilon \) in terms of the function value, i.e., \( \mathbb{E} f(x_\epsilon^0) - f(x_\epsilon) \leq \epsilon \) (this criteria will be called \( \epsilon \)-suboptimality in the expectation in this survey), then the RHS of \( \text{(6)} \) should be of the order of \( \epsilon \approx \sqrt{d} \Delta^{-\frac{1}{2}} R \approx \epsilon \), where \( R = \|x_\epsilon^0 - x_\epsilon\|_2 \). Thus, the maximum admissible level of noise is \( \Delta \leq \epsilon \frac{d}{R} \Delta^{-\frac{1}{2}} \), i.e., the same as it was for the algorithms using the full gradient approximation, see Section 2.

4 Randomized gradient approximation. Non-smooth case

If \( f \) in \( \text{(1)} \) is non-smooth but is Lipschitz continuous in \( p \in [1, 2] \) norm: \( |f(y) - f(x)| \leq M |y - x|_p \) (if \( p = 2 \) this constant will be denoted as \( M_2 \)), then the finite-difference approximation approaches do not work. To resolve this problem, different smoothing techniques are used \( \text{[70, 90, 36, 73, 3, 64, 65, 57, 69, 56]} \). The Gaussian smoothing \( \text{(3)} \) is not considered because it does not give anything new compared to smoothing on the \( l_2 \)-ball \( \text{(23)} \). This section is considered to be the most important part of the survey.

4.1 \( l_2 \)-ball smoothing scheme

This technique goes back to the works of A. Nemirovski, Yu. Er- momsiev, A. Gupal of mid- and late 1970s \( \text{[70, 45]} \), see also \( \text{[90]} \). In this section, the narrative follows \( \text{[34]} \).

The first element of the approach \( \text{[34]} \) is the randomized smoothing for the non-smooth objective \( f: \)
\[
f_f(x) = \mathbb{E}_uf(x + \gamma u),
\]
where \( u \sim RB^2(1) \), i.e., \( u \) is random vector uniformly distributed on a unit euclidean ball \( B^2(1) \). The following theorem is a generalization of the results \( \text{[97, 24]} \) for non-Euclidean norms.

**Theorem 1** (properties of \( f_f \)). For all \( x, y \in Q \)
- \( f(x) \leq f_f(x) \leq f(x) + \gamma M_2 \);  
- \( f_f(x) \) has \( L = \frac{\sqrt{d}M_2}{\gamma} \)-Lipschitz gradient:
\[
\|\nabla f_f(y) - \nabla f_f(x)\|_q \leq \frac{L}{q} |y - x|_p,
\]
where \( q \) is such that \( 1/p + 1/q = 1 \).

The second very important element of the approach \( \text{[34]} \) goes back to O. Shamir \( \text{[87]} \), who proposed a special unbiased stochastic gradient for \( f_f \) with small variance (note that the forward finite-difference has large variance \( \text{(25)} \)):
\[
\nabla f_f(x, e) = d \frac{f(x + ye) - f(x - ye)}{2\gamma} e,
\]
(7)
where \( e \sim RS^2(1) \) is a random vector uniformly distributed on \( S^2 \text{(1)} \) (see also \( \text{[4]} \) for comparison). To simplify the further derivations, here and below (until the end of the section) it is assumed that \( \Delta = 0 \).

The following theorem is a combination of the results from \( \text{[87, 43, 10]} \).

**Theorem 2** (properties of \( \nabla f_f(x, e) \)). For all \( x \in Q \)
- \( \nabla f_f(x, e) \) is an unbiased approximation for \( \nabla f_f(x) \):
\[
\mathbb{E}_e \left( \nabla f_f(x, e) \right) = \nabla f_f(x);
\]
- \( \nabla f_f(x, e) \) has bounded variance (second moment):
\[
\mathbb{E}_e \left[ \|\nabla f_f(x, e)\|^2 \right] \leq \sqrt{d} \min\{q, ln \|d\| \} d^2 M_2^2,
\]
(8)
where \( 1/p + 1/q = 1 \).

Based on the two elements above, everything is ready for the description of a general approach, which for shortness, is called in this paper as the smoothing scheme (technique).

Assume that one has an access to some batched algorithm \( A(L, \sigma^2 \epsilon^2) \) that solves problem \( \text{(1)} \) under the assumption that \( f \) is smooth and satisfies
\[
\|\nabla f(y) - \nabla f(x)\|_2 \leq L |y - x|_p,
\]
and by using a stochastic first-order oracle that depends on a random variable \( \eta \) and returns at a point \( x \) an unbiased stochastic gradient \( \nabla_x f(x, \eta) \) with bounded variance:
\[
\mathbb{E}_\eta \left[ \|\nabla_x f(x, \eta) - \nabla f(x)\|_2^2 \right] \leq \sigma^2.
\]

Further, assume that, to reach \( \epsilon \)-suboptimality in expectation, this algorithm requires \( N(L, \epsilon) \) successive iterations and \( T(L, \sigma^2, \epsilon) \) stochastic first-order oracle calls, i.e., \( A(L, \sigma^2 \epsilon^2) \) allows batch-parallelization with the average batch size
\[
B(L, \sigma^2 \epsilon^2) = \frac{T(L, \sigma^2 \epsilon^2, \epsilon)}{N(L, \epsilon)}.
\]

The approach of \( \text{[34]} \) is to apply \( A(L, \sigma^2 \epsilon^2) \) to the smoothed problem
\[
\min_{x \in \mathbb{R}^{2d}} f_f(x)
\]
with
\[
\gamma = \epsilon/(2M_2)
\]
and \( \eta = \epsilon, \nabla f(x, \eta) = \nabla f_f(x, \epsilon) \), where \( \epsilon > 0 \) is the desired accuracy for solving problem \( \text{(1)} \) in terms of the suboptimality expectation.

According to Theorem \( \text{(1)} \) an \( (\epsilon/2) \)-solution to \( \text{(9)} \) is an \( \epsilon \)-solution to the initial problem \( \text{(1)} \). According to Theorem \( \text{(10)} \) and \( \text{(13)} \), the following inequality holds
\[
L \leq \frac{2 \sqrt{d} M M_2}{\epsilon},
\]
(11)
and Theorem \( \text{(2)} \) implies
\[
\sigma^2 \leq \sigma^2 \epsilon^2 \frac{2}{d \epsilon} = 2 \sqrt{\frac{2}{d}} \frac{\min\{q, \ln \|d\| \}}{\epsilon} d^2 M_2^2.
\]
(12)
Thus, it is shown that \( A(L, \sigma^2 \epsilon^2) \) implemented using stochastic gradient \( \text{(7)} \) is a zeroth-order method for solving non-smooth
problem (1). Moreover, to solve problem (1) with accuracy $\varepsilon > 0$ this method suffices to make

$$N \left( \frac{2 \sqrt{MM_2}}{\varepsilon}, e \right) \text{ successive iterations and}$$

$$2T \left( \frac{2 \sqrt{MM_2}}{\varepsilon}, \sigma^2(p, d), e \right) \text{ zeroth-order oracle calls.}$$

It is important to mention that this approach is flexible and generic as one can take different algorithms as $A(L, \sigma^2)$. For example, if one takes batched Accelerated gradient method [20, 58, 22, 27, 41], then from (11), (12) it follows that

$$N(L, \varepsilon) = O \left( \frac{L^2}{\varepsilon} \right),$$

$$T(L, \sigma^2, \varepsilon) = O \left( \max \left( N(L, \varepsilon), \frac{\sigma^2(p, d)}{\varepsilon} \right) \right) = O \left( \frac{\sigma^2(p, d)}{\varepsilon^2} \right),$$

where $R = O \left( \|x^0 - x\|_p, \min(q, \ln d) \right)$. The last equality on $T$ assumes also that $\varepsilon \leq d^{-1/4}M_2^2R/M^{1/2}$ when $p = 1$.

**Theorem 3.** Based on the batched Accelerated gradient method, the smoothing scheme applied to non-smooth problem (1),

provides a gradient-free method with

$$O \left( \frac{d^{1/4} \sqrt{MM_2} R}{\varepsilon} \right) \text{ successive iterations and}$$

$$O \left( \frac{\min \{ q, \ln d \} \sqrt{M_2^2 R^2}}{\varepsilon^2} \right) \text{ zeroth-order oracle calls, where } 1/p + 1/q = 1.$$
strongly convex problem \(^1\), provides a gradient-free method with \(O\left(\frac{d^2}{\gamma^4} \frac{M}{\varepsilon^2}\right)\) successive iterations and \(O\left(\min\{|q, \ln d|\varepsilon^2\}^2\right)\) zeroth-order oracle calls.

Analogously, all other results in this paper (except the results from the end of Section 4.4 from Section 4.5, and some results on maximum admissible level of noise) can be transferred for strongly convex problem \(^1\).

4.2 \(l_1\)-ball smoothing scheme

In the core of the Euclidean smoothing scheme lies the Stokes’ formula \(^3\) (here \(\nabla f(x)\) is subgradient):

\[
\int_D \nabla f(x) dx = \int_{\partial D} f(x) n(x) dS(x),
\]

where \(\partial D\) is a boundary of \(D\), \(n(x)\) is an outward normal vector at point \(x\) to \(\partial D\), and \(dS(x)\) denotes the surface measure. The Stokes’ formula implies

\[
\nabla f(x) = E_u \left[ \nabla_s f(x + yu) \right] = E_u \left[ d \left( f(x + ye) \right) \right],
\]

where \(u\) is random uniformly distributed on \(B_2^d(1)\) and \(e\) is random uniformly distributed on \(S_2^d(1)\). In \(^3\) independently in \(^3\), it was mentioned that from the Stokes’ formula one can also obtain

\[
\nabla f(x) = E_u \left[ \nabla_s f(x + yu) \right] = E_u \left[ d \left( f(x + ye) \right) \right],
\]

where \(u\) is random uniformly distributed on \(B_2^d(1)\), \(e\) is random uniformly distributed on \(S_2^d(1)\), and \(n(e) = \frac{1}{|e|} \text{sign}(e)\).

Thus, one can introduce a smoothed version of \(f\)

\[
\tilde{f}(x) = E_u(f(x + yu)),
\]

where \(u \sim RB_2^d(1)\), i.e., \(u\) is random vector uniformly distributed on a unit ball in 1-norm denoted as \(B_2^d(1)\). The following theorems are taken from \(^{15, 33, 42}\).

Theorem 5 (properties of \(\tilde{f}(x)\)). For all \(x, y \in Q\)

1. \(f(x) \leq \tilde{f}(x) \leq f(x) + \frac{M}{\gamma} \varepsilon M_2\),

2. \(\tilde{f}(x)\) has \(L = \frac{dM}{\gamma}\)–Lipschitz gradient:

\[
\|\nabla \tilde{f}(y) - \nabla \tilde{f}(x)\|_p \leq L\|y - x\|_p,
\]

where \(|f(y) - f(x)| \leq M\|y - x\|_p\).

Further, a stochastic gradient is introduced as follows:

\[
\nabla \tilde{f}(x, \tilde{e}) = d \frac{f(x + ye) - f(x - ye)}{2\gamma} \text{sign}(\tilde{e}),
\]

where \(\tilde{e} \sim RS_q^d(1)\) is a random vector uniformly distributed on a unit sphere in 1-norm denoted as \(S_q^d(1)\). In comparison with \(^7\), \(\tilde{f}(x, \tilde{e})\) requires less memory (1 float and \(d\) bits). For simplicity in this formula and in the next theorem, it is assumed that \(\Lambda = 0\).

Theorem 6 (properties of \(\nabla \tilde{f}(x, \tilde{e})\)). For all \(x \in Q\)

1. \(\nabla \tilde{f}(x, \tilde{e})\) is an unbiased approximation for \(\nabla \tilde{f}(x)\):

\[
E_\tilde{e} \left[ \nabla \tilde{f}(x, \tilde{e}) \right] = \nabla \tilde{f}(x);
\]

2. \(\nabla \tilde{f}(x, \tilde{e})\) has bounded variance (second moment):

\[
E_\tilde{e} \left[ \|\nabla \tilde{f}(x, \tilde{e})\|_2^2 \right] \leq 288d^2 M_2^2.
\]

Summarizing, the results of Section 4.1 take place for the \(l_1\)-smoothing scheme, but with potentially better (for huge \(d\) – in practice this potential advantage is not observed \(^{42}\)) factor:

\[
2 \sqrt{2} \min\{|q, \ln d|} \to 288, q \geq 2.
\]

Such randomization leads to the number of successive iterations bounded as

\[
O\left(\frac{d^{1/4} \sqrt{MMR}}{\varepsilon}\right),
\]

and the number of oracle calls bounded as (may depend on \(d\) only through the constants \(R = \|x^0 - x_1\|_p \sqrt{\min\{|q, \ln d|} \) and \(M_2\)):

\[
O\left(\frac{M_2^2 R^2}{\varepsilon^2}\right).
\]

The level of maximum admissible noise coincides with \(^{15}\): \(\Delta \leq \frac{\varepsilon}{\sqrt{\gamma^2\varepsilon^2 + \gamma^2\varepsilon^2}}\), where \(R = \|x^0 - x_1\|_p\). Note that for \(p = 1\) the lower bound is not known. One may expect that this bound could be further improved \(^{42}\) (i.e., maximum admissible level of noise could be large).

4.3 Stochastic optimization problems. Two-point feedback

In this section, for certainty, the \(l_2\)-smoothing scheme approach is considered. The same can be done for any other approaches (e.g., for the algorithms corresponding to Section 3 see \(^{33, 42}\)).

If the zeroth-order oracle returns an unbiased noisy stochastic function value \(f(x, \xi)\) \((E_\xi f(x, \xi) = f(x))\), then with two-point oracle \(^{25, 35}\) one can introduce the following counterpart of \(^7\)

\[
\nabla f(x, \xi, \varepsilon) = \frac{d\left(f(x + ye, \xi) - f(x - ye, \xi)\right)}{2\gamma}.
\]

In this case, a stochastic counterpart of Theorem 2 can be formulated by the appropriate changes of \(\nabla f(x, \varepsilon)\) to \(\nabla f(x, \xi, \varepsilon)\), the expectation \(E_\xi\) to the expectation \(E_{\xi, \tilde{e}}\), and the redefinition of \(M_2\) as a constant satisfying \(E_{\xi, \tilde{e}} \|\nabla f(x, \xi, \varepsilon)\|_2^2 \leq M_2^2\) for all \(x \in Q\). Moreover, under this redefinition of constant \(M_2\) one can also reformulate Theorem 3.

Note that for high-probability deviation bounds in the case of deterministic oracle returning the values of \(f(x)\), one can show sub-Gaussian concentration \(^{26}\). But there is a lack of results for stochastic oracle returning the realizations of \(f(x, \xi)\), see also \(^{26}\). Fortunately, in the approach in Section 4.1 one can take \(A(L, \sigma^2)\) to be clipped batched Accelerated gradient method from \(^{19}\) that guarantee almost sub-Gaussian concentration iff \(E_{\xi} \|\nabla f(x, \xi)\|_2^2 \leq M_2^2\).
4.4 Stochastic optimization problems. One-point feedback

In this section, for certainty, the $l_2$-smoothing scheme approach is considered as a starting point.

If the two-point feedback (20) is unavailable, the $l_2$-smoothing technique can utilize the one-point feedback by using the unbiased estimate [70, 30, 35]:

$$\nabla f_q(x, \xi, e) = d \frac{f(x + ye, \xi)}{\gamma} e$$

with [33]

$$\mathbb{E}_{\xi, e} \left[ \|\nabla f_q(x, \xi, e)\|^2 \right] \leq \frac{(q+1)d^2\log \delta^2}{\gamma^2}, \quad q \in [2, 2\ln d] \quad \mathbb{E}_{\xi, e} \left[ \|\nabla f_q(x, \xi, e)\|^2 \right] \leq \frac{2d^2\ln \delta^2}{\gamma^2}, \quad q \in (2\ln d, \infty),$$

where $\gamma$ is defined in (10) and it is assumed that $\mathbb{E}_\xi f(x, \xi) \leq G^2$ for all $x \in Q$. Thus, the Smoothing technique can be generalized to the one-point feedback setup by replacing the RHS of (12) by the above estimate. This leads to the same iteration complexity $N$, but increases the oracle complexity $T$, and, consequently, the batch size $B$ at each iteration.

In particular, for $p = 2$, $q = 2$ and $f = d^2G^2M_2^2e^2/\epsilon^2$ and $T = dG^2M_2^2/\mu^3$ if $f$ is $\mu$-strongly convex in 2-norm [70, 30, 35]. If $f$ has $L$-Lipschitz gradient in 2-norm then [17] holds true and $T \propto \text{O}(\text{poly}(\epsilon^{-2}))$. For non-smooth stochastic problems the lower bound is similar $\propto d^3/\epsilon^2$ [21, 5], but it could not be tight. Here, for the first time, a situation where the strongly convex case and the convex case are not treated in the same way is encountered.

An important observation related to one-point feedback is the following: in some works (see e.g. [32, 51, 3, 4, 56]) assumptions about adversarial noise $\delta$ imply that $\delta$ does not depend on the used randomization $e$ (and $\kappa$, see Section 3.3). In this case, $\mathbb{E}_{\xi, e, [\kappa e]} \equiv 0$ (see also [43]), where additional randomization reduce adversarial noise to stochastic noise) and in terms of convergence of function values in the expectation one can consider this noise $\delta$ to be stochastic unbiased. Moreover, if the maximal level of noise $\Delta = G$ is small enough it makes sense to split the complexities for two-point feedback and one-point stochastic unbiased feedback. This was done in [5, 3] without the acceleration techniques described in the previous sections.

All the results of the survey can be improved in terms of the dependence on $\epsilon$ by worsening the dependence on $d$. Namely, one can use centre of gravity methods with iteration complexity $O(d \ln \epsilon^{-1})$ [70] and estimate (sub-)gradient by using one-point feedback. For instance, the Vayda’s cutting plane method does not accumulate (sub-)gradient error over iterations [38]. For smooth problems with deterministic oracle, this is rather obvious. For example, in oracle complexity bound from Section 2 one can replace $d \epsilon^{-1/2}$ by $d \ln \epsilon^{-1}$. For non-smooth problems, such a change $d \epsilon^{-2} \rightarrow d \ln \epsilon^{-1}$ is not obvious [70], but still holds under proper correction of the approach [79]. For stochastic (one-point) feedback, the best known oracle complexity result is $O\left(d^3 \epsilon^{-2}\right)$ [70, 28] (see also conjecture $O\left(d^3 \epsilon^{-2}\right)$ about the optimal bound in [17]).

If $f$ has $L$-Lipschitz gradient in the 2-norm, then the Kernel-based approach from Section 3.3 allows one to improve oracle complexity to $d^2 \epsilon^{-2}$, when $n$ is large [5, 75, 4]. Note that the lower bound (for $n \geq 2$) is $[5]$

$$\min \left\{ \frac{d^3+1}{d \ln \delta}, \frac{d^2}{\epsilon^2} \right\}.$$ 

Hence, a significant gap still exists between lower bounds and current methods.

4.5 Online optimization

Nowadays, due to numerous applications in the reinforcement learning, online optimization [18, 47, 76] has become increasingly popular. Possibly, the best known online problem formulation with zeroth-order oracle is the multi-armed bandit problem [15, 89, 61], for which the optimal bound $d^2 \epsilon^{-2}$ for the regret is attained. For more general zeroth-order online convex optimization problems (on general convex sets, not necessarily on the unit simplex as for multi-armed bandits), the lower bound is $d^2 \epsilon^{-2}$. This bound was obtained in the class of linear functions [21]. The best known upper bound for general online convex optimization problem with one-point bandit feedback is $O\left(d^3 \epsilon^{-2}\right)$ [17].

For online setup, the iteration complexity typically coincides with the oracle complexity. Moreover, oracle complexity bounds for two-points online feedback are almost the same as for non-online case [35] (up to logarithmic factors). Note that in [35] the smoothness of $f$ was used to estimate the second moment of stochastic gradient. But, it can be easily replaced by [14] for non-smooth case. Also, note that the approach from Section 4.2 (see Theorem 6) can be applied to the online setting. This approach reduces $2 \sqrt{2} \min \{q, \ln d\} \rightarrow 288, q \geq 2$.

To conclude this section, it is important to note that any advances in (online/stochastic) convex optimization usually involve advances in gradient-free optimization. For example, based on [98] it seems to be possible to build optimal adaptive gradient-free methods for convex stochastic online optimization problem with heavy-tails.

5 Further research

In the previous sections, general convex optimization problem (1) are considered. However, many aforementioned results (tricks) can be naturally used for saddle-point problems, sum-type problems, distributed optimization.

5.1 Saddle-point problems

Consider non-smooth convex-concave saddle-point problems:

$$\min_{x \in Q, \xi \in \Delta^k} \max_{y \in Q, \epsilon \in \Delta^k} f(x, y).$$

(21)

Gradient-free methods for convex-concave saddle-point problems were studied in [13, 11, 37, 83, 24, 26, 83].

Applying $l_2$-smoothing technique separately to $x$-variables and $y$-variables, one can obtain almost the same results as for
optimization problems with the only difference in Theorem 1 instead of
\[ f(x) \leq f_j(x) \leq f(x) + \gamma M_2 \]
another inequality is considered:
\[ f(x, y) - \gamma, M_2 \gamma \leq f_j(x, y) \leq f(x, y) + \gamma, M_2 \gamma. \]
This leads to a clear counterpart of (10) for choosing \( \gamma = (\gamma, \gamma) \), where \( M_{2, \gamma}, M_{2, \gamma} \) corresponding Lipschitz constants in 2-norm.

If one takes as \( A(L, \sigma^2) \) the batched Mirror-Prox or the batched Operator extrapolation method or the batched Extra-gradient method \([53, 57, 39]\), using \([11, 12]\), one can obtain the following bounds
\[
N(L, \epsilon) = O \left( \frac{M_2^2}{\epsilon} \right),
\]
\[
T(L, \sigma^2, \epsilon) = O \left( \max \left\{ N(L, \epsilon), \frac{\sigma^2 R^2}{\epsilon^3} \right\} \right) = O \left( \frac{\min \{ \ln d, \ln d \}}{\epsilon^3} M_2^2 \right),
\]
where \( d = \max\{d_1, d_2\} \), \( M_2 = \max\{M_{2,1}, M_{2,2}\} \), \( R \) depends on the approximate optimality criteria. For example, if \( \epsilon \) is expected accuracy in terms of the duality gap \([53]\), then \( R \) is a diameter in the \( p \)-norm of \( Q \otimes Q \), up to a \( \sqrt{\ln d} \) factor, where \( \otimes \) is the Cartesian product of two sets. The last equality assumes that \( d \leq \left( M_2 / M_2 \right)^2 \) when \( p = 1 \). This result is also correct for stochastic saddle-point problems with proper redefinition of what is \( M_2 \), see Section 4.3.

Analogously, one can take \( A(L, \sigma^2) \) to be more specific batched algorithm that takes into account, for example, different constants of strong convexity and concavity \([14, 59, 22]\).

Overall, all of the above results (except Sections 3.2 and 4.5) can be transferred to saddle-point problems. In particular, for the smooth case, a variety of optimal (stochastic) gradient-based algorithms for (stochastic) saddle-point problems are collected in the survey \([12]\). All these algorithms could be extended to gradient-free oracle by using the tricks described above. But there are only few works (mentioned above) where such attempts have been made.

5.2 Sum-type problems

For non-smooth sum-type problems, the \( l_2 \)-smoothing scheme (see Section 4.1) can be applied, see \([54]\) for details. Moreover, all of the above results (except Section 4.5) can be transferred to sum-type problems. Sometimes it requires non-trivial combinations of variance-reduced technique and coordinate-wise randomization technique \([46]\), see also Section 5.1. But there is also a lack of papers in this field.

5.3 Distributed optimization

Distributed optimization (on time-varying networks) is currently a burgeoning area of research, see, e.g., \([43, 52]\).

In distributed optimization, there is one more criterion (number of communications) that typically reduces to the number of successive iterations considered before. So, for smooth problems, the described above results (except Sections 3.1, 3.2, 4.5) make it possible to build optimal gradient-free distributed methods based on gradient ones. Unfortunately, there are no references where this has been done.

For non-smooth stochastic problems, by using the Lan’s sliding \([59]\), in \([10]\) an optimal algorithm was proposed with two-point feedback and in \([92]\) the best-known algorithm was proposed with one-point feedback (see also \([6]\)). Note that the \( l_2 \)-smoothing technique from Section 4.1 gives non-optimal communication steps number, since the number of successive iterations is \( d^{1/4} \epsilon^{-1} \) while for optimal algorithms it should be \( \sim \epsilon^{-1} \). That is why it required to use another technique \([10, 92]\). For this direction one can also observe a lack of papers.

6 Conclusions

This survey discusses the state-of-the-art results and techniques in derivative-free convex optimization. The paper mainly focuses on the theoretical results about the convergence of the methods. In addition to the general convex minimization, the survey contains a discussion of the results for saddle-point problems, sum-type problems, and distributed optimization.

7 Cross-references

- Online Convex Optimization
- Decentralized Convex Optimization over Time-Varying Graphs
- Saddle Point Theory and Optimality Conditions
- Unified analysis of SGD-type methods

8 Acknowledgments

The authors are grateful to Oleg Granichin, Katya Scheinberg, Alexander Tsybakov, Ivan Oseledets and Wouter Jongeneel.

The work was supported by the Ministry of Science and Higher Education of the Russian Federation (Goszadaniye) 075-00337-20-03, project No. 0714-2020-0005.

The paper is dedicated to the memory of Yu. M. Ermoliev (1936 – 2022) – one of the founders of Soviet Stochastic Optimization school.

References

[1] A. Agarwal, O. Dekel, and L. Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *Colt*, pages 28–40. Citeseer, 2010.
[2] A. Agarwal, D. P. Foster, D. J. Hsu, S. M. Kakade, and A. Rakhlin. Stochastic convex optimization with bandit feedback. *Advances in Neural Information Processing Systems*, 24, 2011.
[3] A. Akhavan, E. Chzhen, M. Pontil, and A. B. Tsybakov. A gradient estimator via l1-randomization for online zero-order optimization with derivative-free optimization and continuous bandits. *Advances in Neural Information Processing Systems*, 33, 2020.
[4] A. Akhavan, E. Chzhen, M. Pontil, and A. B. Tsybakov. Gradient-free optimization of highly smooth functions: improved analysis and a new algorithm. *arXiv preprint arXiv:2306.02159*, 2023.
[5] A. Akhavan, M. Pontil, and A. Tsybakov. Exploiting higher order smoothness in derivative-free optimization and continuous bandits. *Advances in Neural Information Processing Systems*, 33:9017–9027, 2020.
[6] A. Akhavan, M. Pontil, and A. Tsybakov. Distributed zero-order optimization under adversarial noise. *Advances in Neural Information Processing Systems*, 34:10209–10220, 2021.
[7] F. Bach and V. Perchet. Highly-smooth zero-th order online optimization. In *Conference on Learning Theory*, pages 257–283, 2016.
