Sarah Frank-Wolfe: Methods for Constrained Optimization with Best Rates and Practical Features

Aleksandr Beznosikov [1 2 3 4] **David Dobre** [5] **Gauthier Gidel** [5 6]

Abstract

The Frank-Wolfe (FW) method is a popular approach for solving optimization problems with structured constraints that arise in machine learning applications. In recent years, stochastic versions of FW have gained popularity, motivated by large datasets for which the computation of the full gradient is prohibitively expensive. In this paper, we present two new variants of the FW algorithms for stochastic finite-sum minimization. Our algorithms have the best convergence guarantees of existing stochastic FW approaches for both convex and non-convex objective functions. Our methods do not have the issue of permanently collecting large batches, which is common to many stochastic projection-free approaches. Moreover, our second approach does not require either large batches or full deterministic gradients, which is a typical weakness of many techniques for finite-sum problems. The faster theoretical rates of our approaches are confirmed experimentally.

1. Introduction

Empirical risk minimization is a cornerstone for training supervised machine learning models such as various regressions, support vector machine, and neural networks (Shalev-Shwartz & Ben-David, 2014). We consider a constrained problem of this type:

$$\min_{x\in\mathcal{X}\subset\mathbb{R}^d} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)\,. \tag{1}$$

The objective function of (1) has the form of a finite sum. Typically, this setting corresponds to the sum of the losses of the model with parameters x applied to a large number of data points, indexed by $i = 1, \ldots, n$. Because n is large, calculating the full gradient of f is expensive. Therefore, stochastic methods which very rarely resort to calling ∇f (or avoid it altogether) are of particular importance. In this problem setting, we assume the set $\mathcal{X} \subset \mathbb{R}^d$ to be convex, that projecting onto this set is expensive, and that it also admits a fast linear minimization oracle (LMO).

The study of methods for (1) that do not require projections has a history of more than half a century. Arguably the most popular projection-free method is the Frank-Wolfe (a.k.a, Conditional Gradient algorithm (Frank & Wolfe, 1956)). This method maintains sparse iterates and only requires a linear minimization oracle that takes into account the specificity of the constraint set $\mathcal{X}$. In particular, the classical version of the method considers a linear approximation of the function at the current point x^k, and minimizes this approximation on the set $\mathcal{X}$:

$$\begin{aligned} s^k &= \arg\min_{s\in\mathcal{X}} \langle \nabla f(x^k), s - x^k\rangle, \\ x^{k+1} &= x^k + \eta_k(s^k - x^k), \end{aligned} \tag{2}$$

where η_k is parameter-free and equal to $\frac{2}{k+2}$.

In the last decade, the Frank-Wolfe-type approaches have attracted increasing interest in the machine-learning community because of their good performance on sparse problems, or on problems where the constraints are complex but structured (e.g., various ℓ_p balls, trace norms), having applications to submodular optimization (Bach, 2011), vision (Miech et al., 2017; Bojanowski et al., 2014), and variational inference (Krishnan et al., 2015).

Due to the significant increase in dataset size and complexity within the machine learning community, stochastic algorithms are of great interest and are the focus of this paper. In particular, we seek to answer the two following questions:

1. Can we improve upon the convergence rates of existing approaches for (1)?

2. Can we avoid the computation of both full gradients and large batches of stochastic gradients?

[1]Innopolis University, Russia [2]Skolkovo Institute of Science and Technology, Russia [3]Moscow Institute of Physics and Technology, Russia [4]Yandex, Russia [5]Universite de' Montreal and Mila, Canada [6]Canada CIFAR AI Chair. Correspondence to: Aleksandr Beznosikov <anbeznosikov@gmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Table 1: Summary of the results on projection free methods for **stochastic constrained minimization problems**.

Reference	Convex case complexity		Non-convex case complexity		No full gradients?	No big batches?
	SFO	LMO	SFO	LMO		
(Frank & Wolfe, 1956)[1] (Lacoste-Julien, 2016)[1]	$\mathcal{O}\left(\frac{n}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
(Hazan & Kale, 2012)	$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗		✓	✗
(Lan & Zhou, 2016)	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon}\right)$	✗		✓	✗
(Lan & Zhou, 2016) [1]	$\mathcal{O}\left(\frac{n}{\sqrt{\varepsilon}}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon}\right)$	✗		✗	✗
(Reddi et al., 2016) Alg. 2	✗		$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✓	✗
(Reddi et al., 2016) Alg. 3	✗		$\mathcal{O}\left(n+\frac{n^{2/3}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
(Reddi et al., 2016) Alg. 4	✗		$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$ [2]	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
(Hazan & Luo, 2016)	$\tilde{\mathcal{O}}\left(n+\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon}\right)$	✗		✗	✗
(Qu et al., 2018) Alg. 3	✗		$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$ [3]	$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$ [3]	✓	✗
(Qu et al., 2018) Alg. 4	✗		$\mathcal{O}\left(n+\frac{n^{2/3}}{\varepsilon^2}\right)$ [3]	$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$ [3]	✗	✗
(Yurtsever et al., 2019)	$\mathcal{O}\left(n+\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(n+\frac{\sqrt{n}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
(Gao & Huang, 2020) Alg. 1	✗		$\mathcal{O}\left(n+\frac{\sqrt{n}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
(Gao & Huang, 2020) Alg. 2	✗		$\mathcal{O}\left(n+\frac{\sqrt{n}}{\varepsilon^2}\right)$ [3]	$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$ [3]	✗	✗
(Mokhtari et al., 2020)	$\mathcal{O}\left(\frac{1}{\varepsilon^3}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^3}\right)$	✗		✓	✓
(Zhang et al., 2020)	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗		✓	✓
(Négiar et al., 2020)[4]	$\mathcal{O}\left(\frac{n}{\varepsilon}\right)$ [5]	$\mathcal{O}\left(\frac{n}{\varepsilon}\right)$ [5]	convergence without rate		✓	✓
(Lu & Freund, 2021)[4]	$\mathcal{O}\left(\frac{n}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{n}{\varepsilon}\right)$	✗		✓	✓
(Akhtar & Rajawat, 2021)	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗		✓	✓
(Weber & Sra, 2022) Alg.2	$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^4}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✓	✗
(Weber & Sra, 2022) Alg.3	$\mathcal{O}\left(n+\frac{n^{2/3}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(n+\frac{n^{2/3}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
(Weber & Sra, 2022) Alg.4	$\mathcal{O}\left(\frac{1}{\varepsilon^3}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^3}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✓	✗
(Hou et al., 2022)	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$\mathcal{O}\left(\exp\left(\frac{1}{\varepsilon}\right)\right)$	$\mathcal{O}\left(\exp\left(\frac{1}{\varepsilon}\right)\right)$	✓	✓
(This paper) Alg. 1	$\tilde{\mathcal{O}}\left(n+\frac{\sqrt{n}}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(\sqrt{n}+\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(n+\frac{\sqrt{n}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✗	✗
(This paper) Alg. 1	$\tilde{\mathcal{O}}\left(n+\frac{\sqrt{n}}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(n+\frac{\sqrt{n}}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$	✗	✓
(This paper) Alg. 2	$\tilde{\mathcal{O}}\left(n+\frac{\sqrt{n}}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(\sqrt{n}+\frac{1}{\varepsilon}\right)$	$\mathcal{O}\left(n+\frac{\sqrt{n}}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	✓	✗
(This paper) Alg. 2	$\tilde{\mathcal{O}}\left(n+\frac{\sqrt{n}}{\varepsilon}\right)$	$\tilde{\mathcal{O}}\left(n+\frac{\sqrt{n}}{\varepsilon}\right)$	$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$	$\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$	✓	✓

[1] fully deterministic; [2] the authors give a different complexity, but it seems to us that their proof contains an error, we try to correct it (see Appendix C.2); [3] in the original papers, the authors give better results, e.g. $\mathcal{O}(\sqrt{n}/\varepsilon)$ instead of $\mathcal{O}(\sqrt{n}/\varepsilon^2)$, but this results violate the lower bounds (see Table 1 from (Li et al., 2021a)]), this is due to the difference in the convergence criterion: in (Li et al., 2021a), the authors use $\|\nabla f\|^2 \sim \varepsilon^2$ and in (Qu et al., 2018; Gao & Huang, 2020) – $\|\nabla f\|^2 \sim \varepsilon$; [4] only for linear models; [5] the authors give a rate in the form κ/ε, where κ is a special constant, which is equal to n in the worst case.
Notation: ε = accuracy of the solution, n = size of the dataset, SFO = stochastic first-order oracle, LMO = linear minimization oracle.

2. Related Works and Our Contributions

After Frank & Wolfe (1956) proposed the FW algorithm, many works improved its theory and extended it to special cases (Levitin & Polyak, 1966; Demianov & Rubinov, 1970; Dunn & Harshbarger, 1978; Patriksson, 1993). About ten years ago, Jaggi (2013); Lacoste-Julien & Jaggi (2015) developed more robust and practical versions of the original FW method, motivated by ML applications with sparsity and structured constraints (see (Braun et al., 2022) for a detailed historical survey).

Motivated by applications with large datasets, the theory of stochastic methods for unconstrained (or projection-friendly) optimization problems has built upon the highly successful SGD method (Robbins & Monro, 1951; Nemirovski et al., 2009) to obtain faster methods for finite sum-problems. Particularly, many so-called variance-reduced variants of SGD have been proposed, including SAG/SAGA (Defazio et al., 2014; Schmidt et al., 2017; Qian et al., 2019), SVRG (Johnson & Zhang, 2013; Allen-Zhu & Yuan, 2016; Yang et al., 2021), MISO (Mairal, 2015), SARAH (Nguyen et al., 2017a; 2021; 2017b; Hu et al., 2019; Li et al., 2021b), SPIDER (Fang et al., 2018), STORM (Cutkosky & Orabona, 2019), PAGE (Li et al., 2021a), and many others.

Extensive research in the theory of deterministic Frank-Wolfe-type methods and stochastic methods for unconstrained problems has led to the development of stochastic versions of projection-free algorithms. Hazan & Kale (2012) proposed an algorithm for online stochastic optimization. Lan & Zhou (2016) developed a projection-free version using sliding. Hazan & Luo (2016); Reddi et al. (2016); Qu et al. (2018); Yurtsever et al. (2019); Gao & Huang (2020); Shen et al. (2019) proposed modifications of the Frank-Wolfe method using variance reduction techniques, namely SVRG, SAGA and SPIDER. Mokhtari et al. (2020); Akhtar & Rajawat (2021); Hou et al. (2022) used the idea of momentum to deal with stochasticity. Négiar et al. (2020) and Lu & Freund (2021) explored stochastic methods for linear predictors. (Weber & Sra, 2022) extended the results of Reddi et al. (2016) from convex sets to manifolds. We summarize and compare the convergence rate of each method in Table 1. Note that for the SAGA-related methods, we report a slightly different result from the one reported by Reddi et al. (2016) because we believe that their proof contains a slight inaccuracy (see App. C.2 for more details). We also do not include the approach from (Shen et al., 2019) in Table 1, since this method uses the hessian of the target function.

Next, we detail our contributions which can be divided into four parts.

• **The best rates in the convex case.** Our convergence guarantees are better than the classical deterministic method (Frank & Wolfe, 1956; Lan & Zhou, 2016) as well as the stochastic methods from (Négiar et al., 2020; Lu & Freund, 2021; Weber & Sra, 2022) in terms of dataset size n. Moreover, the theoretical rates of our methods also surpasses the rest existing results from (Hazan & Kale, 2012; Lan & Zhou, 2016; Hazan & Luo, 2016; Yurtsever et al., 2019; Gao & Huang, 2020; Mokhtari et al., 2020; Akhtar & Rajawat, 2021; Weber & Sra, 2022) in terms of the accuracy ε.

• **No need for full gradients.** Many stochastic methods, especially for finite-sum problems, require the calculation of some full gradients. This makes these techniques less practical because even the infrequent computation of the deterministic gradient can slow down the convergence. Some methods for constrained problems also have this disadvantage (Reddi et al., 2016; Hazan & Luo, 2016; Qu et al., 2018; Yurtsever et al., 2019; Gao & Huang, 2020; Weber & Sra, 2022). While Algorithm 1 also requires the computation of the full gradient, Algorithm 2 removes this issue and uses only stochastic gradients. Note that this modification does not affect convergence: Algorithm 1 and Algorithm 2 have the same theoretical guarantees.

• **Small batches.** Many methods that avoid the computation of the full gradient still use large fixed batch sizes (Hazan & Kale, 2012; Lan & Zhou, 2016; Reddi et al., 2016; Qu et al., 2018; Gao & Huang, 2020; Weber & Sra, 2022) or batch sizes that geometrically increase with iteration number (Hazan & Luo, 2016; Yurtsever et al., 2019), which, like with the collection of full gradients, is a rather strong limitation on the practical applicability of the method. Conversely, our algorithms are guaranteed to converge with all sizes of batches. Large batches are required to get a better dependence in n for the non-convex case. In particular, it is possible to improve the LMO estimate by a factor of n, and the estimate on SFO by a factor of $\sqrt{n}$. Methods dealing with fixed small batches are either only analyzed for linear predictors (Négiar et al., 2020; Lu & Freund, 2021), or have slower convergence rates than our approach (Mokhtari et al., 2020; Akhtar & Rajawat, 2021; Hou et al., 2022).

• **Non-convex analysis.** We give convergence results not only for the convex problem, but also in the case where the target function f in (1) is non-convex. In this setting, our oracle complexity results are the first to be non-exponential (in ϵ) with small mini-batches, and are state-of-the-art with large mini-batches.

3. Notation and Assumptions

We use $\langle x, y\rangle = \sum_{i=1}^{d} x_i y_i$ to denote the standard inner product of vectors $x, y \in \mathbb{R}^d$, where x_i corresponds to the i-th component of x in the standard basis in $\mathbb{R}^n$. With this notation we can introduce the standard ℓ_2-norm in $\mathbb{R}^d$ in the following way: $\|x\| = \sqrt{\langle x, x\rangle}$. We write $[n] = \{1, 2, \ldots, n\}$.

Calls of the stochastic oracle means computing the gradient ∇f_i for some $i \in [n]$.

In order to prove convergence results, we state the following standard assumptions on the problem (1). The first two assumptions relate to the target function f, and the third relates to the constraint set $\mathcal{X}$.

We start with the assumption that the gradients of both the function f and all terms $\{f_i\}_{i=1}^n$ are smooth. This assumption is standard in the optimization literature and widely used in the analysis of Frank-Wolfe-type methods.

Assumption 3.1. The function $f : \mathcal{X} \to \mathbb{R}$, is L-smooth on $\mathcal{X}$, i.e., there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall\, x, y \in \mathcal{X}.$$

Each function $f_i : \mathcal{X} \to \mathbb{R}$, $i \in [n]$, is L_i-smooth on $\mathcal{X}$, i.e., there exists a constant $L_i > 0$ such that,

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i\|x - y\|, \quad \forall\, x, y \in \mathcal{X}.$$

We also define the constant $\tilde{L}$ as $\tilde{L}^2 := \frac{1}{n}\sum_{i=1}^n L_i^2$. By convexity of $x \mapsto x^2$, it is easy to prove that $\tilde{L} \geq L$.

The second assumption is the convexity of the function f.

Assumption 3.2. The function $f : \mathcal{X} \to \mathbb{R}$, is convex, i.e.,

$$f(x) \geq f(y) + \langle \nabla f(y), x - y\rangle, \quad \forall\, x, y \in \mathcal{X}.$$

Note that we consider both convex and non-convex cases of the function f. But even if f is convex, we do not additionally assume that the terms $\{f_i\}_{i=1}^n$ are convex, hence in general they can be non-convex. Naturally, it is common to find settings with convex f and convex f_i, but formulations with non-convex f_i also arise, e.g., in PCA (Garber & Hazan, 2015; Shamir, 2015; Allen-Zhu & Yuan, 2016).

The next assumption is also typical and found in all works on projection-free methods.

Assumption 3.3. The set $\mathcal{X}$ is convex and compact with a diameter D, i.e., for any $x, y \in \mathcal{X}$,

$$\|x - y\| \leq D.$$

4. Main Part

In this section, we present two new algorithms and their convergence guarantees.

4.1. State-of-the-art complexity with Sarah Frank-Wolfe

Previously, Reddi et al. (2016); Hazan & Luo (2016); Weber & Sra (2022) proposed to modify the classical Frank-Wolfe algorithm (2) using the SVRG technique (Johnson & Zhang, 2013). The essence of these modifications is to change the deterministic gradient in the Conditional Gradient method to some stochastic gradient g^k, e.g., calculated according to the SVRG approach:

$$g^k = \nabla f_{i_k}(x^k) - \nabla f_{i_k}(w^k) + \nabla f(w^k),$$

where i_k is randomly generated from $[n]$, and w^k is rarely taken equal to x^k, much more often equal to w^{k-1}. Therefore, when we update $w^k = x^k$, we sometimes need to consider the full deterministic gradient. The update rule for w^k can be deterministic (as in the original version) or randomized, known as the loopless approach (Kovalev et al., 2020). Meanwhile, there are other variance-reduced methods, such as SARAH (Nguyen et al., 2017a):

$$g^k = \nabla f_{i_k}(x^k) - \nabla f_{i_k}(x^{k-1}) + g^{k-1}, \tag{3}$$

where i_k is also randomly generated from $[n]$, and g^k is rarely taken equal to $\nabla f(x^k)$ rather than (3). As noted in the original paper on SARAH, this method has better convergence guarantees and smoother convergence paths with less oscillations than SVRG, making SARAH preferred in both theory and practice. As a result, the SARAH update is also used in the Conditional Gradient method for non-convex problems (Yurtsever et al., 2019; Gao & Huang, 2020; Weber & Sra, 2022). In these works, the authors call (3) the SPIDER technique. We also use SARAH as a base for Algorithm 1, but unlike Yurtsever et al. (2019); Gao & Huang (2020); Weber & Sra (2022), its' loopless version (Li et al., 2021a). First we give the convergence of Algorithm 1 in the convex case.

Theorem 4.1. *Let $\{x^k\}_{k\geq 0}$ denote the iterates of Algorithm 1 for solving problem (1), which satisfies Assumptions 3.1–3.3. Let x^* be the minimizer of f. Then for any K one can choose $\{\eta_k\}_{k\geq 0}$ as follows:*

$$\begin{aligned}
&\text{if } K \leq \frac{2}{p}, && \eta_k = \frac{p}{2},\\
&\text{if } K > \frac{2}{p} \text{ and } k < \left\lceil \frac{K}{2} \right\rceil, && \eta_k = \frac{p}{2},\\
&\text{if } K > \frac{2}{p} \text{ and } k \geq \left\lceil \frac{K}{2} \right\rceil, && \eta_k = \frac{2}{(4/p + k - \lceil K/2 \rceil)}.
\end{aligned}$$

For this choice of η_k, we have the following convergence:

$$\mathbb{E}\left[f(x^K) - f(x^*)\right] = \mathcal{O}\left(\frac{f(x^0) - f(x^*)}{p}\exp\left(-\frac{Kp}{4}\right) + \left[1 + \frac{\tilde{L}}{L}\sqrt{\frac{1-p}{pb}}\right]\frac{LD^2}{K}\right).$$

See the full proof in Section B.1. We highlight an important detail that the convergence is proved not only in terms of $f(x^K) - f(x^*)$, but also for the Lyapunov function, which

Algorithm 1 Sarah Frank-Wolfe

Parameters: step sizes $\{\eta_k\}_{k\geq 0}$, probability p, batch size b;
Initialization: choose $x^0 \in \mathcal{X}$; $g^0 = \nabla f(x^0)$;

1: **for** $k = 0, 1, 2, \ldots K-1$ **do**
2: Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle g^k, s\rangle$;
3: Update $x^{k+1} = x^k + \eta_k(s^k - x^k)$ with η_k
4: Generate batch S_k with size b;
5: Update $g^{k+1} = \begin{cases} \nabla f(x^{k+1}), & \text{with probability } p, \\ g^k + \frac{1}{b}\sum\limits_{i\in S_k}[\nabla f_i(x^{k+1}) - \nabla f_i(x^k)], & \text{with probability } 1-p, \end{cases}$;
6: **end for**

includes additionally $\|g^K - \nabla f(x^K)\|^2$. Therefore, Theorem 4.1 gives guarantees that $\|g^K - \nabla f(x^K)\|^2 \sim \frac{1}{K}$ and hence g^K becomes a good approximation of the full gradient $\nabla f(x^K)$. It is also worth pointing out that the results of Theorem 4.1 depends on $\|g^0 - \nabla f(x^0)\|^2$ in the general case (see Section B.1), but due to our initialization of Algorithm 1, it is equal to zero.

To choose p, one can note that for each iteration, we on average compute the stochastic gradient $(pn + (1-p)\cdot 2b)$ times: with probability p we need the full gradient, with probability $(1-p)$ – a batch of size b in two points x^{k+1} and x^k. If we take p close to 1, the guarantees in Theorem 4.1 gives faster convergence, but the oracle complexity per iteration increases. For example, if we take $p = 1$, we simply obtain a deterministic method, and the estimates for convergence and the number of gradient calculations reproduce the results for the classical Frank-Wolfe method. On the other hand, if p tends to 0, the number of stochastic gradient calls per iteration decreases, but the iterative convergence rate drops. It is optimal to choose p based on the condition: $pn = 2(1-p)b$, i.e. $p = \frac{2b}{n+2b}$. From Theorem 4.1 we can also obtain an estimate on the required number of linear minimizations (LMO complexity). It is equal to the number of iterations of Algorithm 1. Then, the following corollary holds.

Corollary 4.2. *Under the conditions of Theorem 4.1, Algorithm 1 with $p = \frac{2b}{n+2b}$ achieves an ε suboptimality in expectation with*

$$\mathcal{O}\left(\frac{n}{b}\log\frac{1}{\varepsilon} + \left[1 + \frac{\tilde{L}\sqrt{n}}{bL}\right]\frac{LD^2}{\varepsilon}\right) \text{ LMO calls, and}$$

$$\mathcal{O}\left(n\log\frac{1}{\varepsilon} + \left[b + \frac{\tilde{L}\sqrt{n}}{L}\right]\frac{LD^2}{\varepsilon}\right) \text{ stoch. oracle calls.}$$

For any $b \leq \frac{\tilde{L}\sqrt{n}}{L}$, the estimate of the number of calls for the stochastic oracles does not change. Given that $\tilde{L} \geq L$, the smallest batch size $b = 1$ is appropriate for us. In this setting, the required number of the stochastic gradient computations is $\tilde{\mathcal{O}}\left(n + \frac{\sqrt{n}\tilde{L}D^2}{\varepsilon}\right)$. This result is the best in the literature around stochastic projection-free methods, especially since it does not require using large batches (see Table 1). Meanwhile, in the general case (not necessarily projection-free), these estimates can be explicitly improved up to $\tilde{\mathcal{O}}\left(n + \sqrt{\frac{n\tilde{L}D^2}{\varepsilon}}\right)$ (Allen-Zhu, 2018). Whether this is possible in the case of Frank-Wolfe type methods is an attractive question for future consideration. For $p = \frac{2b}{n+2b}$ and $b = \sqrt{n}$, the LMO complexity is $\tilde{\mathcal{O}}\left(\sqrt{n} + \frac{\tilde{L}D^2}{\varepsilon}\right)$, this result is optimal up to an additional factor $\sqrt{n}$ (see Section 2.1.2 from (Braun et al., 2022)). With $b = 1$, the LMO complexity equals $\tilde{\mathcal{O}}\left(n + \frac{\sqrt{n}\tilde{L}D^2}{\varepsilon}\right)$. Note that for many practical examples, the LMO complexity is not the computational bottleneck since the solution of linear minimization problems has a closed-form solution (see, e.g., Algorithm 2 from (Bellet et al., 2015)). It is also important to notice that, based on the above choices for η_k, p, and b, both our method and the original Frank-Wolfe are independent of the objective function parameters (e.g., L or $\tilde{L}$).

Next, we prove the convergence of Algorithm 1 for the non-convex objective function f. We use the *Frank-Wolfe gap* function (Jaggi, 2013) as a criterion for convergence:

$$\text{gap}(y) = \max_{x\in\mathcal{X}} \langle \nabla f(y), y - x\rangle. \tag{4}$$

Such a criterion is standard in the analysis of algorithms for the constrained problems with non-convex functions (Lacoste-Julien, 2016; Reddi et al., 2016). It is easy to check that $\text{gap}(y) \geq 0$ for any $y \in \mathcal{X}$. Moreover, a point $y \in X$ is stationary for (1) if and only if $\text{gap}(y) = 0$. (Lacoste-Julien, 2016) notes that the Frank-Wolfe gap is a meaningful measure of non-stationarity, and also an affine invariant generalization of the more standard convergence criterion $\|\nabla f(y)\|$ that is used for unconstrained non-convex problems. Then the following theorem is valid.

Theorem 4.3. *Let $\{x^k\}_{k\geq 0}$ denote the iterates of Algorithm 1 for solving problem (1), which satisfies Assumptions 3.1,3.3. Let x^* be the global (may be not unique) minimizer of f. Then, if we choose $\eta_k = \frac{1}{\sqrt{K}}$, we have the following*

Algorithm 2 Saga Sarah Frank-Wolfe

Parameters: step sizes $\{\eta_k\}_{k\geq 0}$; momentum λ; batch size b;
Initialization: choose $x^0 \in \mathcal{X}$; $g^0 = \nabla f(x^0)$ or $g^0 = \nabla f_{i_0}(x^0)$; $y_i^0 = \nabla f_i(x^0)$ or $y_i^0 = 0$;

1: **for** $k = 0, 1, 2, \ldots K-1$ **do**
2: Compute $s^k = \arg\min_{s\in\mathcal{X}} \langle g^k, s\rangle$;
3: Update $x^{k+1} = x^k + \eta_k(s^k - x^k)$ with η_k
4: Generate batch S_k with size b
5: Update $g^{k+1} = \frac{1}{b}\sum\limits_{i\in S_k}[\nabla f_i(x^{k+1}) - \nabla f_i(x^k)] + (1-\lambda)g^k + \lambda\left(\frac{1}{b}\sum\limits_{i\in S_k}[\nabla f_i(x^k) - y_i^k] + \frac{1}{n}\sum\limits_{j=1}^{n} y_j^k\right)$;
6: Update $y_i^{k+1} = \begin{cases}\nabla f_i(x^{k+1}), & i \in S_k,\\ y_i^k, & i \notin S_k,\end{cases}$;
7: **end for**

convergence:

$$\mathbb{E}\left[\min_{0\leq k\leq K-1} \text{gap}(x^k)\right] = \mathcal{O}\left(\frac{f(x^0) - f(x^*)}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}}\left[1 + \frac{\tilde{L}}{L}\sqrt{\frac{(1-p)}{pb}}\right]\right).$$

See the proof in Section B.2. In this case, the optimal choice of the parameter p is the same as in Corollary 4.2 of Theorem 4.1.

Corollary 4.4. *Under the conditions of Theorem 4.3, Algorithm 1 with $p = \frac{2b}{n+2b}$ achieves an ε suboptimality in expectation with*

$$\mathcal{O}\left(\left[\frac{h^0}{\varepsilon}\right]^2 + \left[\frac{LD^2}{\varepsilon}\right]^2\left[1 + \frac{\tilde{L}^2 n}{L^2 b^2}\right]\right) \text{ LMO calls, and}$$

$$\mathcal{O}\left(b\left[\frac{h^0}{\varepsilon}\right]^2 + \left[\frac{LD^2}{\varepsilon}\right]^2\left[b + \frac{\tilde{L}^2 n}{L^2 b}\right]\right) \text{ stoch. oracle calls,}$$

where $h^0 := f(x^0) - f(x^*)$.

First, we substitute $b = 1$ in the previous result. This gives us an oracle complexity of $\mathcal{O}\left(\frac{n(\tilde{L}^2+L^2)D^4+(h^0)^2}{\varepsilon^2}\right)$, which corresponds to the Frank-Wolfe complexity. If we choose the batch size $b = \frac{\sqrt{n}\tilde{L}}{L}$ that minimizes the expression $\left[b + \frac{\tilde{L}^2 n}{L^2 b}\right]$, then we need $\mathcal{O}\left(n + \frac{\sqrt{n}\tilde{L}[L^2D^4+(h^0)^2]}{L\varepsilon^2}\right)$ stochastic gradient calls. If one wishes to avoid using $\tilde{L}$ and L constants when selecting b, it is possible to take $b = \sqrt{n}$, and the complexity then becomes $\mathcal{O}\left(n + \frac{\sqrt{n}[(\tilde{L}^2+L^2)D^4+(h^0)^2]}{\varepsilon^2}\right)$. As noted earlier, both of these estimates are the best result of the projection-free methods for the non-convex setup (see Table 1). Moreover, they are optimal according to the lower estimates from Table 1 of (Li et al., 2021a). Additionally, with $p = \frac{2b}{n+2b}$ and $b = 1$, the LMO complexity is equal to $\mathcal{O}\left(\frac{n(\tilde{L}^2+L^2)D^4+(h^0)^2}{\varepsilon^2}\right)$, and with $b = \sqrt{n}$, it is $\mathcal{O}\left(\frac{(\tilde{L}^2+L^2)D^4+(h^0)^2}{\varepsilon^2}\right)$. In terms of lower bounds, only the already mentioned LMO estimates $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ for the convex setting are known. One important detail to note is that in both the convex (see the discussion after Corollary 4.2) and non-convex cases, LMO estimates is better when the batches are chosen quite large. In Section C.3, we give reasoning about this.

4.2. Avoiding full gradient computations with Saga Sarah Frank-Wolfe

The idea of Algorithm 2 is to use a combination (Li et al., 2021b) of the SARAH and SAGA (Defazio et al., 2014) approaches. Both SAGA and SARAH are some of the main variance-reduced methods for finite-sum minimization problems. An important feature of SAGA is that it does not use full gradient calculations, but it has worse convergence guarantees than SARAH (see, e.g., Table 2 in (Nguyen et al., 2017a)). The synergy of SARAH and SAGA brings together the strengths of both methods.

The essence of the SAGA method is similar to SVRG, but where SVRG collects the full gradients at some reference points, SAGA instead maintains a "sliding" version of the full gradient. The gradient $\nabla f(w)$ at the reference point w may become obsolete after a small number of iterations, but in the course of the algorithm we compute newer stochastic gradients for some $i \in [n]$, and one can leverage them to calculate a more recent approximation of the full gradient. To do this, SAGA introduces additional vectors $\{y\}_{i=1}^n$; each such y_i keeps the latest version of the gradient ∇f_i (implemented in line 6). The $\frac{1}{n}\sum_{j=1}^n y_j^k$ term is the aforementioned approximation of the full gradient. The calculation of g_{SAGA}^k is $g_{\text{SAGA}}^k = \left(\frac{1}{b}\sum_{i\in S_k}[\nabla f_i(x^k) - y_i^k] + \frac{1}{n}\sum_{j=1}^n y_j^k\right)$ (similarly to SVRG and SARAH). Line 5 provides a combination of g_{SARAH}^k and g_{SAGA}^k.

Recall that the average number of the stochastic oracle calls per iteration of Algorithm 1 is $(pn + (1-p)\cdot 2b)$, and that Algorithm 2 requires $2b$ computations of the stochastic gra-

dients each iteration. In lines 5 and 6, one need to calculate $\nabla f_i(x^{k+1})$, $\nabla f_i(x^k)$ for $i \in S_k$. Therefore, if $b \leq \frac{n}{2}$, then for any $p \neq 0$, the complexity of one iteration of Algorithm 2 is better than that of Algorithm 1.

In summary, Algorithm 2 does not collect full gradients and has a bit better iteration complexity, but is required to use n extra vectors $\{y_i\}_{i=1}^n$, requiring an additional $\mathcal{O}(nd)$ memory cost compared to Algorithm 1. Once can note that the methods from (Négiar et al., 2020; Lu & Freund, 2021) also use an extra memory size of $\mathcal{O}(nd)$.

For the convex target function f, Algorithm 2 satisfies the following convergence theorem.

Theorem 4.5. *Let $\{x^k\}_{k\geq 0}$ denote the iterates of Algorithm 2 for solving problem (1), which satisfies Assumptions 3.1–3.3. Let x^* be the minimizer of f. Then for any K one can choose $\{\eta_k\}_{k\geq 0}$ as follows:*

$$\begin{aligned}
&\text{if } K \leq \frac{4n}{b}, && \eta_k = \frac{b}{4n},\\
&\text{if } K > \frac{4n}{b} \quad \text{and} \quad k < \left\lceil \frac{K}{2} \right\rceil, && \eta_k = \frac{b}{4n},\\
&\text{if } K > \frac{4n}{b} \quad \text{and} \quad k \geq \left\lceil \frac{K}{2} \right\rceil, && \eta_k = \frac{2}{(8n/b + k - \lceil K/2 \rceil)},
\end{aligned}$$

and $\lambda = \frac{b}{2n}$. For this choice of η_k and λ, we have the following convergence:

$$\mathbb{E}\left[f(x^K) - f(x^*)\right] = \mathcal{O}\left(\frac{n\left[f(x^0) - f(x^*)\right]}{b}\exp\left(-\frac{bK}{8n}\right) + \left[1 + \frac{\tilde{L}\sqrt{n}}{Lb}\right]\frac{LD^2}{K}\right).$$

See the proof in Section B.3. As in the case of Theorem 4.1, here the proof is also obtained in terms of the Lyapunov function containing $\|g^K - \nabla f(x^K)\|^2$, we can guarantee that g^K tends to $\nabla f(x^K)$. The guarantees of Theorem 4.5 depend on $\|g^0 - \nabla f(x^0)\|^2$ and $\sum_{i=1}^n \|y_i^0 - \nabla f_i(x^0)\|$, but because of initialization we put them equal to 0 again. Since we do not need to choose p for Algorithm 2 we proceed directly to the corollary on the oracle complexity.

Corollary 4.6. *Under the conditions of Theorem 4.5, Algorithm 2 achieves an ε suboptimality in expectation with*

$$\mathcal{O}\left(\frac{n}{b}\log\frac{1}{\varepsilon} + \left[1 + \frac{\tilde{L}\sqrt{n}}{bL}\right]\frac{LD^2}{\varepsilon}\right) \text{ LMO calls, and}$$

$$\mathcal{O}\left(n\log\frac{1}{\varepsilon} + \left[b + \frac{\tilde{L}\sqrt{n}}{L}\right]\frac{LD^2}{\varepsilon}\right) \text{ stoch. oracle calls.}$$

This result is exactly the same as Corollary 4.2, so we obtain the same conclusions for choosing the size of b as that Corollary 4.2. In particular, in this case with $b = 1$, the method also have oracle complexity $\tilde{\mathcal{O}}\left(n + \frac{\sqrt{n}\tilde{L}D^2}{\varepsilon}\right)$ – the best among the works on stochastic projection-free methods. The findings on the LMO complexity is also consistent with Theorem 4.1. The reasoning around optimality is also consistent with that given after Corollary 4.2. Note also that Algorithm 2, just the same as Algorithm 1 and the classical Frank-Wolfe method, is free of the target function's parameters.

In the following theorem for the non-convex case of f, as in Theorem 4.3, we use (4) to estimate convergence.

Theorem 4.7. *Let $\{x^k\}_{k\geq 0}$ denote the iterates of Algorithm 2 for solving problem (1), which satisfies Assumptions 3.1,3.3. Let x^* be the global (may be not unique) minimizer of f on $\mathcal{X}$. Then, if we choose $\eta_k = \frac{1}{\sqrt{K}}$ and $\lambda = \frac{b}{2n}$, we have the following convergence:*

$$\mathbb{E}\left[\min_{0\leq k\leq K-1}\text{gap}(x^k)\right] = \mathcal{O}\left(\frac{f(x^0) - f(x^*)}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}}\left[1 + \frac{\tilde{L}\sqrt{n}}{Lb}\right]\right).$$

See the proof in Section B.4.

Corollary 4.8. *Under the conditions of Theorem 4.7, Algorithm 2 achieves an ε suboptimality in expectation with*

$$\mathcal{O}\left(\left[\frac{h^0}{\varepsilon}\right]^2 + \left[\frac{LD^2}{\varepsilon}\right]^2\left[1 + \frac{\tilde{L}^2 n}{L^2 b^2}\right]\right) \text{ LMO calls, and}$$

$$\mathcal{O}\left(b\left[\frac{h^0}{\varepsilon}\right]^2 + \left[\frac{LD^2}{\varepsilon}\right]^2\left[b + \frac{\tilde{L}^2 n}{L^2 b}\right]\right) \text{ stoch. oracle calls,}$$

where $h^0 := f(x^0) - f(x^*)$.

We once again have the same results as for Algorithm 1 in Corollary 4.4, resulting in the same analysis for choosing the batch sizes and for the LMO complexity.

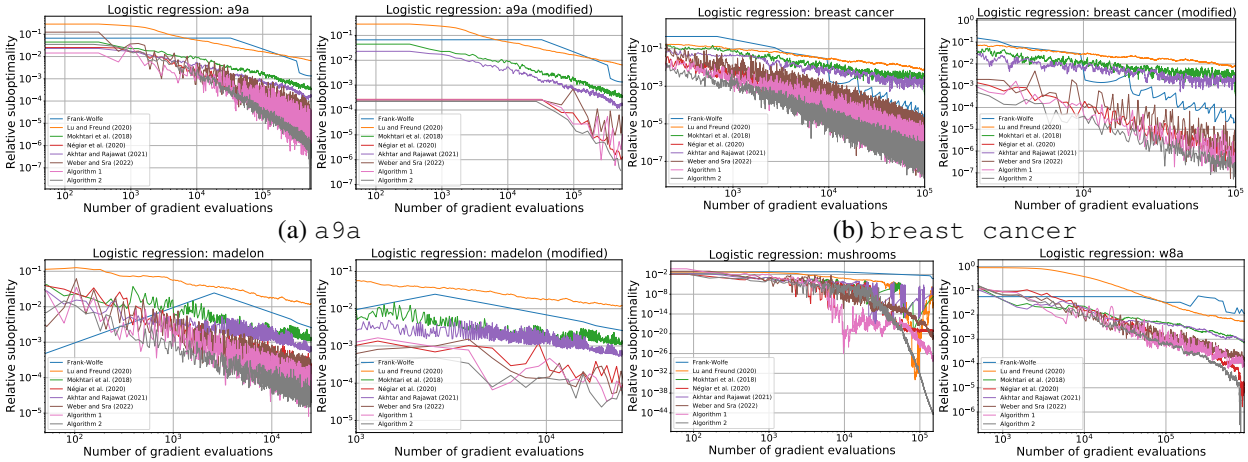
5. Experiments

We conduct our experiments on the constrained empirical risk for a linear model with weights w and on training samples $\{x_i, y_i\}_{i=1}^n$. Then, the logistic regression problem is

$$\min_{w\in\mathcal{C}} f(w) = \frac{1}{n}\sum_{i=1}^n \log(1 + \exp(-y_i w^T x_i)), \quad (5)$$

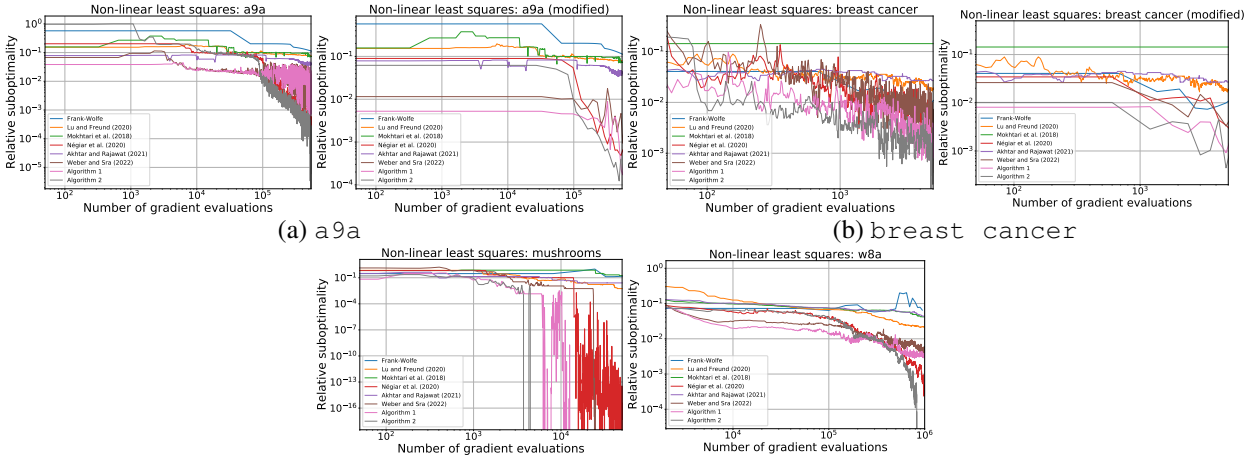
where $y_i \in \{-1, 1\}$, and the non-linear least squares loss is

$$\min_{w\in\mathcal{C}} f(w) = \frac{1}{n}\sum_{i=1}^n (y_i - 1/(1 + \exp(w^T x_i)))^2, \quad (6)$$



(a) a9a (b) breast cancer (c) madelon (d) mushrooms (e) w8a

Figure 1: Comparison of state-of-the-art projection free methods with small batches for (5). The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations. In the modified plots (the right plots in the first three lines), we left only every 100th point for (Négiar et al., 2020), (Weber & Sra, 2022), Algorithm 1 and Algorithm 2.



(a) a9a (b) breast cancer (c) mushrooms (d) w8a

Figure 2: Comparison of state-of-the-art projection free methods with small batches for (6). The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations. In the modified plots (the right plots in the first three lines), we left only every 100th point for (Négiar et al., 2020), (Weber & Sra, 2022), Algorithm 1 and Algorithm 2.

with $y_i \in \{0, 1\}$. We consider two different loss functions to test our algorithms on both convex and nonconvex settings. We choose $\mathcal{C}$ as the ℓ_1 norm ball with radius $R = 2 \cdot 10^3$ (see results with other R in Section D). The LMO for such a constraint set can be computed in a closed-form solution. We take LibSVM (Chang & Lin, 2011) datasets (see Table 2).

Table 2: Datasets from LibSVM in experiments.

Dataset	d	n
a9a	123	22696
breast cancer	10	683
madelon	500	2000
mushrooms	112	8124
w8a	300	49749

For comparison, we consider the methods from Table 1, which do not use large batches: (Mokhtari et al., 2020; Négiar et al., 2020; Lu & Freund, 2021; Akhtar & Rajawat, 2021; Weber & Sra, 2022). Our method is tuned according to the theory (see Sections 4.1 and 4.2), but we take the batchsize $b = \left\lceil \frac{n}{100} \right\rceil$ (similarly as the baselines). For the baselines, we use the implementation of (Négiar et al., 2020) and tune each method accordingly.[1]

[1]Note that the algorithms are not exactly implemented according to the theory of the corresponding works. In particular, instead of randomly selecting the batches, the authors sample them without replacement.

In Figure 1 we plot the relative suboptimality which is defined a $(f(x_t)-f_{\min})/(f_{\max}-f_{\min})$ where $f_{\max}$ is the largest value observed along the optimization and $f_{\min}$ is obtained by running the best algorithm a bit longer than what is plotted. From our results, it is clear that our algorithms are superior or comparable to the baselines, despite the fact that some methods were specifically designed for linear models (e.g. (Négiar et al., 2020; Lu & Freund, 2021)) which is not the case for Algorithm 1 and 2.

6. Conclusion and Future Works

In this paper, we presented two new algorithms for stochastic finite-sum optimization. Our methods are based on the Frank-Wolfe and Sarah approaches. Both of our algorithms are free of target function parameters. In both convex and non-convex target cases, our algorithms have the best stochastic oracle complexity in the literature. Our methods do not need to resort to large batch computation. However, in the non-convex case, it is worth noticing that the methods with large batch sizes give a better oracle complexity estimate. Moreover, Algorithm 2 does not need to collect either large batches or full deterministic gradients at all. Our methods also perform well on different ℓ_1 constrained logistic regression problems.

Ideas from (Jaggi, 2013), and (Lacoste-Julien & Jaggi, 2015) can be noted as a starting point for the future research in order to get fast rates in the strongly convex case. The modifications presented in these papers make the Frank-Wolfe method more practical and faster. Combining these and our approaches can produce a strong synergy that results in new practical algorithms.

Recall that our estimates of LMO in the convex case and SFO in the non-convex setting achieve lower bounds, and thus can be considered as optimal and unimprovable. Meanwhile, the SFO results in the convex setup and the LMO in the non-convex case have an unclosed gap and potential for improvement in finding new algorithms or lower bounds, which give optimality of the current results.

Finally, as mentioned above, the analysis of the algorithms is based on constructing a recursive estimate not only on $f(x) - f(x^*)$, as for classical Frank-Wolfe, but also includes $\|g - \nabla f(x)\|^2$. But this kind of bound is applicable to a number of many modern methods. Here we can mention, for example, distributed optimization methods with compression (Richtarik et al., 2021; Gorbunov et al., 2021), which are quite far from the setting of the current paper. A promising direction for future research is the generalization of the obtained results to a diverse array of optimization methods, with the aim of developing novel modifications of the Conditional Gradient method.

Acknowledgements

The research of A. Beznosikov has been supported by the Analytical Center for the Government of the Russian Federation (Agreement No. 70-2021-00143 dd. 01.11.2021, IGK 000000D730321P5Q0002).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Akhtar, Z. and Rajawat, K. Momentum based projection free stochastic optimization under affine constraints. In *2021 American Control Conference (ACC)*, pp. 2619–2624, 2021. doi: 10.23919/ACC50511.2021.9483167.

Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, 18(221):1–51, 2018.

Allen-Zhu, Z. and Yuan, Y. Improved svrg for non-strongly-convex or sum-of-non-convex objectives. In *International conference on machine learning*, pp. 1080–1089. PMLR, 2016.

Bach, F. Learning with submodular functions: A convex optimization perspective, 2011. URL https://arxiv.org/abs/1111.6453.

Bellet, A., Liang, Y., Garakani, A. B., Balcan, M.-F., and Sha, F. A distributed frank-wolfe algorithm for communication-efficient sparse learning. In *Proceedings of the 2015 SIAM international conference on data mining*, pp. 478–486. SIAM, 2015.

Bojanowski, P., Lajugie, R., Bach, F., Laptev, I., Ponce, J., Schmid, C., and Sivic, J. Weakly supervised action labeling in videos under ordering constraints. In *European Conference on Computer Vision*, pp. 628–643. Springer, 2014.

Braun, G., Carderera, A., Combettes, C. W., Hassani, H., Karbasi, A., Mokhtari, A., and Pokutta, S. Conditional gradient methods. *arXiv preprint arXiv:2211.14103*, 2022.

Chang, C.-C. and Lin, C.-J. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Cutkosky, A. and Orabona, F. Momentum-based variance reduction in non-convex sgd. *arXiv preprint arXiv:1905.10018*, 2019.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Demianov, V. F. and Rubinov, A. M. *Approximate methods in optimization problems*. Number 32. Elsevier Publishing Company, 1970.

Dunn, J. C. and Harshbarger, S. Conditional gradient algorithms with open loop step size rules. *Journal of Mathematical Analysis and Applications*, 62(2):432–444, 1978.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. Spider: Near-optimal non-convex optimization via stochastic path integrated differential estimator. *arXiv preprint arXiv:1807.01695*, 2018.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2): 95–110, 1956.

Gao, H. and Huang, H. Can stochastic zeroth-order frank-Wolfe method converge faster for non-convex problems? In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3377–3386. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/gao20b.html.

Garber, D. and Hazan, E. Fast and simple pca via convex optimization. *arXiv preprint arXiv:1509.05647*, 2015.

Gorbunov, E., Burlachenko, K. P., Li, Z., and Richtarik, P. Marina: Faster non-convex distributed learning with compression. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3788–3798. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/gorbunov21a.html.

Han, Y., Xie, G., and Zhang, Z. Lower complexity bounds of finite-sum optimization problems: The results and construction. *Journal of Machine Learning Research*, 25 (2):1–86, 2024.

Hazan, E. and Kale, S. Projection-free online learning. *arXiv preprint arXiv:1206.4657*, 2012.

Hazan, E. and Luo, H. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pp. 1263–1271. PMLR, 2016.

Hou, J., Zeng, X., Wang, G., Sun, J., and Chen, J. Distributed momentum-based frank-wolfe algorithm for stochastic optimization. *IEEE/CAA Journal of Automatica Sinica*, 2022.

Hu, W., Li, C. J., Lian, X., Liu, J., and Yuan, H. Efficient smooth non-convex stochastic compositional optimization via stochastic recursive gradient descent. 2019.

Jaggi, M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Dasgupta, S. and McAllester, D. (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 427–435, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/jaggi13.html.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/ac1dd209cbcc5e5d1c6e28598e8cbbe8-Paper.pdf.

Kovalev, D., Horváth, S., and Richtárik, P. Don't jump through hoops and remove those loops: Svrg and katyusha are better without the outer loop. In *Algorithmic Learning Theory*, pp. 451–467. PMLR, 2020.

Krishnan, R. G., Lacoste-Julien, S., and Sontag, D. Barrier frank-wolfe for marginal inference. *Advances in Neural Information Processing Systems*, 28, 2015.

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

Lacoste-Julien, S. and Jaggi, M. On the global linear convergence of frank-wolfe optimization variants. *Advances in neural information processing systems*, 28, 2015.

Lan, G. and Zhou, Y. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016. doi: 10.1137/140992382. URL https://doi.org/10.1137/140992382.

Levitin, E. S. and Polyak, B. T. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6(5):1–50, 1966.

Li, Z., Bao, H., Zhang, X., and Richtarik, P. Page: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6286–6295. PMLR, 18–24 Jul 2021a. URL https://proceedings.mlr.press/v139/li21a.html.

Li, Z., Hanzely, S., and Richtárik, P. Zerosarah: Efficient nonconvex finite-sum optimization with zero full gradient computation. *arXiv preprint arXiv:2103.01447*, 2021b.

Lu, H. and Freund, R. M. Generalized stochastic frank–wolfe algorithm with stochastic "substitute" gradient for structured convex optimization. *Mathematical Programming*, 187(1):317–349, 2021.

Mairal, J. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

Miech, A., Alayrac, J.-B., Bojanowski, P., Laptev, I., and Sivic, J. Learning from video and text via large-scale discriminative clustering. In *Proceedings of the IEEE international conference on computer vision*, pp. 5257–5266, 2017.

Mokhtari, A., Hassani, H., and Karbasi, A. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *Journal of machine learning research*, 2020.

Négiar, G., Dresdner, G., Tsai, A., El Ghaoui, L., Locatello, F., Freund, R., and Pedregosa, F. Stochastic frank-wolfe for constrained finite-sum minimization. In *International Conference on Machine Learning*, pp. 7253–7262. PMLR, 2020.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takáč, M. SARAH: a novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, pp. 2613–2621. PMLR, 2017a.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takáč, M. Stochastic recursive gradient algorithm for nonconvex optimization. *arXiv preprint arXiv:1705.07261*, 2017b.

Nguyen, L. M., Scheinberg, K., and Takáč, M. Inexact SARAH algorithm for stochastic optimization. *Optimization Methods and Software*, 36(1):237–258, 2021.

Patriksson, M. Partial linearization methods in nonlinear programming. *Journal of Optimization Theory and Applications*, 78(2):227–246, 1993.

Qian, X., Qu, Z., and Richtárik, P. Saga with arbitrary sampling. In *International Conference on Machine Learning*, pp. 5190–5199. PMLR, 2019.

Qu, C., Li, Y., and Xu, H. Non-convex conditional gradient sliding. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4208–4217. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/qu18a.html.

Reddi, S. J., Sra, S., Póczos, B., and Smola, A. Stochastic frank-wolfe methods for nonconvex optimization. In *2016 54th annual Allerton conference on communication, control, and computing (Allerton)*, pp. 1244–1251. IEEE, 2016.

Richtarik, P., Sokolov, I., and Fatkhullin, I. Ef21: A new, simpler, theoretically better, and practically faster error feedback. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4384–4396. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/231141b34c82aa95e48810a9d1b33a79-Paper.pdf.

Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Schmidt, M., Le Roux, N., and Bach, F. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Shamir, O. A stochastic pca and svd algorithm with an exponential convergence rate. In *International conference on machine learning*, pp. 144–152. PMLR, 2015.

Shen, Z., Fang, C., Zhao, P., Huang, J., and Qian, H. Complexities in projection-free stochastic non-convex minimization. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2868–2876. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/shen19b.html.

Stich, S. U. Unified optimal analysis of the (stochastic) gradient method. *arXiv preprint arXiv:1907.04232*, 2019.

Weber, M. and Sra, S. Projection-free nonconvex stochastic optimization on riemannian manifolds. *IMA Journal of Numerical Analysis*, 42(4):3241–3271, 2022.

Yang, Z., Chen, Z., and Wang, C. Accelerating mini-batch SARAH by step size rules. *Information Sciences*, 558: 157–173, 2021.

Yurtsever, A., Sra, S., and Cevher, V. Conditional gradient methods via stochastic path-integrated differential estimator. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7282–7291. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/yurtsever19b.html.

Zhang, M., Shen, Z., Mokhtari, A., Hassani, H., and Karbasi, A. One sample stochastic frank-wolfe. In *International Conference on Artificial Intelligence and Statistics*, pp. 4012–4023. PMLR, 2020.

A. Technical Facts

Lemma A.1. *For any $x_1, \ldots, x_n \in \mathbb{R}^d$ the following inequality holds:*

$$\left\| \sum_{i=1}^n x_i \right\|^2 \leq n \sum_{i=1}^n \|x_i\|^2.$$

Lemma A.2 (Lemma 1.2.3 from (Nesterov, 2003))**.** *Suppose that f is L-smooth. Then, for any $x, y \in \mathbb{R}^d$,*

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}\|x - y\|^2.$$

Lemma A.3 (Lemma 3 from (Stich, 2019))**.** *Let $\{r_k\}_{k \geq 0}$ is a non-negative sequence, which satisfies the relation*

$$r_{k+1} \leq (1 - \eta_k) r_k + c\eta_k^2.$$

Then there exists stepsizes $\eta_k \leq \frac{1}{d}$, such that:

$$r_K = \mathcal{O}\left(dr_0 \exp\left(-\frac{K}{2d} \right) + \frac{c}{K} \right).$$

In particular, the step sizes $\{\eta\}_{k \geq 0}$ can be chosen as follows

$$\begin{aligned}
&\text{if} \quad K \leq d, && \eta_k = \frac{1}{d}, \\
&\text{if} \quad K > d \quad \text{and} \quad k < k_0, && \eta_k = \frac{1}{d}, \\
&\text{if} \quad K > d \quad \text{and} \quad k \geq k_0, && \eta_k = \frac{2}{(2d + k - k_0)},
\end{aligned}$$

where $k_0 = \lceil \frac{K}{2} \rceil$.

B. Missing Proofs

B.1. Proof of Theorem 4.1

Theorem B.1 (Theorem 4.1)**.** *Let $\{x^k\}_{k \geq 0}$ denote the iterates of Algorithm 1 for solving problem (1), which satisfies Assumptions 3.1–3.3. Let x^* be the minimizer of f. Then for any K one can choose $\{\eta_k\}_{k \geq 0}$ as follows:*

$$\begin{aligned}
&\text{if} \quad K \leq \frac{2}{p}, && \eta_k = \frac{p}{2}, \\
&\text{if} \quad K > \frac{2}{p} \quad \text{and} \quad k < k_0, && \eta_k = \frac{p}{2}, \\
&\text{if} \quad K > \frac{2}{p} \quad \text{and} \quad k \geq k_0, && \eta_k = \frac{2}{(4/p + k - \lceil K/2 \rceil)}.
\end{aligned}$$

For this choice of η_k, we have the following convergence:

$$\mathbb{E}\left[f(x^K) - f(x^*) \right] = \mathcal{O}\left(\frac{1}{p}\left[f(x^0) - f(x^*) \right] \exp\left(-\frac{Kp}{4} \right) + \left[1 + \frac{\tilde{L}}{L}\sqrt{\frac{1-p}{pb}} \right] \frac{LD^2}{K} \right).$$

Proof: Let us start with Assumption 3.1 and Lemma A.2:

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2.$$

Subtracting $f(x^*)$ from both sides, we get

$$f(x^{k+1}) - f(x^*) \leq f(x^k) - f(x^*) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2.$$

With the update of x^{k+1} from line 3 of Algorithm 1, one can obtain

$$\begin{aligned}f(x^{k+1}) - f(x^*) \leq & f(x^k) - f(x^*) + \eta_k \langle \nabla f(x^k), s^k - x^k \rangle + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2 \\ = & f(x^k) - f(x^*) + \eta_k \langle g^k, s^k - x^k \rangle + \eta_k \langle \nabla f(x^k) - g^k, s^k - x^k \rangle \\ & + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2.\end{aligned}$$

The optimal choice of s^k from line 2 gives that $\langle g^k, s^k - x^k \rangle \leq \langle g^k, x^* - x^k \rangle$. Then,

$$\begin{aligned}f(x^{k+1}) - f(x^*) \leq & f(x^k) - f(x^*) + \eta_k \langle g^k, x^* - x^k \rangle + \eta_k \langle \nabla f(x^k) - g^k, s^k - x^k \rangle \\ & + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2 \\ = & f(x^k) - f(x^*) + \eta_k \langle \nabla f(x^k), x^* - x^k \rangle + \eta_k \langle g^k - \nabla f(x^k), x^* - x^k \rangle \\ & + \eta_k \langle \nabla f(x^k) - g^k, s^k - x^k \rangle + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2 \\ = & f(x^k) - f(x^*) + \eta_k \langle \nabla f(x^k), x^* - x^k \rangle + \eta_k \langle \nabla f(x^k) - g^k, s^k - x^* \rangle \\ & + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2.\end{aligned}$$

Applying the Cauchy-Schwartz inequality, we deduce $\langle \frac{\sqrt{\alpha}}{\sqrt{L}}(\nabla f(x^k) - g^k), \frac{\sqrt{L}}{\sqrt{\alpha}}\eta_k(s^k - x^*) \rangle \leq \frac{\alpha}{L}\|\nabla f(x^k) - g^k\|^2 + \frac{L\eta_k^2}{\alpha}\|s^k - x^*\|^2$ with some positive constant α (which we will define below). Thus,

$$\begin{aligned}f(x^{k+1}) - f(x^*) \leq & f(x^k) - f(x^*) + \eta_k \langle \nabla f(x^k), x^* - x^k \rangle + \frac{\alpha}{L}\|\nabla f(x^k) - g^k\|^2 \\ & + \frac{L\eta_k^2}{\alpha}\|s^k - x^*\|^2 + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2.\end{aligned}$$

Using the convexity of the function f (Assumption 3.2): $\langle \nabla f(x^k), x^* - x^k \rangle \leq -(f(x^k) - f(x^*))$, we have

$$\begin{aligned}f(x^{k+1}) - f(x^*) \leq & f(x^k) - f(x^*) - \eta_k(f(x^k) - f(x^*)) + \frac{\alpha}{L}\|\nabla f(x^k) - g^k\|^2 \\ & + \frac{L\eta_k^2}{\alpha}\|s^k - x^*\|^2 + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2 \\ = & (1 - \eta_k)(f(x^k) - f(x^*)) + \frac{\alpha}{L}\|\nabla f(x^k) - g^k\|^2 + \frac{L\eta_k^2}{\alpha}\|s^k - x^*\|^2 \\ & + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2.\end{aligned}$$

Taking the full mathematical expectation, one can obtain

$$\begin{aligned}\mathbb{E}\left[f(x^{k+1}) - f(x^*)\right] \leq & (1 - \eta_k)\mathbb{E}\left[f(x^k) - f(x^*)\right] + \frac{\alpha}{L}\mathbb{E}\left[\|\nabla f(x^k) - g^k\|^2\right] \\ & + \frac{L\eta_k^2}{\alpha}\mathbb{E}\left[\|s^k - x^*\|^2\right] + \frac{L\eta_k^2}{2}\mathbb{E}\left[\|s^k - x^k\|^2\right].\end{aligned} \tag{7}$$

For line 5 we use Lemma 3 from (Li et al., 2021a):

$$\mathbb{E}\left[\|\nabla f(x^{k+1}) - g^{k+1}\|^2\right] \leq (1 - p)\mathbb{E}\left[\|\nabla f(x^k) - g^k\|^2\right] + \frac{1 - p}{b}\mathbb{E}\left[\|\nabla f_{i_k}(x^{k+1}) - \nabla f_{i_k}(x^k)\|^2\right].$$

With Assumption 3.1, we get

$$\begin{aligned}\mathbb{E}\left[\|\nabla f(x^{k+1}) - g^{k+1}\|^2\right] \leq & (1 - p)\mathbb{E}\left[\|\nabla f(x^k) - g^k\|^2\right] + \frac{1 - p}{b}\mathbb{E}\left[L_{i_k}^2\|x^{k+1} - x^k\|^2\right] \\ = & (1 - p)\mathbb{E}\left[\|\nabla f(x^k) - g^k\|^2\right] + \frac{1 - p}{b}\mathbb{E}\left[\mathbb{E}_{i_k}[L_{i_k}^2] \cdot \|x^{k+1} - x^k\|^2\right].\end{aligned}$$

In the last step, we use the independence of i_k and $(x^{k+1}-x^k)$. Taking expectation on i_k, we obtain

$$\mathbb{E}\left[\|\nabla f(x^{k+1})-g^{k+1}\|^2\right] \leq (1-p)\mathbb{E}\left[\|\nabla f(x^k)-g^k\|^2\right] + \frac{1-p}{b}\left(\frac{1}{n}\sum_{i=1}^n L_i^2\right)\mathbb{E}\left[\|x^{k+1}-x^k\|^2\right].$$

The notation of $\tilde{L}$ provides

$$\mathbb{E}\left[\|\nabla f(x^{k+1})-g^{k+1}\|^2\right] \leq (1-p)\mathbb{E}\left[\|\nabla f(x^k)-g^k\|^2\right] + \frac{(1-p)\tilde{L}^2}{b}\mathbb{E}\left[\|x^{k+1}-x^k\|^2\right]. \tag{8}$$

Multiplying (8) by the positive constant M (which we will define below) and summing with (7), we have

$$\begin{aligned}
&\mathbb{E}\big[f(x^{k+1})-f(x^*)+M\cdot\|\nabla f(x^{k+1})-g^{k+1}\|^2\big] \\
&\quad\leq (1-\eta_k)\mathbb{E}\left[f(x^k)-f(x^*)\right] + \left(1-p+\frac{\alpha}{ML}\right)M\cdot\mathbb{E}\left[\|\nabla f(x^k)-g^k\|^2\right] \\
&\qquad + \frac{L\eta_k^2}{\alpha}\mathbb{E}\left[\|s^k-x^*\|^2\right] + \frac{L\eta_k^2}{2}\mathbb{E}\left[\|s^k-x^k\|^2\right] + \frac{M(1-p)\tilde{L}^2\eta_k^2}{b}\mathbb{E}\left[\|s^k-x^k\|^2\right].
\end{aligned}$$

The choice of $M=2\alpha/(pL)$ gives

$$\begin{aligned}
&\mathbb{E}\big[f(x^{k+1})-f(x^*)+M\cdot\|\nabla f(x^{k+1})-g^{k+1}\|^2\big] \\
&\quad\leq (1-\eta_k)\mathbb{E}\left[f(x^k)-f(x^*)\right] + \left(1-\frac{p}{2}\right)M\cdot\mathbb{E}\left[\|\nabla f(x^k)-g^k\|^2\right] \\
&\qquad + \frac{L\eta_k^2}{\alpha}\mathbb{E}\left[\|s^k-x^*\|^2\right] + \frac{L\eta_k^2}{2}\mathbb{E}\left[\|s^k-x^k\|^2\right] + \frac{2\alpha(1-p)\tilde{L}^2\eta_k^2}{pbL}\mathbb{E}\left[\|s^k-x^k\|^2\right] \\
&\quad\leq \max\left\{1-\eta_k, 1-\frac{p}{2}\right\}\mathbb{E}\left[f(x^k)-f(x^*)+M\cdot\|\nabla f(x^k)-g^k\|^2\right] \\
&\qquad + \frac{L\eta_k^2}{\alpha}\mathbb{E}\left[\|s^k-x^*\|^2\right] + \frac{L\eta_k^2}{2}\mathbb{E}\left[\|s^k-x^k\|^2\right] + \frac{2\alpha(1-p)\tilde{L}^2\eta_k^2}{pbL}\mathbb{E}\left[\|s^k-x^k\|^2\right].
\end{aligned}$$

With Assumption 3.3 on the diameter D of $\mathcal{X}$, we get

$$\begin{aligned}
&\mathbb{E}\big[f(x^{k+1})-f(x^*)+M\cdot\|\nabla f(x^{k+1})-g^{k+1}\|^2\big] \\
&\quad\leq \max\left\{1-\eta_k, 1-\frac{p}{2}\right\}\mathbb{E}\left[f(x^k)-f(x^*)+M\cdot\|\nabla f(x^k)-g^k\|^2\right] \\
&\qquad + \frac{LD^2\eta_k^2}{\alpha} + \frac{LD^2\eta_k^2}{2} + \frac{2\alpha(1-p)\tilde{L}^2D^2\eta_k^2}{pbL} \\
&\quad= \max\left\{1-\eta_k, 1-\frac{p}{2}\right\}\mathbb{E}\left[f(x^k)-f(x^*)+M\cdot\|\nabla f(x^k)-g^k\|^2\right] \\
&\qquad + LD^2\eta_k^2\left(\frac{1}{2}+\frac{1}{\alpha}+\frac{2\alpha(1-p)\tilde{L}^2}{pbL^2}\right).
\end{aligned}$$

If we choose $\eta_k \leq \frac{p}{2}$, $\alpha=\sqrt{\frac{pbL^2}{(1-p)\tilde{L}^2}}$, then we have

$$\begin{aligned}
&\mathbb{E}\big[f(x^{k+1})-f(x^*)+M\cdot\|\nabla f(x^{k+1})-g^{k+1}\|^2\big] \\
&\quad\leq (1-\eta_k)\,\mathbb{E}\left[f(x^k)-f(x^*)+M\cdot\|\nabla f(x^k)-g^k\|^2\right] \\
&\qquad + LD^2\eta_k^2\left(\frac{1}{2}+\frac{3\tilde{L}}{L}\sqrt{\frac{1-p}{pb}}\right).
\end{aligned}$$

It remains to use Lemma A.3 with $c=LD^2\left(\frac{1}{2}+\frac{3\tilde{L}}{L}\sqrt{\frac{1-p}{pb}}\right)$, $d=\frac{2}{p}$ and obtain

$$\mathbb{E}\big[f(x^K)-f(x^*)+M\cdot\|\nabla f(x^K)-g^K\|^2\big]$$

$$=\mathcal{O}\left(\frac{1}{p}\left[f(x^0)-f(x^*)+M\cdot\|\nabla f(x^0)-g^0\|^2\right]\exp\left(-\frac{Kp}{4}\right)+\left[1+\frac{\tilde{L}}{L}\sqrt{\frac{1-p}{pb}}\right]\frac{LD^2}{K}\right).$$

Finally, we substitute $g^0=\nabla f(x^0)$:

$$\mathbb{E}\left[f(x^K)-f(x^*)\right]=\mathcal{O}\left(\frac{1}{p}\left[f(x^0)-f(x^*)\right]\exp\left(-\frac{Kp}{4}\right)+\left[1+\frac{\tilde{L}}{L}\sqrt{\frac{1-p}{pb}}\right]\frac{LD^2}{K}\right).$$

This completes the proof.

□

B.2. Proof of Theorem 4.3

Theorem B.2 (Theorem 4.3)**.** *Let $\{x^k\}_{k\geq 0}$ denote the iterates of Algorithm 1 for solving problem (1), which satisfies Assumptions 3.1,3.3. Let x^* be the global (may be not unique) minimizer of f. Then, if we choose $\eta_k=\frac{1}{\sqrt{K}}$, we have the following convergence:*

$$\mathbb{E}\left[\min_{0\leq k\leq K-1}\text{gap}(x^k)\right]=\mathcal{O}\left(\frac{f(x^0)-f(x^*)}{\sqrt{K}}+\frac{LD^2}{\sqrt{K}}\left[1+\frac{\tilde{L}}{L}\sqrt{\frac{(1-p)}{pb}}\right]\right).$$

Proof: Let us start with Assumption 3.1 and Lemma A.2:

$$f(x^{k+1})\leq f(x^k)+\langle\nabla f(x^k),x^{k+1}-x^k\rangle+\frac{L}{2}\|x^{k+1}-x^k\|^2.$$

Subtracting $f(x^*)$ from both sides, we get

$$f(x^{k+1})-f(x^*)\leq f(x^k)-f(x^*)+\langle\nabla f(x^k),x^{k+1}-x^k\rangle+\frac{L}{2}\|x^{k+1}-x^k\|^2.$$

With the update of x^{k+1} from line 3 of Algorithm 1, one can obtain

$$\begin{aligned}f(x^{k+1})-f(x^*)\leq&f(x^k)-f(x^*)+\eta_k\langle\nabla f(x^k),s^k-x^k\rangle+\frac{L\eta_k^2}{2}\|s^k-x^k\|^2\\=&f(x^k)-f(x^*)+\eta_k\langle g^k,s^k-x^k\rangle+\eta_k\langle\nabla f(x^k)-g^k,s^k-x^k\rangle+\frac{L\eta_k^2}{2}\|s^k-x^k\|^2.\end{aligned}$$

The optimal choice of s^k from line 2 gives that $\langle g^k,s^k-x^k\rangle\leq\langle g^k,x-x^k\rangle$ for any $x\in\mathcal{X}$. Then,

$$\begin{aligned}f(x^{k+1})-f(x^*)\leq&f(x^k)-f(x^*)+\eta_k\langle g^k,x-x^k\rangle+\eta_k\langle\nabla f(x^k)-g^k,s^k-x^k\rangle+\frac{L\eta_k^2}{2}\|s^k-x^k\|^2\\=&f(x^k)-f(x^*)+\eta_k\langle\nabla f(x^k),x-x^k\rangle+\eta_k\langle g^k-\nabla f(x^k),x-x^k\rangle\\&+\eta_k\langle\nabla f(x^k)-g^k,s^k-x^k\rangle+\frac{L\eta_k^2}{2}\|s^k-x^k\|^2\\=&f(x^k)-f(x^*)+\eta_k\langle\nabla f(x^k),x-x^k\rangle+\eta_k\langle\nabla f(x^k)-g^k,s^k-x\rangle\\&+\frac{L\eta_k^2}{2}\|s^k-x^k\|^2.\end{aligned}$$

Applying the Cauchy-Schwartz inequality, we deduce $\langle\frac{\sqrt{\alpha}}{\sqrt{L}}(\nabla f(x^k)-g^k),\frac{\sqrt{L}}{\sqrt{\alpha}}\eta_k(s^k-x^*)\rangle\leq\frac{\alpha}{L}\|\nabla f(x^k)-g^k\|^2+\frac{L\eta_k^2}{\alpha}\|s^k-x^*\|^2$ with some positive constant α (which we will define below). Thus,

$$\begin{aligned}f(x^{k+1})-f(x^*)\leq&f(x^k)-f(x^*)+\eta_k\langle\nabla f(x^k),x-x^k\rangle+\frac{\alpha}{L}\|\nabla f(x^k)-g^k\|^2\\&+\frac{L\eta_k^2}{\alpha}\|s^k-x\|^2+\frac{L\eta_k^2}{2}\|s^k-x^k\|^2.\end{aligned}$$

After small rearrangements one can obtain

$$\eta_k\langle\nabla f(x^k), x^k - x\rangle \leq f(x^k) - f(x^*) - \left(f(x^{k+1}) - f(x^*)\right) + \frac{\alpha}{L}\|\nabla f(x^k) - g^k\|^2 + \frac{L\eta_k^2}{\alpha}\|s^k - x\|^2 + \frac{L\eta_k^2}{2}\|s^k - x^k\|^2.$$

Maximizing over all $x \in \mathcal{X}$ and taking the full mathematical expectation, we get

$$\eta_k\mathbb{E}\left[\max_{x\in\mathcal{X}}\langle\nabla f(x^k), x^k - x\rangle\right] \leq \mathbb{E}\left[f(x^k) - f(x^*)\right] - \mathbb{E}\left[f(x^{k+1}) - f(x^*)\right] + \frac{\alpha}{L}\mathbb{E}\left[\|\nabla f(x^k) - g^k\|^2\right] + \frac{L\eta_k^2}{\alpha}\mathbb{E}\left[\max_{x\in\mathcal{X}}\|s^k - x\|^2\right] + \frac{L\eta_k^2}{2}\mathbb{E}\left[\|s^k - x^k\|^2\right]. \tag{9}$$

Multiplying (8) by the positive constant M (which we will define below) and summing with (9), we have

$$\begin{aligned}\eta_k\mathbb{E}\left[\max_{x\in\mathcal{X}}\langle\nabla f(x^k), x^k - x\rangle\right] \leq& \mathbb{E}\left[f(x^k) - f(x^*)\right] + \left(1 - p + \frac{\alpha}{ML}\right)M\mathbb{E}\left[\|\nabla f(x^k) - g^k\|^2\right] \\ &- \mathbb{E}\left[f(x^{k+1}) - f(x^*) + M\|\nabla f(x^{k+1}) - g^{k+1}\|^2\right] \\ &+ \frac{L\eta_k^2}{\alpha}\mathbb{E}\left[\max_{x\in\mathcal{X}}\|s^k - x\|^2\right] + \frac{L\eta_k^2}{2}\mathbb{E}\left[\|s^k - x^k\|^2\right] \\ &+ \frac{M(1-p)\tilde{L}^2\eta_k^2}{b}\mathbb{E}\left[\|s^k - x^k\|^2\right].\end{aligned}$$

The choice of $M = \alpha/(pL)$ gives

$$\begin{aligned}\eta_k\mathbb{E}\left[\max_{x\in\mathcal{X}}\langle\nabla f(x^k), x^k - x\rangle\right] \leq& \mathbb{E}\left[f(x^k) - f(x^*) + M\|\nabla f(x^k) - g^k\|^2\right] \\ &- \mathbb{E}\left[f(x^{k+1}) - f(x^*) + M\|\nabla f(x^{k+1}) - g^{k+1}\|^2\right] \\ &+ \frac{L\eta_k^2}{\alpha}\mathbb{E}\left[\max_{x\in\mathcal{X}}\|s^k - x\|^2\right] + \frac{L\eta_k^2}{2}\mathbb{E}\left[\|s^k - x^k\|^2\right] \\ &+ \frac{\alpha(1-p)\tilde{L}^2\eta_k^2}{pbL}\mathbb{E}\left[\|s^k - x^k\|^2\right].\end{aligned}$$

With Assumption 3.3 on the diameter D of $\mathcal{X}$, we get

$$\begin{aligned}\eta_k\mathbb{E}\left[\max_{x\in\mathcal{X}}\langle\nabla f(x^k), x - x^k\rangle\right] \leq& \mathbb{E}\left[f(x^k) - f(x^*) + M\|\nabla f(x^k) - g^k\|^2\right] \\ &- \mathbb{E}\left[f(x^{k+1}) - f(x^*) + M\|\nabla f(x^{k+1}) - g^{k+1}\|^2\right] \\ &+ LD^2\eta_k^2\left(\frac{1}{2} + \frac{1}{\alpha} + \frac{\alpha(1-p)\tilde{L}^2\eta_k^2}{pbL^2}\right).\end{aligned}$$

With the choice $\alpha = \frac{L}{\tilde{L}}\sqrt{\frac{pb}{1-p}}$, we have

$$\begin{aligned}\eta_k\mathbb{E}\left[\max_{x\in\mathcal{X}}\langle\nabla f(x^k), x^k - x\rangle\right] \leq& \mathbb{E}\left[f(x^k) - f(x^*) + M\|\nabla f(x^k) - g^k\|^2\right] \\ &- \mathbb{E}\left[f(x^{k+1}) - f(x^*) + M\|\nabla f(x^{k+1}) - g^{k+1}\|^2\right] \\ &+ LD^2\eta_k^2\left(\frac{1}{2} + \frac{2\tilde{L}}{L}\sqrt{\frac{(1-p)}{pb}}\right).\end{aligned}$$

Summing over all k from 0 to $K-1$, we have

$$\sum_{k=0}^{K-1}\eta_k\mathbb{E}\left[\max_{x\in\mathcal{X}}\langle\nabla f(x^k), x^k - x\rangle\right] \leq f(x^0) - f(x^*) + M\|\nabla f(x^0) - g^0\|^2$$

$$
\begin{aligned}
&- \mathbb{E}\left[f(x^K) - f(x^*) + M\|\nabla f(x^K) - g^K\|^2\right] \\
&+ LD^2\left(\frac{1}{2} + \frac{2\tilde{L}}{L}\sqrt{\frac{(1-p)q}{pn}}\right)\sum_{k=0}^{K-1}\eta_k^2 \\
\leq & f(x^0) - f(x^*) + \|\nabla f(x^0) - g^0\|^2 \\
&+ LD^2\left(\frac{1}{2} + \frac{2\tilde{L}}{L}\sqrt{\frac{(1-p)}{pb}}\right)\sum_{k=0}^{K-1}\eta_k^2.
\end{aligned}
$$

If we take $\eta_k = \frac{1}{\sqrt{K}}$ and divide both sides by $\sqrt{K}$, then

$$
\begin{aligned}
\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\max_{x\in\mathcal{X}}\langle\nabla f(x^k), x^k - x\rangle\right] \leq & \frac{1}{\sqrt{K}}\cdot\left[f(x^0) - f(x^*) + M\|\nabla f(x^0) - g^0\|^2\right] \\
&+ \frac{LD^2}{\sqrt{K}}\left(\frac{1}{2} + \frac{2\tilde{L}}{L}\sqrt{\frac{(1-p)}{pb}}\right).
\end{aligned}
$$

Finally, we substitute $g^0 = \nabla f(x^0)$:

$$
\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\max_{x\in\mathcal{X}}\langle\nabla f(x^k), x^k - x\rangle\right] \leq \frac{f(x^0) - f(x^*)}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}}\left(\frac{1}{2} + \frac{2\tilde{L}}{L}\sqrt{\frac{(1-p)}{pb}}\right).
$$

The definition of (4) finishes the proof.

□

B.3. Proof of Theorem 4.5

Theorem B.3 (Theorem 4.5)**.** *Let $\{x^k\}_{k\geq 0}$ denote the iterates of Algorithm 2 for solving problem (1), which satisfies Assumptions 3.1–3.3. Let x^* be the minimizer of f. Then for any K one can choose $\{\eta_k\}_{k\geq 0}$ as follows:*

$$
\begin{aligned}
&\text{if } K \leq \frac{4n}{b}, && \eta_k = \frac{b}{4n}, \\
&\text{if } K > \frac{4n}{b} \quad \text{and} \quad k < k_0, && \eta_k = \frac{b}{4n}, \\
&\text{if } K > \frac{4n}{b} \quad \text{and} \quad k \geq k_0, && \eta_k = \frac{2}{(8n/b + k - \lceil K/2\rceil)},
\end{aligned}
$$

and $\lambda = \frac{b}{2n}$. For this choice of η_k and λ, we have the following convergence:

$$
\mathbb{E}\left[f(x^K) - f(x^*)\right] = \mathcal{O}\left(\frac{n}{b}\left[f(x^0) - f(x^*)\right]\exp\left(-\frac{bK}{8n}\right) + \left[1 + \frac{\tilde{L}\sqrt{n}}{Lb}\right]\frac{LD^2}{K}\right).
$$

Proof: The first steps of the proof are the same with Theorem 4.1 (Theorem B.1), therefore we can start from (7). For line 5 we use Lemma 2 from (Li et al., 2021b):

$$
\begin{aligned}
\mathbb{E}\left[\|\nabla f(x^{k+1}) - g^{k+1}\|^2\right] \leq & (1-\lambda)\mathbb{E}\left[\|\nabla f(x^k) - g^k\|^2\right] + \frac{2}{b}\mathbb{E}\left[\|\nabla f_{i_k}(x^{k+1}) - \nabla f_{i_k}(x^k)\|^2\right] \\
&+ \frac{2\lambda^2}{b}\cdot\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[\|\nabla f_j(x^k) - y_j^k\|^2\right].
\end{aligned}
$$

With Assumption 3.1, we get

$$
\mathbb{E}\left[\|\nabla f(x^{k+1}) - g^{k+1}\|^2\right] \leq (1-\lambda)\mathbb{E}\left[\|\nabla f(x^k) - g^k\|^2\right] + \frac{2}{b}\mathbb{E}\left[L_{i_k}^2\|x^{k+1} - x^k\|^2\right]
$$

$$
\begin{aligned}
&+\frac{2\lambda^2}{b}\cdot\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[\|\nabla f_j(x^k)-y_j^k\|^2\right]\\
=&(1-\lambda)\mathbb{E}\left[\|\nabla f(x^k)-g^k\|^2\right]+\frac{2}{b}\mathbb{E}\left[\mathbb{E}_{i_k}[L_{i_k}^2]\cdot\|x^{k+1}-x^k\|^2\right]\\
&+\frac{2\lambda^2}{b}\cdot\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[\|\nabla f_j(x^k)-y_j^k\|^2\right].
\end{aligned}
$$

In the last step, we use the independence of i_k and $(x^{k+1}-x^k)$. Taking expectation on i_k, we obtain

$$
\begin{aligned}
\mathbb{E}\left[\|\nabla f(x^{k+1})-g^{k+1}\|^2\right]\leq&(1-\lambda)\mathbb{E}\left[\|\nabla f(x^k)-g^k\|^2\right]+\frac{2}{b}\left(\frac{1}{n}\sum_{i=1}^{n}L_i^2\right)\mathbb{E}\left[\|x^{k+1}-x^k\|^2\right]\\
&+\frac{2\lambda^2}{b}\cdot\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[\|\nabla f_j(x^k)-y_j^k\|^2\right].
\end{aligned}
$$

The notation of $\tilde{L}$ provides

$$
\begin{aligned}
\mathbb{E}\left[\|\nabla f(x^{k+1})-g^{k+1}\|^2\right]\leq&(1-\lambda)\mathbb{E}\left[\|\nabla f(x^k)-g^k\|^2\right]+\frac{2\tilde{L}^2}{b}\mathbb{E}\left[\|x^{k+1}-x^k\|^2\right]\\
&+\frac{2\lambda^2}{b}\cdot\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[\|\nabla f_j(x^k)-y_j^k\|^2\right].
\end{aligned}
\tag{10}
$$

Additionally, we need Lemma 3 from (Li et al., 2021b) with $\beta_k=\frac{b}{2n}$:

$$
\begin{aligned}
\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[\|\nabla f_j(x^{k+1})-y_j^{k+1}\|^2\right]\leq&\left(1-\frac{b}{2n}\right)\cdot\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[\|\nabla f_j(x^k)-y_j^k\|^2\right]\\
&+\frac{2n}{b}\cdot\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|\nabla f_j(x^{k+1})-\nabla f_j(x^k)\|^2\right].
\end{aligned}
$$

With Assumption 3.1 and the notation of $\tilde{L}$, we get

$$
\begin{aligned}
\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[\|\nabla f_j(x^{k+1})-y_j^{k+1}\|^2\right]\leq&\left(1-\frac{b}{2n}\right)\cdot\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[\|\nabla f_j(x^k)-y_j^k\|^2\right]\\
&+\frac{2n\tilde{L}^2}{b}\mathbb{E}\left[\|x^{k+1}-x^k\|^2\right].
\end{aligned}
\tag{11}
$$

Multiplying (10) by the positive constant M_1, (11) by the positive constant M_2 (M_1, M_2 will be defined below) and summing with (7), we have

$$
\begin{aligned}
&\mathbb{E}\left[f(x^{k+1})-f(x^*)+M_1\cdot\|\nabla f(x^{k+1})-g^{k+1}\|^2+M_2\cdot\frac{1}{n}\sum_{j=1}^{n}\|\nabla f_j(x^{k+1})-y_j^{k+1}\|^2\right]\\
\leq&(1-\eta_k)\mathbb{E}\left[f(x^k)-f(x^*)\right]\\
&+\left(1-\lambda+\frac{\alpha}{M_1L}\right)M_1\cdot\mathbb{E}\left[\|\nabla f(x^k)-g^k\|^2\right]\\
&+\left(1-\frac{b}{2n}+\frac{2M_1\lambda^2}{M_2b}\right)M_2\cdot\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[\|\nabla f_j(x^k)-y_j^k\|^2\right]\\
&+\frac{2(M_1+nM_2)\tilde{L}^2}{b}\mathbb{E}\left[\|x^{k+1}-x^k\|^2\right]+\frac{L\eta_k^2}{\alpha}\mathbb{E}\left[\|s^k-x^*\|^2\right]\\
&+\frac{L\eta_k^2}{2}\mathbb{E}\left[\|s^k-x^k\|^2\right].
\end{aligned}
$$

With $M_1 = \frac{2\alpha}{\lambda L}$ and $M_2 = \frac{8M_1\lambda^2 n}{b^2}$, we obtain

$$\begin{aligned}
&\mathbb{E}\left[f(x^{k+1}) - f(x^*) + M_1 \cdot \|\nabla f(x^{k+1}) - g^{k+1}\|^2 + M_2 \cdot \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^{k+1}) - y_j^{k+1}\|^2\right] \\
&\quad \leq (1-\eta_k)\mathbb{E}\left[f(x^k) - f(x^*)\right] \\
&\qquad + \left(1 - \frac{\lambda}{2}\right) M_1 \cdot \mathbb{E}\left[\|\nabla f(x^k) - g^k\|^2\right] + \left(1 - \frac{b}{4n}\right) M_2 \cdot \frac{1}{n}\sum_{j=1}^n \mathbb{E}\left[\|\nabla f_j(x^k) - y_j^k\|^2\right] \\
&\qquad + \frac{4\alpha\tilde{L}^2}{bL\lambda}\left(1 + \frac{8\lambda^2 n^2}{b^2}\right)\mathbb{E}\left[\|x^{k+1} - x^k\|^2\right] + \frac{L\eta_k^2}{\alpha}\mathbb{E}\left[\|s^k - x^*\|^2\right] + \frac{L\eta_k^2}{2}\mathbb{E}\left[\|s^k - x^k\|^2\right] \\
&\quad \leq (1-\eta_k)\mathbb{E}\left[f(x^k) - f(x^*)\right] \\
&\qquad + \left(1 - \frac{\lambda}{2}\right) M_1 \cdot \mathbb{E}\left[\|\nabla f(x^k) - g^k\|^2\right] + \left(1 - \frac{b}{4n}\right) M_2 \cdot \frac{1}{n}\sum_{j=1}^n \mathbb{E}\left[\|\nabla f_j(x^k) - y_j^k\|^2\right] \\
&\qquad + \frac{4\alpha\tilde{L}^2\eta_k^2}{bL\lambda}\left(1 + \frac{8\lambda^2 n^2}{b^2}\right)\mathbb{E}\left[\|s^k - x^k\|^2\right] + \frac{L\eta_k^2}{\alpha}\mathbb{E}\left[\|s^k - x^*\|^2\right] + \frac{L\eta_k^2}{2}\mathbb{E}\left[\|s^k - x^k\|^2\right].
\end{aligned}$$

With Assumption 3.3 on the diameter D of $\mathcal{X}$, we get

$$\begin{aligned}
&\mathbb{E}\left[f(x^{k+1}) - f(x^*) + M_1 \cdot \|\nabla f(x^{k+1}) - g^{k+1}\|^2 + M_2 \cdot \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^{k+1}) - y_j^{k+1}\|^2\right] \\
&\quad \leq (1-\eta_k)\mathbb{E}\left[f(x^k) - f(x^*)\right] \\
&\qquad + \left(1 - \frac{\lambda}{2}\right) M_1 \cdot \mathbb{E}\left[\|\nabla f(x^k) - g^k\|^2\right] + \left(1 - \frac{b}{4n}\right) M_2 \cdot \frac{1}{n}\sum_{j=1}^n \mathbb{E}\left[\|\nabla f_j(x^k) - y_j^k\|^2\right] \\
&\qquad + LD^2\eta_k^2\left(\frac{1}{2} + \frac{1}{\alpha} + \frac{4\alpha\tilde{L}^2}{bL^2\lambda}\left[1 + \frac{8\lambda^2 n^2}{b^2}\right]\right).
\end{aligned}$$

The choices of $\lambda = \frac{b}{2n}$ and $\alpha = \frac{bL}{5\tilde{L}\sqrt{n}}$ provides

$$\begin{aligned}
&\mathbb{E}\left[f(x^{k+1}) - f(x^*) + M_1 \cdot \|\nabla f(x^{k+1}) - g^{k+1}\|^2 + M_2 \cdot \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^{k+1}) - y_j^{k+1}\|^2\right] \\
&\quad \leq (1-\eta_k)\mathbb{E}\left[f(x^k) - f(x^*)\right] \\
&\qquad + \left(1 - \frac{b}{4n}\right) M_1 \cdot \mathbb{E}\left[\|\nabla f(x^k) - g^k\|^2\right] + \left(1 - \frac{1}{4n}\right) M_2 \cdot \frac{1}{n}\sum_{j=1}^n \mathbb{E}\left[\|\nabla f_j(x^k) - y_j^k\|^2\right] \\
&\qquad + LD^2\eta_k^2\left(\frac{1}{2} + \frac{10\tilde{L}\sqrt{n}}{L}\right) \\
&\quad \leq \max\left\{1 - \eta_k, 1 - \frac{b}{4n}\right\}\mathbb{E}\Bigg[f(x^k) - f(x^*) + M_1 \cdot \|\nabla f(x^k) - g^k\|^2 \\
&\qquad + M_2 \cdot \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^k) - y_j^k\|^2\Bigg] + LD^2\eta_k^2\left(\frac{1}{2} + \frac{10\tilde{L}\sqrt{n}}{Lb}\right).
\end{aligned}$$

If we choose $\eta_k \leq \frac{b}{4n}$, we have

$$\mathbb{E}\left[f(x^{k+1}) - f(x^*) + M_1 \cdot \|\nabla f(x^{k+1}) - g^{k+1}\|^2 + M_2 \cdot \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^{k+1}) - y_j^{k+1}\|^2\right]$$

$$\leq (1-\eta_k)\,\mathbb{E}\left[f(x^k)-f(x^*)+M_1\cdot\|\nabla f(x^k)-g^k\|^2+M_2\cdot\frac{1}{n}\sum_{j=1}^n\|\nabla f_j(x^k)-y_j^k\|^2\right]$$
$$+LD^2\eta_k^2\left(\frac{1}{2}+\frac{10\tilde{L}\sqrt{n}}{Lb}\right).$$

It remains to use Lemma A.3 with $c=LD^2\left(\frac{1}{2}+\frac{10\tilde{L}\sqrt{n}}{Lb}\right)$, $d=\frac{4n}{b}$ and obtain

$$\mathbb{E}\left[f(x^K)-f(x^*)\right]=\mathcal{O}\Bigg(\frac{n}{b}\Bigg[f(x^0)-f(x^*)+M_1\cdot\|\nabla f(x^0)-g^0\|^2$$
$$+M_2\cdot\frac{1}{n}\sum_{j=1}^n\|\nabla f_j(x^0)-y_j^0\|^2\Bigg]\exp\left(-\frac{bK}{8n}\right)+\left[1+\frac{\tilde{L}\sqrt{n}}{Lb}\right]\frac{LD^2}{K}\Bigg).$$

Finally, we substitute $g^0=\nabla f(x^0)$, $y_j^0=\nabla f_j(x^0)$ and get

$$\mathbb{E}\left[f(x^K)-f(x^*)\right]=\mathcal{O}\left(\frac{n}{b}\left[f(x^0)-f(x^*)\right]\exp\left(-\frac{bK}{8n}\right)+\left[1+\frac{\tilde{L}\sqrt{n}}{Lb}\right]\frac{LD^2}{K}\right).$$

This completes the proof.

□

B.4. Proof of Theorem 4.7

Theorem B.4 (Theorem 4.7). *Let $\{x^k\}_{k\geq 0}$ denote the iterates of Algorithm 2 for solving problem (1), which satisfies Assumptions 3.1,3.3. Let x^* be the global (may be not unique) minimizer of f on $\mathcal{X}$. Then, if we choose $\eta_k=\frac{1}{\sqrt{K}}$ and $\lambda=\frac{b}{2n}$, we have the following convergence:*

$$\mathbb{E}\left[\min_{0\leq k\leq K-1}\text{gap}(x^k)\right]=\mathcal{O}\left(\frac{f(x^0)-f(x^*)}{\sqrt{K}}+\frac{LD^2}{\sqrt{K}}\left[1+\frac{\tilde{L}\sqrt{n}}{Lb}\right]\right).$$

Proof: Since lines 2 and 3 of Algorithms 1 and 2 are the same, we start the proof from (9). Multiplying (10) by the positive constant M_1, (11) by the positive constant M_2 (M_1, M_2 will be defined below) and summing with (9), we have

$$\eta_k\mathbb{E}\big[\max_{x\in\mathcal{X}}\langle\nabla f(x^k),x^k-x\rangle\big]$$
$$\leq(1-\eta_k)\mathbb{E}\left[f(x^k)-f(x^*)\right]$$
$$+\left(1-\lambda+\frac{\alpha}{M_1L}\right)M_1\cdot\mathbb{E}\left[\|\nabla f(x^k)-g^k\|^2\right]$$
$$+\left(1-\frac{b}{2n}+\frac{2M_1\lambda^2}{M_2b}\right)M_2\cdot\frac{1}{n}\sum_{j=1}^n\mathbb{E}\left[\|\nabla f_j(x^k)-y_j^k\|^2\right]$$
$$-\mathbb{E}\left[f(x^{k+1})-f(x^*)+M_1\cdot\|\nabla f(x^{k+1})-g^{k+1}\|^2+M_2\cdot\frac{1}{n}\sum_{j=1}^n\|\nabla f_j(x^{k+1})-y_j^{k+1}\|^2\right]$$
$$+\frac{2(M_1+nM_2)\tilde{L}^2\eta_k}{b}\mathbb{E}\left[\|s^k-x^k\|^2\right]+\frac{L\eta_k^2}{\alpha}\mathbb{E}\left[\max_{x\in\mathcal{X}}\|s^k-x\|^2\right]+\frac{L\eta_k^2}{2}\mathbb{E}\left[\|s^k-x^k\|^2\right].$$

With $M_1=\frac{\alpha}{\lambda L}$ and $M_2=\frac{4M_1\lambda^2n}{b^2}$, we obtain

$$\eta_k\mathbb{E}\big[\max_{x\in\mathcal{X}}\langle\nabla f(x^k),x^k-x\rangle\big]$$

$$
\begin{aligned}
&\leq \mathbb{E}\left[f(x^k) - f(x^*) + M_1 \cdot \|\nabla f(x^k) - g^k\|^2 + M_2 \cdot \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^k) - y_j^k\|^2\right] \\
&\quad - \mathbb{E}\left[f(x^{k+1}) - f(x^*) + M_1 \cdot \|\nabla f(x^{k+1}) - g^{k+1}\|^2 + M_2 \cdot \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^{k+1}) - y_j^{k+1}\|^2\right] \\
&\quad + \frac{2\alpha\tilde{L}^2\eta_k^2}{bL\lambda}\left(1 + \frac{4\lambda^2 n^2}{b^2}\right)\mathbb{E}\left[\|s^k - x^k\|^2\right] + \frac{L\eta_k^2}{\alpha}\mathbb{E}\left[\|s^k - x^*\|^2\right] + \frac{L\eta_k^2}{2}\mathbb{E}\left[\|s^k - x^k\|^2\right].
\end{aligned}
$$

Assumption 3.3 on the diameter D of $\mathcal{X}$ gives

$$
\begin{aligned}
&\eta_k \mathbb{E}\big[\max_{x\in\mathcal{X}}\langle \nabla f(x^k), x^k - x\rangle\big] \\
&\quad \leq \mathbb{E}\left[f(x^k) - f(x^*) + M_1 \cdot \|\nabla f(x^k) - g^k\|^2 + M_2 \cdot \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^k) - y_j^k\|^2\right] \\
&\quad\quad - \mathbb{E}\left[f(x^{k+1}) - f(x^*) + M_1 \cdot \|\nabla f(x^{k+1}) - g^{k+1}\|^2 + M_2 \cdot \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^{k+1}) - y_j^{k+1}\|^2\right] \\
&\quad\quad + LD^2\eta_k^2\left(\frac{1}{2} + \frac{1}{\alpha} + \frac{2\alpha\tilde{L}^2}{bL^2\lambda}\left[1 + \frac{4\lambda^2 n^2}{b^2}\right]\right).
\end{aligned}
$$

The choices of $\lambda = \frac{b}{2n}$ and $\alpha = \frac{bL}{3\tilde{L}\sqrt{n}}$ provides

$$
\begin{aligned}
&\eta_k \mathbb{E}\big[\max_{x\in\mathcal{X}}\langle \nabla f(x^k), x^k - x\rangle\big] \\
&\quad \leq \mathbb{E}\left[f(x^k) - f(x^*) + M_1 \cdot \|\nabla f(x^k) - g^k\|^2 + M_2 \cdot \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^k) - y_j^k\|^2\right] \\
&\quad\quad - \mathbb{E}\left[f(x^{k+1}) - f(x^*) + M_1 \cdot \|\nabla f(x^{k+1}) - g^{k+1}\|^2 + M_2 \cdot \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^{k+1}) - y_j^{k+1}\|^2\right] \\
&\quad\quad + LD^2\eta_k^2\left(\frac{1}{2} + \frac{6\tilde{L}\sqrt{n}}{Lb}\right).
\end{aligned}
$$

Summing over all k from 0 to $K-1$, taking $\eta_k = \frac{1}{\sqrt{K}}$ and dividing both sides by

$$
\begin{aligned}
&\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\max_{x\in\mathcal{X}}\langle \nabla f(x^k), x^k - x\rangle\right] \\
&\quad \leq \frac{1}{\sqrt{K}} \cdot \left[f(x^0) - f(x^*) + M_1 \cdot \|\nabla f(x^0) - g^0\|^2 + M_2 \cdot \frac{1}{n}\sum_{j=1}^n \|\nabla f_j(x^0) - y_j^0\|^2\right] \\
&\quad\quad + \frac{LD^2}{\sqrt{K}}\left(\frac{1}{2} + \frac{6\tilde{L}\sqrt{n}}{Lb}\right).
\end{aligned}
$$

Finally, we substitute $g^0 = \nabla f(x^0)$, $y_j^0 = \nabla f_j(x^0)$ and get

$$
\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\max_{x\in\mathcal{X}}\langle \nabla f(x^k), x^k - x\rangle\right] \leq \frac{f(x^0) - f(x^*)}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}}\left(\frac{1}{2} + \frac{6\tilde{L}\sqrt{n}}{Lb}\right).
$$

The definition of (4) finishes the proof.

□

C. Additional Comments

C.1. On the Convergence Criterion in (Qu et al., 2018; Gao & Huang, 2020)

In Table 1, we indicates that the papers by (Qu et al., 2018; Gao & Huang, 2020) considers $\|\nabla f(x)\|^2$ as a convergence criterion. But the authors actually use a more complex criterion $\|G(x, \nabla f(x), \gamma)\|^2$, where $G(x, \nabla f(x), \gamma) = \frac{1}{\gamma}(x - \psi(x, \nabla f(x), \gamma))$ with $\psi(x, \nabla f(x), \gamma) = \arg\min_{y \in C} \left(\langle \nabla f(x), y \rangle + \frac{1}{2\gamma}\|x - y\|^2 \right)$. Let us simplify this criterion. If C is large enough, one can assume that we work on unconstrained setting and $\arg\min_{y \in C} \left(\langle \nabla f(x), y \rangle + \frac{1}{2\gamma}\|x - y\|^2 \right) \approx \arg\min_{y \in R^d} \left(\langle \nabla f(x), y \rangle + \frac{1}{2\gamma}\|x - y\|^2 \right)$, i.e. $\psi(x, \nabla f(x), \gamma) = x - \gamma \nabla f(x)$. Therefore, we get that $G(x, \nabla f(x), \gamma) \approx \nabla f(x)$ and $\|G(x, \nabla f(x), \gamma)\|^2 \approx \|\nabla f(x)\|^2$. As we noted in Table 1, to avoid discrepancies with the lower bounds from (Li et al., 2021a), we slightly modified the result of (Qu et al., 2018; Gao & Huang, 2020). In more details, following (Li et al., 2021a), we assume that want to achieve $\|G(x, \nabla f(x), \gamma)\|^2 \sim \varepsilon^2$. In the original papers (Qu et al., 2018; Gao & Huang, 2020), the authors uses $\|G(x, \nabla f(x), \gamma)\|^2 \sim \varepsilon$.

C.2. Incorrect Proof of Theorem 4 from (Reddi et al., 2016)

As we noted in Section 2, the paper provides another algorithm (Algorithm 4). This method is a modification of the SAGA technique. The proof of convergence of this algorithm, in our opinion, contains a mistake. The authors introduce an additional technical sequence:

$$c_t = (1 - \rho)c_{t+1} + \frac{LD\gamma\sqrt{n}}{\sqrt{b}} \quad \text{with} \quad c_T = 0, \tag{12}$$

and claim that the following estimate is valid:

$$\sum_{t=1}^{T} c_t \leq \frac{LD\gamma\sqrt{n}}{\rho\sqrt{b}}.$$

But if we consider the simplest case with $\rho = 1$, we have that

$$c_t = \frac{LD\gamma\sqrt{n}}{\sqrt{b}} \quad \text{and} \quad \sum_{t=1}^{T} c_t \leq \frac{LD\gamma\sqrt{n}}{\sqrt{b}} \cdot (T - 1),$$

which is larger than the authors' estimate: $\frac{LD\gamma\sqrt{n}}{\sqrt{b}}$.

Let us try to correct this error. Running the recursion (12), we get for all $t = 0, \ldots, (T - 1)$

$$c_t = \frac{LD\gamma\sqrt{n}}{\sqrt{b}} \cdot \sum_{i=T}^{t+1} (1 - \rho)^{T-i} \leq \frac{LD\gamma\sqrt{n}}{\rho\sqrt{b}}$$

And then,

$$\sum_{t=1}^{T} c_t \leq \frac{LD\gamma\sqrt{n}}{\rho\sqrt{b}} \cdot (T - 1).$$

With (13) from (Reddi et al., 2016): $\rho \geq \frac{b}{2n}$, one can obtain

$$\sum_{t=1}^{T} c_t \leq \frac{LD\gamma n^{3/2}}{b^{3/2}} \cdot (T - 1).$$

The result is the following estimate

$$\eta \sum_{k=0}^{K-1} \mathbb{E}\left[\max_{x \in \mathcal{X}} \langle \nabla f(x^k), x^k - x \rangle \right] \leq f(x^0) - f(x^*) + LD^2\eta^2 K \left(1 + \frac{n^{3/2}}{b^{3/2}} \right).$$

If we take $\eta = \frac{1}{\sqrt{K}}$ and divide both sides by $\sqrt{K}$, then

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\max_{x\in\mathcal{X}}\langle\nabla f(x^k), x^k - x\rangle\right] \leq \frac{f(x^0)-f(x^*)}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}}\left(1+\frac{n^{3/2}}{b^{3/2}}\right).$$

The definition of (4) gives

$$\mathbb{E}\left[\min_{0\leq k\leq K-1}\text{gap}(x^k)\right] \leq \frac{f(x^0)-f(x^*)}{\sqrt{K}} + \frac{LD^2}{\sqrt{K}}\left(1+\frac{n^{3/2}}{b^{3/2}}\right).$$

Therefore, the following number of the stochastic oracle calls is needed to achieve the accuracy ε:

$$\mathcal{O}\left(b\left[\frac{f(x^0)-f(x^*)}{\varepsilon}\right]^2 + \left[\frac{LD^2}{\varepsilon}\right]^2\left[b+\frac{n^3}{b^2}\right]\right).$$

With the optimal choice of $b = n$, we get

$$\mathcal{O}\left(n\left[\frac{f(x^0)-f(x^*)}{\varepsilon}\right]^2 + n\left[\frac{LD^2}{\varepsilon}\right]^2\right).$$

C.3. Big Batches and LMO Complexities

Here we study the use of large batches to obtain better estimates on LMO by using of lower bounds. For this purposes, we need to introduce a class of algorithms for which the lower bounds will be valid. Since we work with projection-free methods, the following definition takes this into account.

Definition C.1. We have local memory $\mathcal{M}$ with initialization $\mathcal{M} = \{0\}$. In addition, we also have an auxiliary buffer $\mathcal{H}$, which is also initially equal to $\{0\}$. These memory $\mathcal{M}$ and buffer $\mathcal{H}$ are updated as follows.

• One can sample uniformly and independently batch S of any size b (if $b = n$ – it means that we call the full gradient) from $\{f_i\}$ at some point $x \in \mathcal{M}$, compute stochastic gradient $g = \sum_{j\in S}\nabla f_j(x)$ and adds it to the buffer $\mathcal{H}$ as linear combination of existing vectors in the buffer:

$$\mathcal{H} = \text{span}\{\mathcal{H}, g\}.$$

We can repeat this operation with different batches with any sizes and different points $x \in \mathcal{M}$.

• Using information in the buffer, we can update our memory $\mathcal{M}$ by adding to $\mathcal{M}$ a finite number of points x', satisfying

$$x' \in \text{span}\{x, \text{LMO}(h, \mathcal{X})\},$$

where we can take any $x \in \mathcal{M}$, $h \in \mathcal{H}$ and LMO is the linear minimization oracle.

• The final global output is calculated as $x \in \mathcal{M}$.

The next step in constructing lower bounds is to create a "bad" problem on which all methods perform poorly. In our case, the problem consists of two parts: a function f with its division by f_i, and an optimization set $\mathcal{C}$. The functions can be taken from works on lower bounds for the unconstrained case, e.g., in the convex (Han et al., 2024) and non-convex (Fang et al., 2018; Li et al., 2021a) setups. As an optimization set we choose ℓ_1-ball with 0 center and size $R = 1$. For this ball:

$$\text{LMO}(h, \mathcal{X}) = -\text{sign}(h_i)e_i \quad \text{with} \quad i = \arg\max_j |g_j|.$$

If the solution of argmax is not unique, we choose the smallest one.

The essence of the lower bounds is classical (Nesterov, 2013) – how the final output close to the real solution is measured in the number of non-zero coordinates in the output. On non-zero coordinates we can (in the best case) get a number corresponding to the real solution, and on zero coordinates we cannot. How it works for unconstrained optimization methods: each gradient call "open" a new non-zero coordinate, then as many gradients we compute – that is how many non-zero coordinates we have.

In our case, we have also LMO in the update rule. If $\mathcal{H} = \text{span}\{e_1, \ldots, e_k\}$, then we can guarantee that LMO for any vector from $\mathcal{H}$ lies in $\text{span}\{e_1, \ldots, e_k\}$, since LMO for ℓ_1-ball is a maximum absolute value of coordinates and if there are two maximums we choose the smaller one. It means that LMO for our set does give any progress in terms of non-zero coordinates, but we can make this progress in $\mathcal{H}$ and using LMO we transfer a new non-zero coordinate to $\mathcal{M}$. Therefore, we need to understand how batching affects $\mathcal{H}$, and then we immediately understand how it affects LMO.

Proposition C.2. *After K LMO computations, only the first K coordinates of the global output can be non-zero while the rest of the $d - K$ coordinates are strictly equal to zero.*

Proof: We compute full gradients in the best case, these full gradients add one more non-zero coordinate in $\mathcal{H}$ and then using LMO in $\mathcal{M}$.

□

Proposition C.3. *If between LMO computations and $\mathcal{M}$ updates we collect batch size of 1 in only one point, then after K LMO computations, in expectation only the first $\frac{K}{n}$ coordinates of the global output can be non-zero while the rest of the $d - \frac{K}{n}$ coordinates are strictly equal to zero.*

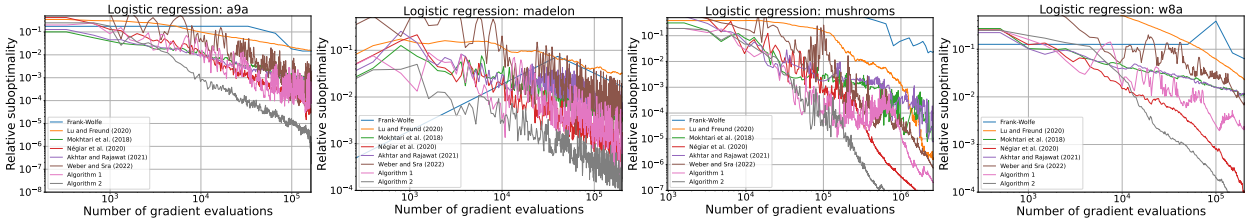
Proof: Now we cannot compute full gradients, we compute only batch of size 1. The key problem with such a batch is that we set f_i in such a way that different parts of the whole f are stored on different f_i, and only one of all f_is can increase the number of non-zero coordinates (Fang et al., 2018; Li et al., 2021a; Han et al., 2024). But by virtue of the fact that we choose this function randomly, then we get a new non-zero coordinate with probability $1/n$. It turns out that we can call LMO for nothing and not get new non-zero coordinates in the output of $\mathcal{M}$.

□

Propositions C.2 and C.3 show that with different batching we can get different number of non-zero coordinates. These propositions are not a rigorous justification for the use of large (e.g., $\sqrt{n}$) or even full batches, but they provide a semi-formalized intuition for why this might be the case.

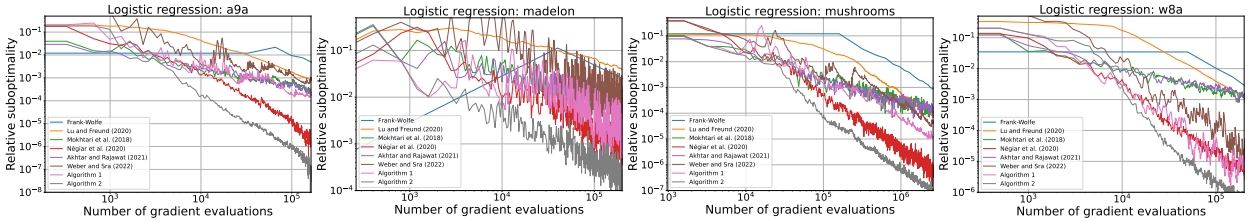
D. Additional Experiments

Here we give experiments on logistic regression but unlike the main part we consider other sizes of the ℓ_1-ball $R = 200, 20, 2$. These experiments also verify the superiority of our algorithms.



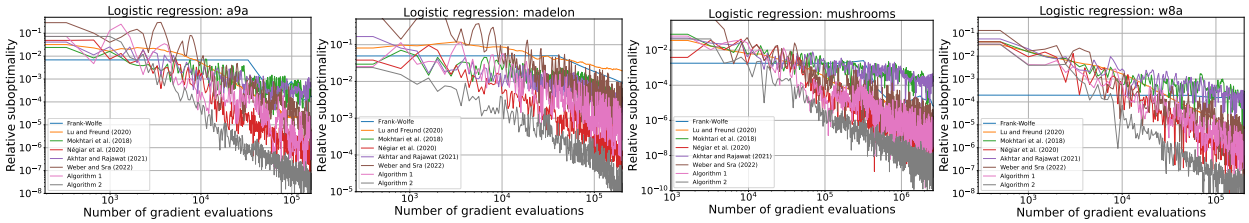
(a) a9a (b) breast cancer (c) mushrooms (d) w8a

Figure 3: Comparison of state-of-the-art projection free methods with small batches for (5) with $R = 200$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations.



(a) a9a (b) breast cancer (c) mushrooms (d) w8a

Figure 4: Comparison of state-of-the-art projection free methods with small batches for (5) with $R = 20$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations.



(a) a9a (b) breast cancer (c) mushrooms (d) w8a

Figure 5: Comparison of state-of-the-art projection free methods with small batches for (5) with $R = 2$. The comparison is made on the real datasets from LibSVM. The criterion is the number of full gradients computations.