Bandit Convex Optimization in Non-stationary Environments

Peng Zhao, Guanghui Wang, Lijun Zhang, Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210023, China
{zhaop, wanggh, zhanglj, zhouzh}@lamda.nju.edu.cn

Abstract

Bandit Convex Optimization (BCO) is a fundamental framework for modeling sequential decision-making with partial information, where the only feedback available to the player is the one-point or two-point function values. In this paper, we investigate BCO in non-stationary environments and choose the *dynamic regret* as the performance measure, which is defined as the difference between the cumulative loss incurred by the algorithm and that of any feasible comparator sequence. Let T be the time horizon and P_T be the path-length of the comparator sequence that reflects the non-stationarity of environments. We propose a novel algorithm that achieves $O(T^{3/4}(1+P_T)^{1/2})$ and $O(T^{1/2}(1+P_T)^{1/2})$ dynamic regret respectively for the one-point and two-point feedback models. The latter result is optimal, matching the $\Omega(T^{1/2}(1+P_T)^{1/2})$ lower bound established in this paper. Notably, our algorithm is more adaptive to non-stationary environments since it does not require prior knowledge of the path-length P_T ahead of time, which is generally unknown.

1 Introduction

Online Convex Optimization (OCO) is a powerful tool for modeling sequential decision-making problems, which can be regarded as an iterative game between the player and environments [Shalev-Shwartz, 2012]. At iteration t, the player commits a decision $\mathbf{x}_t$ from a convex feasible set $\mathcal{X} \subseteq \mathbb{R}^d$, simultaneously, a convex function $f_t : \mathcal{X} \mapsto \mathbb{R}$ is revealed by environments, and then the player will suffer an instantaneous loss $f_t(\mathbf{x}_t)$. The standard performance measure is the *regret*,

$$\text{S-Regret}_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) \qquad (1)$$

which is the difference between the cumulative loss of the player and that of the best *fixed* decision in hindsight. To emphasize the fact that the comparator in (1) is fixed, it is called *static* regret.

There are two setups for online convex optimization according to the information that environments reveal [Hazan, 2016]. In the *full-information* setup, the player has all the information of the function f_t, including the gradients of f_t over $\mathcal{X}$. By contrast, in the *bandit* setup, the instantaneous loss is the only feedback available to the player. In this paper, we focus on the latter case, which is referred to as the bandit convex optimization (BCO).

BCO has attracted considerable attention because it successfully models many real-world scenarios where the feedback available to the decision maker is partial or incomplete [Hazan, 2016]. The key challenge lies in the limited feedback, i.e., the player has no access to gradients of the function. In the standard *one-point feedback* model, the only feedback is the one-point function value, based on which Flaxman et al. [2005] constructed an unbiased estimator of the gradient and then appealed to the online gradient descent algorithm that developed in the full-information setting [Zinkevich, 2003] to establish an $O(T^{3/4})$ expected regret. Another common variant is the *two-point feedback* model, where the player is allowed to query function values of two points at each iteration. Agarwal et al. [2010] demonstrated an optimal $O(\sqrt{T})$ regret for convex functions under this feedback model. Algorithms and regret bounds are further developed in later studies [Saha and Tewari, 2011, Hazan and Levy, 2014, Bubeck et al., 2015, Dekel et al., 2015, Yang and Mohri, 2016, Bubeck et al., 2017].

Note that the static regret in (1) compares with a fixed benchmark, so it implicitly assumes that there

Proceedings of the 23$^{\text{rd}}$International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

Table 1: Comparisons of dynamic regret for BCO problems. In the table, the column of "Parm-Free" indicates whether the algorithm requires to know the path-length in advance. Meanwhile, T is the time horizon, $P_T = P_T(\mathbf{u}_1, \ldots, \mathbf{u}_T)$ and $P_T^* = \max_{\mathbf{x}_1,\ldots,\mathbf{x}_T \in \mathcal{X}} P_T(\mathbf{x}_1, \cdots, \mathbf{x}_T)$.

Feedback model	Dynamic regret	Type	Parm-Free	Reference
one-point	$O\big(T^{\frac{3}{4}}(1+P_T^*)\big)$	worst-case	NO	[Chen and Giannakis, 2019]
one-point	$O\big(T^{\frac{3}{4}}(1+P_T)^{\frac{1}{2}}\big)$	universal	YES	This work
two-point	$O\big(\sqrt{T(1+P_T^*)}\big)$	worst-case	NO	[Yang et al., 2016]
two-point	$O\big(\sqrt{T}(1+P_T^*)\big)$	worst-case	NO	[Chen and Giannakis, 2019]
two-point	$O\big(\sqrt{T(1+P_T)}\big)$	universal	YES	This work

is a reasonably good decision over all iterations. Unfortunately, this may not be true in non-stationary environments, where the underlying distribution of online functions changes. To address this limitation, the notion of *dynamic regret* is introduced by Zinkevich [2003] and defined as the difference between the cumulative loss of the player and that of a comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$,

$$\text{D-Regret}_T(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t). \quad (2)$$

In contrast to a fixed benchmark in the static regret, dynamic regret compares with a *changing* comparator sequence and therefore is more suitable in non-stationary environments. We remark that (2) is also called the *universal* dynamic regret, since it holds universally for any feasible comparator sequence. In the literature, there is a variant named the *worst-case* dynamic regret [Besbes et al., 2015], which specifies the comparator sequence to be minimizers of online functions, namely, $\mathbf{u}_t = \mathbf{x}_t^* \in \arg\min_{\mathbf{x}\in\mathcal{X}} f_t(\mathbf{x})$. As pointed out by Zhang et al. [2018], the universal dynamic regret is more desired, because the worst-case dynamic regret is typically too pessimistic while the universal one is more adaptive to the non-stationarity of environments. Moreover, the universal dynamic regret is more general since it accommodates the worst-case dynamic regret and static regret as special cases.

Recently, there are some studies on the dynamic regret of BCO problems [Yang et al., 2016, Chen and Giannakis, 2019]. They provide the worst-case dynamic regret only, and the algorithms require some quantities as the input which are generally unknown in advance. Therefore, it is desired to design algorithms that enjoy *universal* dynamic regret for BCO problems.

In this paper, we start with the bandit gradient descent (BGD) algorithm of Flaxman et al. [2005], and analyze its universal dynamic regret. We demonstrate that the optimal parameter configuration of vanilla BGD also requires prior information of the unknown path-length. To address this issue, we propose the Parameter-free Bandit Gradient Descent algorithm (PBGD), which is inspired by the strategy of maintaining multiple learning rates [van Erven and Koolen, 2016]. Our approach is essentially an online ensemble method [Zhou, 2012], consisting of meta-algorithm and expert-algorithm. The basic idea is to maintain a pool of candidate parameters, and then invoke multiple instances of the expert-algorithm simultaneously, where each expert-algorithm is associated with a candidate parameter. Next, the meta-algorithm combines predictions from expert-algorithms by an expert-tracking algorithm [Cesa-Bianchi and Lugosi, 2006]. However, it is prohibited to run multiple expert-algorithms with different parameters simultaneously in BCO problems, since the player is only allowed to query one/two points in the bandit setup. To overcome this difficulty, we carefully design a surrogate function, as the linearization of the smoothed version of the loss function in the sense of expectation, and make the strategy suitable for bandit convex optimization. Our algorithm and analysis accommodate one-point and two-point feedback models, and Table 1 summarizes existing dynamic regret for BCO problems and our results. The main contributions of this work are listed as follows.

- We establish the first *universal* dynamic regret that supports to compare with any feasible comparator sequence for the bandit gradient descent algorithm, in a unified analysis framework.
- We propose a *parameter-free* algorithm, which does not require to know the upper bound of the path-length P_T ahead of time, and meanwhile enjoys the state-of-the-art dynamic regret.
- We establish the *first* minimax lower bound of universal dynamic regret for BCO problems.

2 Related Work

In this section, we briefly introduce related work of bandit convex optimization and dynamic regret.

2.1 Bandit Convex Optimization

In the bandit convex optimization setting, the player is only allowed to query function values of one point or two points, and the gradient information is not accessible as opposed to the full-information setting.

For the one-point feedback model, the seminal work of Flaxman et al. [2005] constructed an unbiased gradient estimator and established an $O(T^{3/4})$ expected regret for convex and Lipschitz functions. A similar result was independently obtained by Kleinberg [2004]. Later, an $O(T^{2/3})$ rate was shown to be attainable with either strong convexity [Agarwal et al., 2010] or smoothness [Saha and Tewari, 2011]. When functions are both strongly convex and smooth, Hazan and Levy [2014] designed a novel algorithm that achieves a regret of $O(\sqrt{T \log T})$ based on the follow-the-regularized-leader framework with self-concordant barriers, matching the $\Omega(\sqrt{T})$ lower bound [Shamir, 2013] up to logarithmic factors. Furthermore, recent breakthroughs [Bubeck et al., 2015, 2017] showed that $O(\text{ploy}(\log T)\sqrt{T})$ regret is attainable for convex and Lipschitz functions, though with a high dependence on the dimension d.

BCO with two-point feedback was proposed and studied by Agarwal et al. [2010], and was also independently studied in the context of stochastic optimization [Nesterov, 2011]. Agarwal et al. [2010] first established the expected regret of $O(d^2\sqrt{T})$ and $O(d^2 \log T)$ for convex Lipschitz and strongly convex Lipschitz functions, respectively. These bounds are proved to be minimax optimal in T [Agarwal et al., 2010], and the dependence on d was later improved to be optimal [Shamir, 2017].

Besides, bandit linear optimization is a special case of BCO where the feedback is assumed to be a linear function of the chosen decision, and has been studied extensively [Awerbuch and Kleinberg, 2004, McMahan and Blum, 2004, Dani et al., 2007, Abernethy et al., 2008, Bubeck et al., 2012].

2.2 Dynamic Regret

There are two types of dynamic regret as aforementioned. The universal dynamic regret holds universally for any feasible comparator sequence, while the worst-case one only compares with the sequence of the minimizers of online functions.

For the universal dynamic regret, existing results are only limited to the full-information setting. Zinkevich [2003] showed that OGD achieves an $O(\sqrt{T}(1 + P_T))$ regret, where $P_T = P_T(\mathbf{u}_1, \ldots, \mathbf{u}_T)$ is the path-length of comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T$,

$$P_T(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=2}^{T} \|\mathbf{u}_{t-1} - \mathbf{u}_t\|_2. \tag{3}$$

Recently, Zhang et al. [2018] demonstrated that this upper bound is not optimal by establishing an $\Omega(\sqrt{T(1 + P_T)})$ lower bound, and further proposed an algorithm that attains an optimal $O(\sqrt{T(1 + P_T)})$ dynamic regret for convex functions. However, there is no universal dynamic regret in the bandit setting.

For the worst-case dynamic regret, there are many studies in the full-information setting [Besbes et al., 2015, Jadbabaie et al., 2015, Yang et al., 2016, Mokhtari et al., 2016, Zhang et al., 2017] as well as a few works in the bandit setting [Gur et al., 2014, Yang et al., 2016, Luo et al., 2018, Auer et al., 2019, Cheung et al., 2019, Chen and Giannakis, 2019]. In the bandit convex optimization, when the upper bound of P_T^* is known, Yang et al. [2016] established an $O(\sqrt{T(1 + P_T^*)})$ dynamic regret for the two-point feedback model. Here, $P_T^* = \max_{\mathbf{x}_1,\ldots,\mathbf{x}_T \in \mathcal{X}} P_T(\mathbf{x}_1, \cdots, \mathbf{x}_T)$ is the longest path-length of the feasible comparator sequence. Later, Chen and Giannakis [2019] applied BCO techniques in the dynamic Internet-of-Things management, showing $O(T^{3/4}(1 + P_T^*))$ and $O(T^{1/2}(1 + P_T^*))$ dynamic regret bounds respectively for one-point and two-point feedback models.

3 Bandit Gradient Descent (BGD)

In this section, we provide assumptions used in the paper, then present the bandit gradient descent (BGD) algorithm for BCO problems, as well as its universal dynamic regret. To the best of our knowledge, this is the first work that analyzes the universal dynamic regret of BGD.

3.1 Assumptions

We make following common assumptions for BCO [Flaxman et al., 2005, Agarwal et al., 2010].

Assumption 1 (Bounded Region)**.** The feasible set $\mathcal{X}$ contains the ball of radius r centered at the origin and is contained in the ball of radius R,

$$r\mathbb{B} \subseteq \mathcal{X} \subseteq R\mathbb{B} \tag{4}$$

where $\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_2 \leq 1\}$.

Assumption 2 (Bounded Function Value)**.** The absolute values of all the functions are bounded by C,

$$\forall t \in [T], \quad \max_{\mathbf{x} \in \mathcal{X}} |f_t(\mathbf{x})| \leq C. \tag{5}$$

Assumption 3 (Lipschitz Continuity)**.** All the functions are L-Lipschitz continuous over domain $\mathcal{X}$, that is, for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, we have

$$\forall t \in [T], \quad |f_t(\mathbf{x}) - f_t(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|_2. \tag{6}$$

Meanwhile, we consider loss functions and the comparator sequence are chosen by an oblivious adversary.

3.2 Algorithm and Regret Analysis

In this part, we present algorithm and regret analysis of the bandit gradient descent.

We start from the online gradient descent (OGD) developed in the full-information setting [Zinkevich, 2003]. OGD begins with any $\mathbf{x}_1 \in \mathcal{X}$ and performs

$$\mathbf{x}_{t+1} = \text{Proj}_{\mathcal{X}}[\mathbf{x}_t - \eta \nabla f_t(\mathbf{x}_t)] \tag{7}$$

where $\eta > 0$ is the step size and $\text{Proj}_{\mathcal{X}}[\cdot]$ denotes the projection onto the nearest point in $\mathcal{X}$.

The key challenge of BCO problems is the lack of gradients. Therefore, Flaxman et al. [2005] and Agarwal et al. [2010] proposed to replace $\nabla f_t(\mathbf{x}_t)$ in (7) with a gradient estimator $\widetilde{g}_t$, obtained by evaluating the function at one (in the one-point feedback model) or two random points (in the two-point feedback model) around $\mathbf{x}_t$. Details will be presented later. We unify their algorithms in Algorithm 1, called the Bandit Gradient Descent (BGD). Notice that in lines 8 and 14 of the algorithm, the projection of $\mathbf{y}_{t+1}$ is on a slightly smaller set $(1-\alpha)\mathcal{X}$ instead of $\mathcal{X}$ to ensure that the final decision $\mathbf{x}_{t+1}$ lies in the feasible set $\mathcal{X}$, and the idea is originated in the works of tracking the best experts [Herbster and Warmuth, 1998, 2001].

In the following, we describe the gradient estimator and analyze the universal dynamic regret for each model.

One-Point Feedback Model. Flaxman et al. [2005] proposed the following gradient estimator,

$$\widetilde{g}_t = \frac{d}{\delta} f_t(\mathbf{y}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t \tag{8}$$

where $\mathbf{s}_t$ is a unit vector selected uniformly at random and $\delta > 0$ is the perturbation parameter. Then, the following lemma [Flaxman et al., 2005, Lemma 2.1] guarantees that (8) is an unbiased gradient estimator of the smoothed version of the loss function f_t.

Lemma 1. *For any convex (but not necessarily differentiable) function $f : \mathcal{X} \mapsto \mathbb{R}$, define its* smoothed *version $\widehat{f}(\mathbf{x}) = \mathbb{E}_{\mathbf{v}\in\mathbb{B}}[f(\mathbf{x}+\delta\mathbf{v})]$. Then, for any $\delta > 0$,*

$$\mathbb{E}_{\mathbf{s}\in\mathbb{S}}[f(\mathbf{x}+\delta\mathbf{s}) \cdot \mathbf{s}] = \frac{\delta}{d} \nabla \widehat{f}(\mathbf{x}) \tag{9}$$

where $\mathbb{S}$ is the unit sphere centered around the origin, namely, $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^d | \|\mathbf{x}\|_2 = 1\}$.

Therefore, we adopt $\widetilde{g}_t$ to perform the online gradient descent in (7). The main update procedures of the one-point feedback model are summarized in the case 1 (line 4-7) of Algorithm 1. We have the following result regarding its universal dynamic regret.

Algorithm 1 Bandit Gradient Descent (BGD)

```
Input: time horizon $T$, perturbation parameter $\delta$,
    shrinkage parameter $\alpha$, step size $\eta$
 1: Let $\mathbf{y}_1 = \mathbf{0}$
 2: for $t = 1$ to $T$ do
 3:     Select a unit vector $\mathbf{s}_t$ uniformly at random
        {Case 1. One-Point Feedback Model}
 4:     Submit $\mathbf{x}_t = \mathbf{y}_t + \delta\mathbf{s}_t$
 5:     Receive $f_t(\mathbf{x}_t)$ as the feedback
 6:     Construct the gradient estimator by (8)
 7:     $\mathbf{y}_{t+1} = \text{Proj}_{(1-\alpha)\mathcal{X}}[\mathbf{y}_t - \eta\widetilde{g}_t]$
        {Case 2. Two-Point Feedback Model}
 8:     Submit $\mathbf{x}_t^{(1)} = \mathbf{y}_t + \delta\mathbf{s}_t$ and $\mathbf{x}_t^{(2)} = \mathbf{y}_t - \delta\mathbf{s}_t$
 9:     Receive $f_t(\mathbf{x}_t^{(1)})$ and $f_t(\mathbf{x}_t^{(2)})$ as the feedback
10:     Construct the gradient estimator by (11)
11:     $\mathbf{y}_{t+1} = \text{Proj}_{(1-\alpha)\mathcal{X}}[\mathbf{y}_t - \eta\widetilde{g}_t]$
12: end for
```

Theorem 1. *Under Assumptions 1, 2, and 3, for any $\delta > 0$, $\eta > 0$, and $\alpha = \delta/r$, the expected dynamic regret of* BGD(T,δ,α,η) *for the one-point feedback model satisfies*

$$\begin{aligned} &\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \\ &\le \frac{7R^2 + RP_T}{4\eta} + \frac{\eta d^2 C^2 T}{2\delta^2} + \big(3L + \frac{LR}{r}\big)\delta T, \end{aligned} \tag{10}$$

for any *feasible comparator sequence* $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$.

Remark 1. By setting $\eta = ((7R^2 + RP_T)/T)^{3/4}$ and $\delta = \eta^{1/3}$, we obtain an $O(T^{3/4}(1+P_T)^{1/4})$ dynamic regret. However, such a configuration requires prior knowledge of P_T, which is generally unavailable. We will develop a parameter-free algorithm to eliminate the undesired dependence later.

Two-Point Feedback Model. In this setup, the player is allowed to query two points, $\mathbf{x}_t^{(1)} = \mathbf{y}_t + \delta\mathbf{s}_t$ and $\mathbf{x}_t^{(2)} = \mathbf{y}_t - \delta\mathbf{s}_t$. Then, the function values $f_t(\mathbf{x}_t^{(1)})$ and $f_t(\mathbf{x}_t^{(2)})$ are revealed as the feedback. We use the following gradient estimator [Agarwal et al., 2010],

$$\widetilde{g}_t = \frac{d}{2\delta}\left(f_t(\mathbf{y}_t + \delta\mathbf{s}_t) - f_t(\mathbf{y}_t - \delta\mathbf{s}_t)\right) \cdot \mathbf{s}_t. \tag{11}$$

The major limitation of the one-point gradient estimator (8) is that it has a potentially large magnitude, proportional to the $1/\delta$ which is usually quite large since the perturbation parameter δ is typically small. This is avoided in the two-point gradient estimator (11), whose magnitude can be upper bounded by Ld, independent of the perturbation parameter δ. This crucial advantage leads to the substantial improvement in the dynamic regret (also static regret).

Theorem 2. *Under Assumptions 1, 2, and 3, for any $\delta > 0$, $\eta > 0$, and $\alpha = \delta/r$, the expected dynamic regret of* $\mathrm{BGD}(T, \delta, \alpha, \eta)$ *for the two-point feedback model satisfies*

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T}\frac{1}{2}\big(f_t(\mathbf{x}_t^{(1)}) + f_t(\mathbf{x}_t^{(2)})\big)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}_t)\\ &\leq \frac{7R^2 + RP_T}{4\eta} + \frac{\eta L^2 d^2}{2}T + \big(3L + \frac{LR}{r}\big)\delta T\end{aligned} \tag{12}$$

for any *feasible comparator sequence* $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$.

Remark 2. By setting $\eta = \sqrt{(7R^2 + RP_T)/(2L^2d^2T)}$ and $\delta = 1/\sqrt{T}$, BGD algorithm achieves an $O(T^{1/2}(1+P_T)^{1/2})$ dynamic regret. However, this configuration has an unpleasant dependence on the unknown quantity P_T, which will be removed in the next part.

4 Parameter-Free BGD

From Theorems 1 and 2, we observe that the optimal parameter configurations of BGD algorithm require to know the path-length P_T in advance, which is generally unknown. In this section, we develop a parameter-free algorithm to address this limitation.

4.1 Algorithm

The fundamental obstacle in obtaining universal dynamic regret guarantees is that the path-length P_T remains unknown even after all iterations, since the comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T$ can be chosen arbitrarily from the feasible set. Therefore, the well-known doubling trick [Cesa-Bianchi et al., 1997] is not applicable to remove the dependence on the unknown path-length. Another possible technique to overcome this difficulty is to grid search the optimal parameter by maintaining multiple learning rates in parallel and using expert-tracking algorithms to combine predictions and track the best parameter [van Erven and Koolen, 2016]. However, it is infeasible to directly apply this method to bandit convex optimization because of the inherent difficulty of bandit setting — it is only allowed to query the function value *once* at each iteration.

To address this issue, we need a closer investigation of dynamic regret analysis of BCO problems. Taking the one-feedback model as an example, the expected dynamic regret can be decomposed into three terms,

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{u}_t)\\ &= \underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\Big(\widehat{f}_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{v}_t)\Big)\right]}_{\texttt{term (a)}} + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\Big(f_t(\mathbf{x}_t) - \widehat{f}_t(\mathbf{y}_t)\Big)\right]}_{\texttt{term (b)}}\\ &\quad + \underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\Big(\widehat{f}_t(\mathbf{v}_t) - f_t(\mathbf{u}_t)\Big)\right]}_{\texttt{term (c)}},\end{aligned} \tag{13}$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_T$ is the scaled comparator sequence set as $\mathbf{v}_t = (1-\alpha)\mathbf{u}_t$. It turns out that term (b) and term (c) can be bounded by $2L\delta T$ and $(L\delta + L\alpha R)T$ respectively, without involving the unknown path-length. Hence, it suffices to design parameter-free algorithms to optimize term (a), i.e., the dynamic regret of the smoothed loss function $\widehat{f}_t$.

However, it remains infeasible to maintain multiple learning rates for optimizing dynamic regret of $\widehat{f}_t$. Suppose there are in total N experts where each expert is associated with a learning rate (step size), then at iteration t, expert-algorithms will require the information of $\nabla \widehat{f}_t(\mathbf{y}_t^1), \nabla \widehat{f}_t(\mathbf{y}_t^2), \ldots, \nabla \widehat{f}_t(\mathbf{y}_t^N)$ to perform the bandit gradient descent. This necessitates to query N function values of original loss f_t, which is prohibited in bandit convex optimization.

Fortunately, we discover that the expected dynamic regret of $\widehat{f}_t$ can be upper bounded by that of a linear function, as demonstrated in the following proposition.

Proposition 1.

$$\mathbb{E}[\widehat{f}_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{v}_t)] \leq \mathbb{E}[\langle \widetilde{g}_t, \mathbf{y}_t - \mathbf{v}_t \rangle]. \tag{14}$$

This feature motivates us to design the following *surrogate loss* function $\ell_t : (1-\alpha)\mathcal{X} \mapsto \mathbb{R}$,

$$\ell_t(\mathbf{y}) = \langle \widetilde{g}_t, \mathbf{y} - \mathbf{y}_t \rangle, \tag{15}$$

which can be regarded as a linearization of smoothed function $\widehat{f}_t$ on the point $\mathbf{y}_t$ in terms of expectation. Furthermore, the surrogate loss function enjoys the following two properties.

Property 1. $\forall \mathbf{y} \in (1-\alpha)\mathcal{X}$, $\nabla \ell_t(\mathbf{y}) = \widetilde{g}_t$.

Property 2. $\forall \mathbf{v} \in (1-\alpha)\mathcal{X}$,

$$\mathbb{E}[\widehat{f}_t(\mathbf{y}_t) - \widehat{f}_t(\mathbf{v})] \leq \mathbb{E}[\ell_t(\mathbf{y}_t) - \ell_t(\mathbf{v})]. \tag{16}$$

Property 1 follows from the definition of surrogate loss, and Proposition 1 immediately implies Property 2. These two properties are simple yet quite useful, and they together make the grid search feasible in bandit convex optimization. Concretely speaking,

- Property 1 implies that we can now initialize N experts to perform the bandit gradient descent *over the surrogate loss* where each expert is associated with a specific learning rate, since all the gradients $\nabla \ell_t(\mathbf{y}_t^1), \nabla \ell_t(\mathbf{y}_t^2), \ldots, \nabla \ell_t(\mathbf{y}_t^N)$ essentially equal to $\widetilde{g}_t$, which can be obtained by querying the function value of f_t only once.

- Property 2 guarantees the expected dynamic regret of smoothed functions $\widehat{f}_t$'s is upper bounded by that of the surrogate loss ℓ_t's.

Consequently, we propose to optimize surrogate loss ℓ_t instead of original loss f_t (or its smoothed version $\widehat{f}_t$). We note that the idea of constructing surrogate loss for maintaining multiple learning rates is originally proposed by van Erven and Koolen [2016] but for different purposes. They construct a quadratic upper bound for original loss f_t as surrogate loss, with the aim to adapt to the potential curvature of online functions in full-information online convex optimization. In this paper, we design the surrogate loss as linearization of smoothed function $\widehat{f}_t$ in terms of expectation, to make the grid search of optimal parameter doable in bandit convex optimization. To the best of our knowledge, this is the first time to optimize surrogate loss for maintaining multiple learning rates in *bandit* setup.

In the following, we describe the design details of parameter-free algorithms for the one-point feedback model, and will present configurations of BCO with two-point feedback model in a longer version.

In the one-point feedback model, the optimal step size is $\eta^* = \sqrt{7R^2 + RP_T}/(dCT^{3/4})$, whose value is unavailable due to the unknown path-length P_T. Nevertheless, we confirm that

$$\frac{\sqrt{7}R}{dCT^{3/4}} \le \eta^* \le \frac{\sqrt{7+2T}R}{dCT^{3/4}} \tag{17}$$

always holds from the non-negativity and boundedness of the path-length ($0 \le P_T \le 2RT$). Hence, we first construct the following pool of candidate step sizes $\mathcal{H}$ to discretize the range of optimal parameter in (17),

$$\mathcal{H} = \left\{ \eta_i = 2^{i-1}\frac{\sqrt{7}R}{dCT^{3/4}} \,\Big|\, i = 1, \ldots, N \right\}, \tag{18}$$

where $N = \lceil \frac{1}{2}\log_2(1+2T/7)\rceil + 1$. The above configuration ensures there exists an index $k \in \{1, \ldots, N-1\}$ such that $\eta_k \le \eta^* \le \eta_{k+1} = 2\eta_k$. More intuitively, there is a step size in the pool $\mathcal{H}$ that is not optimal but sufficiently close to η^*. Next, we instantiate N expert-algorithms, where the i-th expert is a BGD algorithm with parameters $\eta_i \in \mathcal{H}$ and $\delta = T^{-1/4}$. Finally, we adopt an expert-tracking algorithm as the meta-algorithm to combine predictions from all the experts to produce the final decision. Owing to nice theoretical guarantees of the meta-algorithm, dynamic regret of final decisions is comparable to that of the best expert, i.e., the expert-algorithm with near-optimal step size.

We present descriptions for expert-algorithm and meta-algorithm of PBGD as follows.

Expert-algorithm. For each candidate step size from the pool $\mathcal{H}$, we initialize an expert, and the expert $i \in [N]$ performs the online gradient descent over the surrogate loss defined in (15),

$$\begin{aligned} \mathbf{y}_{t+1}^i &= \mathrm{Proj}_{(1-\alpha)\mathcal{X}}[\mathbf{y}_t^i - \eta_i \nabla \ell_t(\mathbf{y}_t^i)] \\ &= \mathrm{Proj}_{(1-\alpha)\mathcal{X}}[\mathbf{y}_t^i - \eta_i \widetilde{g}_t], \end{aligned} \tag{19}$$

where η_i is the step size of the expert i, shown in (18).

Algorithm 2 PBGD: Meta-algorithm

```
Input: time horizon $T$, the pool of candidate step sizes $\mathcal{H}$, learning rate of the meta-algorithm $\epsilon$
Run expert-algorithms (19) with different step sizes simultaneously
Initialize the weight of each expert as
    $$w_1^i = \frac{N+1}{N} \cdot \frac{1}{i(i+1)}, \quad \forall i \in [N]$$
for $t = 1$ to $T$ do
    Receive $\mathbf{y}_t^i$ from each expert $i \in [N]$
    Obtain $\mathbf{y}_t = \sum_{i \in [N]} w_t^i \mathbf{y}_t^i$
    Submit $\mathbf{x}_t = \mathbf{y}_t + \delta \mathbf{s}_t$ and incur loss $f_t(\mathbf{x}_t)$
    Compute gradient estimator $\widetilde{g}_t$ by (8)
    Construct surrogate loss $\ell_t(\cdot)$ as (15)
    Update the weight of each expert $i \in [N]$ by
        $$w_{t+1}^i = \frac{w_t^i \exp(-\epsilon \ell_t(\mathbf{y}_t^i))}{\sum_{i \in [N]} w_t^i \exp(-\epsilon \ell_t(\mathbf{y}_t^i))}$$
    Send the gradient estimator $\widetilde{g}_t$ to each expert
end for
```

The above update procedure once again demonstrates the necessity of constructing the surrogate loss. Due to the nice property of surrogate loss (Property 1), at each iteration, all the experts can perform the *exact* online gradient descent in the same direction $\widetilde{g}_t$. By contrast, suppose each expert is conducted over the smoothed loss function $\widehat{f}_t$, then at each iteration it requires to query multiple gradients $\nabla \widehat{f}_t(\mathbf{y}_t^i)$, or equivalently, to query multiple function values $f_t(\mathbf{x}_t^i)$, which are unavailable in bandit convex optimization.

Meta-algorithm. To combine predictions returned from various experts, we adopt the exponentially weighted average forecaster algorithm [Cesa-Bianchi and Lugosi, 2006] with nonuniform initial weights as the meta-algorithm, whose input is the pool of candidate step sizes $\mathcal{H}$ in (18) and its own learning rate ϵ. The nonuniform initialization of weights aims to make regret analysis tighter, which will be clear in the proof. Algorithm 2 presents detailed procedures. Note that the meta-algorithm itself does not require any prior information of the unknown path-length P_T.

The meta-algorithm in Algorithm 2, together with the expert-algorithm (19), gives PBGD (short for *Parameter-free Bandit Gradient Descent*).

4.2 Regret Analysis

The following theorem states the dynamic regret of the proposed PBGD algorithm.

Theorem 3. *Under Assumptions 1, 2, and 3, with a proper setting of the pool of candidate step sizes $\mathcal{H}$ and the learning rate ϵ,* PBGD *algorithm enjoys the following expected dynamic regret,*

- *One-Point Feedback Model:* $O\big(T^{\frac{3}{4}}(1+P_T)^{\frac{1}{2}}\big)$;
- *Two-Point Feedback Model:* $O\big(T^{\frac{1}{2}}(1+P_T)^{\frac{1}{2}}\big)$.

The above results hold universally for any *feasible comparator sequence* $\mathbf{u}_1,\ldots,\mathbf{u}_T\in\mathcal{X}$.

Remark 3. Theorem 3 shows that the dynamic regret can be improved from $O\big(T^{\frac{3}{4}}(1+P_T)^{\frac{1}{2}}\big)$ to $O\big(T^{\frac{1}{2}}(1+P_T)^{\frac{1}{2}}\big)$ when it is allowed to query two points at each iteration. The attained dynamic regret (though in expectation) of BCO with two-point feedback, surprisingly, is in the same order with that of the full-information setting [Zhang et al., 2018]. This extends the claim argued by Agarwal et al. [2010] *knowing the value of each loss function at two points is almost as useful as knowing the value of each function everywhere* to dynamic regret analysis. Furthermore, we will show that the obtained dynamic regret for the two-point feedback model is minimax optimal in the next section.

Due to the limitation of space, we only present the proof sketch of one-point feedback model. More details and all the other proofs (for two-point feedback model) will be presented in a longer version.

Proof Sketch. To obtain dynamic regret, it suffices to bound term (a) as shown in (13). Proposition 1 implies that term (a) can be upper bounded by $\mathbb{E}\big[\sum_{t=1}^T(\ell_t(\mathbf{y}_t)-\ell_t(\mathbf{v}_t))\big]$, expected dynamic regret in terms of surrogate loss, denoted by D_T. Notice that D_T can be further decomposed into two terms,

$$D_T=\underbrace{\sum_{t=1}^T\big(\ell_t(\mathbf{y}_t)-\ell_t(\mathbf{y}_t^k)\big)}_{\texttt{meta-regret}}+\underbrace{\sum_{t=1}^T\big(\ell_t(\mathbf{y}_t^k)-\ell_t(\mathbf{v}_t)\big)}_{\texttt{expert-regret}}.$$

First, we bound the meta-regret. Since the path-length P_T is bounded by $2RT$ and the optimal tuning is $\eta^*=((7R^2+RP_T)/T)^{3/4}$, there exists an index $k\in[N]$ such that $\eta_k\le\eta^*\le\eta_{k+1}$ with

$$k\le\left\lceil\frac{1}{2}\log_2\Big(1+\frac{P_T}{7R}\Big)\right\rceil+1. \tag{20}$$

The regret analysis of exponentially weighted average forecaster algorithm implies

$$\begin{aligned}\texttt{meta-regret}&\le\widetilde{G}R\sqrt{2T}\,\big(1+\ln(1/w_1^k)\big)\\&\le\frac{dCR}{\delta}\sqrt{2T}\big(1+2\ln(k+1)\big). \qquad (21)\end{aligned}$$

The last inequality holds due to $w_1^k=\frac{C}{k(k+1)}\ge\frac{C}{(k+1)^2}$ and $\widetilde{G}$ is the magnitude of the gradient estimator.

Next, we bound the expert-regret. At each iteration, each expert performs deterministic OGD over surrogate loss, so we can employ the existing dynamic regret guarantee for OGD and obtain that

$$\begin{aligned}\texttt{expert-regret}&\le\frac{7R^2+RP_T}{4\eta_k}+\frac{\eta_k\widetilde{G}^2T}{2} &(22)\\&\le\frac{7R^2+RP_T}{2\eta^*}+\frac{\eta^*d^2C^2T}{2\delta^2} &(23)\\&=\frac{3\sqrt{2}}{4}dCT^{\frac{3}{4}}\sqrt{7R^2+RP_T}\end{aligned}$$

where (22) follows from the dynamic regret guarantee of OGD, (23) holds due to $\eta_k\le\eta^*\le2\eta_k$, and the last equation holds due to the setting of η^* and $\delta=T^{-1/4}$.

Therefore, by combining upper bounds of meta-regret and expert-regret, we conclude that the term (a) is upper bounded by

$$\sqrt{2}dCRT^{3/4}\big(1+2\ln(k+1)+3\sqrt{7R^2+RP_T}/4\big).$$

Finally, we bound the expected dynamic regret of original loss functions by combining above results,

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^T f_t(\mathbf{x}_t)\right]-\sum_{t=1}^T f_t(\mathbf{u}_t)\\&\overset{(13)}{=}\texttt{term (a)}+\texttt{term (b)}+\texttt{term (c)}\\&\le\texttt{term (a)}+2L\delta T+(L\delta+L\alpha R)T\\&\le\sqrt{2}dCRT^{3/4}\big(1+2\ln(k+1)+3\sqrt{7R^2+RP_T}/4\big)\\&\qquad+(3L+LR/r)T^{3/4}\\&=O\big(T^{3/4}(1+P_T)^{1/2}\big),\end{aligned}$$

where the first inequality makes use of the fact that term (b) and term (c) are bounded by $2L\delta T$ and $(L\delta+L\alpha R)T$, respectively. □

5 Lower Bound and Extension

In this section, we investigate the attainable dynamic regret for BCO problems, and then extend our algorithm to an anytime version, that is, an algorithm without requiring the time horizon in advance.

5.1 Lower Bound

We have the following minimax lower bound of universal dynamic regret for BCO problems.

Theorem 4. *For any $\tau \in [0, 2RT]$, there exists a comparator sequence $\mathbf{u}_1, \dots, \mathbf{u}_T \in \mathcal{X}$ satisfying Assumption 1 whose path-length P_T is less than τ, and a sequence of functions satisfying Assumption 3, such that for any algorithm designed for BCO with one-/two-point feedback who returns $\mathbf{x}_1, \dots, \mathbf{x}_T$,*

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \ge C \cdot dL\sqrt{(R^2 + R\tau)T}, \quad (24)$$

where C is a positive constant independent of T.

From the above lower bound and the upper bounds in Theorem 3, we know that our dynamic regret for the two-point feedback model is optimal, while the rate for one-point feedback model remains sub-optimal, where the desired rate is of order $O(T^{3/4}(1 + P_T)^{1/4})$ as demonstrated in Remark 1. Note that the desired bound does not contradict with the minimax lower bound, since $O(T^{3/4}(1 + P_T)^{1/4}) = O(T^{1/2}T^{1/4}(1 + P_T)^{1/4})$ is larger than the $\Omega(T^{1/2}(1 + P_T)^{1/2})$ lower bound by noticing that $P_T = o(T)$.

Our attained $O(T^{3/4}(1 + P_T)^{1/2})$ dynamic regret exhibits a square-root dependence on the path-length, and it will become meaningless when $P_T \ge \sqrt{T}$, though the path-length is typically small. The challenge is that the grid search technique cannot support to approximate the optimal perturbation parameter δ^* which is also dependent on P_T. Otherwise, we have to query the function more than once at each iteration. We will investigate a sharper bound for BCO with one-point feedback in the future.

Remark 4. The lower bound holds even all the functions f_t's are strongly convex and smooth in BCO with one-point feedback. This is to be contrasted with that in the full-information setting. The reason is that the minimax static regret of BCO with one-point feedback can neither benefit from strongly convexity nor smoothness [Shamir, 2013]. This implies the inherent difficulty of learning with bandit feedback.

5.2 Extension to Anytime Algorithm

Notice that the proposed PBGD algorithm requires the time horizon T as an input, which is not available in advance. In this part, we remove the undesired dependence and develop an *anytime* algorithm.

Our method is essentially a standard implementation of the doubling trick [Cesa-Bianchi et al., 1997]. Specifically, the idea is to initialize the interval by 2, and once the actual number of iterations exceeds the current counts, double the counts and restart the algorithm. So there will be $K = \lfloor \log T \rfloor + 1$ epochs and the i-th epoch contains 2^i iterations. We have the following regret guarantees for the above anytime algorithm.

Theorem 5. *Under the same conditions with Theorem 3, the anytime version of* PBGD *enjoys the following expected dynamic regret,*

- *One-Point Feedback Model: $O\big(T^{\frac{3}{4}}(\log T + P_T)^{\frac{1}{2}}\big)$;*
- *Two-Point Feedback Model: $O\big(T^{\frac{1}{2}}(\log T + P_T)^{\frac{1}{2}}\big)$.*

The above results hold universally for any *feasible comparator sequence $\mathbf{u}_1, \dots, \mathbf{u}_T \in \mathcal{X}$.*

We take the one-point feedback model as an example and provide a brief analysis as follows. Actually, by the strategy of doubling trick, we can bound the dynamic regret of the anytime algorithm by

$$\sum\nolimits_{i=1}^{K} T_i^{\frac{3}{4}}(1 + P_i)^{\frac{1}{2}} \le \sqrt{\sum\nolimits_{i=1}^{K} T_i^{\frac{3}{2}}}\sqrt{\sum\nolimits_{i=1}^{K}(1 + P_i)}$$
$$= \sqrt{\sum\nolimits_{i=1}^{K} 2^{\frac{3i}{2}}}\sqrt{\log T + P_T} = O\big(T^{\frac{3}{4}}(\log T + P_T)^{\frac{1}{2}}\big).$$

Compared with the $O(T^{3/4}(1 + P_T)^{1/2})$ rate of the original PBGD algorithm, we observe that an extra $\log T$ term is suffered due to the anytime demand.

6 Conclusion and Future Work

In this paper, we study the bandit convex optimization (BCO) problems in non-stationary environments. We propose the Parameter-free Bandit Gradient Descent (PBGD) algorithm that achieves the state-of-the-art $O(T^{3/4}(1 + P_T)^{1/2})$ and $O(T^{1/2}(1 + P_T)^{1/2})$ dynamic regret bounds for one-point and two-point feedback models respectively. The regret bounds hold universally for any feasible comparator sequence. Meanwhile, the algorithm does not need to know prior information of the path-length, which is unknown but required in previous studies. Furthermore, we demonstrate the regret bound for the two-point feedback model is minimax optimal by establishing the first lower bound for the universal dynamic regret in the bandit convex optimization setup. We also extend the algorithm to an anytime version.

In the future, we will investigate a sharper bound for BCO with one-point feedback. Moreover, we will consider incorporating other properties, like strong convexity and smoothness, to further enhance the dynamic regret for bandit convex optimization.

Acknowledgment

This research was supported by the National Science Foundation of China (61921006), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

J. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pages 263–274, 2008.

A. Agarwal, O. Dekel, and L. Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *Proceedings of the 23rd Conference on Learning Theory (COLT)*, pages 28–40, 2010.

P. Auer, Y. Chen, P. Gajane, C.-W. Lee, H. Luo, R. Ortner, and C.-Y. Wei. Achieving optimal dynamic regret for non-stationary bandits without prior information. In *Proceedings of the 32nd Conference on Learning Theory*, pages 159–163, 2019.

B. Awerbuch and R. D. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 45–53, 2004.

O. Besbes, Y. Gur, and A. J. Zeevi. Non-stationary stochastic optimization. *Operations Research*, 63(5): 1227–1244, 2015.

S. Bubeck, N. Cesa-Bianchi, and S. M. Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, 2012.

S. Bubeck, O. Dekel, T. Koren, and Y. Peres. Bandit convex optimization: $\sqrt{T}$ regret in one dimension. In *Proceedings of the 28th Conference on Learning Theory (COLT)*, volume 40, pages 266–278, 2015.

S. Bubeck, Y. T. Lee, and R. Eldan. Kernel-based methods for bandit convex optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 72–85, 2017.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3): 427–485, 1997.

T. Chen and G. B. Giannakis. Bandit convex optimization for scalable and dynamic IoT management. *IEEE Internet of Things Journal*, 6(1):1276–1286, 2019.

W. C. Cheung, D. Simchi-Levi, and R. Zhu. Learning to optimize under non-stationarity. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1079–1087, 2019.

V. Dani, T. P. Hayes, and S. M. Kakade. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 345–352, 2007.

O. Dekel, R. Eldan, and T. Koren. Bandit smooth convex optimization: Improving the bias-variance tradeoff. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 2926–2934, 2015.

A. Flaxman, A. T. Kalai, and H. B. 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 (SODA)*, pages 385–394, 2005.

Y. Gur, A. J. Zeevi, and O. Besbes. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 199–207, 2014.

E. Hazan. Introduction to Online Convex Optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

E. Hazan and K. Y. Levy. Bandit convex optimization: Towards tight bounds. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 784–792, 2014.

M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2):151–178, 1998.

M. Herbster and M. K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.

A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan. Online optimization : Competing with dynamic comparators. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

R. D. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems 17 (NIPS)*, pages 697–704, 2004.

H. Luo, C.-Y. Wei, A. Agarwal, and J. Langford. Efficient contextual bandits in non-stationary worlds. In *Proceedings of the 31st Conference On Learning Theory (COLT)*, pages 1739–1776, 2018.

H. B. McMahan and A. Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *Proceedings of the 17th Annual Conference on Learning Theory (COLT)*, pages 109–123, 2004.

A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, pages 7195–7201, 2016.

Y. Nesterov. Random gradient-free minimization of convex functions. Technical report, Université catholique de Louvain, Center for Operations Research and Econometrics (ECORE), 2011.

A. Saha and A. Tewari. Improved regret guarantees for online smooth convex optimization with bandit feedback. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 636–642, 2011.

S. Shalev-Shwartz. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

O. Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT)*, pages 3–24, 2013.

O. Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, 18:52:1–52:11, 2017.

T. van Erven and W. M. Koolen. Metagrad: Multiple learning rates in online learning. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 3666–3674, 2016.

S. Yang and M. Mohri. Optimistic bandit convex optimization. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 2289–2297, 2016.

T. Yang, L. Zhang, R. Jin, and J. Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 449–457, 2016.

L. Zhang, T. Yang, J. Yi, R. Jin, and Z.-H. Zhou. Improved dynamic regret for non-degeneracy functions. In *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017.

L. Zhang, S. Lu, and Z.-H. Zhou. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 1330–1340, 2018.

Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC Press, 2012.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 928–936, 2003.