Online Frank-Wolfe with Arbitrary Delays

Yuanyu Wan[1,2], Wei-Wei Tu[3,4], Lijun Zhang[4,*]

[1]School of Software Technology, Zhejiang University, Ningbo, China

[2]Zhejiang University-China Southern Power Grid Joint Research Centre on AI, Hangzhou, China

[3]4Paradigm Inc., Beijing, China

[4]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

wanyy@zju.edu.cn, tuweiwei@4paradigm.com, zhanglj@lamda.nju.edu.cn

Abstract

The online Frank-Wolfe (OFW) method has gained much popularity for online convex optimization due to its projection-free property. Previous studies show that OFW can attain an $O(T^{3/4})$ regret bound for convex losses and an $O(T^{2/3})$ regret bound for strongly convex losses. However, they assume that each gradient queried by OFW is revealed immediately, which may not hold in practice and limits the application of OFW. To address this limitation, we propose a delayed variant of OFW, which allows gradients to be delayed by arbitrary rounds. The main idea is to perform an update similar to OFW after receiving any delayed gradient, and play the latest decision for each round. Despite its simplicity, we prove that our delayed variant of OFW is able to achieve an $O(T^{3/4}+dT^{1/4})$ regret bound for convex losses and an $O(T^{2/3}+d\log T)$ regret bound for strongly convex losses, where d is the maximum delay. This is quite surprising since under a relatively large amount of delay (e.g., $d=O(\sqrt{T})$ for convex losses and $d=O(T^{2/3}/\log T)$ for strongly convex losses), the delayed variant of OFW enjoys the same regret bound as that of the original OFW.

1 Introduction

Online convex optimization (OCO) has become a leading paradigm for online learning due to its capability to model various problems from diverse domains such as online routing, online collaborative filtering, and online advertisement [Hazan, 2016]. In general, it is formulated as a structured repeated game between a player and an adversary. In each round t, the player first chooses a decision $\mathbf{x}_t$ from a convex decision set $\mathcal{K}\subseteq\mathbb{R}^n$, where n is the dimensionality. Then, the adversary selects a convex function $f_t(\mathbf{x}):\mathcal{K}\mapsto\mathbb{R}$, and the player suffers a loss $f_t(\mathbf{x}_t)$. The player aims to choose decisions such that the regret $R(T)=\sum_{t=1}^{T}f_t(\mathbf{x}_t)-\min_{\mathbf{x}\in\mathcal{K}}\sum_{t=1}^{T}f_t(\mathbf{x})$ is sublinear in the number of total rounds T. Online gradient descent (OGD) is a standard method for OCO, which enjoys an $O(\sqrt{T})$ regret bound for convex losses [Zinkevich, 2003] and an $O(\log T)$ regret bound for strongly convex losses [Hazan et al., 2007]. However, it needs to compute a projection onto the decision set to ensure the feasibility of each decision, which is computationally expensive for complex decision sets [Hazan and Kale, 2012].

To tackle this computational issue, Hazan and Kale [2012] propose the online Frank-Wolfe (OFW) method, which has become one of the most commonly used algorithms for OCO over complex decision sets. The main advantage of OFW is its projection-free property: instead of performing the projection operation, it utilizes a linear optimization step to select a feasible decision, which could be much more efficient. For example, in the problem of online collaborative filtering, the decision set

*Lijun Zhang is the corresponding author.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

consists of matrices with a bounded trace norm, and the linear optimization step is at least an order of magnitude faster than the projection operation [Hazan and Kale, 2012]. Moreover, it has been shown that OFW achieves an $O(T^{3/4})$ regret bound for convex losses [Hazan and Kale, 2012, Hazan, 2016] and an $O(T^{2/3})$ regret bound for strongly convex losses [Wan and Zhang, 2021, Garber and Kretzu, 2021], which are the best known regret bounds of projection-free methods without further assumptions.

However, OFW requires that the gradient $\nabla f_t(x_t)$ is revealed immediately after making the decision x_t, which is not necessarily satisfied in reality. For example, in the previously mentioned online collaborative filtering [Hazan and Kale, 2012], the decision is a prediction of a user-item rating matrix, and the corresponding gradient depends on the true rating of a user on a item, which may not be decided by the user immediately. Therefore, it is natural to consider a more practical setting, where the gradient $\nabla f_t(x_t)$ arrives at the end of round $t + d_t - 1$, and $d_t \geq 1$ denotes an arbitrary delay. To handle this setting, one potential way is to combine OFW with an existing black-box technique for converting any traditional OCO algorithm into this delayed setting [Joulani et al., 2013]. To be precise, this black-box technique is to pool independent instances of OFW, each of which acts as a learner in the non-delayed setting over a subsequence of rounds. In each round, a single instance will be taken out from the pool, which makes a decision and then waits for its feedback before rejoining the pool. If the pool is empty, a new instance of OFW will be added to it. Moreover, according to Joulani et al. [2013], their black-box technique is able to attain a regret bound of $dR(T/d)$ by combining with a traditional OCO algorithm with $R(T)$ regret, where d is the maximum delay. As a result, combining it with OFW will attain an $O(d^{1/4}T^{3/4})$ regret bound for convex losses and an $O(d^{1/3}T^{2/3})$ regret bound for strongly convex losses, which magnify the regret bounds of OFW in the non-delayed setting by a coefficient depending the delay. Thus, it is natural to ask whether the effect of delay can be further reduced.

In this paper, we give an affirmative answer by developing a simple method called delayed OFW, which is robust to a relatively large amount of delay for both convex and strongly convex losses. Different from the black-box technique that needs to maintain multiple instances of OFW [Joulani et al., 2013], our delayed OFW is a natural extension of OFW in the delayed setting, which updates the decision similar to OFW after receiving any delayed gradient, and plays the latest decision for each round. Our theoretical contributions are summarized as follows.

- First, we prove that our delayed OFW attains an $O(T^{3/4} + dT^{1/4})$ regret bound for convex losses, where d is the maximum delay, which matches the $O(T^{3/4})$ regret bound in the non-delayed setting as long as d does not exceed $O(\sqrt{T})$.
- Second, we prove that our delayed OFW attains an $O(T^{2/3} + d\log T)$ regret bound for strongly convex losses, which matches the $O(T^{2/3})$ regret bound in the non-delayed setting as long as d does not exceed $O(T^{2/3}/\log T)$.

Therefore, our regret bounds are strictly better than those achieved by combining the black-box technique [Joulani et al., 2013] with OFW, when the term involving d in them is not dominant. Furthermore, simulation experiments are conducted to verify the performance of our delayed OFW.

2 Related work

In this section, we briefly review related work on projection-free algorithms for OCO, and OCO under delayed feedback.

2.1 Projection-free algorithms for OCO

The OFW method [Hazan and Kale, 2012, Hazan, 2016] is the first projection-free algorithm for OCO, which is an online extension of the classical Frank-Wolfe method [Frank and Wolfe, 1956, Jaggi, 2013]. For convex losses, OFW first chooses an arbitrary $x_1 \in \mathcal{K}$, and then iteratively updates its decision by the following linear optimization step

$$v_t \in \operatorname*{argmin}_{x \in \mathcal{K}} \langle \nabla F_t(x_t), x \rangle, \quad x_{t+1} = x_t + \sigma_t(v_t - x_t) \tag{1}$$

where $F_t(x)$ is a surrogate loss function defined as

$$F_t(x) = \sum_{i=1}^{t} \langle \nabla f_i(x_i), x \rangle + \|x - x_1\|_2^2 \tag{2}$$

and ; $_t$ are two parameters. By setting parameters appropriately, it can attain an $O(T^{3/4})$ regret bound for convex losses.

If losses are convex and smooth, Hazan and Minasyan [2020] propose a randomized projection-free method, which is based on a classical OCO method called follow the perturbed leader [Kalai and Vempala, 2005], and achieve a regret bound of $O(T^{2/3})$. Recently, Wan and Zhang [2021] prove that OFW can achieve an $O(T^{2/3})$ regret bound for strongly convex losses. Specifically, to utilize the strong convexity of losses, they redefine $F_t(x)$ in (2) to

$$F_t(x) = \sum_{i=1}^{t} \quad x_1\|_4)$$

in Weinberger and Ordentlich [2002], it also needs to run multiple instances of a traditional OCO algorithm, which could be prohibitively resource-intensive. Many studies [Quanrud and Khashabi, 2015, Joulani et al., 2016, Li et al., 2019, Flaspohler et al., 2021, Wan et al., 2022a] have proposed delayed OCO algorithms, which only require the same storage and computational resources as in the non-delayed setting, but do not consider projection-free algorithms.

3 Main results

In this section, we first introduce necessary preliminaries including the problem setting, definitions, and assumptions. Then, we present our delayed OFW and the corresponding theoretical guarantees for convex and strongly convex losses, respectively.

3.1 Preliminaries

We consider the problem of OCO with arbitrary delays [Joulani et al., 2013, Quanrud and Khashabi, 2015]. Similar to the standard OCO, in each round $t = 1, \ldots, T$, the player first chooses a decision x_t from the decision set $\mathcal{K}$, and then the adversary selects a convex function $f_t(x)$. However, different from the standard OCO, the gradient $g_t = \nabla f_t(x_t)$ is revealed at the end of round $t + d_t - 1$, where $d_t \geq 1$

Algorithm 1 Delayed OFW for Convex Losses

```
Input:
Initialization: choose an arbitrary vector y_1 ∈ K and set τ = 1; ḡ_0 = 0
for t = 1, 2, ..., T do
    Play x_t = y_τ and query g_t = ∇f_t(x_t)
    Receive a set of delayed gradients {g_k | k ∈ F_t}
    for k ∈ F_t do
        Update ḡ_τ = ḡ_{τ-1} + g_k and define F_τ(y) = η⟨ḡ_τ, y⟩ + ‖y − y_1‖_2^2
        Compute v_τ ∈ argmin_{y∈K} ⟨∇F_τ(y_τ), y⟩
        Update y_{τ+1} = y_τ + σ_τ(v_τ − y_τ) with σ_τ in (4) and set τ = τ + 1
    end for
end for
```

Finally, we update $\tau = \tau + 1$ so that τ still indexes the latest intermediate decision.

The detailed procedures are summarized in Algorithm 1, which is named as delayed OFW for convex losses. Let $d = \max\{d_1, \ldots, d_T\}$. We establish the following theorem with respect to the regret of Algorithm 1.

Theorem 1 For any $x^* \in \mathcal{K}$, under Assumptions 1 and 2, Algorithm 1 with $\eta = \frac{D}{\sqrt{2}G(T+2)^{3/4}}$ has

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) = O(T^{3/4} + dT^{1/4}).$$

Theorem 1 shows that without knowing the value of d, our Algorithm 1 can attain an $O(T^{3/4} + dT^{1/4})$ regret bound for convex losses with arbitrary delays. This bound matches the $O(T^{3/4})$ regret bound of OFW in the non-delayed setting [Hazan, 2016], as long as d does not exceed $O(\sqrt{T})$. Moreover, it is better than the $O(d^{1/4}T^{3/4})$ regret bound achieved by combining the technique of Joulani et al. [2013] and the $O(T^{3/4})$ regret bound of OFW for convex losses, as long as d does not exceed $O(T^{2/3})$.

3.3 Delayed OFW for strongly convex losses

We proceed to handle α-strongly convex losses by slightly modifying Algorithm 1. Recall that in the standard OCO without delays, the critical idea of utilizing the strong convexity of losses is to replace the surrogate loss function in (1) by that in (3) [Wan and Zhang, 2021]. The main difference is that the regularization term in (3) is about all historical decisions, instead of only the initial decision.

Inspired by (3), we first redefine $F_\tau(y)$ in Algorithm 1 to

$$F_\tau(y) = \langle \bar{g}_\tau, y\rangle + \sum_{i=1}^{\tau} \frac{\alpha}{2}\|y - y_i\|_2^2.$$

Second, since $F_\tau(y)$ is modified, we adjust the line search rule to

$$\sigma_\tau = \mathop{\arg\min}_{\sigma\in[0,1]} \langle \sigma(v_\tau - y_\tau), \nabla F_\tau(y_\tau)\rangle + \frac{\sigma^2\alpha\tau}{2}\|v_\tau - y_\tau\|_2^2. \tag{5}$$

The detailed procedures are summarized in Algorithm 2, which is named as delayed OFW for strongly convex losses. Then, we establish the following theorem about the regret of Algorithm 2.

Theorem 2 Suppose all losses are α-strongly convex and Assumptions 1 and 2 hold. For any $x^* \in \mathcal{K}$, Algorithm 2 has

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) = O(T^{2/3} + d\log T).$$

Theorem 2 shows that our Algorithm 2 can attain an $O(T^{2/3} + d\log T)$ regret bound for strongly convex losses with arbitrary delays. First, this bound is better than the $O(T^{3/4} + dT^{1/4})$ regret bound

Algorithm 2 Delayed OFW for Strongly Convex Losses

Input:
Initialization: choose an arbitrary vector $y_1 \in \mathcal{K}$ and set $= 1; g_0 = 0$
for $t = 1, 2, \ldots, T$ **do**
  Play $x_t = y$ and query $g_t = \nabla f_t(x_t)$
  Receive a set of delayed gradients $\{g_k | k \in \mathcal{F}_t\}$
  for $k \in \mathcal{F}_t$ **do**
    Update $g = g_{-1} + g_k$ and define $F(y) = \langle g, y \rangle + \sum_{i=1} \frac{}{2} \|y - y_i\|_2^2$
    Compute $v \in \operatorname{argmin}_{y \in \mathcal{K}} \langle \nabla F(y), y \rangle$
    Update $y_{+1} = y + (v - y)$ with in (5) and set $= + 1$
  end for
end for

in Theorem 1, which is established by only using the convexity condition. Second, it matches the O(Tend f 0 6 w660.9626 yh29626 y(xi6-2507e1)23 7.876 7.02 0 0 1 1ndition.the

By using this notation, $F_\tau(y)$ defined in Algorithms 1 and 2 are respectively equivalent to

$$F_\tau(y) = \eta\sum_{i=1}^{\tau}\langle g_{c_i}, y\rangle + \|y - y_1\|_2^2; \quad (11)$$

$$F_\tau(y) = \sum_{i=1}^{\tau}\langle g_{c_i}, y\rangle + \sum_{i=1}^{\tau}\frac{\alpha}{2}\|y - y_i\|_2^2. \quad (12)$$

4.2 Proof of Theorem 1

Let $t' = t + d_t - 1$ for any $t \in [T]$. According to the convexity of $f_t(x)$, we have

$$\begin{aligned} f_t(x_t) - f_t(x^*) &\le \langle g_t, x_t - x^*\rangle = \langle g_t, x_{t'} - x^*\rangle + \langle g_t, x_t - x_{t'}\rangle \\ &\le \langle g_t, x_{t'} - x^*\rangle + G\|x_t - x_{t'}\|_2 = \langle g_t, y_{\tau_{t'}} - x^*\rangle + G\|y_{\tau_t} - y_{\tau_{t'}}\|_2 \end{aligned}$$

where the last equality is due to (6).

Let $A = \sum_{t=1}^{T} G\|y_{\tau_t} - y_{\tau_{t'}}\|_2$. By summing over $t = 1, \ldots, T$, we have

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) \le \sum_{t=1}^{T}\langle g_t, x_t - x^*\rangle \le \sum_{t=1}^{T}\langle g_t, y_{\tau_{t'}} - x^*\rangle + A. \quad (13)$$

Then, we bound the first term in the right side of (13) as follows

$$\begin{aligned} \sum_{t=1}^{T}\langle g_t, y_{\tau_{t'}} - x^*\rangle &= \sum_{t=s}^{T+d-1}\sum_{i\in\mathcal{F}_t}\langle g_i, y_{\tau_{i+d_i-1}} - x^*\rangle \\ &= \sum_{t=s}^{T+d-1}\sum_{i\in\mathcal{F}_t}\langle g_i, y_{\tau_t} - x^*\rangle = \sum_{t=s}^{T+d-1}\sum_{i=\tau_t}^{\tau_{t+1}-1}\langle g_{c_i}, y_{\tau_t} - x^*\rangle \\ &= \sum_{t=s}^{T+d-1}\sum_{i=\tau_t}^{\tau_{t+1}-1}\left(\langle g_{c_i}, y_i - x^*\rangle + \langle g_{c_i}, y_{\tau_t} - y_i\rangle\right) \\ &= \sum_{t=1}^{T}\langle g_{c_t}, y_t - x^*\rangle + \sum_{t=s}^{T+d-1}\sum_{i=\tau_t}^{\tau_{t+1}-1}\langle g_{c_i}, y_{\tau_t} - y_i\rangle \end{aligned} \quad (14)$$

where the first equality is due to (9), the second equality is due to $i + d_i - 1 = t$ for any $i \in \mathcal{F}_t$, the third equality is due to (10), and the last equality is due to (7) and (8).

Then, let $B = \sum_{t=1}^{T}\langle g_{c_t}, y_t - x^*\rangle$ and $C = \sum_{t=s}^{T+d-1}\sum_{i=\tau_t}^{\tau_{t+1}-1} G\|y_{\tau_t} - y_i\|_2$. By combining (13), (14), and $\langle g_{c_i}, y_{\tau_t} - y_i\rangle \le G\|y_{\tau_t} - y_i\|_2$, we have

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) \le A + B + C. \quad (15)$$

Next, we proceed to bound terms A, B, and C. Specifically, we first establish the following bound for the sum of terms A and C by carefully analyzing the distance $\|y_{\tau_t} - y_{\tau_{t'}}\|_2$ in the term A and the distance $\|y_{\tau_t} - y_i\|_2$ in the term C.

Lemma 1 Let $y_t^* = \operatorname{argmin}_{y\in\mathcal{K}} F_{t-1}(y)$ for any $t \in [T+1]$, where $F_t(y)$ is defined in (11). Suppose Assumption 1 and 2 hold, and there exist some constants $\gamma > 0$ and $0 < \alpha \le 1$ such that $F_{t-1}(y_t) - F_{t-1}(y_t^*) \le \gamma(t+2)^{-\alpha}$ for any $t \in [T+1]$. Algorithm 1 ensures

$$A + C \le 3dGD + 4Gd\sqrt{\gamma} + \frac{8G\sqrt{\gamma}}{2-\alpha}T^{1-\alpha/2} + \frac{3\eta G^2 dT}{2}$$

where $A = \sum_{t=1}^{T} G\|y_{\tau_t} - y_{\tau_{t'}}\|_2$ and $C = \sum_{t=s}^{T+d-1}\sum_{i=\tau_t}^{\tau_{t+1}-1} G\|y_{\tau_t} - y_i\|_2$.

Note that Lemma 1 introduces an assumption about y_t and $F_{t-1}(y)$. According to our Algorithm 1, y_t is actually generated by approximately minimizing $F_{t-1}(y)$ with a linear optimization step. Therefore, by following the analysis of the original OFW [Hazan, 2016], we show that this assumption can be satisfied with $\gamma = 8D^2$ and $\alpha = 1/2$.

Lemma 2 Let $y_t^* = \arg\min_{y\in\mathcal{K}} F_{t-1}(y)$ for any $t \in [T+1]$, where $F_t(y)$ is defined in (11). Under Assumptions 1 and 2, for any $t\in[T+1]$, Algorithm 1 with $\eta = \frac{D}{\sqrt{2}G(T+2)^{3/4}}$ has

$$F_{t-1}(y_t) - F_{t-1}(y_t^*) \le \frac{8D^2}{\eta\sqrt{t+2}}.$$

Then, by combining $\eta = \frac{D}{\sqrt{2}G(T+2)^{3/4}}$ with Lemmas 1 and 2, we have

$$A + C \le (3+8\sqrt{2})GDd + \frac{32\sqrt{2}GD}{3}T^{3/4} + \frac{3GDdT^{1/4}}{2\sqrt{2}} = O(T^{3/4} + dT^{1/4}). \quad (16)$$

Furthermore, by following the analysis of the original OFW [Hazan, 2016], we establish an upper bound for the term B.

Lemma 3 Under Assumptions 1 and 2, for any $x^*\in\mathcal{K}$, Algorithm 1 with $\eta = \frac{D}{\sqrt{2}G(T+2)^{3/4}}$ ensures

$$\sum_{t=1}^{T}\langle g_{c_t}, y_t - x^*\rangle \le \frac{11\sqrt{2}GD(T+2)^{3/4}}{3} + \frac{GDT^{1/4}}{\sqrt{2}}.$$

Finally, by combining (15), (16), and Lemma 3, we complete this proof.

4.3 Proof of Theorem 2

Since $f_t(x)$ is α-strongly convex, we have

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) \le \sum_{t=1}^{T}\langle g_t, x_t - x^*\rangle - \sum_{t=1}^{T}\frac{\alpha}{2}\|x_t - x^*\|_2^2. \quad (17)$$

Then, we note that the first term in the right side of (17) can be bounded by reusing (13) and (14). Specifically, we have

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) \le A + C + \sum_{t=1}^{T}\langle g_{c_t}, y_t - x^*\rangle - \sum_{t=1}^{T}\frac{\alpha}{2}\|x_t - x^*\|_2^2 \quad (18)$$

where terms A and C are defined in the proof of Theorem 1.

Next, we consider the last term in the right side of (18). For any $y_t, x_t, x^*\in\mathcal{K}$, we have

$$\begin{aligned}\|y_t - x^*\|_2^2 &= \|y_t - x_t\|_2^2 + \|x_t - x^*\|_2^2 + 2\langle y_t - x_t, x_t - x^*\rangle \\ &\le 3D\|y_t - x_t\|_2 + \|x_t - x^*\|_2^2\end{aligned}$$

where the last inequality is due to $2\langle y_t - x_t, x_t - x^*\rangle \le 2\|y_t - x_t\|_2\|x_t - x^*\|_2$ and Assumption 2. Let $B' = \sum_{t=1}^{T}\langle g_{c_t}, y_t - x^*\rangle - \sum_{t=1}^{T}\frac{\alpha}{2}\|y_t - x^*\|_2^2$ and $E = \sum_{t=1}^{T}\frac{3\alpha D}{2}\|y_t - y_t^*\|_2$. By combining the above inequality and (6) with (18), we have

$$\sum_{t=1}^{T}[f_t(x_t) - f_t(x^*)] \le A + C + B' + E. \quad (19)$$

Then, we proceed to establish upper bounds for terms A, C, and E by carefully analyzing the distance $\|y_t^* - y_{t'}^*\|_2$ in the term A, the distance $\|y_t^* - y_i\|_2$ in the term C, and the distance $\|y_t - y_t^*\|_2$ in the term E.

Lemma 4 Let $y_t^* = \arg\min_{y\in\mathcal{K}} F_{t-1}(y)$ for any $t = 2, \ldots, T+1$, where $F_t(y)$ is defined in (12). Suppose Assumption 1 and 2 hold, all losses are α-strongly convex, and there exist some constants $\gamma > 0$ and $0 \le \beta < 1$ such that $F_{t-1}(y_t) - F_{t-1}(y_t^*) \le \gamma(t-1)^{\beta}$ for any $t = 2, \ldots, T+1$. Algorithm 1 ensures

$$E \le 3dD\sqrt{\frac{2\gamma}{\alpha}} + \frac{6D\sqrt{2\gamma\alpha}}{1+\beta}T^{(1+\beta)/2} + 3\alpha dD^2 + 3D(G+\alpha D)d\ln T;$$

$$A + C \le 3dGD + \frac{4G(G+\alpha D)d(1+\ln T)}{\alpha} + 4dG\sqrt{\frac{2\gamma}{\alpha}} + \sqrt{\frac{2\gamma}{\alpha}}\frac{8G}{1+\beta}T^{(1+\beta)/2};$$

where $A = \sum_{t=1}^{T} G\|y_{t} \;\; y_{t^0}\|_2$, $C = \sum_{t=s}^{T+d \;\; 1} \sum_{i=\;\; t}^{t+1 \;\; 1} G\|y_{t} \;\; y_i\|_2$, and $E = \sum_{t=1}^{T} \frac{3\;D}{2}\|y_t \;\; y_{t}\|_2$.

Note that Lemma 4 also introduces an assumption about y_t and $F_{t \;\; 1}(y)$. According to our Algorithm 2, y_t is actually generated by approximately minimizing $F_{t \;\; 1}(y)$ with a linear optimization step. Therefore, by following the analysis of OFW for strongly convex losses [Wan and Zhang, 2021], we show that this assumption is satisfied with $= 16(G + 2\;D)^2 =$ and $= 1=3$.

Lemma 5 Let $y_t = \mathrm{argmin}_{y \in K} F_{t \;\; 1}(y)$ for any $t = 2; \dots; T+1$, where $F_t(y)$ is defined in (12). Suppose Assumption 1 and 2 hold, and all losses are strongly convex. For any $t = 2; \dots; T+1$, Algorithm 2 has

$$F_{t \;\; 1}(y_t) \quad F_{t \;\; 1}(y_t) \quad \frac{16(G + 2\;D)^2(t \quad 1)^{1=3}}{}:$$

By combining Lemmas 4 and 5, we have

$$A + C + E = O\left(T^{(1+ \quad)=2} + d\log T\right) = O(T^{2=3} + d\log T): \qquad (20)$$

Furthermore, by following the analysis of OFW for strongly convex losses [Wan and Zhang, 2021], we establish an upper bound for the third term in (19).

Lemma 6 Suppose Assumption 1 and 2 hold, and all losses are strongly convex. For any $x \in K$, Algorithm 2 ensures

$$B^0 \quad \frac{6\sqrt{2}(G + 2\;D)^2 T^{2=3}}{} + \frac{2(G + 2\;D)^2 \ln T}{} + (G + \;D)D$$

where $B^0 = \sum_{t=1}^{T}$

(a) Different d (b) $d = 501$

Figure 1: Comparisons of our DOFW and DOFW$_{sc}$ against BOLD-OFW and BOLD-OFW$_{sc}$.

and BOLD-OFW$_{sc}$ will maintain several instances of OFW for convex and strongly convex losses, respectively. Note that in the non-delayed case with $d = 1$ our delayed OFW actually reduces to the original OFW. Therefore, our Theorems 1 and 2 can also be utilized to choose the parameters for each instance of OFW maintained in BOLD-OFW and BOLD-OFW$_{sc}$. To be precise, in BOLD-OFW, we set $\eta = \frac{D}{\sqrt{2}G(T/d+2)^{3/4}}$ for each instance of OFW for convex losses, since the total rounds of each instance is roughly T/d. In BOLD-OFW$_{sc}$, we only need to set $\eta = 2$ for each instance of OFW for strongly convex losses. Moreover, for our delayed OFW and each instance of OFW, the initial decision is set to $\mathbf{1}/50$, where $\mathbf{1}$ denotes the all-ones vector.

Fig. 1(a) shows the total loss of T rounds for each algorithm under different values of the maximum delay d. First, when $d = 1$, the total loss of our DOFW is the same as that of BOLD-OFW and the total loss of our DOFW$_{sc}$ is the same as that of BOLD-OFW$_{sc}$, which is reasonable because in this case DOFW and BOLD-OFW reduce to the original OFW for convex losses, and DOFW$_{sc}$ and BOLD-OFW$_{sc}$ reduce to the original OFW for strongly convex losses. Second, for $d = 51, 101, 151, \dots, 501$, our DOFW and DOFW$_{sc}$ are better than BOLD-OFW and BOLD-OFW$_{sc}$, respectively, which clearly verifies the advantage of our algorithms in the delayed setting. It is worthy to notice that $d = 501$ is larger than $T^{2/3}$. Moreover, for our DOFW and DOFW$_{sc}$, when d increases from 1 to 501, the growth of the total loss is very slow, which is consistent with the dependence of our regret bounds on d. Fig. 1(b) shows the cumulative loss for each algorithm when $d = 501$. As the number of iterations increases, the cumulative loss of BOLD-OFW and BOLD-OFW$_{sc}$ increase much faster than that of our algorithms.

6 Conclusion and future work

In this paper, we propose delayed OFW for OCO with arbitrary delays. For convex losses, we show that it attains an $O(T^{3/4} + dT^{1/4})$ regret bound, which matches the $O(T^{3/4})$ regret bound of OFW in the non-delayed setting, as long as d does not exceed $O(\sqrt{T})$. When losses are strongly convex, we further prove that it can attain an $O(T^{2/3} + d \log T)$ regret bound, which matches the $O(T^{2/3})$ regret bound of OFW in the non-delayed setting, as long as d does not exceed $O(T^{2/3}/\log T)$. Simulation experiments demonstrate the performance of delayed OFW in the delayed setting.

This paper only extends the classical OFW to the delayed setting. In the future, we will investigate how to develop delayed variants for other projection-free online algorithms.

Acknowledgments

This work was partially supported by NSFC (61921006, 62122037), and JiangsuSF (BK20200064).

References

Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge University Press, 2004.

Lin Chen, Mingrui Zhang, and Amin Karbasi. Projection-free bandit convex optimization. In Proceedings of the 22nd International Conference on Arti cial Intelligence and Statistics, pages 2047–2056, 2019.

Genevieve E Flaspohler, Francesco Orabona, Judah Cohen, Soukayna Mouatadid, Miruna Oprescu, Paulo Orenstein, and Lester Mackey. Online learning with optimism and delay. In Proceedings of the 38th International Conference on Machine Learning, pages 3363–3373, 2021.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. Naval Research Logistics Quarterly, 3(1–2):95–110, 1956.

Dan Garber and Elad Hazan. A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. SIAM Journal on Optimization, 26(3):1493–1528, 2016.

Dan Garber and Ben Kretzu. Improved regret bounds for projection-free bandit convex optimization. In Proceedings of the 23rd International Conference on Arti cial Intelligence and Statistics, pages 2196–2206, 2020.

Dan Garber and Ben Kretzu. Revisiting projection-free online learning: the strongly convex case. In Proceedings of the 24th International Conference on Arti cial Intelligence and Statistics, pages 3592–3600, 2021.

Dan Garber and Ben Kretzu. New projection-free algorithms for online convex optimization with adaptive regret guarantees. In Proceedings of 35th Conference on Learning Theory, pages 2326–2359, 2022.

Elad Hazan. Introduction to online convex optimization. Foundations and Trends in Optimization, 2 (3–4):157–325, 2016.

Elad Hazan and Satyen Kale. Projection-free online learning. In Proceedings of the 29th International Conference on Machine Learning, pages 1843–1850, 2012.

Elad Hazan and Edgar Minasyan. Faster projection-free online learning. In Proceedings of the 33rd Annual Conference on Learning Theory, pages 1877–1893, 2020.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. Machine Learning, 69(2):169–192, 2007.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In Proceedings of the 30th International Conference on Machine Learning, pages 427–435, 2013.

Pooria Joulani, András György, and Csaba Szepesvári. Online learning under delayed feedback. In Proceedings of the 30th International Conference on Machine Learning, pages 1453–1461, 2013.

o4 T1 [(59timizatJ8(pr50(on)-2561 0 0 1 117.963 282.43 Tm1 [(686fer)37(emputelayed)-Csaba)-25Sy(prmba)-25

Zakaria Mhammedi. Efficient projection-free online convex optimization with membership oracle. In *Proceedings of 35th Conference on Learning Theory*, pages 5314–5390, 2022.

Kent Quanrud and Daniel Khashabi. Online learning with adversarial delays. In *Advances in Neural Information Processing Systems 28*, pages 1270–1278, 2015.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

Yuanyu Wan and Lijun Zhang. Projection-free online learning over strongly convex sets. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 10076–10084, 2021.

Yuanyu Wan, Wei-Wei Tu, and Lijun Zhang. Projection-free distributed online convex optimization with $O(\sqrt{T})$ communication complexity. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9818–9828, 2020.

Yuanyu Wan, Wei-Wei Tu, and Lijun Zhang. Online strongly convex optimization with unknown delays. *Machine Learning*, 111(3):871–893, 2022a.

Yuanyu Wan, Guanghui Wang, Wei-Wei Tu, and Lijun Zhang. Projection-free distributed online learning with sublinear communication complexity. *Journal of Machine Learning Research*, 23 (172):1–53, 2022b.

Marcelo J. Weinberger and Erik Ordentlich. On delayed prediction of individual sequences. *IEEE Transactions on Information Theory*, 48(7):1959–1976, 2002.

Wenpeng Zhang, Peilin Zhao, Wenwu Zhu, Steven C. H. Hoi, and Tong Zhang. Projection-free distributed online learning in networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 4054–4062, 2017.

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

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately re ect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This work is mostly theoretical and the societal impacts discussion is not applicable.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identi able information or offensive content? [N/A] Only synthetic data are used in this work, which do not contain personally identi able information and offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]