(Bandit) Convex Optimization with Biased Noisy Gradient Oracles

Xiaowei Hu[1] **Prashanth L.A.**[2] **András György**[3] **Csaba Szepesvári**[1]

[1] Department of Computing Science, University of Alberta
[2] Institute for Systems Research, University of Maryland
[3] Department of Electrical and Electronic Engineering, Imperial College London

Abstract

Algorithms for bandit convex optimization and online learning often rely on constructing noisy gradient estimates, which are then used in appropriately adjusted first-order algorithms, replacing actual gradients. Depending on the properties of the function to be optimized and the nature of "noise" in the bandit feedback, the bias and variance of gradient estimates exhibit various tradeoffs. In this paper we propose a novel framework that replaces the specific gradient estimation methods with an abstract oracle. With the help of the new framework we unify previous works, reproducing their results in a clean and concise fashion, while, perhaps more importantly, the framework also allows us to formally show that to achieve the optimal root-n rate either the algorithms that use existing gradient estimators, or the proof techniques used to analyze them have to go beyond what exists today.

1 Introduction

Convex optimization is widely studied under different models concerning what can be observed about the objective function. These models range from accessing the full gradient to only observing samples from the objective function (cf. [27, 13, 18, 29, 17, 1, 2, 28, 3, 22, 24, 33, 34, 14, 5, 15]). In this paper, we present and analyze a novel framework for convex optimization with biased gradient oracles, which encompasses most of what has been done in the bandit framework where the algorithms observe noisy point-evaluations of the objective function and use these to construct gradient estimators.

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 41.

In our framework, an optimization algorithm can query an oracle repeatedly to get a noisy and biased version of the gradient (or a subgradient for non-differentiable functions), where the algorithm querying the oracle also sets a parameter that controls the tradeoff between the bias and the variance of the gradient estimate.

Gradient oracles have been considered in the literature before: Several previous works assume that the accuracy requirements hold with probability one [10, 4, 13] or consider adversarial noise [31]. Gradient oracles with stochastic noise, which is central to our development, were also considered [21, 19, 16]; however, these papers assume that the bias and the variance are controlled separately, and consider the performance of special algorithms (in some cases in special setups).

The main feature of our model is that we allow stochastic noise, control of the bias and the variance, and we also consider lower bounds on the achievable error rates. Our gradient oracle model applies to several gradient estimation techniques extensively used in the literature, mostly for the case when the gradient is estimated only based on noisy observations of the objective function [22, 24, 33, 34, 14, 5, 15]. A particularly interesting application of our model is the widely studied bandit convex optimization problem, mentioned above, where most previous algorithms essentially use gradient estimates and first-order methods [29, 17, 1, 2, 28, 3, 18].

In this paper, we consider the optimization accuracy in both the stochastic and online bandit convex optimization (BCO) setting. We provide upper and lower bounds on the minimax optimization error (or regret) for several oracle models, which correspond to different ways of quantifying the bias-variance tradeoff of the gradient estimate. In particular, we provide matching upper and lower bounds for optimizing smooth, convex functions. We do not claim to invent methods for proving upper bounds, as the methods we use have been known previously for special cases (see the references above), but our main contribution lies in abstracting away the properties of gradient estimation procedures, thereby unifying previous analysis, providing a concise summary and an explanation of differences between previous works. More importantly, our framework also allows to prove lower bounds for any algorithm that relies on gradient estimation oracles of the type our framework captures (earlier work of Chen [9] considered a related lower bound on the converge of the iterate instead of the function value).

Note that our oracle model does not capture the full strength of the gradient estimates used in previous work, but it fully describes the properties of the estimates that *so far have been used in their analysis*. As a consequence, our lower bounds show that the known minimax regret of $\sqrt{T}$ [7, 8, 32] of online and stochastic bandit convex optimization cannot be shown to hold for any algorithm that uses current gradient estimation procedures, unless the proof exploited finer properties of the gradient estimators than used in prior works. In particular, our lower bounds even invalidate the claimed (weaker) upper bound of Dekel et al. [11].

The organization of the paper is as follows: We introduce the oracle model in Section 2, upper and lower bounds are provided in Section 3, with applications to online and stochastic BCO in Sections 4 and 5. All proofs can be found in the extended version of the paper [20].

2 Problem Setup

Notation: Capital letters will denote random variables. For $i \le j$ positive integers, we use the notation $a_{i:j}$ to denote the sequence $(a_i, a_{i+1}, \dots, a_j)$.

We let $\|\cdot\|$ denote some norm on $\mathbb{R}^d$, whose dual is denoted by $\|\cdot\|_*$. Let $\mathcal{K} \subset \mathbb{R}^d$ be a non-empty closed convex set. Given the function $f : \mathcal{K} \to \mathbb{R}$ which is differentiable [1] in $\mathcal{K}^\circ$ [2], f is said to be μ-strongly convex w.r.t. $\|\cdot\|$ ($\mu \ge 0$) if $\frac{\mu}{2}\|x-y\|^2 \le \mathcal{D}_f(x,y) \doteq f(x) - f(y) - \langle \nabla f(y), x - y \rangle$, for all $x \in \mathcal{K}, y \in \mathcal{K}^\circ$. Similarly, f is μ-strongly convex w.r.t. a *function* $\mathcal{R}$ if $\frac{\mu}{2} D_{\mathcal{R}}(x,y) \le \mathcal{D}_f(x,y)$ for all $x \in \mathcal{K}, y \in \mathcal{K}^\circ$, where $\mathcal{K}^\circ \subseteq \mathrm{dom}(\mathcal{R})$ and $\mathcal{R}$ is differentiable over $\mathcal{K}^\circ$. A function f is L-smooth w.r.t. $\|\cdot\|$ for some $L > 0$ if $D_f(x,y) \le \frac{L}{2}\|x-y\|^2$, for all $x \in \mathcal{K}, y \in \mathcal{K}^\circ$. This latter condition is equivalent to that ∇f is L-Lipschitz [27, Theorem 2.1.5]. We let $\mathcal{F}_{L,\mu,\mathcal{R}}(\mathcal{K})$ denote the class of functions that are μ-strongly convex w.r.t. $\mathcal{R}$ and L-smooth w.r.t. some norm $\|\cdot\|$ on the set $\mathcal{K}$. We also let $\mathcal{F}_{L,\mu}(\mathcal{K})$ be $\mathcal{F}_{L,\mu,\mathcal{R}}(\mathcal{K})$ with $\mathcal{R}(\cdot) = \frac{1}{2}\|\cdot\|_2^2$. Then, the set of convex and L-smooth functions is $\mathcal{F}_{L,0}(\mathcal{K})$.

[1] Here we assume the differentiablity for simplicity of definition, actually it is easy to generalize to non-differentiable functions by using sub-gradient.

[2] For $A \subset \mathbb{R}^d$, A° denotes the interior of A.

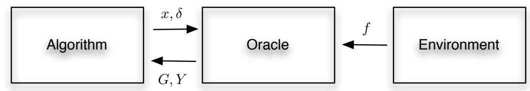

Figure 1: The Interaction of the algorithms with the gradient estimation oracle and the environment. For more information, see the text.

In this paper, we consider convex optimization in a novel setting, both for stochastic as well as online BCO. In the online BCO setting, the environment chooses a sequence of loss functions $f_1, \dots, f_n$ belonging to a set $\mathcal{F}$ of convex functions over a common, non-empty convex closed domain $\mathcal{K} \subset \mathbb{R}^d$. In the stochastic BCO setting, a single fixed loss function $f \in \mathcal{F}$ is chosen. An algorithm chooses a sequence of points $X_1, \dots, X_n \in \mathcal{K}$ in a serial fashion. The novelty of our setting is that the algorithm, upon selecting point X_t, receives a noisy and potentially biased estimate $G_t \in \mathbb{R}^d$ of the gradient of the loss function f (more generally, an estimate of a subgradient of f, in case f is not differentiable at X_t). To control the bias and the variance, the algorithm can choose a *tolerance parameter* $\delta_t > 0$ (in particular, we allow the algorithms to choose the tolerance parameter sequentially). A smaller δ_t results in a smaller "bias" (for the precise meaning of bias, we will consider two definitions below), while typically with a smaller δ_t, the "variance" of the gradient estimate increases. Notice that in the online BCO setting, the algorithm suffers the loss $f_t(Y_t)$ in round t, where $Y_t \in \mathcal{K}$[3] is guaranteed to be in the δ_t-vicinity of X_t. The goal in the online BCO setting is to keep the expected *regret*, $R_n = \mathbb{E}\left[\sum_{t=1}^n f_t(Y_t)\right] - \inf_{x \in \mathcal{K}} \sum_{t=1}^n f_t(x)$, small. In the stochastic BCO setting, the algorithm is also required to select a point $\hat{X}_n \in \mathcal{K}$ once the nth round is over (in both settings, n is given to the algorithms) and the algorithm's performance is quantified using the *optimization error*, $\Delta_n = \mathbb{E}\left[f(\hat{X}_n)\right] - \inf_{x \in \mathcal{K}} f(x)$.

The main novelty of the model is that the information flow between the algorithm and the environment (holding f, or $f_{1:n}$) is mediated by a gradient estimation oracle. As we shall see, numerous existing approaches to online learning and optimization based on noisy pointwise information fit in this framework.

We will use two classes of oracles. In both cases, the oracles are specified by two functions $c_1, c_2 : [0, \infty) \to [0, \infty)$, which will be assumed to be continuous, monotonously increasing (resp., decreasing) with $\lim_{\delta \to 0} c_1(\delta) = 0$ and $\lim_{\delta \to 0} c_2(\delta) = +\infty$. Typical choices for c_1, c_2 are $c_1(\delta) = C_1 \delta^p$, $c_2(\delta) = C_2 \delta^{-q}$ with $p, q > 0$. Our type-I oracles are defined as follows:

[3] For simplicity, in some cases we allow f to be defined outside of $\mathcal{K}$ and allow Y_t to be in a small vicinity of $\mathcal{K}$.

Definition 1 ((c_1, c_2) type-I oracle) We say that γ is a (c_1, c_2) type-I oracle for $\mathcal{F}$, if for any function $f \in \mathcal{F}$, $x \in \mathcal{K}, 0 < \delta \leq 1$, γ returns $G \in \mathbb{R}^d$ and $Y \in \mathcal{K}$ random elements such that $\|x - Y\| \leq \delta$ almost surely (a.s.),

1. $\|\mathbb{E}[G] - \nabla f(x)\|_* \leq c_1(\delta)$ (bias); and
2. $\mathbb{E}\left[\|G - \mathbb{E}[G]\|_*^2\right] \leq c_2(\delta)$. □

The upper bound on δ is arbitrary: by changing the norm, any other value can also be accommodated. Also, the upper bound only matters when $\mathcal{K}$ is bounded and the functions in $\mathcal{F}$ are defined only in a small vicinity of $\mathcal{K}$.

The second type of oracles considered is as follows:

Definition 2 ((c_1, c_2) type-II oracle) We say that γ is a (c_1, c_2) type-II oracle for $\mathcal{F}$, if for any function $f \in \mathcal{F}$, $x \in \mathcal{K}, 0 < \delta \leq 1$, γ returns $G \in \mathbb{R}^d$ and $Y \in \mathcal{K}$ random elements such that $\|x - Y\| \leq \delta$ a.s.,

1. There exists $\tilde{f} \in \mathcal{F}$ such that $\|\tilde{f} - f\|_\infty \leq c_1(\delta)$ and $\mathbb{E}[G] = \nabla \tilde{f}(x)$ (bias); and
2. $\mathbb{E}\left[\|G - \mathbb{E}[G]\|_*^2\right] \leq c_2(\delta)$ (variance). □

Note that our definition allows the same oracle γ to respond to the same inputs (x, δ, f) with a differently constructed pair (e.g., to have memory), though most often the oracles constructed in practice will map the triples (x, δ, f) to a gradient estimate-point pair using a fixed stochastic mapping.[4] Examples will be discussed in Section 4. We will denote the set of type-I (type-II) oracles satisfying the (c_1, c_2)-requirements given a function $f \in \mathcal{F}$ by $\Gamma_1(f, c_1, c_2)$ (resp., $\Gamma_2(f, c_1, c_2)$).

Here, we will study the minimax regret in the online convex optimization setting, while we study the minimax error in the stochastic convex optimization setting (sometimes, also called as the "simple regret"). Both are defined with respect to a class of loss functions $\mathcal{F}$, and the bias/variance control functions c_1, c_2. The *minimax expected regret* for $(\mathcal{F}, c_1, c_2)$ with type-I oracles is

$$R^*_{\mathcal{F},n}(c_1, c_2) = \inf_{\mathcal{A}} \sup_{f_{1:n} \in \mathcal{F}^n} \sup_{\substack{\gamma_t \in \Gamma_1(f_t, c_1, c_2) \\ 1 \leq t \leq n}} R_n^{\mathcal{A}},$$

where $\mathcal{A}$ ranges through all algorithms that interact with the loss sequence $f_{1:n} = (f_1, \ldots, f_n)$ through the oracles $\gamma_{1:n}$ (in round t, oracle γ_t is used), and we use $R_n^{\mathcal{A}}$ to denote the expected regret of $\mathcal{A}$ (against $f_{1:n}, \gamma_{1:n}$). The minimax regret for type-II oracles is defined analogously.

[4]For oracles with memory, in the definition the expectation should be replaced with an expectation that is conditioned on the past.

In the stochastic BCO setting, the minimax error is defined through

$$\Delta^*_{\mathcal{F},n}(c_1, c_2) = \inf_{\mathcal{A}} \sup_{f \in \mathcal{F}} \sup_{\gamma \in \Gamma_1(f, c_1, c_2)} \Delta_n^{\mathcal{A}}(f, \gamma), \quad (1)$$

where, again, $\mathcal{A}$ ranges through all algorithms that interact with an oracle and $\Delta_n^{\mathcal{A}}(f, \gamma)$ is the optimization error that $\mathcal{A}$ suffers after n rounds of interaction with f through (a single) γ as described earlier. The minimax error for type-II oracles is defined analogously.

When the set $\mathcal{K}$ is bounded and the function set $\mathcal{F}$ is invariant to linear shifts, every (c_1, c_2) type-I oracle is also an (Rc_1, c_2) type-II oracle, where $R = \sup_{x \in \mathcal{K}} \|x\|$: simply consider $\tilde{f}(y) = f(y) + (\mathbb{E}[G] - \nabla f(x))^\top y$, where G is the gradient estimate returned by the oracle. Thus, if for some set $\Delta_n^{\text{type-I}}$, $\Delta_n^{\text{type-II}}$ denote the appropriate minimax errors and $R = 1$ then $\Delta^{\text{type-I}}_{\mathcal{F},n}(c_1, c_2) \leq \Delta^{\text{type-II}}_{\mathcal{F},n}(c_1, c_2)$. As a result, when proving lower bounds, we shall consider type-I, while when proving upper bounds we will consider type-II oracles. Also note that for either type of oracles, $\Delta^*_{\mathcal{F},n}(c_1, c_2) \leq R^*_{\mathcal{F},n}(c_1, c_2)/n$. This follows by the well known construction that turns an online convex optimization method $\mathcal{A}$ for regret minimization into an optimization method by running the method and at the end choosing as $\hat{X}_n$ the average of the points $X_1, \ldots, X_n$ queried by $\mathcal{A}$ during the n rounds. Indeed, then $f(\hat{X}_n) \leq \frac{1}{n}\sum_{t=1}^n f(X_t)$ by Jensen's inequality, hence the average regret of $\mathcal{A}$ will upper bound the error of choosing $\hat{X}_n$ at the end. A consequence of this relation is that a lower bound for $\Delta^*_{\mathcal{F},n}(c_1, c_2)$ will also be a lower bound for $R^*_{\mathcal{F},n}(c_1, c_2)/n$ and an upper bound on $R^*_{\mathcal{F},n}(c_1, c_2)$ leads to an upper bound on $\Delta^*_{\mathcal{F},n}(c_1, c_2)$. This explains why we allowed taking supremum over time-varying oracles in the definition of the regret and why we used a static oracle for the optimization error: to maximize the strength of the bounds we obtain.

3 Main Results

In this section we provide our main results in forms of upper and lower bounds on the minimax error. First we give an upper bound by analyzing a mirror-descent algorithm given in Algorithm 1. In the algorithm, we assume that the regularizer function $\mathcal{R}$ is α-strongly convex and the target function f is smooth or smooth and strongly convex. A standard analysis yields the following upper bounds:

Theorem 1 (*Upper bound*) *Consider the class $\mathcal{F}_{L,0}$ of convex, L-smooth functions on the bounded, convex domain $\mathcal{K} \neq \emptyset$, $\mathcal{K} \subset \mathbb{R}^d$. Assume that the regularization function $\mathcal{R}$ is α-strongly convex w.r.t. some norm $\|\cdot\|$, and $\mathcal{K}^\circ \subseteq \mathrm{dom}(\mathcal{R})$. For any (c_1, c_2) type-II*

Oracle type	Convex + Smooth		Strongly Convex + Smooth	
	Upper bound	Lower bound	Upper bound	Lower bound
δ-bias, δ^{-2}-variance $(p=1, q=2)$	$\left(\frac{C_1^2 C_2 D}{n}\right)^{1/4}$	$\left(\frac{C_1^2 C_2 d^2}{n}\right)^{1/4}$	$\left(\frac{C_1^2 C_2}{n}\right)^{1/3}$	$\left(\frac{C_1^2 C_2}{n}\right)^{1/2}$
δ^2-bias, δ^{-2}-variance $(p=2, q=2)$	$\left(\frac{C_1 C_2 D}{n}\right)^{1/3}$	$\left(\frac{C_1 C_2 \sqrt{d^3}}{n}\right)^{1/3}$	$\left(\frac{C_1 C_2}{n}\right)^{1/2}$	$\left(\frac{C_1 C_2}{n}\right)^{2/3}$

Table 1: Summary of upper and lower bounds for different smooth function classes and gradient oracles for the settings of Theorem 1 and Theorem 2. Note that when $\mathcal{R}$ is the squared norm and $\mathcal{K}$ is the hypercube (as in the lower bounds), $D = \theta(d)$ in the upper bounds and also that C_1, C_2 may hide dimension-dependent quantities for the common gradient estimators, as will be discussed later.

Algorithm 1 Mirror Descent with Type-I/II Oracle.

Input: Closed convex set $\mathcal{K} \neq \emptyset$, regularization function $\mathcal{R} : \mathrm{dom}(\mathcal{R}) \to \mathbb{R}$, $\mathcal{K}^\circ \subset \mathrm{dom}(\mathcal{R})$, tolerance parameter δ, learning rates $\{\eta_t\}_{t=1}^{n-1}$.
Initialize $X_1 \in \mathcal{K}$ arbitrarily.
for $t = 1, 2, \cdots, n-1$ **do**
  Query the oracle at X_t to receive G_t, Y_t.
  Set $X_{t+1} = \mathrm{argmin}_{x \in \mathcal{K}} \left[\eta_t \langle G_t, x\rangle + D_{\mathcal{R}}(x, X_t)\right]$.
Return: $\hat{X}_n = \frac{1}{n}\sum_{t=1}^n X_t$.

oracle with $c_1(\delta) = C_1\delta^p$, $c_2(\delta) = C_2\delta^{-q}$, $p, q > 0$, if Algorithm 1 is run with $\eta_t = \alpha/(a_t + L)$ and $\delta = \left(\frac{C_2}{4aC_1}\frac{2p+q}{p}\frac{n}{n+1}\right)^{\frac{1}{p+q}} n^{-\frac{1}{2p+q}}$, where $a^{2p+q} = 2^{q-p}\left(2+\frac{q}{p}\right)^p \left(1+\frac{1}{n}\right)^q \left(\frac{\alpha}{D}\right)^{p+q} C_1^q C_2^p$, $a_t = at^{\frac{p+q}{2p+q}}$, for $t = 1, 2, \cdots, n-1$, then the regret and the minimax error can be bounded as

$$\Delta^*_{\mathcal{F}_{L,0},n}(c_1, c_2) \leq \frac{R^*_{\mathcal{F}_{L,0},n}(c_1,c_2)}{n} = O\left(\left(\frac{DC_1^{\frac{q}{p}} C_2}{n}\right)^{\frac{p}{2p+q}}\right),$$

where $D = \sup_{x,y\in\mathcal{K}} D_{\mathcal{R}}(x, y)$.

For the class $\mathcal{F} = \mathcal{F}_{L,\mu,\mathcal{R}}$ of μ-strongly convex (w.r.t. $\mathcal{R}$) and L-smooth functions, with $\alpha > 2L/\mu$, $\eta_t = 2/(\mu t)$, and $\delta^{p+q} = \frac{C_2\left(\log n + 1 + \frac{\alpha\mu}{\alpha\mu - 2L}\right)}{2\alpha\mu C_1(n+1)}$, the regret and the minimax error satisfy

$$\Delta^*_{\mathcal{F}_{L,\mu,\mathcal{R}},n}(c_1, c_2) \leq \frac{R^*_{\mathcal{F}_{L,\mu,\mathcal{R}},n}(c_1,c_2)}{n} = \tilde{O}\left(\left(\frac{C_1^{\frac{q}{p}} C_2}{n}\right)^{\frac{p}{p+q}}\right).$$

In the bounds $O(\cdot)$ hides a constant that is a function of p, q, α, L and μ.[5] □

[5]In the second bound, $\tilde{O}(f(n))$ denotes $O(\log(n) f(n))$.

We next state lower bounds for both convex as well as strongly convex function classes. In particular, we observe that for convex and smooth functions the upper bound for the mirror descent scheme matches the lower bound, up to constants, whereas there is a gap for strongly convex+smooth functions. Filling the gap is left for future work.

Theorem 2 (***Lower bound***) *Let $n > 0$ be an integer, $p, q > 0$, $C_1, C_2 > 0$, $\mathcal{K} \subset \mathbb{R}^d$ convex, closed, with $[+1, -1]^d \subset \mathcal{K}$. Then, for any algorithm that observes n random elements from a (c_1, c_2) type-I oracle with $c_1(\delta) = C_1\delta^p$, $c_2(\delta) = C_2\delta^{-q}$, the minimax error (and hence the regret) satisfies the following bounds:*
$\mathcal{F}_{L,0}(\mathcal{K})$ (Convex and smooth) with $L \geq \frac{1}{2}$:

$$\Delta^*_{\mathcal{F}_{L,0},n}(c_1, c_2) \geq K_1\sqrt{d}\left(\frac{C_1^{\frac{q}{p}} C_2}{n}\right)^{\frac{p}{2p+q}},$$

where $K_1 = \frac{(2p+q)^2}{2q^{\frac{q}{2p+q}}(4p+q)^{\frac{4p+q}{2p+q}}}$.
$\mathcal{F}_{L,1}(\mathcal{K})$ (1-strongly convex and smooth) with $L \geq 1$:

$$\Delta^*_{\mathcal{F}_{L,1},n}(c_1, c_2) \geq K_2\left(\frac{C_1^{\frac{q}{p}} C_2}{n}\right)^{\frac{p}{p+q/2}},$$

where $K_2 = 2^{\frac{2p-q}{2p+q}} \frac{(2p+q)^3}{q^{\frac{2q}{2p+q}}(6p+q)^{\frac{6p+q}{2p+q}}}$. □

By continuity, the above claim can be extended to cover the case of $q = 0$ (constant variance). For the special case of $p = 0$ and $C_1 > 0$, which implies a constant bias, it is possible to derive an $\Omega(1)$ lower bound by tweaking the proof. On the other hand, the case of $p = 0$ and $C_1 = 0$ (no bias) leads to an $\Omega(d/\sqrt{n})$ lower bound. The proof of the lower bound is obtained in the usual way by presenting a family of functions and a type-I oracle such that any algorithm suffers at least the stated error on one of the functions. In particular, for $\mathcal{F}_{L,0}$ with $L \geq 1/2$

we use $f_{v,\epsilon}(x) = \epsilon\,(x-v) + 2\epsilon^2 \ln\left(1 + e^{-\frac{x-v}{\epsilon}}\right)$ with $v = \pm 1$, $\epsilon > 0$, and $x \in \mathcal{K} \subset \mathbb{R}$ for appropriate ϵ. Note that for *any* $\epsilon > 0$, $f_{v,\epsilon} \in \mathcal{F}_{1/2,0} \setminus \cup_{0<\lambda<1/2} \mathcal{F}_{\lambda,0}$.

Remark 1 (***Scaling***) For any function class $\mathcal{F}$, by the definition of the minimax error (1), it is easy to see that

$$\Delta_n^*(\mu\mathcal{F}, c_1, c_2) = \mu\Delta_n^*\left(\mathcal{F}, c_1/\mu, c_2/\mu^2\right),$$

where μF denotes the function class comprised of functions in $\mathcal{F}$, each scaled by $\mu > 0$. In particular, this relation implies that the bound for μ-strongly convex function class is only a constant factor away from the bound for 1-strongly convex function class. □

Table 2 presents the upper and lower bounds for two specific choices of p and q (relevant in applications, as we shall see later). These bounds can be inferred from the results in Theorems 1 and 2. Some specific examples will be discussed in the next section.

4 Applications to Stochastic BCO

The main application of the biased noisy gradient oracle based convex optimization of the previous section is bandit convex optimization. We introduce here briefly the stochastic version of the problem, while online BCO will be considered in Section 5. Readers familiar with these problems and the associated gradient estimation techniques, may skip this description to jump directly to Theorem 4, and come back here only if clarifications are needed.

In the *stochastic BCO* setting, the algorithm sequentially chooses the points $X_1, \ldots, X_n \in \mathcal{K}$ while observing the loss function at these points in noise. In particular, in round t, the algorithm chooses X_t based on the earlier observations $Z_1, \ldots, Z_{t-1} \in \mathbb{R}$ and $X_1, \ldots, X_{t-1}$, after which it observes Z_t, where Z_t is the value of $f(X_t)$ corrupted by "noise".

Previous research considered several possible constraints connecting Z_t and $f(X_t)$. One simple assumption is that $\{Z_t - f(X_t)\}_t$ is an $\{\mathcal{F}_t\}_t = \{\sigma(X_{1:t}, Z_{1:t-1})\}_t$-adapted martingale difference sequence (with favorable tail properties). A specific case is when $Z_t - f(X_t) = \xi_t$, where (ξ_t) is a sequence of independent and identically distributed (i.i.d.) variables. A stronger assumption, common in stochastic programming, is that

$$Z_t = F(X_t, \Psi_t), \quad f(x) = \int F(x,\psi) P_\Psi(d\psi), \quad (2)$$

where $\Psi_t \in \mathbb{R}$ is chosen by the algorithm and in particular the algorithm can draw Ψ_t at random from P_Ψ. As in [15], we assume that the function $F(\cdot, \psi)$ is L_ψ-smooth P_Ψ-a.s. and the quantity $\overline{L}_\Psi = \sqrt{\mathbb{E}[L_\Psi^2]}$ is finite. Note that the algorithm is aware of P_Ψ, but does not know how different values of ψ affect the noise $\xi(x,\psi) = F(x,\psi) - f(x)$. Nevertheless, as the algorithm can control ψ and thus ξ, we refer to this as *controlled noise* setting and to the others as the case of *uncontrolled noise*. As we will see, and is well known in the simulation optimization literature [23, 15], this extra structure allows the algorithm to reduce the variance of the noise of its gradient estimates by reusing the same Ψ_t in consecutive measurements, while measuring the gradient at the same point, an instance of the method of the method of common random variables. As creating an estimate from K points (which is equivalent to the so-called "multi-point feedback setup" from the literature where K points are queried in each round) changes the number of rounds from n to n/K, which does not change the convergence rate as long as K is fixed.

4.1 Estimating the Gradient

A common popular idea in bandit convex optimization is to use the bandit feedback to construct noisy (and biased) estimates of the gradient. In the following, we provide a few examples for oracles that construct gradient estimates for function classes that are increasingly general: from smooth, convex to non-differentiable functions.

One-point feedback Given $x \in \mathcal{K}$, $0 < \delta \leq 1$, common gradient estimates that are based on a single query to the function evaluation oracle (the so-called "one-point feedback") take the form

$$G = \frac{Z}{\delta} V, \text{ where } Z = f(x + \delta U) + \xi, \quad (3)$$

where $(U, V, \xi) \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}$ are jointly distributed random variables, ξ is the function evaluation noise whose distribution may depend on $x + \delta U$ but $\mathbb{E}[\xi|V] = 0$, and G is the estimate of $\nabla f(x)$ ($f : \mathcal{K} \to \mathbb{R}$).

In all oracle constructions we will use the following assumption:

Assumption 1 *Let $\mathcal{K} \subset \mathcal{D}^\circ \subset \mathbb{R}^d$, where $f : \mathcal{D} \to \mathbb{R}$. For any $x \in \mathcal{K}$, $x + \delta U \in \mathcal{D}$ a.s., and* $\mathbb{E}\left[\|V\|_*^2\right]$, $\mathbb{E}\left[\|U\|^3\right] < +\infty$.

Note that here the function domain $\mathcal{D}$ can be larger than or equal to the set $\mathcal{K}$, where the algorithm chooses x. This is to ensure that the oracle will not receive invalid inputs, that is, queries where f is not defined. When the functions are defined over $\mathcal{K}$ only and $\mathcal{K}$ is bounded, the above constructions only work for δ small enough. In this case, the best approach perhaps is to use Dikin ellipsoids to construct the oracles, as done by Hazan and Levy [18].

The next proposition, whose proof is based on ideas from Spall [33] shows that the above one-point gradient estimator leads to a type-I (and, hence, also type-II) oracle.

Proposition 1 *Let Assumption 1 hold and let γ be the one-point feedback oracle defined in (3). Assume further that U is symmetrically distributed, $V = h(U)$, where $h : \mathbb{R}^d \to \mathbb{R}^d$ is an odd function, $\mathbb{E}[V] = 0$, and $\mathbb{E}[VU^\intercal] = I$. Then, in the uncontrolled noise case, γ is a $(c_1(\delta), c_2(\delta))$ type-I oracle given in Table 2, where $C_2 = 4\mathbb{E}\left[\|V\|_*^2\right]\left(\operatorname{ess\,sup}\mathbb{E}[\xi^2|V] + \sup_{x\in\mathcal{D}} f^2(x)\right)$, and $C_1 = \frac{L}{2}\mathbb{E}[\|V\|_*\|U\|^2]$ when $f \in \mathcal{F}_{L,0}$ and $C_1 = \frac{B_3}{6}\mathbb{E}\left[\|V\|_*\|U\|^3\right]$ for $f \in \mathcal{C}^3$ where $B_3 = \sup_{x\in D}\|\nabla^3 f(x)\|_T$ where $\|\cdot\|_T$ denotes the implied norm for rank-3 tensors.*

Another possibility is to use the so-called smoothing technique [29, 17, 18] to obtain type-II oracles. Following the analysis in Flaxman et al. [17], one gets the following result, which improves the bias of the previous result from $O(\delta)$ to $O(\delta^2)$ in the smooth+convex case:

Proposition 2 *Let Assumption 1 hold and let γ be the one-point feedback oracle defined in (3). Define $V = n_W(U)\frac{|\partial W|}{|W|}$, where $W \subset \mathbb{R}^n$ is a convex body with boundary ∂W, U is uniformly distributed on ∂W, $n_W(U)$ denotes the normal vector of ∂W at U, and $|\cdot|$ denotes the appropriate volume. Let $C_2 > 0$ be defined as in Proposition 1. Then, if f is L_0-Lipschitz, γ is a type-II oracle with $c_1(\delta) = C_1\delta$, $c_2(\delta) = C_2/\delta^2$ where $C_1 = L_0\sup_{w\in W}\|w\|$. Further, assuming W is symmetric w.r.t. the origin, if f is L-smooth, then γ is a type-I (and type-II oracle) with $c_1(\delta) = C_1\delta^2$, $c_2(\delta) = C_2/\delta^2$ where $C_1 = (L/|W|)\int_W \|w\|^2 dw$, and, if in addition f is also convex (i.e., $f \in \mathcal{F}_{L,0}$) then γ is a type-I oracle with $c_1(\delta) = C_1\delta^2/2$ and $c_2(\delta) = C_2/\delta^2$.* □

Note that the improvement did not even require convexity. Also, the bias is smaller for smoother functions, a property that will be enjoyed by all the gradient estimators.

Two-point feedback While the one-point estimators are intriguing, in the optimization setting one can also always group two consecutive observations and obtain similar smoothing-type estimates at the price of reducing the number of rounds by a factor of two only, which does not change the rate of convergence. Next we present an oracle that uses two function evaluations to obtain a gradient estimate. As will be discussed later, this oracle encompasses several simultaneous perturbation methods (see 5): Given the inputs $x \in \mathcal{K}$, $0 < \delta \le 1$, the gradient estimate is

$$G = \frac{Z^+ - Z^-}{2\delta} V\,, \tag{4}$$

Noise → Function ↓	**Controlled** (see (2))	**Uncontrolled** (see (5))
Convex + Smooth	$(C_1\delta, C_2)$	Props 1,3: $(C_1\delta, \frac{C_2}{\delta^2})$ Prop 2: $(C_1'\delta^2, \frac{C_2}{\delta^2})$
$\boldsymbol{f \in \mathcal{C}^3}$	$(C_1\delta^2, \frac{C_2}{\delta^2})$	Props 1,3: $(C_1\delta^2, \frac{C_2}{\delta^2})$

Table 2: Gradient oracles for different function classes and noise categories. The constants C_1, C_1', C_2 are defined in Propositions 1–3.

where $Z^{\pm} = f(X^{\pm}) + \xi^{\pm}$, $X^{\pm} = x \pm \delta U$, $U, V \in \mathbb{R}^d$, $\xi^{\pm} \in \mathbb{R}$ are random, jointly distributed random variables, U, V chosen by the oracle in the uncontrolled case and chosen by the algorithm in the controlled case from some fixed distribution characterizing the oracle, and $\xi^{\pm}$ being the noise of the returned feedback $Z^{\pm}$ at points $X^{\pm}$. For the following proposition we consider $4 = 2 \times 2$ cases. First, the function is either assumed to be L-smooth and convex (i.e., the derivative of f is L-Lipschitz w.r.t. $\|\cdot\|_*$), or it is assumed to be three times continuously differentiable ($f \in C^3$). The other two options are either the controlled noise setting of (2), or, in the uncontrolled case, we make the alternate assumptions

$$\begin{aligned}\mathbb{E}[\xi^+ - \xi^- \,|\, U, V] &= 0 \quad \text{and} \\ \mathbb{E}[(\xi^+ - \xi^-)^2 \,|\, V] &\le \sigma_\xi^2 < \infty\,.\end{aligned} \tag{5}$$

The following proposition, whose proof is based on [33, Lemma 1] and [15, Lemma 1], provides conditions under which the bias-variance parameters (c_1, c_2) can be bounded as shown in Table 2:

Proposition 3 *Let Assumption 1 hold and let γ be a two-point feedback oracle defined by (4). Suppose furthermore that $\mathbb{E}[VU^\intercal] = I$. Then γ is a type-I oracle with the pair $(c_1(\delta), c_2(\delta))$ given by Table 2. For uncontrolled noise and for controlled noise with $f \in \mathcal{C}^3$, C_1 is as in Proposition 1 and C_2 is $4C_2$ from Proposition 1. For the controlled noise case with $f \in \mathcal{F}_{L,0}$, $C_1 = \frac{L_\Psi}{2}\mathbb{E}[\|V\|_*\|U\|^2]$ and $C_2 = 2B_1^2 + \frac{L_\Psi^2}{2}\mathbb{E}\left[\|V\|_*^2\|U\|^4\right]$, with $B_1 = \sup_{x\in\mathcal{K}}\|\nabla f(x)\|_*$.* □

Popular choices for U and V:

• If we set U_i to be independent, symmetric ± 1-valued random variables and $V_i = 1/U_i$, then we recover the popular SPSA scheme proposed by Spall [33]. It is easy to see that $\mathbb{E}[VU^\intercal] = I$ holds in this case. When the norm $\|\cdot\|$ is the 2-norm, $C_1 = O(d^2)$ and $C_2 = O(d)$. If we set $\|\cdot\|$ to be the max-norm, $C_1 = O(\sqrt{d})$ and $C_2 = O(d)$.

• If we set $V = U$ with U chosen uniform at random on the surface of a sphere with radius $\sqrt{d}$, then we recover the RDSA scheme proposed by Kushner and Clark

[24, pp. 58–60]. In particular, the (U_i) are identically distributed with $\mathbb{E}[U_i U_j] = 0$ if $i \neq j$ and $\mathbb{E}[U^\intercal U] = d$, hence $\mathbb{E}\left[U_i^2\right] = 1$. Thus, if we choose $\|\cdot\|$ to be the 2-norm, $C_1 = O(d^2)$ and $C_2 = O(d)$.

• If we set $V = U$ with U the standard d-dimensional Gaussian with unit covariance matrix, we recover the smoothed functional (SF) scheme proposed by Katkovnik and Kulchitsky [22]. Indeed, in this case, by definition, $\mathbb{E}[VU^\intercal] = \mathbb{E}[UU^\intercal] = I$. When $\|\cdot\|$ is the 2-norm, $C_1 = O(d^2)$ and $C_2 = O(d)$. This scheme can also be interpreted as a smoothing operation that convolves the gradient of f with a Gaussian density.

4.2 Achievable Results for Stochastic BCO

We now consider stochastic BCO with L-smooth functions over a convex, closed non-empty domain $\mathcal{K}$. Let $\mathcal{F}$ denote the set of these functions. Duchi et al. [15] proves that the minimax expected optimization error for the functions $\mathcal{F}$ with uncontrolled noise is lower bounded by $\Omega(n^{-1/2})$. They also give an algorithm which uses two-point gradient estimates which matches this lower bound for the case of *controlled noise*. For controlled noise, the constructions in the previous section give that for two-point estimators $c_1(\delta) = C_1\delta^p$ and $c_2(\delta) = C_2\delta^{-q}$ with $p = 1$ and $q = 0$. Plugging this into Theorem 1 we get the rate $O(n^{-1/2})$ (which is unsurprising given that the algorithms and the upper bound proof techniques are essentially the same as that of Duchi et al. [15]). However, when the noise is uncontrolled, the best that we get is $p = 2$ and $q = 2$. From Theorem 2 we get that with such oracles, no algorithm can get better rate than $\Omega(n^{-1/3})$, while from Theorem 1 we get that these rates are matched by mirror descent. We can summarize these findings as follows:

Theorem 3 *Consider $\mathcal{F}_{L,0}$, the space of convex, L-smooth functions over a convex, closed non-empty domain $\mathcal{K}$. Then, we have the following:*
Uncontrolled noise*: Take any (δ^2, δ^{-2}) type-I oracle γ. There exists an algorithm that uses γ and achieves the rate $O(n^{-1/3})$. Furthermore, no algorithm using γ can achieve better error than $\Omega(n^{-1/3})$ for every (δ^2, δ^{-2}) type-I oracle γ.*
Controlled noise*: Take any $(\delta, 1)$ type-I oracle γ. There exists an algorithm that uses γ an achieves the rate $O(n^{-1/2})$. Furthermore, no algorithm using γ can achieve better error than $\Omega(n^{-1/2})$ for every $(\delta, 1)$ type-I oracle γ.* □

For stochastic BCO with uncontrolled noise, Agarwal et al. [3] analyze a variant of the well-known ellipsoid method and provide regret bounds for the case of convex, 1-Lipschitz functions over the unit ball. Their regret bound implies a minimax error (1) bound of order $O\left(\sqrt{d^{32}/n}\right)$. Liang et al. [25] provide an algorithm based on random walks (and not using gradient estimates) for the setting of convex, bounded functions whose domain is contained in the unit cube and their algorithm results in a bound of the order $\mathcal{O}\left((d^{14}/n)^{1/2}\right)$ for the minimax error. These bounds decrease faster in n than the bound available in Theorem 3, while showing a much worse dependence on the dimension. However, what is more interesting is that our results also shows that an $O(n^{-1/2})$ upper bound *cannot* be achieved solely based on the oracle properties of the gradient estimates considered. Since the analysis of all gradient algorithms for stochastic BCO does this, it is no wonder that the best known upper bound for convex+smooth functions is $O(n^{-1/3})$ [30]. (We will comment on the recent paper of Dekel et al. [11] later.)

The above result also shows that the gradient oracle based algorithms are optimal for smooth problems, under a controlled noise setting. While Duchi et al. [15] suggests that it is the power of two-point gradient estimators that helps to achieve this, we need to add that having controlled noise is also critical.

Finally, let us make some remarks on the early literature on this problem. A finite time lower bound for stochastic, smooth BCO is presented by Chen [9] for convex functions on the real line. When applied to our setting in the uncontrolled noise case, his results imply that $\mathbb{E}\left[|\hat{X}_n - x^*|\right]$, that is, the distance of the estimate to the optimum, is at least $\Omega(n^{-1/3})$. Note that this is larger than the error achieved by the algorithms of Liang et al. [25], Bubeck et al. [7], Bubeck and Eldan [8], but the apparent contradiction is easily resolved by noticing the difference in their error measure: distance to the optimum vs. error in the function value (in particular, compressing the range of functions makes locating the minimizer harder). Polyak and Tsybakov [29], who also considered distance to optimum, proved that mirror descent with gradient estimation achieves asymptotically optimal rates for functions that enjoy high order smoothness.

5 Applications to Online BCO

In the *online BCO* setting a learner sequentially chooses the points $X_1, \ldots, X_n \in \mathcal{K}$ while observing the losses $f_1(X_1), \ldots, f_n(X_n)$. More specifically, in round t, having observed $f_1(X_1), \ldots, f_{t-1}(X_{t-1})$ of the previous rounds, the learner chooses $X_t \in \mathcal{K}$, after which it observes $f_t(X_t)$. The learner's goal is to minimize its expected regret $\mathbb{E}\left[\sum_{t=1}^n f_t(X_t) - \inf_{x\in\mathcal{K}} \sum_{t=1}^n f_t(x)\right]$. This problem is also called online convex optimization with one-point feedback. A slightly different problem is obtained if we allow the learner to choose multiple points in every round, at which points the function f_t is observed. The loss is suffered at X_t. The points where the function is observed ("observation points" for short) may or may not be tied to X_t. One possibility is that X_t is one of the observation points. Another possibility is that X_t is the average of the observation points (e.g., Agarwal et al. [2]). Yet another possibility is that there is no relationship between them.

The oracle constructions from the previous section also apply to the online BCO setting where the algorithm is evaluated at Y_t, though in this case one cannot employ two-point feedback as the functions change between rounds. This also rules out the controlled noise case. Thus, for the online BCO setting, one should consider type-I (and II) oracles with $c_1(\delta) = C_1\delta^p$ and $c_2(\delta) = C_2\delta^{-q}$ with $p = q = 2$. For these type of oracles, the results from Theorem 2 give the following result:

Theorem 4 *Let $\mathcal{F}_{L,0}$ be the space of convex, L-smooth functions over a convex non-empty domain $\mathcal{K}$. No algorithm that relies on (δ^2, δ^{-2}) type-I oracles can achieve better regret than $\Omega(n^{2/3})$.* □

With a noisy gradient oracle of Proposition 2, Theorem 4 implies that this regret rate is achievable, essentially recovering, and in some sense proving optimality of the result of Saha and Tewari [30]:

Theorem 5 *For zeroth order noisy optimization with smooth convex functions, the gradient estimator of Proposition 2 together with mirror descent (see Algorithm 1) achieve $\mathcal{O}(n^{2/3})$ regret.* □

This optimality result shows that with the usual analysis of the current gradient estimation techniques, no gradient method can achieve the optimal regret $O(n^{1/2})$ for online bandit convex optimization, established by Bubeck et al. [7], Bubeck and Eldan [8]. Note that this shows a contradiction to the recent result of Dekel et al. [11], who claimed to achieve $\tilde{O}(n^{5/8})$ regret with the same (δ^2, δ^{-2}) type-II gradient oracle as Saha and Tewari [30], but their proof only used the (δ^2, δ^{-2}) tradeoff in the bias and variance properties of the oracle.

6 Related Work

Gradient oracle models have been studied in a number of previous papers [10, 4, 31, 13]. A full comparison between these oracle models is given by Devolder et al. [13]. For illustration, here we only review the model of this latter paper as a typical example of these previous works. The model of Devolder et al. [13] assumes a first-order approximation to the function with parameters (δ, L). In particular, given (x, δ, L) and the convex function f, the oracle gives a pair $(t, g) \in \mathbb{R} \times \mathbb{R}^d$ such that $t + \langle g, \cdot - x\rangle$ is a linear lower approximation to $f(\cdot)$ in the sense that $0 \le f(y) - \{t + \langle g, y - x\rangle\} \le \frac{L}{2}\|y - x\|^2 + \delta$. Devolder et al. [13] argue that this notion appears naturally in several optimization problems and study whether the so-called accelerated gradient techniques are still superior to their non-accelerated counterparts (and find a negative answer). The authors study both lower and upper rates of convergence, similarly to our paper. A major difference between the previous and our settings is that we allow stochastic noise (and bias), which the algorithms can control, while the oracle in these previous paper must guarantee that the accuracy requirements hold in each time step with probability one. This is a much stronger requirement, which may be impossible to satisfy in some problems, such as when the only information available about the functions is noise contaminated.

Some works, such as Schmidt et al. [31] allow arbitrary sequences of errors and show error bounds as a function of the accumulated errors. Our proof technique is actually essentially the same (as can be expected). However, the noisy case requires special care. For example, Proposition 3 of Schmidt et al. [31] bounds the optimization error for the smooth, convex case by $O(1/n^2(\|x_1 - x^*\|^2 + A_n^2))$ where $A_n = O(\sum_{t=1}^n t\|e_t\|)$, e_t being the error of the approximate gradient. This expression becomes $\Theta(\frac{1}{n^2}\sum_{t=1}^n t^2) \approx n$ assuming that errors' noise level is a positive constant (in all our result, this holds). This clearly shows that the noisy case requires (somewhat) special treatment.

Similar, but simpler noisy oracle models were introduced [21, 19, 16], but these models lack the bias-variance tradeoff central to this paper (i.e., they assume the variance and bias can be controlled independently of each other). The results in these papers are upper bounds on the error of certain gradient methods (also to some very specific problem for [19]), and they correspond to the bounds we obtained with $q = 0$.

7 Conclusions

We presented a novel noisy gradient oracle model for convex optimization. The oracle model covers several gradient estimation methods in the literature designed for algorithms that can observe only noisy function values, while allowing to handle explicitly the bias-variance tradeoff of these estimators. The framework allows to derive sharp upper and lower bounds on the minimax optimization error and the regret in the online case. From our lower bounds it follows that the current state of the art in designing and analyzing noisy gradient methods for stochastic and online smooth bandit convex optimization are suboptimal.

Acknowledgements

This work was supported by the Alberta Innovates Technology Futures through the Alberta Ingenuity Centre for Machine Learning and RLAI, NSERC, the National Science Foundation (NSF) under Grants CMMI-1434419, CNS-1446665, and CMMI-1362303, and by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-15-10050.

References

[1] Abernethy, J., Hazan, E., and Rakhlin, A. (2008). Competing in the dark: An efficient algorithm for bandit linear optimization. In *COLT*, pages 263–274.

[2] Agarwal, A., Dekel, O., and Xiao, L. (2010). Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pages 28–40.

[3] Agarwal, A., Foster, D. P., Hsu, D., Kakade, S. M., and Rakhlin, A. (2013). Stochastic convex optimization with bandit feedback. *SIAM Journal on Optimization*, 23(1):213–240.

[4] Baes, M. (2009). Estimate sequence methods: Extensions and approximations. Technical report, IFOR Internal report, ETH Zurich, Switzerland.

[5] Bhatnagar, S., Prasad, H. L., and Prashanth, L. A. (2013). *Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods (Lecture Notes in Control and Information Sciences)*, volume 434. Springer.

[6] Bubeck, S. (2014). Theory of convex optimization for machine learning. Technical report, Microsoft Research.

[7] Bubeck, S., Dekel, O., Koren, T., and Peres, Y. (2015). Bandit convex optimization: $o(\sqrt{T})$ regret in one dimension. In *COLT*, pages 266–278.

[8] Bubeck, S. and Eldan, R. (2015). Multi-scale exploration of convex functions and bandit convex optimization. Technical report, Microsoft Research.

[9] Chen, H. (1988). Lower rate of convergence for locating a maximum of a function. *The Annals of Statistics*, 16(3):1330–1334.

[10] d'Aspremont, A. (2008). Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19:1171–1183.

[11] Dekel, O., Eldan, R., and Koren, T. (2015). Bandit smooth convex optimization: Improving the bias-variance tradeoff. In *NIPS*, pages 2926–2934.

[12] Dekel, O., Gilad-Bachrach, R., Shamir, O., and Xiao, L. (2012). Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(1):165–202.

[13] Devolder, O., Glineur, F., and Nesterov, Y. (2014). First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146:37–75.

[14] Dippon, J. (2003). Accelerated randomized stochastic optimization. *The Annals of Statistics*, 31(4):1260–1281.

[15] Duchi, J. C., Jordan, M., Wainwright, M. J., and Wibisono, A. (2015). Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806.

[16] Dvurechensky, P. and Gasnikov, A. (2015). Stochastic intermediate gradient method: Convex and strongly convex cases. arXiv:1411.2876.

[17] Flaxman, A. D., Kalai, A. T., and McMahan, H. B. (2005). Online convex optimization in the bandit setting: gradient descent without a gradient. In *SODA*, pages 385–394.

[18] Hazan, E. and Levy, K. (2014). Bandit convex optimization: Towards tight bounds. In *NIPS*, pages 784–792.

[19] Honorio, J. (2012). Convergence rates of biased stochastic optimization for learning sparse ising models. In *ICML*, pages 257–264, New York, NY, USA. Omnipress.

[20] Hu, X., L.A., P., György, A., and Szepesvári, C. (2016). (Bandit) convex optimization with biased noisy gradient oracles. arXiv.

[21] Juditsky, A. and Nemirovski, A. (2011). First-order methods for nonsmooth convex large-scale optimization, i: General purpose methods. In Sra, S., Nowozin, S., and Wright, S., editors, *Optimization for Machine Learning*, pages 121–147. MIT press.

[22] Katkovnik, V. Y. and Kulchitsky, Y. (1972). Convergence of a class of random search algorithms. *Automation Remote Control*, 8:1321–1326.

[23] Kleinman, N. L., Spall, J. C., and Naiman, D. Q. (1999). Simulation-based optimization with stochastic approximation using common random numbers. *Management Science*, 45(11):1570–1578.

[24] Kushner, H. J. and Clark, D. S. (1978). *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer Verlag, New York.

[25] Liang, T., Narayanan, H., and Rakhlin, A. (2014). On zeroth-order stochastic convex optimization via random walks. arXiv preprint1402.2667.

[26] Mahdavi, M. (2014). *Exploiting Smoothness in Statistical Learning, Sequential Prediction, and Stochastic Optimization*. PhD thesis, Michigan State University.

[27] Nesterov, Y. (2004). *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media.

[28] Nesterov, Y. and Spokoiny, V. (2011). Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, pages 1–40.

[29] Polyak, B. and Tsybakov, A. (1990). Optimal orders of accuracy for search algorithms of stochastic optimization. *Problems in Information Transmission*, pages 126–133.

[30] Saha, A. and Tewari, A. (2011). Improved regret guarantees for online smooth convex optimization with bandit feedback. In *AISTATS*, pages 636–642.

[31] Schmidt, M. W., Roux, N. L., and Bach, F. R. (2011). Convergence rates of inexact proximal-gradient methods for convex optimization. In *NIPS*, pages 1458–1466.

[32] Shamir, O. (2012). On the complexity of bandit and derivative-free stochastic convex optimization. In *COLT*.

[33] Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341.

[34] Spall, J. C. (1997). A one-measurement form of simultaneous perturbation stochastic approximation. *Automatica*, 33(1):109–112.

[35] Yao, A. C. C. (1977). Probabilistic computations: Toward a unified measure of complexity. In *FOCS*, pages 222–227.