OPT2020: 12th Annual Workshop on Optimization for Machine Learning

Optimal Nonsmooth Frank-Wolfe method for Stochastic Regret Minimization

Kiran Koshy Thekumparampil THEKUMP2@ILLINOIS.EDU
University of Illinois at Urbana-Champaign, USA

Prateek Jain PRAJAIN@MICROSOFT.COM
Praneeth Netrapalli PRANEETH@MICROSOFT.COM
Microsoft Research, India

Sewoong Oh SEWOONG@CS.WASHINGTON.EDU
University of Washington, Seattle, USA

Abstract

The current best-known algorithm for convex constrained nonsmooth online stochastic regret minimization using a Linear Minimization Oracle (LMO, a la Frank-Wolfe) and a Stochastic First-order Oracle (SFO) achieves a regret of $\mathcal{O}(K^{3/4})$, where K is the number of iterations [26]. We provide two novel single-loop nonsmooth Frank-Wolfe methods, P-MOLES & PD-MOLES, which achieve the nearly-optimal online stochastic (non-adversarial) regret of $\mathcal{O}(\sqrt{K}\ln(K))$ for this problem with a Lipschitz continuous function. Our methods only need a mild assumption that the function remains Lipschitz and can be queried on a slightly larger neighborhood around the constraint set. Further, the last-iterates of our methods are guaranteed to be ε-suboptimal feasible solutions just after using $\mathcal{O}(\varepsilon^{-2})$ LMO calls and $\mathcal{O}(\varepsilon^{-2})$ SFO calls. These *offline oracle calls complexities* are optimal, and compared to the state-of-the-art offline method, MOLES [48], P-MOLES & PD-MOLES have the added advantage of being single-loop methods which use only one LMO call and one SFO call per iteration. This kind of simplicity is much preferred in practice, especially in the online setting.

1. Introduction

In this paper we study the constrained online stochastic nonsmooth convex optimization (OSNCO) problem where the objective is to minimize the stochastic cumulative regret:

$$\text{Stochastic Regret, } \mathcal{SR}_K := \sum_{k=1}^{K} [\mathbb{E}[f(x_{k-1})] - \min_{x\in\mathcal{X}} f(x)] \tag{1}$$

for some iterates $\{x_{k-1}\}_{k=1}^{K}$ from a compact convex constrained set $\mathcal{X} \subset \mathbb{R}^d$, when given access to a Stochastic First-order Oracle (SFO) of a Lipschitz continuous convex function $f : \mathcal{X} \to \mathbb{R}$ and a Linear Minimization Oracle (LMO) for the convex constrained set $\mathcal{X}$. An SFO returns an unbiased stochastic estimator of the subgradient of f at x when queried at x, i.e. $\mathbb{E}[\text{SFO}(x) \,|\, x] \in \partial f(x)$. An LMO returns a minimizer of any linear functional $\langle g, \cdot\rangle$ over the constraint set $\mathcal{X}$.

$$\text{LMO}\,(g) \in \operatorname*{argmin}_{s\in\mathcal{X}} \langle g, s\rangle \tag{2}$$

Table 1: Comparison of the online regrets of LMO-based algorithms for stochastic/adversarial settings for smooth (Lipschitz ∇f)/nonsmooth (Lipschitz f) problems. We only show single-loop algorithms utilizing one LMO and one SFO call per iteration, except for 1-SFW [55] & ORGFW [52]. Both exceptions require two correlated SFO calls (using the same stochastic realization) per iteration, which is impossible with a general black-box SFO.

Algorithms	**Regret**	**Assumptions**
Online Adversarial setting	(†we tightened dimension d terms from [26, Corollary 3.3])	
Online Conditional Gradient [23]	$\mathcal{O}(K^{3/4})$	Lipschitz f_t in $\mathcal{X}$
OSPF [26]	$\mathcal{O}(d^{1/4}\, K^{3/4})^\dagger$	Lipschitz f_t in $\mathcal{X}$
OSPF [26]	$\mathcal{O}(d^{1/3}\, K^{2/3})^\dagger$	Lipschitz f_t & ∇f_t in $\mathcal{X}$
Online Stochastic setting	(‡requires SFO at two points under the same stochastic realization)	
1-SFW [55] & ORGFW [52]	$\mathcal{O}(\sqrt{K})$	ERM SFO‡, Lipschitz f & ∇f in $\mathcal{X}$
OSFW [11]	$\mathcal{O}(K^{2/3})$	Lipschitz f & ∇f in $\mathcal{X}$
P-MOLES & PD-MOLES (ours)	$\mathcal{O}(\sqrt{K}\ln(K))$	Lipschitz f in $\mathcal{X}'$

At each iteration we play a feasible random action $x_{k-1} \in \mathcal{X}$ and consequently incur an expected regret of $\mathbb{E}[f(x_{k-1})] - \min_{x\in\mathcal{X}} f(x)$. Then we are can to query the SFO and LMO once before playing the next round. The goal is to obtain a non-trivial $o(K)$ cumulative $\mathcal{SR}_K$ after K rounds.

Traditionally, OSNCO has been studied, assuming access to the Projection Oracle (PO) to the constraint set $\mathcal{X}$, $\text{PO}(x) = \mathcal{P}_\mathcal{X}(x) = \text{argmin}_{y\in\mathcal{X}} \|y - x\|_2^2$ [23]. This was successfully applied to many important stochastic learning applications [46], such as support vector machines (SVM) [7], robust learning [29], and utility maximization in finance [49]. However, in higher-dimensional applications, even calling the PO once can be computationally prohibitive.

As an alternative to using the PO, Frank and Wolfe [15], in their seminal work, proposed the Frank-Wolfe (FW) or *conditional gradient* method to minimize smooth deterministic convex functions, which uses an LMO to access the constraint set. Of late, FW method and its variants have found a resurgence in popularity in Machine Learning, as linear minimization is much faster than projection in many modern applications such as a nuclear norm ball constrained problems [9], video-narration alignment [1], structured SVM [31], and multiple sequence alignment and motif discovery [53]. LMO based methods also have an added benefit of preserving some desirable atomic structures, such sparsity and low-rankness, in their solution [13].

However, until very recently [48], only *smooth* problems had efficient LMO-based algorithms. For a minimization problem $\min_{x\in\mathcal{X}} f(x)$, we measure the efficiency of an algorithm using its LMO/SFO calls complexity (LMO-CC/SFO-CC), which is defined as the worst-case number of LMO/SFO calls the algorithm makes so as to find a feasible ε-suboptimal solution, $\widehat{x}$, i.e. $f(\widehat{x}) - \min_{x\in\mathcal{X}} f(x) \le \varepsilon$. In the deterministic smooth case, FW method has an LMO-CC and SFO-CC of $O(1/\varepsilon)$. Over the last decade several algorithms have been proposed for the stochastic smooth setting. One of the best known algorithm, Stochastic Conditional Gradient Sliding (SCGS) method, obtains the optimal LMO-CC of $O(\varepsilon^{-1})$ and an SFO-CC of $O(\varepsilon^{-2})$ [36]. SCGS is not online friendly because it involves multiple loops and multiple SFO and LMO calls per iteration, and the most general online setting restricts the algorithm from making more than a constant number of LMO and SFO calls per iteration.

The current best online algorithms for stochastic smooth problems are variance reduction-based 1-SFW [55] and ORGFW [52] methods, and they achieve the nearly-optimal regret of $\mathcal{O}(\sqrt{K})$ using only one LMO and two SFO calls per iteration ($\mathcal{O}$ suppresses constants and polylog factors). However, both these methods assume that the SFO is not a black-box and SFO can be queried at two different points under the same realization of stochasticity, like in the case of empirical risk minimization (ERM). Additionally, 1-SFW assumes bounded and Lipschitz function, and ORGFW assumes bounded initial regret: $f(x_0) - \min_{x\in\mathcal{X}} f(x)$ and bounded stochasticity in the SFO: $\|\text{SFO}\,(x) - \nabla f(x)\|$. With a black-box SFO, the current best regret $\mathcal{O}(K^{2/3})$ in stochastic smooth setting is achieved by OSFW [11], which additionally assume Lipschitzness of the function and bounded initial regret. Our P-MOLES & PD-MOLES methods (assuming bounded gradients) achieve the nearly-optimal $\tilde{\mathcal{O}}(\sqrt{K})$ stochastic regret in the smooth setting, without any explicit variance-reduction.

Online adversarial optimization is another problem closely related to and much stricter than OSNCO. In the adversarial setting, one also assumes that the loss functions f_k are also adversarially chosen after the player plays her random action x_{k-1}, i.e. we aim to minimize the cumulative regret:

$$\text{Adversarial Regret, } \mathcal{R}_K := \sum_{k=1}^{K} \mathbb{E}[f_k(x_{k-1})] - \min_{x\in\mathcal{X}} \sum_{k=1}^{K} f_k(x) \tag{3}$$

Note that, usually an algorithm with a particular adversarial regret also achieves the same stochastic regret. In the smooth setting, the best online adversarial regret is achieved by OSPF algorithm [26], although it achieves a dimension d dependent regret $\mathcal{O}(d^{1/3}K^{2/3})$.

For offline stochastic nonsmooth Lipschitz continuous problems, for many decades the known best combined LMO and SFO calls complexity was $\mathcal{O}(\varepsilon^{-4})$, which was achieved by FW-PGD (folklore) [48, Theorem 4], Randomized FW [34], and Online Conditional Gradient [23] methods. The gap to a known lower bound of $O(\varepsilon^{-2})$ LMO-CC was closed by MOLES [48], which achieves the optimal LMO-CC and SFO-CC of $\mathcal{O}(\varepsilon^{-2})$.

MOLES [48], casts the constrained minimization problem into a composite optimization problem. This separates the nonsmooth objective and the constraint into two parts of a composite objective consisting of a nonsmooth unconstrained function f and a simple smooth constrained function. Under the right conditions an approximate minimizer of this composite objective is also an approximate minimizer of the original minimization problem. MOLES is a multi-loop scheme which uses multiple LMO/SFO calls per iteration to solve this composite problem using the optimal $\mathcal{O}(\varepsilon^{-2})$ number of LMO and SFO calls. Hence, it is difficult to obtain an online regret for this algorithm and the method could be challenging to implement and slow in practice.

There are only two known online algorithms for the OSNCO problem and they address the more general adversarial setting. They are Online Conditional Gradient [24] and OSPF [26] methods which achieve worst-case adversarial regrets of $\mathcal{O}(K^{3/4})$ and $\mathcal{O}(d^{1/4}\,K^{3/4})$ respectively. A natural question to ask here is: *can we achieve a better regret for the OSNCO problem?*

We answer this question in the affirmative by providing two novel algorithm, P-MOLES (primal averaging MOLES) and PD-MOLES (primal-dual averaging MOLES), which achieve a worst-case online stochastic regret of $\mathcal{O}(\sqrt{K}\ln(K))$. This regret is optimal up to polylog factors [23]. Our algorithm P-MOLES (PD-MOLES) solves the same composite objective as MOLES, but we employ a novel scheme which combines a primal averaging variant of the FW: PA-FW (primal-dual averaging variant of the FW: PDA-FW) [34] and the Accelerated Stochastic Approximation: AC-SA [33] (Accelerated Dual-Averaging [51]) scheme to solve it. This simplifies the multi-loop MOLES [48] into a single-loop algorithm which only uses one LMO and SFO call per iteration.

Contributions: We summarize our contributions below and in Table 1. We assume that the function f extends to a slightly larger neighborhood $\mathcal{X}'$ of the constraint set $\mathcal{X}$ i.e., f continues to be Lipschitz continuous and SFO can be queried in this neighborhood.

- We provide two novel single-loop algorithms P-MOLES & PD-MOLES, both of which achieve the optimal last iterate LMO-CC and SFO-CC of $\mathcal{O}(\varepsilon^{-2})$ for finding a *feasible* ε-suboptimal solution of a non-smooth stochastic convex problem.
- Our P-MOLES & PD-MOLES are also the first algorithms which achieves the nearly-optimal regret of $\mathcal{O}(\sqrt{K}\ln(K))$ for the OSNCO problem. This regret is optimal up to polylog factors. Surprisingly, these methods do not use any explicit variance reduction techniques, like other algorithms which achieve this regret for smooth problems [52, 55].

2. Related Work

Frank-Wolfe methods: After the resurgence of interest in the FW or *conditional gradient* method [15, 38] for machine learning, several of its variants and their analyses have been proposed [3, 8, 16, 18, 34, 37, 41], and FW has been extended to stochastic or nonconvex [4, 22, 25, 30, 44, 45] settings. However these methods provide dimension-free LMO-CC and SFO-CC only for smooth functions.
Nonsmooth Frank-Wolfe: After [50] posed the question of optimizing nonsmooth functions using LMO, few works studied the problem, however, apart from the optimal MOLES [48], all of them were either inefficient [34, 48] or assumed more stricter assumptions [12, 43, 50]. Notably, the special case of nonsmooth functions admitting *smooth* convex-concave minimax saddle point reformulation, has been extensively studied [10, 14, 19–21, 27, 28, 39, 42, 47].
Online Frank-Wolfe: There has been several works which studied online stochastic or adversarial regrets of LMO-based methods [11, 17, 23, 24, 26, 32, 52]. However, apart from the ones mentioned earlier [11, 23, 24, 26, 52], none of the methods provide online regrets under the the general setting of black-box LMO and SFO of our paper, using only one LMO and one SFO call per iteration.

3. Preliminaries and Notations

We consider Nonsmooth Convex Optimization with SFO (4) and LMO (2) accesses. Let $\mathcal{X} \subset \mathbb{R}^d$ be a closed convex set of diameter $D_{\mathcal{X}} := \max_{x_1,x_1\in\mathcal{X}} \|x_1 - x_2\|$, where $\|\cdot\|$ is the Euclidean norm which corresponds to the inner product $\langle\cdot,\cdot\rangle$. Let $\mathcal{X}$ be enclosed in a closed convex set $\mathcal{X}' \subseteq \mathbb{R}^d$ to which it is easy to project, i.e. $\mathcal{X} \subset \mathcal{X}'$. For simplicity, let $\mathcal{X}'$ be a Euclidean ball of radius R around origin. We can satisfy $R = D_{\mathcal{X}}$ by re-centering $\mathbb{R}^d$ around any feasible point of $\mathcal{X}$. We assume $f : \mathcal{X}' \to \mathbb{R}$ to be a proper, lower semi-continuous (l.s.c.), convex Lipschitz function. We use $\partial f(x)$ to denote sub-differential of f at x, and if f is differentiable we use $\nabla f(x)$ to denote its gradient at x. Below we provide the definitions of Lipschitzness, smoothness and SFO.

Definition 1 *A function $f : \mathcal{X}' \to \mathbb{R}$ is G-Lipschitz if and only if, $|f(y) - f(x)| \le G\|y - x\|$ for all $x, y \in \mathcal{X}'$. For a convex f, this is equivalent to:* $\max_{x\in\mathcal{X}'} \max_{g\in\partial f(x)} \|g\| \le G$.

Definition 2 *A function $f : \mathcal{X}' \to \mathbb{R}$ is μ-strongly convex if and only if, $\frac{\mu}{2}\|y-x\|^2 + \langle g, y-x\rangle + f(x) \le f(y)$, for all $x, y \in \mathcal{X}'$ and $g \in \partial f(x)$. Similarly, a differentiable function $f : \mathcal{X}' \to \mathbb{R}$ is said to be L-smooth if and only if, $f(y) \le f(x) + \langle \nabla f(x), y-x\rangle + \frac{L}{2}\|y-x\|^2$ for all $x, y \in \mathcal{X}'$.*

We consider problems with black-box Stochastic First-order Oracle (SFO) access, which computes unbiased stochastic subgradient of a point x with a variance σ^2, as defined below:

$$\text{SFO}\,(x) := \widehat{g}, \text{ where } \mathbb{E}[\widehat{g}\,|\,x] = g \text{ for some } g \in \partial f(x), \text{ and } \mathbb{E}[\|\widehat{g} - g\|^2\,|\,x] \le \sigma^2. \tag{4}$$

We define ε-suboptimal minimizer (solution) of $\min\limits_{x\in\mathcal{X}} f(x)$ and the online stochastic regret as follows.

Definition 3 *We say that $x_\varepsilon \in \mathcal{X}$ is an ε-suboptimal minimizer of the constrained optimization problem $\min_{x\in\mathcal{X}} f(x)$ if, $f(x_\varepsilon) - f(x) \le \varepsilon$ for all $x \in \mathcal{X}$. An algorithm which plays the actions $\{x_k\}_{k=0}^{K-1}$ by querying one LMO and one SFO after each round, is said to achieve the stochastic online regret $\mathcal{SR}_K := \sum_{k=1}^{K} \mathbb{E}[f(x_{k-1}) - \min_{x\in\mathcal{X}} f(x)]$ for the function f.*

Moreau Envelope: The key idea behind MOLES [48] and our method is to use "smoothed" version of the function via its Moreau envelope [40, 54] defined below.

Definition 4 *For a proper l.s.c. convex function $f : \mathcal{X}' \to \mathbb{R} \cup \{\infty\}$ defined on a closed convex set $\mathcal{X}'$ and $\lambda > 0$, its* Moreau-(Yosida) envelope *function, $f_\lambda : \mathcal{X}' \to \mathbb{R}$, is given by*

$$f_\lambda(x) \quad = \quad \min_{x'\in\mathcal{X}'} f(x') + \frac{1}{2\lambda}\|x - x'\|^2, \quad \textit{for all } x \in \mathcal{X}' \; . \tag{5}$$

Furthermore, the prox *operator is defined:* $\text{prox}_{\lambda f}(x) := \text{argmin}_{x'\in\mathcal{X}'} f(x') + \frac{1}{2\lambda}\|x - x'\|^2$.

Note that this definition of Moreau envelope is not standard as x' is constrained to $\mathcal{X}' \subseteq \mathbb{R}^d$. However, Lemma 7 (in Appendix) shows that this Moreau envelope and the prox operator still satisfies most useful properties of the standard definition. This lemma implies that, to find an feasible ε-suboptimal solution of a nonsmooth f, one can instead minimize f_λ and achieve a faster convergence by exploiting its smoothness. Concretely, if f is G-Lipschitz and $\lambda = O(\varepsilon/G^2)$, then the Lemma 7 ensures that solving f_λ up to $O(\varepsilon)$ accuracy guarantees $O(\varepsilon)$ accuracy in the minimization of the original function f (Lemma 8 in Appendix). This insight allows us to design a simple method that can achieve optimal LMO-CC while the maintaining optimal SFO-CC.

4. Improved MOreau LMO Efficient Subgradient (MOLES) methods

In order to achieve the optimal LMO-CC and SFO-CC, MOLES [48] directly optimize the Moreau envelope through the following joint optimization.

$$\min_{x\in\mathcal{X}, x'\in\mathcal{X}'} [\Psi_\lambda(x, x') := f(x') + \psi_\lambda(x, x')] \quad \text{where} \quad \psi_\lambda(x, x') := \frac{1}{2\lambda}\|x' - x\|^2, \tag{6}$$

where the function $\Psi_\lambda : \mathcal{X}' \times \mathcal{X}' \to \mathbb{R}$ is convex in the joint variable (x, x'). The main advantage of this new form is that, this is a composite optimization problem with a nonsmooth part (corresponding to $f(x')$) and a $2/\lambda$-smooth part (corresponding to $(1/2\lambda)\|x' - x\|^2$) with the constrained variable $x \in \mathcal{X}$ only appearing in the smooth part. Now, by the Lemma 8 (in Appendix), if $\lambda = \mathcal{O}(\varepsilon)$, an ε-suboptimal minimizer of Ψ_λ, is also an approximate minimizer of the original function f.

4.1. Primal Averaging MOLES (P-MOLES)

MOLES essentially solves (6) simultaneously using Gradient Sliding [35] and Conditional Gradient Sliding [36] frameworks, which are optimal for minimizing this composite problem (6). However this is a multi-loop scheme with multiple calls to LMO and SFO per iteration, which is challenging to implement and may be slow in practice. Therefore, we simplify this to a single-loop algorithm, P-MOLES (Algorithm 1) by applying the the accelerated primal averaging FW (PA-FW) [34] scheme on the constrained variable x and the AC-SA scheme [33] on the unconstrained variable $x' \in \mathcal{X}$. The iterates of the algorithm satisfy the following guarantees (a proof in Appendix A).

```
Algorithm 1: P-MOLES: Primal Averaging MOLES [48] method
Input: $f$, $\mathcal{X}$, $\mathcal{X}'$, $G$, $D_{\mathcal{X}}$, $R$, $x_0$, $K$, $\lambda$, $\{\eta_k\}_{k\in[K]}$,
Set $x'_0 = z'_0 = x_0 = z_0 = x_0$
for $k = 1, \ldots, K$ do
    Set $\beta_k = (\frac{8}{k\lambda} + \frac{1}{k\eta_k})$ , and $\gamma_k = \frac{2}{k+1}$ .
    Set $(y_k, y'_k) = (1-\gamma_k)\,(x_{k-1}, x'_{k-1}) + \gamma_k(z_{k-1}, z'_{k-1})$
    Set $z_k = \mathrm{LMO}\,(\nabla_{y_k}\Psi_\lambda(y_k, y'_k))$ (2)        // Note $\nabla_{y_k}\Psi_\lambda(y_k, y'_k) = \frac{y_k - y'_k}{\lambda}$
    Set $\widehat{g}_k = \mathrm{SFO}\,(y'_k)$ (4)
    Set $\widehat{z}'_k = z'_{k-1} - \frac{1}{\beta_k}\cdot(\nabla_{y'_k}\psi_\lambda(y_k, y'_k) + \widehat{g}_k)$        // $\nabla_{y'_k}\psi_\lambda(y_k, y'_k) = \frac{y'_k - y_k}{\lambda}$
    Set $z'_k = \widehat{z}'_k \cdot \min(1, R/\|\widehat{z}'_k\|)$
    Set $(x_k, x'_k) = (1-\gamma_k)\,(x_{k-1}, x'_{k-1}) + \gamma_k(z_k, z'_k)$
end
Output: $(x_K, x'_K)$
```

Theorem 5 *Let $f : \mathcal{X}' \to \mathbb{R}$ be a G-Lipschitz continuous convex function equipped with an SFO with variance σ^2, and $\mathcal{X} \subseteq \mathcal{X}'$ be a compact convex set of diameter $D_{\mathcal{X}}$ equipped with an LMO and be enclosed by the Euclidean ball $\mathcal{X}'$ of radius R ($R < D_{\mathcal{X}}$) around the origin. Then, the iterates of P-MOLES (Algorithm 1) run for K iterations with $\lambda = \frac{2\sqrt{2}D_{\mathcal{X}}}{G\sqrt{K}}$, $\eta_k \le \eta_{k-1}$ for all $k \in [K]$ and:*

(a) stepsize choice: $\eta_k = \eta = \frac{\sqrt{3}\,D_{\mathcal{X}}}{\sqrt{2\,K^3(4G^2+\sigma^2)}}$ for all $1 \le k \le K$ satisfy

$$\mathbb{E}[f\,(x_K)] - \min_{x\in\mathcal{X}} f(x) \le [2\sqrt{2}\,GD_{\mathcal{X}} + (2/\sqrt{3})\sqrt{2}\sqrt{4G^2+\sigma^2}D_{\mathcal{X}}](K^{-1/2}) := \varepsilon_K^{(a)} \quad (7)$$

(b) stepsize choice: $\eta_k = \frac{\sqrt{3}\,R}{\sqrt{k^3\,(4G^2+\sigma^2)}}$ for all $1 \le k \le K$ satisfy

$$\mathbb{E}[f\,(x_K)] - \min_{x\in\mathcal{X}} f(x) \le [2\sqrt{2}\,GD_{\mathcal{X}} + (8/\sqrt{3})\,\sqrt{4G^2+\sigma^2}R](K^{-1/2}) := \varepsilon_K^{(b)}, \quad \textit{and} \quad (8)$$

$$\mathcal{SR}_K = \mathbb{E}\big[\sum_{k=1}^{K} f\,(x_{k-1}) - \min_{x\in\mathcal{X}} f(x^*)\big] \le \sqrt{2}GD_{\mathcal{X}}\sqrt{K}(\ln(K)+2) +$$

$$(16/\sqrt{3})\sqrt{4G^2+\sigma^2}R\sqrt{K} + 2GD_{\mathcal{X}}\ln(K+1) := \Delta_K^{(b)} \quad (9)$$

Remarks: The theorem implies that after $K = \mathcal{O}(\frac{(G^2+\sigma^2)D_{\mathcal{X}}^2}{\varepsilon^2})$ iterations, the algorithm can output x_K such that $\mathbb{E}[f(x_K)] - f(x^*) \le \varepsilon$. Therefore, P-MOLES uses a total of $\mathcal{O}(\frac{(G^2+\sigma^2)D_{\mathcal{X}}^2}{\varepsilon^2})$ SFO and LMO calls to reach ε-suboptimal solution. Although, this achieves the same optimal oracle calls complexities as the MOLES [48], our method being a single loop online algorithm may be more practical and may have better performance. Using the second stepsizes choice, we can achieve a nearly-optimal [23] online stochastic regret of $\mathcal{O}(GD_{\mathcal{X}}\sqrt{K}\ln(K))$ when the time horizon K is known a priori. This regret better than that of other algorithms for our black-box oracle setting (Table 1). For unknown K, we can use the doubling trick [2] to obtain a regret of the same order.

4.2. Primal-Dual Averaging MOLES (PD-MOLES)

Next, we provide a dual-averaging variant of the P-MOLES, PD-MOLES (Algorithm 2) which combines primal-dual averaging FW (PDA-FW) [34] and Accelerated Dual-Averaging [51] schemes and achieve the following convergence and regret guarantees (a proof in Appendix B).

Algorithm 2: PD-MOLES: Primal-Dual Averaging MOLES [48] method

```
Input: $f$, $\mathcal{X}$, $\mathcal{X}'$, $G$, $D_{\mathcal{X}}$, $R$, $x_0$, $K$, $\lambda$, $\{\eta_k\}_{k\in[K]}$,
Use the same steps as P-MOLES (Algorithm 1), but replace Line 5 and Line 7 with:
5 Set $z_k = \mathrm{LMO}\left(\sum_{j=1}^{k} \theta_j \cdot \nabla_{y_j}\Psi_\lambda(y_j, y_j')\right)$ (2)        // Note $\nabla_{y_k}\Psi_\lambda(y_k, y_k') = \frac{y_k - y_k'}{\lambda}$
7 Set $\widehat{z}_k' = z_0' - \frac{1}{k\,\beta_k} \cdot (\sum_{j=1}^{k} \theta_j \cdot (\nabla_{y_j'}\psi_\lambda(y_j, y_j') + \widehat{g}_j))$        // $\nabla_{y_k'}\psi_\lambda(y_k, y_k') = \frac{y_k' - y_k}{\lambda}$
```

Theorem 6 *Under the same assumptions and real numbers defined $\varepsilon_K^{(a)}$ (7), $\varepsilon_K^{(b)}$ (8) and $\Delta_K^{(b)}$ (9) as in Theorem 5, the iterates of PD-MOLES (Algorithm 2) run for K iterations with $\lambda = \frac{2\sqrt{2}D_{\mathcal{X}}}{G\sqrt{K}}$, $\eta_k \le \eta_{k-1}$ for all $k \in [K]$ and*

(a) the same stepsize choice as Theorem 5 (a), satisfy $\mathbb{E}[f(x_K)] - \min_{x\in\mathcal{X}} f(x) \le \varepsilon_K^{(a)}$

(b) the same stepsize choice as Theorem 5 (b) satisfy $\mathbb{E}[f(x_K)] - \min_{x\in\mathcal{X}} f(x) \le \varepsilon_K^{(b)} + \frac{\sqrt{4G^2+\sigma^2}D_{\mathcal{X}}^2}{\sqrt{3}\sqrt{K}\,R}$ and $\mathcal{SR}_K = \mathbb{E}\big[\sum_{k=1}^{K} f(x_{k-1}) - \min_{x\in\mathcal{X}} f(x)\big] \le \Delta_K^{(b)} + \frac{2\sqrt{4G^2+\sigma^2}D_{\mathcal{X}}^2\sqrt{K}}{\sqrt{3}\,R}$

Remarks: Although the above worst-case guarantee of the dual-averaging PD-MOLES is similar to that of P-MOLES, as with other dual-averaging algorithms [51], in practice we expect PD-MOLES to work better than P-MOLES.

5. Conclusion

We provide two novel single-loop algorithms which obtain the nearly-optimal $\mathcal{O}(K\ln(K))$ online stochastic regret for the OSNCO (1) problem. The same algorithms also achieve the optimal offline last-iterate LMO and SFO calls complexities of $\mathcal{O}(\varepsilon^{-2})$, which was earlier obtained by a multi-loop scheme, MOLES [48]. This makes our new methods more practical and easy to tune than the latter. We note that, just like MOLES [48], our dimension-free results are limited only to the Euclidean norm, since our results crucially depends on smoothness of the Moreau envelope and its regularizer, which is not known for non-Euclidean geometry [6]. However, due to equivalence of norms we can easily obtain dimension-dependent rates and regrets for any non-Euclidean geometry.

References

[1] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016.

[2] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 322–331. IEEE, 1995.

[3] Francis Bach. Duality between subgradient and conditional gradient methods. *SIAM Journal on Optimization*, 25(1):115–129, 2015.

[4] Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. In *Advances in Neural Information Processing Systems*, pages 3455–3464, 2018.

[5] Nikhil Bansal and Anupam Gupta. Potential-function proofs for first-order methods. *arXiv preprint arXiv:1712.04581*, 2017.

[6] Heinz H Bauschke, Minh N Dao, and Scott B Lindstrom. Regularizing with bregman–moreau envelopes. *SIAM Journal on Optimization*, 28(4):3208–3228, 2018.

[7] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[8] Gábor Braun, Sebastian Pokutta, and Daniel Zink. Lazifying conditional gradient algorithms. *Journal of Machine Learning Research*, 20(71):1–42, 2019.

[9] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982, 2010.

[10] Cheng Chen, Luo Luo, Weinan Zhang, and Yong Yu. Efficient projection-free algorithms for saddle point problems. *Advances in Neural Information Processing Systems*, 33, 2020.

[11] Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In *International Conference on Machine Learning*, pages 814–823, 2018.

[12] Edward Cheung and Yuying Li. Nonsmooth frank-wolfe using uniform affine approximations. *arXiv preprint arXiv:1710.05776*, 2017.

[13] Kenneth L Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):1–30, 2010.

[14] Bruce Cox, Anatoli Juditsky, and Arkadi Nemirovski. Decomposition techniques for bilinear saddle point problems and variational inequalities with affine monotone operators. *Journal of Optimization Theory and Applications*, 172(2):402–435, 2017.

[15] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[16] Robert M Freund and Paul Grigas. New analysis and results for the frank–wolfe method. *Mathematical Programming*, 155(1-2):199–230, 2016.

[17] Dan Garber and Elad Hazan. A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. *arXiv preprint arXiv:1301.4666*, 2013.

[18] Dan Garber and Elad Hazan. Faster rates for the frank-wolfe method over strongly-convex sets. In *32nd International Conference on Machine Learning, ICML 2015*, 2015.

[19] Gauthier Gidel, Tony Jebara, and Simon Lacoste-Julien. Frank-wolfe algorithms for saddle point problems. In *Artificial Intelligence and Statistics*, pages 362–371. PMLR, 2017.

[20] Janice H Hammond. *Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms*. PhD thesis, Massachusetts Institute of Technology, 1984.

[21] Zaid Harchaoui, Anatoli Juditsky, and Arkadi Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152(1-2): 75–112, 2015.

[22] Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Zebang Shen. Stochastic conditional gradient++: (non-)convex minimization and continuous submodular maximization. *arXiv preprint arXiv:1902.06992*, 2019.

[23] Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

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

[25] Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271, 2016.

[26] Elad Hazan and Edgar Minasyan. Faster projection-free online learning. *arXiv preprint arXiv:2001.11568*, 2020.

[27] Niao He and Zaid Harchaoui. Semi-proximal mirror-prox for nonsmooth composite minimization. In *Advances in Neural Information Processing Systems*, pages 3411–3419, 2015.

[28] Niao He and Zaid Harchaoui. Stochastic semi-proximal mirror-prox. Workshop on Optimization for Machine Learning, 2015. URL https://opt-ml.org/papers/OPT2015_paper_27.pdf.

[29] Peter J Huber. *Robust statistical procedures*, volume 68. SIAM, 1996.

[30] Simon Lacoste-Julien. Convergence rate of frank-wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

[31] Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. Block-coordinate frank-wolfe optimization for structural svms. In *Proceedings of the 30th international conference on machine learning*, pages 53–61, 2013.

[32] Jean Lafond, Hoi-To Wai, and Eric Moulines. On the online frank-wolfe algorithms for convex and non-convex optimizations. *arXiv preprint arXiv:1510.01171*, 2015.

[33] Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

[34] Guanghui Lan. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv preprint arXiv:1309.5550*, 2013.

[35] Guanghui Lan. Gradient sliding for composite optimization. *Mathematical Programming*, 159 (1-2):201–235, 2016.

[36] Guanghui Lan and Yi Zhou. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016.

[37] Guanghui Lan, Sebastian Pokutta, Yi Zhou, and Daniel Zink. Conditional accelerated lazy stochastic gradient descent. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1965–1974, 2017.

[38] Evgeny S Levitin and Boris T Polyak. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6(5):1–50, 1966.

[39] Francesco Locatello, Alp Yurtsever, Olivier Fercoq, and Volkan Cevher. Stochastic frank-wolfe for composite convex minimization. In *Advances in Neural Information Processing Systems*, pages 14246–14256, 2019.

[40] Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

[41] Yu Nesterov. Complexity bounds for primal-dual methods minimizing the model of objective function. *Mathematical Programming*, 171(1-2):311–330, 2018.

[42] Federico Pierucci, Zaid Harchaoui, and Jérôme Malick. A smoothing approach for composite conditional gradient with nonsmooth loss. Technical report, [Research Report] RR-8662, INRIA Grenoble, 2014.

[43] Sathya N Ravi, Maxwell D Collins, and Vikas Singh. A deterministic nonsmooth frank wolfe algorithm with coreset guarantees. *Informs Journal on Optimization*, 1(2):120–142, 2019.

[44] Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic frank-wolfe methods for nonconvex optimization. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1244–1251. IEEE, 2016.

[45] Anit Kumar Sahu, Manzil Zaheer, and Soummya Kar. Towards gradient free and projection free stochastic optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3468–3477, 2019.

[46] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT*, 2009.

[47] Arun Suggala and Praneeth Netrapalli. Follow the perturbed leader: Optimism and fast parallel algorithms for smooth minimax games. *Advances in Neural Information Processing Systems*, 33, 2020.

[48] Kiran K Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. Projection efficient subgradient method and optimal nonsmooth frank-wolfe method. *Advances in Neural Information Processing Systems*, 33, 2020.

[49] RB Vinter and H Zheng. Some finance problems solved with nonsmooth optimization techniques. *Journal of optimization theory and applications*, 119(1):1–18, 2003.

[50] DJ White. Extension of the frank-wolfe algorithm to concave nondifferentiable objective functions. *Journal of optimization theory and applications*, 78(2):283–301, 1993.

[51] Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010.

[52] Jiahao Xie, Zebang Shen, Chao Zhang, Boyu Wang, and Hui Qian. Efficient projection-free online methods with stochastic recursive gradient. In *AAAI*, pages 6446–6453, 2020.

[53] Ian En-Hsu Yen, Xin Lin, Jiong Zhang, Pradeep Ravikumar, and Inderjit Dhillon. A convex atomic-norm approach to multiple sequence alignment and motif discovery. In *International Conference on Machine Learning*, pages 2272–2280, 2016.

[54] Kōsaku Yosida. *Functional analysis*. Springer Verlag, 1965.

[55] Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. One sample stochastic frank-wolfe. In *International Conference on Artificial Intelligence and Statistics*, pages 4012–4023. PMLR, 2020.

Appendix

Lemma 7 (Lemma 1, [48]) *For a closed convex set $\mathcal{X}'$, a convex proper l.s.c. function $f : \mathcal{X}' \to \mathbb{R} \cup \{\infty\}$ and $\lambda > 0$, the following hold true for any $x \in \mathcal{X}'$ and its Moreau envelope f_λ and the prox operator $\widehat{x}_\lambda(x) := \mathrm{prox}_{f\lambda}(x)$ as given in Definition 4.*
(a) $\hat{x}_\lambda(x)$ is unique and $f(\hat{x}_\lambda(x)) \leq f_\lambda(x) \leq f(x)$,
(b) f_λ is convex, differentiable, $1/\lambda$-smooth and $\nabla f_\lambda(x) = (1/\lambda)(x - \hat{x}_\lambda(x))$, and,
(c) if f is G-Lipschitz continuous, then, $\|\hat{x}_\lambda(x) - x\| \leq G\lambda$, and $f(x) \leq f_\lambda(x) + G^2\lambda/2$.

Lemma 8 (Lemma 2, [48]) *Under the same assumptions as in Lemma 7, let $\mathcal{X} \subseteq \mathcal{X}'$ be a convex subset and Ψ_λ be defined as in (6). Then, (i) $\min_{x\in\mathcal{X}} \min_{x\in\mathcal{X}'} \Psi_\lambda(x, x') = \min_{x\in\mathcal{X}} f_\lambda(x) \leq \min_{x\in\mathcal{X}} f(x)$, and (ii) for any random vectors $(x_\varepsilon, x'_\varepsilon) \in \mathcal{X}\times\mathcal{X}'$, $\mathbb{E}[f(x_\varepsilon)]-G^2\lambda/2 \leq \mathbb{E}[f_\lambda(x_\varepsilon)] \leq \mathbb{E}[\Psi_\lambda(x_\varepsilon, x'_\varepsilon)]$.*

Appendix A. Proof of Theorem 5

Proof [Proof of Theorem 5] For some arbitrary $(x, x') \in \mathcal{X} \times \mathcal{X}'$, consider the following potential (Lyapunov) function:

$$\Phi_k := k(k+1)(\Psi_\lambda(x_k, x'_k) - \Psi_\lambda(x, x')) + (\frac{8}{\lambda} + \frac{1}{\eta_k})\|z'_k - x'\|^2 \tag{10}$$

This potential is similar to the one used to analyze the standard AGD for a $2/\lambda$-smooth function [5]. Below we prove that this potential satisfies an approximate descent guarantee. Now, let $k \geq 1$ and notice that by $2/\lambda$-smoothness and convexity of ψ_λ

$$\begin{aligned}
\psi_\lambda(x_k, x'_k) &\leq \psi_\lambda(y_k, y'_k) + \langle \nabla_k, (x_k, x'_k) - (y_k, y'_k)\rangle + \frac{1}{\lambda}\|(x_k, x'_k) - (y_k, y'_k)\|^2 \\
&\leq (1-\gamma_k)[\psi_\lambda(y_k, y'_k) + \langle \nabla_k, (x_{k-1}, x'_{k-1}) - (y_k, y'_k)\rangle] \\
&\quad \gamma_k[\psi_\lambda(y_k, y'_k) + \langle \nabla_k, (z_k, z'_k) - (y_k, y'_k)\rangle + \frac{\gamma_k}{\lambda}\|(z_k, z'_k) - (z_{k-1}, z'_{k-1})\|^2] \\
&\leq (1-\gamma_k)\psi_\lambda(x_{k-1}, x'_{k-1}) + \\
&\quad \gamma_k[\psi_\lambda(y_k, y'_k) + \langle \nabla_k, (z_k, z'_k) - (y_k, y'_k)\rangle + \frac{\gamma_k}{\lambda}\|(z_k, z'_k) - (z_{k-1}, z'_{k-1})\|^2]
\end{aligned} \tag{11}$$

where we use the shorthand $\nabla_k := [\nabla^T_{k,x} \nabla^T_{k,x'}]^T := [\nabla_x \psi_\lambda(y_k, y'_k)^T \; \nabla_{x'}\psi_\lambda(y_k, y'_k)^T]^T$, and the second inequality uses Lines 4 and 9 of Algorithm 1. Similarly, using G-Lipschitzness, convexity of f in $\mathcal{X}'$ and Line 6 of Algorithm 1, and assuming $\mathbb{E}[\widehat{g}_k \mid x] = g_k \in \partial f(z'_k)$ (4) and $\widetilde{g}_k \in \partial f(x'_k)$ we get that

$$\begin{aligned}
f(x'_k) &\leq f(y'_k) + \langle g_k, x'_k - y'_k\rangle + \langle \widetilde{g}_k - g_k, x'_k - y'_k\rangle \\
&\leq f(y'_k) + \langle g_k, (1-\gamma_k)x'_{k-1} + (\gamma_k)z'_k - y'_k\rangle + 2G\,\|\gamma_k(z'_k - z'_{k-1})\| \\
&\leq (1-\gamma_k)[f(y'_k) + \langle g_k, x'_{k-1} - y'_k\rangle] + 2G\,\gamma_k\|z'_k - z'_{k-1}\| + \\
&\quad \gamma_k[f(y'_k) + \langle \widehat{g}_k, z'_k - y'_k\rangle - \langle \delta_k, z'_{k-1} - y'_k\rangle - \langle \delta_k, z'_k - z'_{k-1}\rangle] \\
&\leq (1-\gamma_k)f(x'_{k-1}) + \gamma_k[f(y'_k) + \langle \widehat{g}_k, z'_k - y'_k\rangle] + \\
&\quad -\gamma_k\,\langle \delta_k, z'_{k-1} - y'_k\rangle + \gamma_k(2G + \|\delta_k\|)\|z'_k - z'_{k-1}\|
\end{aligned} \tag{12}$$

where we use Lines 4 and 9 of Algorithm 1 and $\delta_k = \widehat{g}_k - g_k$. Now multiplying (11) and (12) with $k(k+1)$ and summing them, using the fact that $\gamma_k/\lambda = 2/\lambda(k+1) \le 2/\lambda k = \beta_k/2 - 1/(2k\,\eta_k)$, (Line 3 of Algorithm 1) we get that

$$\begin{aligned} & k(k+1)\Psi_\lambda(x_k, x'_k) \\ \le\ & k(k-1)\Psi_\lambda(x_{k-1}, x'_{k-1}) + 2k\Psi_\lambda(y_k, y'_k) + 2k\left\langle \nabla_{k,x}, z_k - y_k \right\rangle + \\ & 2k[\left\langle \nabla_{k,x'} + \widehat{g}_k, z'_k - y'_k \right\rangle + \frac{\beta_k}{2}\|z'_k - z'_{k-1}\|^2] + \\ & \frac{4}{\lambda}\|z_k - z_{k-1}\|^2 + 2k[-\frac{1}{2k\,\eta_k}\|z'_k - z'_{k-1}\|^2 + (2G + \|\delta_k\|)\|z'_k - z'_{k-1}\| - \left\langle \delta_k, z'_{k-1} - y'_k \right\rangle] \end{aligned} \tag{13}$$

Using the elementary inequality $-bc^2/2 + ac \le a^2/2b$ for any real numbers a, b and c, we get

$$-\frac{1}{2k\,\eta_k}\|z'_k - z'_{k-1}\|^2 + (2G + \|\delta_k\|)\|z'_k - z'_{k-1}\| \le \frac{k\,\eta_k}{2}(2G + \|\delta_k\|)^2 \tag{14}$$

As z_k is a corner point and the output of LMO $(\nabla_{y_k}\Psi_\lambda(y_k, y'_k))$ (Line 5 of Algorithm 1) we get

$$\left\langle \nabla_x \Psi_\lambda(y_k, y'_k), z_k - y_k \right\rangle \le \left\langle \nabla_x \Psi_\lambda(y_k, y'_k), x - y_k \right\rangle . \tag{15}$$

Line 8 of Algorithm 1 implies that $z'_k \in \mathcal{X}'$ since $\mathcal{X}'$ is a ball of radius R. Thus, using Lines 7 and 8 of Algorithm 1 and $x' \in \mathcal{X}'$ we get

$$\begin{aligned} & z'_k \in \operatorname*{argmin}_{x' \in \mathcal{X}'} \frac{\beta_k}{2}\|x' - (z'_{k-1} - (\nabla_{k,x'} + \widehat{g}_k)/\beta_k)\|^2 \\ \implies & \left\langle \nabla_{k,x'} + \widehat{g}_k, z'_k \right\rangle + \frac{\beta_k}{2}\|z'_k - z'_{k-1}\|^2 \le \left\langle \nabla_{k,x'} + \widehat{g}_k, x' \right\rangle + \frac{\beta_k}{2}(\|z'_{k-1} - x'\|^2 - \|z'_k - x'\|^2) \end{aligned} \tag{16}$$

Now substituting (14), (15) and (16) into (13) and using the fact that $\delta_k = \widehat{g}_k - g_k$ we get

$$\begin{aligned} k(k+1)\Psi_\lambda(x_k, x'_k) \le\ & k(k-1)\Psi_\lambda(x_{k-1}, x'_{k-1}) + 2k\Psi_\lambda(y_k, y'_k) + 2k\left\langle \nabla_{k,x}, x - y_k \right\rangle + \\ & 2k[\left\langle \nabla_{k,x'} + g_k, x' - y'_k \right\rangle + \frac{\beta_k}{2}(\|z'_{k-1} - x'\|^2 - \|z'_k - x'\|^2)] + \\ & \frac{4}{\lambda}\|z_k - z_{k-1}\|^2 + 2k[\frac{k\,\eta_k}{2}(2G + \|\delta_k\|)^2 + \left\langle \delta_k, x' - z'_{k-1} \right\rangle] \\ \le\ & k(k-1)\Psi_\lambda(x_{k-1}, x'_{k-1}) + 2k\Psi_\lambda(x, x') + \\ & (\frac{4}{\lambda} + \frac{1}{\eta_{k-1}})\|z'_{k-1} - x'\|^2 - (\frac{4}{\lambda} + \frac{1}{\eta_k})\|z'_k - x'\|^2 + \\ & (\frac{1}{\eta_k} - \frac{1}{\eta_{k-1}})\|z'_{k-1} - x'\|^2 + \\ & \frac{4}{\lambda}\|z_k - z_{k-1}\|^2 + \eta_k k^2(2G + \|\delta_k\|)^2 + 2k\left\langle \delta_k, x' - z'_{k-1} \right\rangle . \end{aligned} \tag{17}$$

where the last inequality uses $\Psi_\lambda = f + \psi_\lambda$ and the convexity of Ψ_λ, and the definition of $\beta_k = \frac{4}{k\lambda} + \frac{1}{k\eta_k}$ (Line 3 of Algorithm 1). This proves the following approximate descent guarantee for the potential Φ_k (10):

$$\Phi_k \leq \Phi_{k-1} + \frac{4}{\lambda}\|z_k - z_{k-1}\|^2 + (\frac{1}{\eta_k} - \frac{1}{\eta_{k-1}})\|z'_{k-1} - x'\|^2 + \eta_k k^2 (2G + \|\delta_k\|)^2 + 2k\left\langle \delta_k, x' - z'_{k-1}\right\rangle . \tag{18}$$

Summing these descent lemmas for $j = 1, \ldots, k$ and taking expectation, with respect to the randomness in the stochastic subgradients $(\widehat{g}_k)_{k=1}^K$ used in the algorithm, on both sides we get

$$\begin{aligned}\mathbb{E}[\Phi_k] \leq \Phi_0 + &\sum_{j=1}^{k} \mathbb{E}[\frac{4}{\lambda}\|z_j - z_{j-1}\|^2 + (\frac{1}{\eta_j} - \frac{1}{\eta_{j-1}})\|z'_{j-1} - x'\|^2] + \\ &\sum_{j=1}^{k} \mathbb{E}[\eta_k\, j^2 (2G + \|\delta_j\|)^2 + 2j\left\langle \delta_j, x' - z'_{j-1}\right\rangle] .\end{aligned} \tag{19}$$

Then the expectation of the sum of last two term can be bounded as follows

$$\mathbb{E}[\eta_j\, j^2\, (2G + \|\delta_j\|)^2 - 2j\left\langle \delta_j, x' - z'_{j-1}\right\rangle] \leq \eta_j\, 2\, j^2\, (4G^2 + \sigma^2) + 0\,, \tag{20}$$

where we use linearity of expectation, $(a + b)^2 \leq 2(a^2 + b^2)$, variance of stochastic gradient $\mathbb{E}[\|\delta_k\|^2] = \sigma^2$ (4), and the fact that expectation of the second term becomes zero. The latter follows from the definition of stochastic gradient $\mathbb{E}[\widehat{g}_j \mid y'_j] = g_j$ (4), which in turn implies that

$$\mathbb{E}[\left\langle \delta_j, x' - z'_{j-1}\right\rangle] = \mathbb{E}\big[\,\mathbb{E}[\left\langle \widehat{g}_j - g_j, x' - z'_{j-1}\right\rangle \mid z'_{j-1}, y'_j]\,\big] = \mathbb{E}[\left\langle 0, z'_{k-1} - y'_k\right\rangle] = 0\,. \tag{21}$$

Next using (20) in (19) we get that

$$\mathbb{E}[\Phi_k] \leq \Phi_0 + \sum_{j=1}^{k} \mathbb{E}[\frac{4}{\lambda}\|z_j - z_{j-1}\|^2 + (\frac{1}{\eta_j} - \frac{1}{\eta_{j-1}})\|z'_{j-1} - x'\|^2] + 2(4G^2 + \sigma^2)\sum_{j=1}^{k} \eta_j\, j^2 \tag{22}$$

This directly implies the following convergence guarantee for Ψ_λ for all $k \geq 1$.

$$\begin{aligned}\mathbb{E}[\Psi_\lambda\left(x_k, x'_k\right)] - \Psi_\lambda(x, x') \leq (\frac{4}{\lambda} + \frac{1}{\eta_0})\frac{\|x_0 - x'\|^2}{k(k+1)} + \frac{4}{\lambda k(k+1)}\sum_{j=1}^{k}\mathbb{E}[\|z_j - z_{j-1}\|^2] + \\ \sum_{j=1}^{k}(\frac{1}{\eta_j} - \frac{1}{\eta_{j-1}})\frac{\mathbb{E}[\|z'_{j-1} - x'\|^2]}{k(k+1)} + \frac{2(4G^2 + \sigma^2)}{k(k+1)}\sum_{j=1}^{k}\eta_j\, j^2\end{aligned} \tag{23}$$

where we use the initialization $z'_0 = x_0$. Next setting $x = x' = x^* \in \mathcal{X} \subset \mathcal{X}'$ for some $x^* \in \operatorname{argmin}_{x \in \mathcal{X}} f(x)$, assuming $\eta_j \leq \eta_{j-1}$ and $\eta_0^{-1} \in \mathbb{R}$, and then using Lemma 8, (6) and the diameters

$D_{\mathcal{X}}$ of $\mathcal{X}$ and $2R$ of $\mathcal{X}'$ we get that

$$\begin{aligned}\mathbb{E}[f\,(x_k)] - f(x^*) \le\, & (\frac{4}{\lambda} + \frac{1}{\eta_0})\frac{\|x_0 - x^*\|^2}{k(k+1)} + \frac{4}{\lambda k(k+1)}\sum_{j=1}^{k}\mathbb{E}[\|z_j - z_{j-1}\|^2] + G^2\frac{\lambda}{2} + \\ & \sum_{j=1}^{k}(\frac{1}{\eta_j} - \frac{1}{\eta_{j-1}})\frac{\mathbb{E}[\|z'_{j-1} - x^*\|^2]}{k(k+1)} + \frac{2(4G^2+\sigma^2)}{k(k+1)}\sum_{j=1}^{k}\eta_j\, j^2 \\ = & \frac{4D_{\mathcal{X}}^2}{\lambda k} + G^2\frac{\lambda}{2} + \frac{D_{\mathcal{X}}^2}{\eta_0\, k(k+1)} + \frac{4R^2}{k(k+1)}(\frac{1}{\eta_k} - \frac{1}{\eta_0}) + \frac{2(4G^2+\sigma^2)}{k(k+1)}\sum_{j=1}^{k}\eta_j\, j^2\,.\end{aligned} \tag{24}$$

For part (a) setting $k = K$ and using the the given parameter choices, $\lambda = \frac{2\sqrt{2}\,D_{\mathcal{X}}}{G\sqrt{K}}$ and $\eta_j = \eta = \frac{\sqrt{3}\,D_{\mathcal{X}}}{\sqrt{2\,K^3\,(4G^2+\sigma^2)}}$ for all $0 \le j \le K$, and $\sum_{j=1}^{K} j^2 = \frac{K(K+1)(2K+1)}{6}$ we get

$$\mathbb{E}[f\,(x_K)] - f(x^*) \le \frac{2\sqrt{2}\,GD_{\mathcal{X}}}{\sqrt{K}} + \frac{2\sqrt{2}\sqrt{4G^2+\sigma^2}D_{\mathcal{X}}}{\sqrt{3}\,\sqrt{K}} \tag{25}$$

For part (b) using the the given parameter choices, $\lambda = \frac{2\sqrt{2}\,D_{\mathcal{X}}}{G\sqrt{K}}$, $\eta_j = \frac{\sqrt{3}\,R}{\sqrt{j^3\,(4G^2+\sigma^2)}}$ for all $1 \le j \le K$, $\eta_0^{-1} := 0$, $\sum_{j=1}^{k}\sqrt{j} \le (2/3)\sqrt{k}(k+1)$, we get

$$\begin{aligned}\mathbb{E}[f\,(x_k)] - f(x^*) \le & \frac{\sqrt{2}GD_{\mathcal{X}}}{\sqrt{K}}(\frac{K}{k} + 1) + \frac{4R^2\sqrt{k}}{\eta_1(k+1)} + \frac{\eta_1 4(4G^2+\sigma^2)}{\sqrt{3}\,\sqrt{k}} + \\ \le & \frac{\sqrt{2}GD_{\mathcal{X}}}{\sqrt{K}}(\frac{K}{k} + 1) + \frac{8\sqrt{4G^2+\sigma^2}R}{\sqrt{3}\,\sqrt{k}}\,.\end{aligned} \tag{26}$$

Setting, $k = K$, we get a last iterate convergence result like part (a)

$$\mathbb{E}[f\,(x_K)] - f(x^*) \le \frac{2\sqrt{2}\,GD_{\mathcal{X}}}{\sqrt{K}} + \frac{8\,\sqrt{4G^2+\sigma^2}R}{\sqrt{3}\,\sqrt{K}}\,. \tag{27}$$

Finally, summing (26) inequality for $k = 1, \ldots, K$, and using $\sum_{k=1}^{K}\frac{1}{k} \le \ln(K)+1$ and $\sum_{k=1}^{K}\frac{1}{\sqrt{k}} \le 2\sqrt{K}$, we get the desired regret bound

$$\sum_{k=1}^{K}\mathbb{E}\big[f\,(x_k) - f(x^*)\big] \le \sqrt{2}GD_{\mathcal{X}}\sqrt{K}(\ln(K)+2) + \frac{16}{\sqrt{3}}\sqrt{4G^2+\sigma^2}R\sqrt{K}\,. \tag{28}$$

Then using the above bound, the diameter $D_{\mathcal{X}}$ of $\mathcal{X}$, and the fact that $\sum_{k=1}^{K}\gamma_k = \sum_{k=1}^{K} 2/(k+1) \leq 2\ln(K+1)$ we get

$$\begin{aligned}\sum_{k=1}^{K}\mathbb{E}\big[f(x_{k-1}) - f(x^*)\big] &\leq \sum_{k=1}^{K}\mathbb{E}\big[f\left(x_k\right) - f(x^*)\big] + G\,\mathbb{E}[\|x_{k-1} - x_k\|] \\ &\leq \sum_{k=1}^{K}\mathbb{E}\big[f\left(x_k\right) - f(x^*)\big] + \sum_{k=1}^{K}\gamma_k G\,\mathbb{E}[\|x_{k-1} - z_k\|] \\ &\leq \sqrt{2}GD_{\mathcal{X}}\sqrt{K}(\ln(K)+2) + \frac{16}{\sqrt{3}}\sqrt{4G^2+\sigma^2}R\sqrt{K} + \\ &\quad 2GD_{\mathcal{X}}\ln(K+1) \end{aligned} \tag{29}$$

■

Appendix B. Proof of Theorem 6

Proof [Proof of Theorem 6] For some arbitrary $(x, x') \in \mathcal{X}\times\mathcal{X}'$, consider the following potential function which is different from (10).

$$\Phi_k := k(k+1)\Psi_\lambda(x_k, x_k') - 2l_k(z_k, z_k') - k\,\beta_k\|z_k' - z_0'\|^2\,, \tag{30}$$

where

$$l_k(z_k, z_k') := [\sum_{j=1}^{k} j(\langle\nabla_{j,x}, z_k - y_k\rangle + \left\langle\nabla_{j,x'} + \widehat{g}_j, z_k' - y_k'\right\rangle)]\,. \tag{31}$$

Using (13) we get,

$$\begin{aligned}&k(k+1)\Psi_\lambda(x_k, x_k') \\ \leq\;& k(k-1)\Psi_\lambda(x_{k-1}, x_{k-1}') + 2k\Psi_\lambda(y_k, y_k') + 2k\left\langle\nabla_{k,x}, z_k - y_k\right\rangle + \\ & 2k[\left\langle\nabla_{k,x'} + \widehat{g}_k, z_k' - y_k'\right\rangle + \frac{\beta_k}{2}\|z_k' - z_{k-1}'\|^2] + \frac{4}{\lambda}\|z_k - z_{k-1}\|^2 + \\ & 2k[-\frac{1}{2k\,\eta_k}\|z_k' - z_{k-1}'\|^2 + (2G + \|\delta_k\|)\|z_k' - z_{k-1}'\| - \left\langle\delta_k, z_{k-1}' - y_k'\right\rangle]\end{aligned} \tag{32}$$

We can bound the third and fourth terms on the RHS above as follows. First using $z_{k-1} = \text{LMO}\left(\sum_{j=1}^{k-1} j\cdot\nabla_{y_j}\Psi_\lambda(y_j, y_j')\right)$ (Line 5 of Algorithm 2) we get

$$\begin{aligned}2k\,\left\langle\nabla_{k,x}, z_k - y_k\right\rangle &= 2[\sum_{j=1}^{k} j\,\left\langle\nabla_{j,x}, z_k - y_j\right\rangle] - 2[\sum_{j=1}^{k-1} j\,\left\langle\nabla_{j,x}, z_k - y_j\right\rangle] \\ &\leq 2[\sum_{j=1}^{k} j\,\left\langle\nabla_{j,x}, z_k - y_j\right\rangle] - 2[\sum_{j=1}^{k-1} j\,\left\langle\nabla_{j,x}, z_{k-1} - y_j\right\rangle]\,.\end{aligned} \tag{33}$$

Line 8 of Algorithm 2 implies that $z'_{k-1} \in \mathcal{X}'$ since $\mathcal{X}'$ is a ball of radius R. Thus, using Lines 7 and 8 of Algorithm 2 and $x' \in \mathcal{X}'$ we get

$$z'_i \in \operatorname*{argmin}_{x' \in \mathcal{X}'} \frac{i\,\beta_i}{2} \|x' - (z'_0 - \frac{1}{i\,\beta_i} \sum_{j=1}^{i} j \cdot (\nabla_{j,x'} + \widehat{g}_k))\|^2 \tag{34}$$

Using the $(k-1)\beta_{k-1}$-strong convexity of the above projection problem at $i = k-1$ and the optimality of the projection z'_{k-1} we get

$$\begin{aligned} & k \left\langle \nabla_{k,x'} + \widehat{g}_k, z'_k - y'_k \right\rangle \\ & \leq [\sum_{j=1}^{k} j \cdot \left\langle \nabla_{j,x'} + \widehat{g}_j, z'_k - y'_k \right\rangle] - [\sum_{j=1}^{k-1} j \cdot \left\langle \nabla_{j,x'} + \widehat{g}_j, z'_k - y'_k \right\rangle] \\ & \leq [\sum_{j=1}^{k} j \cdot \left\langle \nabla_{j,x'} + \widehat{g}_j, z'_k - y'_k \right\rangle] + \frac{(k-1)\beta_{k-1}}{2} \|z'_k - z'_0\|^2 \\ & \quad - [\sum_{j=1}^{k-1} j \cdot \left\langle \nabla_{j,x'} + \widehat{g}_j, z'_{k-1} - y'_k \right\rangle] - \frac{(k-1)\beta_{k-1}}{2} (\|z'_{k-1} - z'_0\|^2 + \|z'_k - z'_{k-1}\|^2) \end{aligned} \tag{35}$$

Now substituting (14), (33) and (35) into (32) we get

$$\begin{aligned} k(k+1)\Psi_\lambda(x_k, x'_k) \leq {} & k(k-1)\Psi_\lambda(x_{k-1}, x'_{k-1}) - 2l_{k-1}(z_{k-1}, z'_{k-1}) - (k-1)\beta_{k-1}\|z'_{k-1} - z'_0\|^2 + \\ & 2l_k(z_k, z'_k) + k\beta_k \|z'_k - z'_0\|^2 + (\frac{1}{\eta_k} - \frac{1}{\eta_{k-1}})\|z'_k - z'_{k-1}\|^2 + \\ & \frac{4}{\lambda}\|z_k - z_{k-1}\|^2 + \eta_k k^2 (2G + \|\delta_k\|)^2 + 2k \left\langle \delta_k, y'_{k-1} - z'_{k-1} \right\rangle \end{aligned} \tag{36}$$

where the last inequality uses the definition of $\beta_k = \frac{8}{k\lambda} + \frac{1}{k\eta_k}$ (Line 3 of Algorithm 2). This proves the following approximate descent guarantee for the potential Φ_k (30):

$$\Phi_k \leq \Phi_{k-1} + \frac{4}{\lambda}\|z_k - z_{k-1}\|^2 + (\frac{1}{\eta_k} - \frac{1}{\eta_{k-1}})\|z'_j - z'_{j-1}\|^2 + \eta_k 2k^2 (2G + \|\delta_k\|)^2 + 2k \left\langle \delta_k, y'_{k-1} - z'_{k-1} \right\rangle . \tag{37}$$

Then using arguments similar to the ones used to get (23) we can get the following convergence guarantee

$$\begin{aligned} & \mathbb{E}[\Psi_\lambda\left(x_k, x'_k\right) - \frac{2l_k(z_k, z'_k)}{k(k+1)} - \frac{2k\beta_k}{2k(k+1)}\|z'_k - z'_0\|^2] \\ & \leq \frac{4}{\lambda k(k+1)} \sum_{j=1}^{k} \mathbb{E}[\|z_j - z_{j-1}\|^2] + \sum_{j=1}^{k} (\frac{1}{\eta_j} - \frac{1}{\eta_{j-1}}) \frac{\mathbb{E}[\|z'_j - z'_{j-1}\|^2]}{k(k+1)} + \frac{2(4G^2 + \sigma^2)}{k(k+1)} \sum_{j=1}^{k} \eta_j\, j^2 . \end{aligned} \tag{38}$$

Next, using the definition of l_k (13), the $k\,\beta_k$-strong convexity of the projection problem (34) at $i = k$ and the optimality of the projection z'_k, and $z_k = \text{LMO}\left(\sum_{j=1}^{k} j \cdot \nabla_{y_j} \Psi_\lambda(y_j, y'_j)\right)$ (Line 5 of Algorithm 2) we get

$$l_k(x, x') + \frac{k\,\beta_k}{2}\|z'_k - x_0\|^2 \leq l_k(x, x') + \frac{k\,\beta_k}{2}(\|x' - z'_0\|^2 - \|x' - z'_k\|^2) . \tag{39}$$

Using the definition of stochastic gradient $\mathbb{E}[\widehat{g}_j \mid y_j'] = g_j$ (4), $\Psi_\lambda = f + \psi_\lambda$, and convexity of Ψ_λ we can show that

$$\begin{aligned}
\mathbb{E}[l_k(z_k, z_k')] &= \mathbb{E}[\sum_{j=1}^{k} j(\Psi_\lambda(y_j, y_j') + \langle \nabla_{j,x}, x - y_j \rangle + \left\langle \nabla_{j,x'} + \widehat{g}_j, x' - y_j' \right\rangle)] \\
&= \mathbb{E}[\sum_{j=1}^{k} j(\Psi_\lambda(y_j, y_j') + \langle \nabla_{j,x}, x - y_j \rangle + \mathbb{E}[\left\langle \nabla_{j,x'} + \widehat{g}_j, x' - y_j' \right\rangle \mid y_j'])] \\
&= \mathbb{E}[\sum_{j=1}^{k} j(\Psi_\lambda(y_j, y_j') + \langle \nabla_{j,x}, x - y_j \rangle + \left\langle \nabla_{j,x'} + g_j, x' - y_j' \right\rangle)] \\
&\le \frac{k(k+1)}{2} \Psi_\lambda(x, x') \,.
\end{aligned} \tag{40}$$

Substituting (39), (40) and $k\beta_k = \frac{4}{\lambda} + \frac{1}{\eta_k}$ (Line 3 of Algorithm 2) in (38) and using the initialization $z_0' = x_0$ we get

$$\begin{aligned}
\mathbb{E}[\Psi_\lambda\left(x_k, x_k'\right)] - \Psi_\lambda(x, x') \le (\frac{4}{\lambda} + \frac{1}{\eta_k}) \frac{\|x_0 - x'\|^2}{k(k+1)} + \frac{4}{\lambda k(k+1)} \sum_{j=1}^{k} \mathbb{E}[\|z_j - z_{j-1}\|^2] + \\
\sum_{j=1}^{k} (\frac{1}{\eta_j} - \frac{1}{\eta_{j-1}}) \frac{\mathbb{E}[\|z_j' - z_{j-1}'\|^2]}{k(k+1)} + \frac{2(4G^2 + \sigma^2)}{k(k+1)} \sum_{j=1}^{k} \eta_j \, j^2
\end{aligned} \tag{41}$$

Next setting $x = x' = x^* \in \mathcal{X} \subset \mathcal{X}'$ for some $x^* \in \operatorname{argmin}_{x \in \mathcal{X}} f(x)$, assuming $\eta_j \le \eta_{j-1}$ and $\eta_0^{-1} \in \mathbb{R}$, and then using Lemma 8, (6) and the diameters $D_{\mathcal{X}}$ of $\mathcal{X}$ and $2R$ of $\mathcal{X}'$ we get that

$$\begin{aligned}
\mathbb{E}[f\left(x_k\right)] - f(x^*) \le (\frac{4}{\lambda} + \frac{1}{\eta_k}) \frac{\|x_0 - x^*\|^2}{k(k+1)} + \frac{4}{\lambda k(k+1)} \sum_{j=1}^{k} \mathbb{E}[\|z_j - z_{j-1}\|^2] + G^2 \frac{\lambda}{2} + \\
\sum_{j=1}^{k} (\frac{1}{\eta_j} - \frac{1}{\eta_{j-1}}) \frac{\mathbb{E}[\|z_j' - z_{j-1}'\|^2]}{k(k+1)} + \frac{2(4G^2 + \sigma^2)}{k(k+1)} \sum_{j=1}^{k} \eta_j \, j^2 \\
= \frac{4D_{\mathcal{X}}^2}{\lambda k} + G^2 \frac{\lambda}{2} + \frac{D_{\mathcal{X}}^2}{\eta_k \, k(k+1)} + \frac{4R^2}{k(k+1)} (\frac{1}{\eta_k} - \frac{1}{\eta_0}) + \frac{2(4G^2 + \sigma^2)}{k(k+1)} \sum_{j=1}^{k} \eta_j \, j^2 \,.
\end{aligned} \tag{42}$$

Part (a) can be proved exactly like we the proved Theorem 5 (a) (25). Similarly, for part (b), we can obtain the following inequalities, using similar arguments like in the proof of Theorem 5 (b), by noting that the above inequality (42) differs from (24) only in the third term, as this term scales with $1/\eta_k$ instead of $1/\eta_0$.

$$\mathbb{E}[f\left(x_k\right)] - f(x^*) \le \frac{\sqrt{2} G D_{\mathcal{X}}}{\sqrt{K}} \left(\frac{K}{k} + 1\right) + \frac{\sqrt{4G^2 + \sigma^2} D_{\mathcal{X}}^2}{\sqrt{3}\,\sqrt{k} R} + \frac{8\sqrt{4G^2 + \sigma^2} R}{\sqrt{3}\,\sqrt{k}} \tag{43}$$

$$\mathbb{E}[f\left(x_K\right)] - f(x^*) \le \frac{2\sqrt{2}\, G D_{\mathcal{X}}}{\sqrt{K}} + \frac{\sqrt{4G^2 + \sigma^2} R}{\sqrt{3}\,\sqrt{K}} \left(8 + \frac{D_{\mathcal{X}}^2}{R^2}\right) \tag{44}$$

$$\sum_{k=1}^{K} \mathbb{E}\big[f(x_{k-1}) - f(x^*)\big] \leq \sqrt{2}GD_{\mathcal{X}}\sqrt{K}(\ln(K)+2) + \frac{2}{\sqrt{3}}\sqrt{4G^2+\sigma^2}R\sqrt{K}\Big(8 + \frac{D_{\mathcal{X}}^2}{R^2}\Big) + 2GD_{\mathcal{X}}\ln(K+1) \tag{45}$$

■