Projection-Free Bandit Convex Optimization over Strongly Convex Sets

Chenxu Zhang[1], Yibo Wang[2], Peng Tian[3], Xiao Cheng[3], Yuanyu Wan[1](✉), and Mingli Song[1]

[1] Zhejiang University, Hangzhou 310058, China
{zhangchenxu,wanyy,brooksong}@zju.edu.cn
[2] Nanjing University, Nanjing 210023, China
[3] State Grid Chongqing Electric Power Research Institute, Chongqing 401123, China

Abstract. Projection-free algorithms for bandit convex optimization have received increasing attention, due to the ability to deal with the bandit feedback and complicated constraints simultaneously. The state-of-the-art ones can achieve an expected regret bound of $O(T^{3/4})$. However, they need to utilize a blocking technique, which is unsatisfying in practice due to the delayed reaction to the change of functions, and results in a logarithmically worse high-probability regret bound of $O(T^{3/4}\sqrt{\log T})$. In this paper, we study the special case of bandit convex optimization over strongly convex sets, and present a projection-free algorithm, which keeps the $O(T^{3/4})$ expected regret bound without employing the blocking technique. More importantly, we prove that it can enjoy an $O(T^{3/4})$ high-probability regret bound, which removes the logarithmical factor in the previous high-probability regret bound. Furthermore, empirical results on synthetic and real-world datasets have demonstrated the better performance of our algorithm.

Keywords: Projection-Free · Bandit Convex Optimization · Strongly Convex Sets

1 Introduction

Online convex optimization (OCO) plays an important role in many industrial applications with large-scale and streaming data, such as recommendation systems [26] and packet routing [3]. Specifically, it can be deemed as a repeated game between a learner and an adversary [18], in which the learner needs to first select a decision $\mathbf{x}_t$ from a convex set $\mathcal{K} \subseteq \mathbb{R}^d$ at each round t, and then the adversary chooses a convex loss function $f_t(\cdot) : \mathcal{K} \rightarrow \mathbb{R}$. The learner suffers a loss $f_t(\mathbf{x}_t)$ at each round t, and pursue that the regret defined below

$$\mathrm{Regret}(T) = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x})$$

D.-N. Yang et al. (Eds.): PAKDD 2024, LNAI 14647, pp. 118–129, 2024.
https://doi.org/10.1007/978-981-97-2259-4_9

is sublinear in the time horizon T. Over the past decades, various algorithms, such as online gradient descent (OGD) [36] and follow the regularized leader (FTRL) [28], have been proposed to achieve the optimal regret bound of $O(\sqrt{T})$.

However, there exist two common limitations in these algorithms. One is that they require the full information or the gradient of loss functions, which is not available in real applications with a black-box model. The other is that a projection operation is required at each round, which is time-consuming and even intractable when the constraint set is complicated [23]. To address the first limitation, there has been a growing research interest in OCO with the bandit feedback, where only the loss value $f_t(\mathbf{x}_t)$ is available at each round, which is also known as bandit convex optimization (BCO) [5,6,10,11,20,27]. On the other hand, to alleviate the second limitation, projection-free algorithms, which employ efficient operations such as the linear optimization in lieu of projections, have also attracted ever-increasing attention [9,14–16,19,21,32].

Nonetheless, there are only a few studies that simultaneously tackle the above two limitations. Specifically, Chen et al. [9] develop the first projection-free algorithm for BCO (PF-BCO), which combines the FTRL algorithm with the one-point gradient estimator [11] and the Frank-Wolfe (FW) iteration [12,23] for utilizing the bandit feedback and avoiding the projection, respectively. Unfortunately, this algorithm can only achieve an expected regret bound of $O(T^{4/5})$, which is worse than both the expected $O(T^{3/4})$ regret bound attained by existing algorithms only using the one-point gradient estimator [11] and the $O(T^{3/4})$ regret bound attained by existing algorithms only using the FW iteration [19]. Intuitively, this gap is caused because the variance of the one-point gradient estimator can increase the approximation error of solving the objective at each round of FTRL via the FW iteration.

To fill the gap, Garber and Kretzu [15] propose a novel algorithm called block bandit conditional gradient (BBCG), which enjoys an expected regret bound of $O(T^{3/4})$. The key technique for this improvement is a blocking technique—dividing total T rounds into size-equal blocks and only updating the decision at the end of each block, which can reduce the variance of the gradient estimator such that the approximation error of FW keeps unchanged. Recently, based on the blocking technique, a bandit and projection-free variant of OGD has also been developed to achieve the $O(T^{3/4})$ expected regret bound [16]. However, despite the improvement in the regret, the blocking technique could inevitably sacrifice the performance in practice due to the mismatch between the fixed action and the changing loss functions over each block. Moreover, we notice that due to the blocking technique, BBCG can only achieve a high-probability regret bound of $O(T^{3/4}\sqrt{\log T})$ [29],[1] which is logarithmically worse than the expected one.

To address these limitations, in this paper, we aim to improve the regret of PF-BCO without using the blocking technique. Specifically, different from the blocking technique that indirectly controls the approximation error of FW via

[1] Although Wan et al. [29] originally establish such bound for a decentralized variant of BBCG, it is easy to extend this result for BBCG.

reducing the variance of the gradient estimator, our main idea is to directly reduce the approximation error of FW. Note that in the offline setting, Garber and Hazan [13] have shown that FW can converge faster over strongly convex sets by utilizing a line search rule to select the step size. Inspired by this result, we propose an improved variant of PF-BCO by combining it with the line search, namely BFW-LS. Theoretical analysis demonstrates that our algorithm enjoys the $O(T^{3/4})$ regret bound in both expectation and high probability over strongly convex sets. Compared with previous improvements on PF-BCO, the blocking technique is dismissed, and the logarithmical term in the high-probability bound is removed. Furthermore, empirical results on synthetic and real-world datasets have verified the effectiveness of our algorithm.

2 Related Work

In this section, we briefly review related work on projection-free OCO algorithms and bandit convex optimization.

2.1 Projection-Free OCO Algorithms

To handle OCO with complicated constraints, Hazan and Kale [19] propose the first projection-free algorithm called online Frank-Wolfe (OFW), and achieve a regret bound of $O(T^{3/4})$ for the general case. The essential idea is to replace the projection operation required by FTRL [28] with an iteration of FW [12]. Specifically, following FTRL, in each round t, OFW first defines an objective function

$$F_t(\mathbf{x}) = \eta \sum_{\tau=1}^{t} \langle \nabla f_\tau(\mathbf{x}_\tau), \mathbf{x} \rangle + \|\mathbf{x} - \mathbf{x}_1\|_2^2 \tag{1}$$

with a parameter η. Then, it updates the decision by minimizing $F_t(\mathbf{x})$ via an iteration of FW, i.e.,

$$\begin{aligned} \mathbf{v}_t &= \operatorname*{argmin}_{\mathbf{x}\in\mathcal{K}} \langle \nabla F_t(\mathbf{x}_t), \mathbf{x} \rangle \\ \mathbf{x}_{t+1} &= \mathbf{x}_t + \sigma_t(\mathbf{v}_t - \mathbf{x}_t) \end{aligned} \tag{2}$$

where $\sigma_t \in [0, 1]$ is a step size. Note that only a linear optimization is required by the update, which can be implemented more efficiently than the projection over many complicated constraints [18].

Later, plenty of projection-free OCO algorithms have been proposed to establish tighter regret bounds by leveraging additional assumptions of the constraint set [14,16,32] and loss functions [17,21,25,32]. The most related one is the variant of OFW over strongly convex sets [32], which adopts the following line search rule to select the step size of OFW

$$\sigma_t = \operatorname*{argmin}_{\sigma\in[0,1]} \langle \sigma\left(\mathbf{v}_t - \mathbf{x}_t\right), \nabla F_t\left(\mathbf{x}_t\right) \rangle + \sigma^2 \|\mathbf{v}_t - \mathbf{x}_t\|_2^2. \tag{3}$$

By exploiting the faster convergence of FW over strongly convex sets [13], Wan and Zhang [32] establish a regret bound of $O(T^{2/3})$, which is better than the $O(T^{3/4})$ regret bound of the original OFW. Although our paper makes a similar exploitation as them, we want to emphasize that in the bandit setting, more careful analyses are required to deal with the variance of the gradient estimator.

Additionally, projection-free OCO algorithms have also been extended into more practical scenarios with decentralized agents [29,30,34], dynamic environments [16,24,31,33,35], and the bandit setting discussed below.

2.2 Bandit Convex Optimization

The first method for the bandit convex optimization (BCO) is proposed by Flaxman et al. [11], and attains an expected regret bound of $O(T^{3/4})$ for convex loss function. The significant contribution of this study is to introduce a profound technique called one-point gradient estimator, which can approximate the gradient with only a single loss value. Based on this technique, subsequent studies establish several improved bounds for different types of loss functions, such as the linear function [1,4], the smooth function [27], the strong convex function [2], and the smooth and strong convex function [20,22].

However, these methods rely on the projection operation or more time-consuming operations, which is unacceptable for applications with complicated constraint sets. To address this issue, Chen et al. [9] propose the PF-BCO method by combining OFW with the one-point gradient estimator, which attains an expected regret bound of $O(T^{4/5})$ for convex loss functions. Later, by employing the blocking technique, Garber and Kretzu [15] propose a refined variant of this method, namely BBCG, which reduces the expected regret bound to $O(T^{3/4})$ for the same case. Similarly, Garber and Kretzu [16] propose a bandit and projection-free variant of OGD, namely blocked online gradient descent with linear optimization oracle (LOO-BBGD), and establish the same expected regret bound of $O(T^{3/4})$. Moreover, besides the expected regret bound, Wan et al. [29] have shown that BBCG can achieve a high-probability regret bound of $O(T^{3/4}\sqrt{\log T})$.

As mentioned before, although the blocking technique is utilized to improve the expected regret of projection-free BCO, it is unsatisfying in practice due to the delayed reaction to the change of functions, and results in a logarithmical term in the high-probability regret bound. In this paper, we focus on the special case with strongly convex sets, and develop an improved variant of PF-BCO without using the blocking technique.

3 Main Results

In this section, we first introduce necessary preliminaries including basic definitions, common assumptions, and algorithmic ingredients. Then, we present our algorithm and its theoretical guarantees. Due to the limitation of space, we defer the proof of theoretical results to the supplementary material.[2]

3.1 Preliminaries

We first recall the standard definition for strongly convex sets [13].

Definition 1. *A convex set $\mathcal{K} \subseteq \mathbb{R}^d$ is called α_K-strongly convex with respect to a norm $\|\cdot\|$ if for any $\mathbf{x}, \mathbf{y} \in \mathcal{K}, \gamma \in [0, 1]$ and $\mathbf{z} \in \mathbb{R}^d$ such that $\|\mathbf{z}\| = 1$, it holds that*

$$\gamma\mathbf{x} + (1-\gamma)\mathbf{y} + \gamma(1-\gamma)\frac{\alpha_K}{2}\|\mathbf{x}-\mathbf{y}\|^2\mathbf{z} \in \mathcal{K}.$$

Next, we introduce three common assumptions in BCO [11].

Assumption 1. *The convex decision set $\mathcal{K}$ is full dimensional and contains the origin. Moreover, there exist two constants $r, R > 0$ such that*

$$r\mathcal{B}^d \subseteq \mathcal{K} \subseteq R\mathcal{B}^d$$

where $\mathcal{B}^d$ denotes the unit Euclidean ball centered at the origin in $\mathbb{R}^d$.

Assumption 2. *At each round t, the loss function $f_t(\mathbf{x})$ is G-Lipschitz over $\mathcal{K}$, i.e., for any $\mathbf{x}, \mathbf{y} \in \mathcal{K}$*

$$|f_t(\mathbf{x}) - f_t(\mathbf{y})| \leq G\|\mathbf{x}-\mathbf{y}\|_2.$$

Assumption 3. *At each round t, each loss function $f_t(\mathbf{x})$ is bounded over $\mathcal{K}$, i.e., for any $\mathbf{x} \in \mathcal{K}$*

$$|f_t(\mathbf{x})| \leq M. \tag{4}$$

Moreover, all loss functions are chosen beforehand, i.e., the adversary is oblivious.

Last, we recall the one-point gradient estimator [11], which is commonly utilized to deal with the bandit feedback. Specifically, for a function $f(\mathbf{x})$, we can define its δ-smooth version as

$$\widehat{f}_\delta(\mathbf{x}) = \mathbb{E}_{\mathbf{u}\sim\mathcal{B}^d}[f(\mathbf{x}+\delta\mathbf{u})] \tag{5}$$

which satisfies the following lemma.

Lemma 1 (Lemma 1 in Flaxman et al. [11]). *Let $\delta > 0$, $\widehat{f}_\delta(\mathbf{x})$ defined in (5) satisfies*

$$\nabla\widehat{f}_\delta(\mathbf{x}) = \mathbb{E}_{\mathbf{u}\sim\mathcal{S}^d}\left[\frac{d}{\delta}f(\mathbf{x}+\delta\mathbf{u})\mathbf{u}\right] \tag{6}$$

where $\mathcal{S}^d$ denotes the unit sphere in $\mathbb{R}^d$.

According to this lemma, the one-point gradient estimator is to make an unbiased estimation of $\nabla\widehat{f}_\delta(\mathbf{x})$ as $\frac{d}{\delta}f(\mathbf{x}+\delta\mathbf{u})\mathbf{u}$ by leveraging the single value $f(\mathbf{x}+\delta\mathbf{u})$.

[2] https://github.com/zcx-xxx/PAKDD-2024/blob/main/PAKDD-2024-Zhang-S.pdf.

Algorithm 1. Bandit Frank-Wolfe with Line Search

Input: $\mathcal{K}, \delta, \eta$
Initialization: $\mathbf{y}_1 \in \mathcal{K}_\delta$
for $t = 1, 2, \ldots, T$ **do**
  Play $\mathbf{x}_t = \mathbf{y}_t + \delta \mathbf{u}_t$, where $\mathbf{u}_t \sim \mathcal{S}^d$
  Observe $f_t(\mathbf{x}_t)$ and compute $\mathbf{g}_t$ according to (7)
  Construct $F_t(\mathbf{y})$ as in (8).
  Compute $\mathbf{v}_t = \operatorname*{argmin}_{\mathbf{y} \in \mathcal{K}_\delta} \langle \nabla F_t(\mathbf{y}_t), \mathbf{y} \rangle$
  Compute σ_t according to (10)
  $\mathbf{y}_{t+1} = \mathbf{y}_t + \sigma_t (\mathbf{v}_t - \mathbf{y}_t)$
end for

3.2 Our Proposed Algorithm

Now, we introduce our improved variant of PF-BCO, which is still a combination of the OFW algorithm and the one-point gradient estimator. Specifically, we first define a subset of the convex set $\mathcal{K}$ as

$$\mathcal{K}_\delta = (1 - \delta/r)\mathcal{K} = \{(1 - \delta/r)\mathbf{x} \mid \mathbf{x} \in \mathcal{K}\}$$

where $0 < \delta < r$ is a parameter. Following previous BCO algorithms [8,11,15], the decision $\mathbf{x}_t$ at each round t is divided into two parts, i.e.,

$$\mathbf{x}_t = \mathbf{y}_t + \delta \mathbf{u}_t$$

where $\mathbf{y}_t$ is an auxiliary decision learning from historical information and $\mathbf{u}_t$ is uniformly sampled from $\mathcal{S}^d$. Note that according to Assumption 1, it is easy to verify the feasibility of the above $\mathbf{x}_t$, i.e., $\mathbf{x}_t \in \mathcal{K}$. Moreover, in this way, the loss value of $f_t(\mathbf{x}_t) = f_t(\mathbf{y}_t + \delta \mathbf{u}_t)$ is observed at each round t. According to the one-point gradient estimator, it can be utilized to generate an approximate gradient as

$$\mathbf{g}_t = \frac{d}{\delta} f_t(\mathbf{y}_t + \delta \mathbf{u}_t)\mathbf{u}_t. \tag{7}$$

To further combine the OFW algorithm, we need to reconstruct the objective function in (1) as

$$F_t(\mathbf{y}) = \eta \sum_{\tau=1}^{t} \langle \mathbf{g}_\tau, \mathbf{y} \rangle + \|\mathbf{y} - \mathbf{y}_1\|_2^2 . \tag{8}$$

Then, similar to (2) in OFW, we update the auxiliary decision by minimizing $F_t(\mathbf{y})$ via a FW iteration over $\mathcal{K}_\delta$, i.e.,

$$\begin{aligned} \mathbf{v}_t &= \operatorname*{argmin}_{\mathbf{y} \in \mathcal{K}_\delta} \langle \nabla F_t(\mathbf{y}_t), \mathbf{y} \rangle \\ \mathbf{y}_{t+1} &= \mathbf{y}_t + \sigma_t (\mathbf{v}_t - \mathbf{y}_t) . \end{aligned} \tag{9}$$

We notice that the above procedures have been utilized in the PF-BCO algorithm [9]. However, they make use of a decaying step size, i.e., $\sigma_t = t^{-2/5}$, which cannot exploit the strong convexity of $\mathcal{K}$ to reduce the approximation error of the FW iteration. By contrast, inspired by (3) utilized in the full information setting [32], we employ a line search rule to select the step size as

$$\sigma_t = \operatorname*{argmin}_{\sigma \in [0,1]} \langle \sigma (\mathbf{v}_t - \mathbf{y}_t), \nabla F_t(\mathbf{y}_t) \rangle + \sigma^2 \|\mathbf{v}_t - \mathbf{y}_t\|_2^2 \tag{10}$$

which is able to make FW converge faster over the strongly convex set [13]. The detailed procedures of our algorithm are summarized in Algorithm 1, and it is named as bandit Frank-Wolfe with line search (BFW-LS).

3.3 Theoretical Guarantees

Next, we proceed to present the theoretical guarantees of our BFW-LS over strongly convex sets. Although in the full information setting, Wan and Zhang [32] have shown the advantage of the line search over the strongly convex set. It is worth noting that their result is not applicable in the bandit setting due to the following two challenges.

- In the full information setting, Wan and Zhang [32] directly utilize the faster convergence of FW over strongly convex $\mathcal{K}$. However, in the bandit setting, the FW iteration is performed over the shrunk set $\mathcal{K}_\delta$. It is unclear whether $\mathcal{K}_\delta$ is also strongly convex.
- In the bandit setting, the objective function in (8) is defined based on the estimated gradient, the variance of which makes it more difficult for us to minimize (8) with one FW iteration than the objective function (1) in the full information setting.

To address the above challenges, we first derive the following lemma, which implies that $\mathcal{K}_\delta$ is also strongly convex.

Lemma 2. *If $\mathcal{K}$ is α_K-strongly convex with respect to a norm $\|\cdot\|$, then $\mathcal{K}_\delta$ is $\frac{\alpha_K}{1-(\delta/r)}$-strongly convex with respect to the norm $\|\cdot\|$.*

Based on the above lemma, we establish an upper bound for the approximation error of minimizing the objective function in (8) over strongly convex sets.

Lemma 3. *Let $\mathcal{K}$ be an α_K-strongly convex set with respect to the ℓ_2 norm. Let $\mathbf{y}_t^* = \operatorname{argmin}_{\mathbf{y} \in \mathcal{K}_\delta} F_{t-1}(\mathbf{y})$ for any $t \in [T+1]$, where $F_t(\mathbf{y})$ is define in (8). Under Assumptions 1, 2, and 3, for any $t \in [T+1]$, Algorithm 1 with $\eta = \frac{cR}{dM(T+2)^{3/4}}$ and $\delta = cT^{-1/4}$ has*

$$F_{t-1}(\mathbf{y}_t) - F_{t-1}(\mathbf{y}_t^*) \leq \epsilon_t = \frac{C}{\sqrt{t+2}}$$

where $c > 0$ is a constant such that $\delta < r$ and $C = max\left(16R^2, \frac{4096}{3\alpha_K^2}\right)$.

Remark 1. We find that the approximation error of the FW iteration is upper bound by $O(t^{-1/2})$ for our algorithm over strongly convex sets. For a clear comparison, we note that this approximation error for PF-BCO over general convex sets has a worse bound of $O(t^{-2/5})$ (See the proof of Theorem 1 in Chen et al. [8]).

By applying Lemma 3, we prove that our method enjoys the following regret bound over strongly convex sets.

Theorem 1. *Let $\mathcal{K}$ be an α_K-strongly convex set with respect to the ℓ_2 norm and $C = max\left(16R^2, \frac{4096}{3\alpha_K^2}\right)$. Let $c > 0$ be a constant such that $\delta = cT^{-1/4} \le r$. Under Assumptions 1, 2, and 3, Algorithm 1 with $\eta = \frac{cR}{dM(T+2)^{3/4}}$ and $\delta = cT^{-1/4}$ ensures*

$$\mathbb{E}[\text{Regret}(T)] \le \frac{4RdM(T+2)^{3/4}}{c} + \frac{RdMT^{3/4}}{c} + 3cGT^{3/4} + \frac{4\sqrt{C}G(T+2)^{3/4}}{3} + \frac{cGRT^{3/4}}{r}.$$

Remark 2. From Theorem 1, our Algorithm 1 achieves an expected regret bound of $O(T^{3/4})$ over strongly convex sets. Note that this bound is the same as the expected regret bound of BBCG [15] and LOO-BBGD [16]. Although our result does not hold in general convex case like their results, we dismiss the compromising blocking technique required by them.

Furthermore, although Theorem 1 has provided an expected regret bound, one may still wonder whether this bound can hold at most of the time. For this reason, we also establish a high-probability regret bound for Algorithm 1.

Theorem 2. *Let $\mathcal{K}$ be an α_K-strongly convex set with respect to the ℓ_2 norm and $C = max\left(16R^2, \frac{4096}{3\alpha_K^2}\right)$. Let $c > 0$ be a constant such that $\delta = cT^{-1/4} \le r$ and $\eta = \frac{cR}{dM(T+2)^{3/4}}$. Under Assumption 1, 2, and 3, with probability at least $1 - \gamma$, Algorithm 1 ensures*

$$\text{Regret}(T) \le 2RG\sqrt{2\ln\frac{1}{\gamma}}T^{1/2} + \frac{2RdM}{c}\sqrt{2\ln\frac{1}{\gamma}}T^{3/4} + \frac{4RdM(T+2)^{3/4}}{c} + \frac{RdMT^{3/4}}{c} + \frac{4\sqrt{C}G(T+2)^{3/4}}{3} + \frac{cGRT^{3/4}}{r} + 3cGT^{3/4}.$$

Remark 3. Theorem 2 implies that our algorithm also enjoys a high-probability regret bound of $O(T^{3/4})$ over strongly convex sets. It is worth noting that it removes the logarithmic factor in the $O(T^{3/4}\sqrt{\log T})$ high-probability regret bound achieved by BBCG [15,29], which demonstrate a theoretical advantage of removing the blocking technique.

4 Experiments

In this section, we present experimental results on synthetic and real-world data. All experiments are conducted on a Linux machine with 2.3 GHz CPU and 125 GB RAM.

4.1 Problem Settings

We first consider the problem of online quadratic programming (OQP) with synthetic data [9], where the total number of iterations is set as $T = 40000$, and the dimensionality is set as $d = 10$. At each iteration $t \in [T]$, the learner first selects $\mathbf{x}_t \in \mathcal{K}$, and then suffers a quadratic loss

$$f_t(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{G}_t^\top \mathbf{G}_t \mathbf{x} + \mathbf{w}_t^\top \mathbf{x}$$

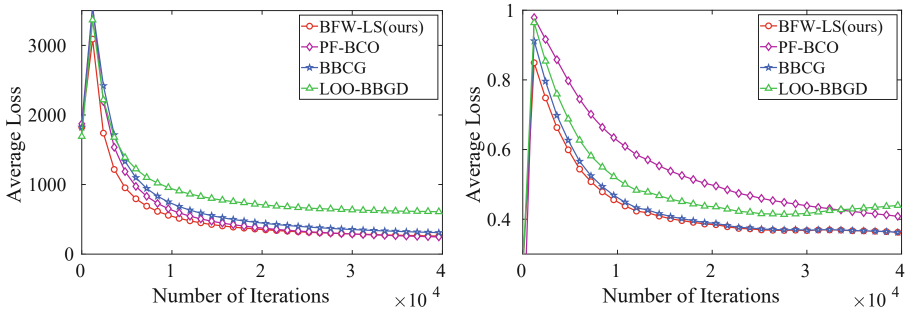
where each element of $\mathbf{G}_t \in \mathbb{R}^{d\times d}$ and $\mathbf{w}_t \in \mathbb{R}^d$ is sampled from the standard normal distribution. Second, we consider the problem of online binary classification (OBC) with a real-world dataset ijcnn1 [7], which consists of 49990 instances and each instance having 22 features, i.e., $d = 22$. To make the block size K of BBCG and LOO-BBGD be an integer, we randomly select $T = 40000$ instances from the original dataset. At each round t, the learner receives a single example $\mathbf{e}_t \in \mathbb{R}^d$ and chooses a decision $\mathbf{x}_t \in \mathcal{K}$. Then, the true class label $y_t \in \{-1, 1\}$ is revealed, and the learner suffers the hinge loss

$$f_t(\mathbf{x}_t) = \max\left\{1 - y_t \mathbf{e}_t^\top \mathbf{x}_t, 0\right\}.$$

More specifically, we set $\mathcal{K} = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_p \le \tau\}$ with $p = 1.5, \tau = 50$ for the OQP experiment and $p = 1.5, \tau = 30$ for the OBC experiment. One can verify that this set is strongly convex for any $p \in (1, 2)$ [13], and satisfies Assumption 1 with $r = \frac{\tau}{d^{1/p-1/2}}$ and $R = \tau$.

4.2 Experimental Results

We compare our BFW-LS against existing projection-free BCO algorithms including PF-BCO [9], BBCG [15], and LOO-BBGD [16]. The parameters of all algorithms are set according to what their corresponding theories suggest. Specifically, all of them depend on two parameters η and δ, which are set as $\eta = c_1 T^{-3/4}$ and $\delta = c_2 T^{-1/4}$ for BBCG, LLO-BBGD, and our BFW-LS, and $\eta = c_1 T^{-4/5}, \delta = c_2 T^{-1/5}$ for PF-BCO. The constants c_1 and c_2 are selected from $\{1e-2, 1e-1, \ldots, 1e3\}$ and $\{20, 40, \ldots, 120, 140\}$, respectively. Moreover, BBCG adopts two additional parameters K and ϵ to control the block size and the error tolerance in each block, which are set as $\epsilon = 16R^2T^{-1/2}$ and $K = \sqrt{T}$. The same block size is also utilized in LOO-BBGD. Figure 1a and Fig. 1b show the average loss of each algorithm, i.e., $\frac{1}{t}\sum_{i=1}^{t} f_i(\mathbf{x}_i)$ at each iteration t, on the OQP and OBC experiments, respectively. We find that the average loss of our BFW-LS is lower than that of these baselines. Additionally, although the regret



(a) OQP with synthetic data (b) OBC with ijcnn1

Fig. 1. Average Loss of Our and Previous Projection-free BCO Algorithms.

bound of BBCG and LOO-BBGD is better than that of PF-BCO, they fail to exhibit significantly better performance in comparison to PF-BCO, especially on the OQP experiment. This phenomenon partially suggests the shortcoming of the blocking technique.

5 Conclusion

In this paper, we propose a projection-free algorithm called BFW-LS for BCO, which achieves an $O(T^{3/4})$ expected regret bound over strongly convex sets, without using the blocking technique required by previous projection-free BCO algorithms with the same expected bound. Furthermore, we also show that our BFW-LS has a high-probability regret bound of $O(T^{3/4})$, which removes the logarithmic factor in the $O(T^{3/4}\sqrt{\log T})$ high-probability regret bound achieved by previous projection-free BCO algorithms. Finally, experimental results verify the advantage of our BFW-LS.

Acknowledgments. This work was supported by State Grid science and technology project (5700-202327286A-1-1-ZN).

Disclosure of Interests. The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Abernethy, J., Hazan, E., Rakhlin, A.: Competing in the dark: an efficient algorithm for bandit linear optimization. In: Proceedings of the 21st Conference on Learning Theory, pp. 263–274 (2008)
2. Agarwal, A., Dekel, O., Xiao, L.: Optimal algorithms for online convex optimization with multi-point bandit feedback. In: Proceedings of the 23rd Conference on Learning Theory, pp. 28–40 (2010)

3. Awerbuch, B., Kleinberg, R.: Online linear optimization and adaptive routing. J. Comput. Syst. Sci. **74**(1), 97–114 (2008)
4. Bubeck, S., Cesa-Bianchi, N., Kakade, S.M.: Towards minimax policies for online linear optimization with bandit feedback. In: Proceedings of the 25th Conference on Learning Theory, pp. 41.1–41.14 (2012)
5. Bubeck, S., Dekel, O., Koren, T., Peres, Y.: Bandit convex optimization: $\sqrt{t}$ regret in one dimension. In: Proceedings of the 28th Conference on Learning Theory, pp. 266–278 (2015)
6. Bubeck, S., Eldan, R., Lee, Y.T.: Kernel-based methods for bandit convex optimization. In: Proceedings of the 49th Annual ACM Symposium on Theory of Computing, pp. 72–85 (2019)
7. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(27), 1–27 (2011)
8. Chen, L., Harshaw, C., Hassani, H., Karbasi, A.: Projection-free online optimization with stochastic gradient: from convexity to submodularity. In: Proceedings of the 35th International Conference on Machine Learning, pp. 814–823 (2018)
9. Chen, L., Zhang, M., Karbasi, A.: Projection-free bandit convex optimization. In: Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, pp. 2047–2056 (2019)
10. Dekel, O., Eldan, R., Koren, T.: Bandit smooth convex optimization: Improving the bias-variance tradeoff. In: Advances in Neural Information Processing Systems 28, pp. 2926–2934 (2015)
11. Flaxman, A.D., Kalai, A.T., McMahan, H.B.: Online convex optimization in the bandit setting: gradient descent without a gradient. In: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 385–394 (2005)
12. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Res. Logistics Quart. **3**(1–2), 95–110 (1956)
13. Garber, D., Hazan, E.: Faster rates for the frank-wolfe method over strongly-convex sets. In: Proceedings of the 32nd International Conference on Machine Learning, pp. 541–549 (2015)
14. Garber, D., Hazan, E.: A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. SIAM J. Optim. **26**(3), 1493–1528 (2016)
15. Garber, D., Kretzu, B.: Improved regret bounds for projection-free bandit convex optimization. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, pp. 2196–2206 (2020)
16. Garber, D., Kretzu, B.: New projection-free algorithms for online convex optimization with adaptive regret guarantees. In: Proceedings of the 35th Conference on Learning Theory, pp. 2326–2359 (2022)
17. Garber, D., Kretzu, B.: Projection-free online exp-concave optimization. In: Proceedings of the 36th Conference on Learning Theory (2023)
18. Hazan, E.: Introduction to online convex optimization. Found. Trends Optim. **2**(3–4), 157–325 (2016)
19. Hazan, E., Kale, S.: Projection-free online learning. In: Proceedings of the 29th International Conference on Machine Learning, pp. 1843–1850 (2012)
20. Hazan, E., Levy, K.Y.: Bandit convex optimization: towards tight bounds. In: Advances in Neural Information Processing Systems 27, pp. 784–792 (2014)
21. Hazan, E., Minasyan, E.: Faster projection-free online learning. In: Proceedings of the 33rd Conference on Learning Theory, pp. 1877–1893 (2020)

22. Ito, S.: An optimal algorithm for bandit convex optimization with strongly-convex and smooth loss. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, pp. 2229–2239 (2020)
23. Jaggi, M.: Revisiting Frank-Wolfe: projection-free sparse convex optimization. In: Proceedings of the 30th International Conference on Machine Learning, pp. 427–435 (2013)
24. Kalhan, D.S., et al.: Dynamic online learning via frank-wolfe algorithm. IEEE Trans. Signal Process. **69**, 932–947 (2021)
25. Kretzu, B., Garber, D.: Revisiting projection-free online learning: the strongly convex case. In: Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, pp. 3592–3600 (2021)
26. McMahan, H.B., et al.: Ad click prediction: a view from the trenches. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1222–1230 (2013)
27. Saha, A., Tewari, A.: Improved regret guarantees for online smooth convex optimization with bandit feedback. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, pp. 636–642 (2011)
28. Shalev-Shwartz, S., Singer, Y.: A primal-dual perspective of online learning algorithm. Mach. Learn. **69**(2–3), 115–142 (2007)
29. Wan, Y., Tu, W.W., Zhang, L.: Projection-free distributed online convex optimization with $\mathcal{O}(\sqrt{T})$ communication complexity. In: Proceedings of the 37th International Conference on Machine Learning, pp. 9818–9828 (2020)
30. Wan, Y., Wang, G., Tu, W.W., Zhang, L.: Projection-free distributed online learning with sublinear communication complexity. J. Mach. Learn. Res. **23**(172), 1–53 (2022)
31. Wan, Y., Xue, B., Zhang, L.: Projection-free online learning in dynamic environments. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence Advances, pp. 10067–10075 (2021)
32. Wan, Y., Zhang, L.: Projection-free online learning over strongly convex set. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence Advances, pp. 10076–10084 (2021)
33. Wan, Y., Zhang, L., Song, M.: Improved dynamic regret for online frank-wolfe. In: Proceedings of the 36th Conference on Learning Theory (2023)
34. Wang, Y., Wan, Y., Zhang, S., Zhang, L.: Distributed projection-free online learning for smooth and convex losses. In: Proceedings of the 37th AAAI Conference on Artificial Intelligence, pp. 10226–10234 (2023)
35. Wang, Y., et al.: Non-stationary projection-free online learning with dynamic and adaptive regret guarantees. ArXiv e-prints arXiv:2305.11726 (2023)
36. Zinkevich, M.: Online convex programming and generalized infinitesimal gradient ascent. In: Proceedings of the 20th International Conference on Machine Learning, pp. 928–936 (2003)