Spring 2021

Scalable Projection-Free Optimization

Mingrui Zhang
Yale University Graduate School of Arts and Sciences, pkusmszmr@gmail.com

Abstract

Scalable Projection-Free Optimization

Mingrui Zhang

2021

As a projection-free algorithm, Frank-Wolfe (FW) method, also known as conditional gradient, has recently received considerable attention in the machine learning community. In this dissertation[1], we study several topics on the FW variants for scalable projection-free optimization.

We first propose `1-SFW`, the first projection-free method that requires only one sample per iteration to update the optimization variable and yet achieves the best known complexity bounds for convex, non-convex, and monotone DR-submodular settings. Then we move forward to the distributed setting, and develop `Quantized Frank-Wolfe` (QFW), a general communication-efficient distributed FW framework for both convex and non-convex objective functions. We study the performance of QFW in two widely recognized settings: 1) stochastic optimization and 2) finite-sum optimization. Finally, we propose `Black-Box Continuous Greedy`, a derivative-free and projection-free algorithm, that maximizes a monotone continuous DR-submodular function over a bounded convex body in Euclidean space.

[1]This dissertation includes the following publications [1, 2, 3]. Some passages in this dissertation have been quoted verbatim from the above papers.

Scalable Projection-Free Optimization

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Mingrui Zhang

Dissertation Director: Amin Karbasi

June 2021

Acknowledgments

During the six years for pursuing a Ph.D. degree, I have received a lot of help, support and encouragement.

First and foremost I would like to express my sincere gratitude to my advisors, Professor Amin Karbasi and Professor Sekhar Tatikonda, for their constant guidance, regular meetings, helpful advice and support for my research and this dissertation. I am deeply grateful to Professor Huibin Zhou, the other member in my dissertation committee. I would also like to thank all my research collaborators: Professor Hamed Hassani, Lin Chen, Yifei Min, Aryan Mokhtari, Zebang Shen.

I would like to express my sincere thanks to all my teachers from the elementary school to the graduate school. Especially I would like to offer my thanks to the following people, listed in a chronological order: Lan Yang, Xiabing Shen, Weizi Zheng, Panfeng Wang, Professor Chunwei Song, Professor Houhong Fan, Professor Minping Qian, Professor Daquan Jiang.

I would also like to thank my friends at Yale University, Peking University, Kaifeng High School, etc. Here I would like to give thanks to the following people, listed in an alphabetic order: Peiliang Bai, Lin Chen, Siyuan Dong, Xialiang Dou, Bei Fan, Peizhen Guo, Chris Harshaw, Bo Hu, Wenmian Hua, Dingjue Ji, Pengcheng Li, Yitong Li, Yupeng Li, Naijia Liu, Zhan Liu, Marko Mitrovic, Zhichao Peng, Weiqi Shi, Chuan Tian, Kaizheng Wang, Qingcan Wang, Shuai Wang, Xingyan Wang, Zeyu Wang, Zhixin Wang, Zheng Wei, Yujun Xie, Wenjie Xiong, Ruitu Xu, Sheng Xu, Yihan Zhou, Yun Zhou, Zihan Zhuo.

In the end, I would like to express my deepest gratitude to my parents for their endless love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In many modern machine learning scenarios, the task of learning is usually converted to an optimization problem, where the loss function is defined as the empirical loss function (plus some possible regularization). In many cases, the corresponding constraint set is not the whole Euclidean space, but some bounded convex set.

In order to solve these *constrained* optimization problems, methods like Projected Gradient Descent (PGD) are quite popular and effective in practice. In these methods, projection oracle is applied once the proposed iterates land outside the feasibility region. However, the projection operation can be computationally expensive for some special constraint sets. For example, in recommender systems and matrix completion, projections amount to expensive linear algebraic operations. Similarly, projections onto matroid polytopes with exponentially many linear inequalities are daunting tasks in general. This difficulty has motivated the use of projection-free algorithms.

As a projection-free algorithm for various constrained convex [4, 5, 6, 7, 8] and non-convex [9, 10, 11, 12, 13] optimization problems, the Frank-Wolfe (FW) method [14], also known as conditional gradient, has recently received considerable attention in the machine learning community.

In this dissertation, we investigate various topics on the FW variants for scalable projection-free optimization.

1.1 One-Sample Stochastic Frank-Wolfe

Although Frank-Wolfe (FW) methods have been widely used for solving constrained optimization problems [4, 5, 14], exact gradient evaluations are required in order to guarantee convergence. In many cases, however, exact gradients are difficult to compute or even inaccessible. This challenge motivates the study of FW variants which can be fed with stochastic gradient information.

Indeed, extending the original FW methods to the stochastic setting is a challenging task as it is known that FW-type methods are highly sensitive to stochasticity in gradient computation [15]. To resolve this issue, several stochastic variants of FW methods have been studied in the literature [7, 10, 13, 15, 16, 17, 18, 19]. In all these stochastic methods, the basic idea is to provide an accurate estimate of the gradient by using some variance reduction techniques that typically rely on large mini-batches of samples where the size grows with the number of iterations or is reciprocal of the desired accuracy. A growing mini-batch, however, is undesirable in practice as requiring a large collection of samples per iteration may easily prolong the duration of each iterate without updating optimization parameters frequently enough [20]. A notable exception to this trend is the the work of [8] which employs a momentum variance reduction technique requiring only one sample per iteration; however, this method suffers from suboptimal convergence rates.

In this dissertation, we present the first projection-free method that requires only one sample per iteration to update the optimization variable and yet achieves the best known complexity bounds for convex, non-convex, and monotone DR-submodular settings.

1.2 Communication-Efficient Frank-Wolfe in the Distributed Setting

Thanks to the numerous information-sensors and many other modern information technologies, the sizes of available datasets have been growing fast recently. As a result, efficient FW methods are also motivated to be applied to large-scale problems (*e.g.*, training deep neural networks [21, 22, 23], RBMs [24]). To this end, distributed FW variants have been proposed for specific problems, *e.g.*, online learning [25], learning low-rank matrices [26], and optimization under block-separable constraint sets [27].

As is well known, a significant performance bottleneck of distributed optimization methods is the cost of communicating gradients, which is typically handled by using a parameter-server framework. Intuitively, if each worker in the distributed system transmits the entire gradient, then at least d floating-point numbers are communicated for each worker, where d is the dimension of the problem. This communication cost can be a huge burden on the performance of parallel optimization algorithms [28, 29, 30]. To circumvent this drawback, communication-efficient parallel algorithms have received significant attention. One major approach is to quantize the gradients while maintaining sufficient information [31, 32, 33]. For *unconstrained* optimization, when projection is not required for implementing Stochastic Gradient Descent (SGD), several communication-efficient distributed methods have been proposed, including QSGD [34], SIGN-SGD [35], and Sparsified-SGD [36].

In the constrained setting, and in particular for distributed FW methods, the communication-efficient versions were only studied for specific problems such as sparse learning [37, 38]. In this dissertation, we develop `Quantized Frank-Wolfe` (QFW), a general communication-efficient distributed FW framework for both convex and non-convex objective functions. We also study the performance of QFW in two widely recognized settings:

1) stochastic optimization and 2) finite-sum optimization.

1.3 Black-Box Submodular Maximization

Black-Box optimization, also known as zeroth-order or derivative-free optimization[1], has been extensively studied in the literature [40, 41, 42, 43]. In this setting, we assume that the objective function is unknown and we can only obtain zeroth-order information such as (stochastic) function evaluations.

Fueled by a growing number of machine learning applications, black-box optimization methods are usually considered in scenarios where gradients (*i.e.*, first-order information) are 1) difficult or slow to compute, *e.g.*, graphical model inference [44], structure predictions [45, 46], or 2) inaccessible, *e.g.*, hyper-parameter turning for natural language processing or image classifications [47, 48], black-box attacks for finding adversarial examples [49, 50]. Even though heuristics such as random or grid search, with undesirable dependencies on the dimension, are still used in some applications (*e.g.*, parameter tuning for deep networks), there have been a growing number of rigorous methods to address the convergence rate of black-box optimization in convex and non-convex settings [51, 52, 53].

Continuous DR-submodular functions are an important subset of non-convex functions that can be minimized exactly [54, 55] and maximized approximately [12, 13, 56, 57, 58, 59]. This class of functions generalizes the notion of diminishing returns, usually defined over discrete set functions, to the continuous domains. They have found numerous applications in machine learning including MAP inference in determinantal point processes (DPPs) [60], experimental design [61], resource allocation [62], mean-field inference in

[1]We note that black-box optimization (BBO) and derivative-free optimization (DFO) are not identical terms. Audet and Hare [39] defined DFO as "the mathematical study of optimization algorithms that do not use derivatives" and BBO as "the study of design and analysis of algorithms that assume the objective and/or constraint functions are given by blackboxes". However, as the differences are nuanced in most scenarios, this dissertation uses them interchangeably. For a detailed review of DFO and BBO, interested readers refer to book [39].

probabilistic models [63], among many others.

In this dissertation, we propose a derivative-free FW method for continuous DR-submodular maximization over a bounded convex body, which also avoids the expensive projection operations.

1.4 Organization

In this dissertation, we study several topics on scalable projection-free optimization algorithms. The rest of this dissertation is organized as follows.

In Chapter 2, we introduce important background and related work.

In Chapter 3, we present the first one-sample stochastic Frank-Wolfe method, called `1-SFW`, which attains the best known complexity bounds for convex, non-convex, and monotone DR-submodular settings, while requiring only *one single* stochastic oracle query per iteration and avoiding large batch sizes altogether. In particular, we show that `1-SFW` achieves the optimal convergence rate of $\mathcal{O}(1/\epsilon^2)$ for reaching an ϵ-suboptimal solution in the stochastic convex setting, and a $(1-1/e)-\epsilon$ approximate solution for a stochastic monotone DR-submodular maximization problem. In a general non-convex setting, `1-SFW` finds an ϵ-first-order stationary point after at most $\mathcal{O}(1/\epsilon^3)$ iterations. We also empirically validate the efficiency of `1-SFW` algorithm by comparing it with baseline methods in Section 3.5. This chapter is based on our work in [1][2].

In Chapter 4, we propose a novel distributed projection-free framework, `Quantized Frank-Wolfe` (QFW), which handles quantization for constrained convex and non-convex optimization problems in finite-sum and stochastic cases. We show that with quantized gradients, we can obtain a provably convergent method which preserves the convergence rates of the state-of-the-art vanilla centralized methods in all the considered

[2]This work was done in collaboration with Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. I proposed the algorithms and also completed the work of theoretical analysis.

cases [12, 13, 18, 19]. In Section 4.5, we evaluate the performance of algorithms by visualizing their loss vs. the number of transmitted bits on two problems: multinomial logistic regression and three-layer neural network under ℓ_1 constraint. This chapter is based on our work in [2][3].

In Chapter 5, we propose a derivative-free and projection-free algorithm `Black-Box Continuous Greedy` (BCG), that maximizes a monotone continuous DR-submodular function over a bounded convex body in Euclidean space. We study three scenarios:

(1) In the deterministic setting, where function evaluations can be obtained exactly, BCG achieves the tight $[(1-1/e)OPT-\epsilon]$ approximation guarantee with $\mathcal{O}(d/\epsilon^3)$ function evaluations.

(2) In the stochastic setting, where function evaluations are noisy, BCG achieves the tight $[(1-1/e)OPT-\epsilon]$ approximation guarantee with $\mathcal{O}(d^3/\epsilon^5)$ function evaluations.

(3) In the discrete setting, `Discrete Black-Box Greedy` (DBG), the discrete version of BCG, achieves the tight $[(1-1/e)OPT-\epsilon]$ approximation guarantee with $\mathcal{O}(d^5/\epsilon^5)$ function evaluations.

In Section 5.5, numerical experiments show that empirically, our proposed algorithm often requires significantly fewer function evaluations and less running time compared with baselines, while achieving a practically similar utility. This chapter is based on our work in [3][4].

[3]This work was done in collaboration with Lin Chen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. I proposed the algorithms and also completed the work of theoretical analysis.

[4]This work was done in collaboration with Lin Chen, Hamed Hassani, and Amin Karbasi. I proposed the algorithms, completed the work of theoretical analysis and numerical experiments jointly with Lin Chen.

Chapter 2

Preliminaries

In this chapter, we present important preliminaries and related work for the topics of this dissertation.

2.1 Frank-Wolfe Algorithm

Frank-Wolfe method [14], also known as the conditional gradient method, has been studied for both convex optimization [4, 5, 6, 7, 8] and non-convex optimization problems [9, 10, 11, 13, 64].

As a projection-free algorithm, FW method replaces the projection operations by solving a linear optimization problem for each iteration. To be precise, suppose that we want to solve the following constrained convex optimization problem:

$$\min_{x \in \mathcal{K}} F(x),$$

where $\mathcal{K} \subseteq \mathbb{R}^d$ is a compact convex set, and F is convex. Assuming that we have access to the *exact* gradient ∇F, the FW method starts from some initial point $x^{(1)} \in \mathcal{K}$, and at the

k-th iteration, solves a linear optimization problem

$$v^{(k)} \leftarrow \arg\min_{v \in \mathcal{K}} \langle v, \nabla F(x^{(k)}) \rangle,$$

which is used to update $x^{(k+1)} \leftarrow x^{(k)} + \eta_k (v^{(k)} - x^{(k)})$, where $\eta_k \in [0, 1]$ is the step size. Note that $x^{(k+1)}$ is a convex combination of $x^{(k)}$ and $v^{(k)}$, which are defined to fall in $\mathcal{K}$. Therefore, we have $x^{(k+1)} \in \mathcal{K}$, thus avoid the projection operation which is used to guarantee all the iterates land inside the constraint set.

Since linear optimization problems can usually be solved fast by various algorithms, thus for many practical problems, FW methods can be much more computationally efficient than the projected gradient-based methods like PGD.

In terms of theoretical convergence rate, it can be shown that the iterates of the FW method above satisfies that

$$F(x^{(k)}) - F(x^*) \leq \mathcal{O}(\frac{1}{k}),$$

where x^* is the global minimizer of the convex function F in $\mathcal{K}$ [14, 65]. FW methods can also be utilized to solve non-convex minimization and monotone continuous DR-submodular maximization problems with slight modifications [9, 57].

When fed with *stochastic* gradient, however, FW methods may diverge [7, 8]. In order to establish guaranteed convergences, stochastic FW methods are usually incorporated with various variance reduction techniques. A detailed summary of convergence rates for various stochastic FW-type algorithms for convex minimization, non-convex minimization, and monotone continuous DR-submodular maximization problems can be found in Table 3.1.

2.1.1 Related Work on Frank-Wolfe Algorithm

Frank-Wolfe methods are very sensitive to noisy gradients. This issue was recently resolved in centralized [8] and online settings [66, 67]. In large-scale settings, distributed FW methods were proposed to solve specific problems, including optimization under block-separable constraint set [27], and learning low-rank matrices [26]. The communication-efficient distributed FW variants were proposed for specific sparse learning problems in [37, 38], and for general constrained optimization problems in our paper [2]. Zeroth-order FW methods were studied in [3, 52, 53].

Stochastic FW methods are strongly associated with variance reduction techniques. Several works have studied different ideas for reducing variance. The SVRG method was proposed by [68] for the convex setting and then extended to the non-convex setting in [10, 69, 70]. The StochAstic Recursive grAdient algoritHm (SARAH) was studied in [71, 72]. Then as a variant of SARAH, the Stochastic Path-Integrated Differential Estimator (SPIDER) technique was proposed by [73]. Based on SPIDER, various algorithms for convex and non-convex optimization problems have been studied [13, 18, 19].

2.2 Submodular Functions

We say a set function $f : 2^{\Omega} \to \mathbb{R}$ is submodular, if it satisfies the diminishing returns property: for any $A \subseteq B \subseteq \Omega$ and $x \in \Omega \setminus B$, we have

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B). \tag{2.1}$$

In words, the marginal gain of adding an element x to a subset A is no less than that of adding x to its superset B. A submodular set function $f : 2^{\Omega} \to \mathbb{R}$ is called monotone if for any two sets $A \subseteq B \subseteq \Omega$ we have $f(A) \leq f(B)$.

For the continuous analogue, consider a function $F : \mathcal{X} \to \mathbb{R}_+$, where $\mathcal{X} = \Pi_{i=1}^d \mathcal{X}_i$, and each $\mathcal{X}_i$ is a compact subset of $\mathbb{R}_+$. We define F to be continuous submodular if F is continuous and for all $x, y \in \mathcal{X}$, we have

$$F(x) + F(y) \geq F(x \vee y) + F(x \wedge y), \tag{2.2}$$

where $\vee$ and $\wedge$ are the component-wise maximizing and minimizing operators, respectively.

For two vectors $x, y \in \mathbb{R}^d$, we write $x \leq y$ if $x_i \leq y_i$ holds for every $i \in \{1, 2, \cdots, d\}$, where x_i is the i-th coordinate of x. Then a continuous function F is called DR-submodular [57] if F is differentiable and for all $x \leq y$: $\nabla F(x) \geq \nabla F(y)$. The function F is called monotone if for $x \leq y$, we have $F(x) \leq F(y)$.

An important implication of DR-submodularity is that the function F is concave in any non-negative directions, *i.e.*, for $x \leq y$, we have

$$F(y) \leq F(x) + \langle \nabla F(x), y - x \rangle. \tag{2.3}$$

For continuous DR-submodular maximization, it has been shown that approximated solution within a factor of $(1 - e^{-1} + \epsilon)$ can not be obtained in polynomial time [57].

2.2.1 Related Work on Submodular Functions

Submodular functions [74], that capture the intuitive notion of diminishing returns, have become increasingly important in various machine learning applications. Examples include graph cuts in computer vision [75, 76], data summarization [77, 78, 79, 80, 81], influence maximization [82, 83, 84], feature compression [85], network inference [86], active and semi-supervised learning [87, 88, 89], crowd teaching [90], dictionary learning [91], fMRI parcellation [92], compressed sensing and structured sparsity [93, 94], fairness in machine learning [95, 96], learning causal structures [97, 98], experimental design [61], MAP inference in determinantal point processes (DPPs) [60], and mean-field inference in probabilistic models [99], to name a few.

Continuous DR-submodular functions naturally extend the notion of diminishing returns to the continuous domains [57]. Monotone continuous DR-submodular functions can be minimized exactly [55, 100], and maximized approximately [13, 56, 57, 58, 59, 101, 102, 103]. Among those works, Bach [100] derived connections between continuous submodularity and convexity, while Bian et al. [57] studied the offline continuous DR-submodular maximization and proposed a variant of the Frank-Wolfe algorithm to achieve the tight $(1 - 1/e)$ approximation ratio. A derivative-free and projection-free algorithm for monotone continuous DR-submodular maximization was proposed in our work [3].

In the online setting, maximization of submodular set functions was studied in [104, 105]. Adaptive submodular bandit maximization was analyzed in [106]. The linear submodular bandit problems were studied in [107, 108]. The first online and bandit algorithms for general continuous submodular maximization problems were proposed in our work [109].

Chapter 3

One Sample Stochastic Frank-Wolfe

3.1 Introduction

Recall that FW algorithms are very sensitive to noisy gradients [7, 8]. As a result, many stochastic FW variants are fed with an accurate estimation of gradients by utilizing various variance reduction techniques [7, 10, 13, 15, 16, 17, 18, 19]. These variance reduction methods usually rely on large mini-batches of samples where the size grows with the number of iterations or is reciprocal of the desired accuracy. A growing mini-batch, however, is undesirable in practice as requiring a large collection of samples per iteration may easily prolong the duration of each iterate without updating optimization parameters frequently enough [20]. A notable exception to this trend is the the work of [8] which employs a momentum variance-reduction technique requiring only one sample per iteration; however, this method suffers from suboptimal convergence rates. At the heart of this chapter[1] is the answer to the following question:

> *Can we achieve the best known complexity bounds for a stochastic variant of Frank-Wolfe while using a single stochastic sample per iteration?*

We show that the answer to the above question is positive and present the first projection-

[1]This chapter is based on our work in [1].

free method that requires only one sample per iteration to update the optimization variable and yet achieves the best known complexity bounds for convex, non-convex, and monotone DR-submodular settings.

More formally, we focus on a general *non-oblivious* constrained stochastic optimization problem

$$\min_{x\in\mathcal{K}} F(x) \triangleq \min_{x\in\mathcal{K}} \mathbb{E}_{z\sim p(z;x)}[\tilde{F}(x;z)], \tag{3.1}$$

where $x \in \mathbb{R}^d$ is the optimization variable, $\mathcal{K} \subseteq \mathbb{R}^d$ is the convex constraint set, and the objective function $F : \mathbb{R}^d \to \mathbb{R}$ is defined as the expectation over a set of functions $\tilde{F}$. The function $\tilde{F} : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$ is determined by x and a random variable $z \in \mathcal{Z}$ with distribution $z \sim p(z;x)$. We refer to problem (3.1) as a non-oblivious stochastic optimization problem as the distribution of the random variable z depends on the choice of x. When the distribution p is independent of x, we are in the standard oblivious stochastic optimization regime where the goal is to solve

$$\min_{x\in\mathcal{K}} F(x) \triangleq \min_{x\in\mathcal{K}} \mathbb{E}_{z\sim p(z)}[\tilde{F}(x;z)]. \tag{3.2}$$

Hence, the oblivious problem (3.2) can be considered as a special case of the non-oblivious problem (3.1). Note that non-oblivious stochastic optimization has broad applications in machine learning, including multi-linear extension of a discrete submodular function [13], MAP inference in determinantal point processes (DPPs) [60], and reinforcement learning [64, 110, 111, 112].

Our goal is to propose an efficient FW-type method for the non-oblivious optimization problem (3.1). Here, the efficiency is measured by the number of stochastic oracle queries, *i.e.*, the sample complexity of z. As we mentioned earlier, among the stochastic variants of FW, the momentum stochastic Frank-Wolfe method proposed in [8, 101] is the only method that requires only one sample per iteration. However, the stochastic oracle complexity of this algorithm is suboptimal, *i.e.*, $\mathcal{O}(1/\epsilon^3)$ stochastic queries are required for both convex minimization and monotone DR-submodular maximization problems. This suboptimal rate is due to the fact that the gradient estimator in momentum FW is biased and it is necessary to use a more conservative averaging parameter to control the effect of the bias term.

To resolve this issue, we propose a one-sample stochastic Frank-Wolfe method, called `1-SFW`, which modifies the gradient approximation in momentum FW to ensure that the resulting gradient estimation is an unbiased estimator of the gradient (Section 3.2). This goal has been achieved by adding an unbiased estimator of the gradient variation $\Delta_t = \nabla F(x_t) - \nabla F(x_{t-1})$ to the gradient approximation vector (Section 3.2.1). We later explain why coming up with an unbiased estimator of the gradient difference Δ_t could be a challenging task in the non-oblivious setting and show how we overcome this difficulty (Section 3.2.2). We also characterize the convergence guarantees of `1-SFW` for convex minimization, non-convex minimization, and monotone DR-submodular maximization (Section 3.3). In particular, we show that `1-SFW` achieves the optimal convergence rate of $\mathcal{O}(1/\epsilon^2)$ for reaching an ϵ-suboptimal solution in the stochastic convex setting, and a $(1-1/e)-\epsilon$ approximate solution for a stochastic monotone DR-submodular maximization problem. Moreover, in a general non-convex setting, `1-SFW` finds an ϵ-first-order stationary point after at most $\mathcal{O}(1/\epsilon^3)$ iterations, achieving the current best known convergence rate. Finally, we study the oblivious problem in (3.2) and show that our proposed `1-SFW` method becomes significantly simpler and the corresponding theoretical results hold under less strict assumptions. For example, in the non-oblivious setting, we require second-order information as the nature of the problems requires; while in the oblivious setting, we only need access to first-order information (Theorem 4). We further highlight the similarities between the variance reduced method in [113] also known as STORM and the oblivious variant of `1-SFW`. Indeed, our algorithm has been originally inspired by STORM.

Theoretical results of `1-SFW` and other related works are summarized in Table 3.1. The complexity shows the required number of stochastic queries to obtain an ϵ-suboptimal

Table 3.1: Convergence guarantees of stochastic Frank-Wolfe methods for constrained convex minimization, non-convex minimization, and stochastic monotone continuous DR-submodular function maximization.

Function	Ref.	Batch	Complexity	Non-oblivious	Utility
Convex	[15]	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^4)$	✗	-
Convex	[7]	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^3)$	✗	-
Convex	[8]	1	$\mathcal{O}(1/\epsilon^3)$	✗	-
Convex	[19]	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^2)$	✗	-
Convex	[13]	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^2)$	✓	-
Convex	**This diss.**	1	$\mathcal{O}(1/\epsilon^2)$	✓	-
Non-convex	[7]	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon^4)$	✗	-
Non-convex	[7]	$\mathcal{O}(1/\epsilon^{4/3})$	$\mathcal{O}(1/\epsilon^{10/3})$	✗	-
Non-convex	[18]	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^3)$	✗	-
Non-convex	[19]	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^3)$	✗	-
Non-convex	[13]	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^3)$	✓	-
Non-convex	**This diss.**	1	$\mathcal{O}(1/\epsilon^3)$	✓	-
Submodular	[58]	1	$O(1/\epsilon^2)$	✗	$(1/2)\mathrm{OPT}-\epsilon$
Submodular	[8]	1	$O(1/\epsilon^3)$	✗	$(1-1/e)\mathrm{OPT}-\epsilon$
Submodular	[13]	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(1/\epsilon^2)$	✓	$(1-1/e)\mathrm{OPT}-\epsilon$
Submodular	**This diss.**	1	$\mathcal{O}(1/\epsilon^2)$	✓	$(1-1/e)\mathrm{OPT}-\epsilon$

solution in convex case; an ϵ-first-order stationary point in non-convex case; and an $\alpha \cdot$ OPT $- \epsilon$ utility in monotone DR-submodular case, where $\alpha = 1/2$ or $(1-1/e)$. These results show that `1-SFW` attains the best known complexity bounds in all the considered settings, while requiring only *one single* stochastic oracle query per iteration and avoiding large batch sizes altogether. Even though the focus of this chapter is the fundamental theory behind `1-SFW`, we provide some empirical evidence in Section 3.5. All the proofs in this chapter are provided in Section 3.7.

3.2 One Sample SFW Algorithm

3.2.1 Stochastic Gradient Approximation

In our work, we build on the momentum variance reduction approach proposed in [8, 101] to reduce the variance of the one-sample method. To be more precise, in the momentum FW method [101], we update the gradient approximation d_t at round t as follows

$$d_t = (1 - \rho_t)d_{t-1} + \rho_t \nabla \tilde{F}(x_t; z_t), \tag{3.3}$$

where ρ_t is the averaging parameter and $\nabla \tilde{F}(x_t; z_t)$ is a *one-sample* estimation of the gradient. Since d_t is a weighted average of the previous gradient estimation d_{t-1} and the newly updated stochastic gradient, it has a lower variance comparing to one-sample estimation $\nabla \tilde{F}(x_t; z_t)$. In particular, it was shown in [101] that the variance of gradient approximation in (3.3) approaches zero at a sublinear rate of $O(t^{-2/3})$. The momentum approach reduces the variance of gradient approximation, but it leads to a *biased* gradient approximation, *i.e.*, d_t is not an unbiased estimator of the gradient $\nabla F(x_t)$. Consequently, it is necessary to use a conservative averaging parameter ρ_t for momentum FW to control the effect of the bias term which leads to a sublinear error rate of $\mathcal{O}(t^{-1/3})$ and overall complexity of $\mathcal{O}(1/\epsilon^3)$.

To resolve this issue and come up with a faster momentum based FW method for the non-oblivious problem in (3.1), we slightly modify the gradient estimation in (3.3) to ensure that the resulting gradient estimation is an unbiased estimator of the gradient $\nabla F(x_t)$. Specifically, we add the term $\tilde{\Delta}_t$, which is an unbiased estimator of the gradient variation $\Delta_t = \nabla F(x_t) - \nabla F(x_{t-1})$, to d_{t-1}. This modification leads to the following gradient approximation

$$d_t = (1 - \rho_t)(d_{t-1} + \tilde{\Delta}_t) + \rho_t \nabla \tilde{F}(x_t; z_t). \tag{3.4}$$

To verify that d_t is an unbiased estimator of $\nabla F(x_t)$ we can use a simple induction argument. Assuming that d_{t-1} is an unbiased estimator of $\nabla F(x_t)$ and $\tilde{\Delta}_t$ is an unbiased estimator of $\nabla F(x_t) - \nabla F(x_{t-1})$ we have $\mathbb{E}[d_t] = (1 - \rho_t)(\nabla F(x_{t-1}) + (\nabla F(x_t) - \nabla F(x_{t-1}))) + \rho_t \nabla F(x_t) = \nabla F(x_t)$. Hence, the gradient approximation in (3.4) leads to an unbiased approximation of the gradient. Let us now explain how to compute an unbiased estimator of the gradient variation $\Delta_t = \nabla F(x_t) - \nabla F(x_{t-1})$ in the non-oblivious setting.

3.2.2 Gradient Variation Estimation

The most natural approach for estimating the gradient variation $\Delta_t = \nabla F(x_t) - \nabla F(x_{t-1})$ using only one sample z is computing the difference of two consecutive stochastic gradients, *i.e.*, $\nabla \tilde{F}(x_t; z) - \nabla \tilde{F}(x_{t-1}; z)$. However, this approach leads to an unbiased estimator of the gradient variation Δ_t only in the oblivious setting where $p(z)$ is independent of the choice of x, and would introduce bias in the more general non-oblivious case. To better highlight this issue, assume that z is sampled according to distribution $p(z; x_t)$. Note that $\nabla \tilde{F}(x_t; z)$ is an unbiased estimator of $\nabla F(x_t)$, *i.e.*, $\mathbb{E}[\nabla F(x_t; z)] = \nabla F(x_t)$, however, $\nabla \tilde{F}(x_{t-1}; z)$ is not an unbiased estimator of $\nabla F(x_{t-1})$ since $p(z; x_{t-1})$ may be different from $p(z; x_t)$.

To circumvent this obstacle, an *unbiased* estimator of Δ_t was introduced in [13]. To explain their proposal for approximating the gradient variation using only one sample, note that the difference $\Delta_t = \nabla F(x_t) - \nabla F(x_{t-1})$ can be written as

$$\begin{aligned}\Delta_t &= \int_0^1 \nabla^2 F(x_t(a))(x_t - x_{t-1})\mathrm{d}a \\ &= \left[\int_0^1 \nabla^2 F(x_t(a))\mathrm{d}a\right](x_t - x_{t-1}),\end{aligned}$$

where $x_t(a) = ax_t + (1-a)x_{t-1}$ for $a \in [0, 1]$. According to this expression, one can find an unbiased estimator of $\int_0^1 \nabla^2 F(x_t(a))\mathrm{d}a$ and use its product with $(x_t - x_{t-1})$ to find an unbiased estimator of Δ_t. It can be easily verified that $\nabla^2 F(x_t(a))(x_t - x_{t-1})$ is an unbiased estimator of Δ_t if a is chosen from $[0, 1]$ uniformly at random. Therefore, all we need is to come up with an unbiased estimator of the Hessian $\nabla^2 F$.

By basic calculus, we can show that for all $x \in \mathcal{K}$ and z with distribution $p(z; x)$, the matrix $\tilde{\nabla}^2 F(x; z)$ defined as

$$\begin{aligned}\tilde{\nabla}^2 F(x; z) = \tilde{F}(x; z)&[\nabla \log p(z; x)][\nabla \log p(z; x)]^\top \\ &+ \nabla^2 \tilde{F}(x; z) + [\nabla \tilde{F}(x; z)][\nabla \log p(z; x)]^\top \\ &+ \tilde{F}(x; z)\nabla^2 \log p(z; x) \\ &+ [\nabla \log p(z; x)][\nabla \tilde{F}(x; z)]^\top, \end{aligned} \tag{3.5}$$

is an *unbiased* estimator of $\nabla^2 F(x)$. Note that the above expression requires only one sample of z. As a result, we can construct $\tilde{\Delta}_t$ as an unbiased estimator of Δ_t using only one sample

$$\tilde{\Delta}_t \triangleq \tilde{\nabla}_t^2 (x_t - x_{t-1}), \tag{3.6}$$

where $\tilde{\nabla}_t^2 = \tilde{\nabla}^2 F(x_t(a); z_t(a))$, and $z_t(a)$ follows the distribution $p(z_t(a); x_t(a))$. By using this procedure, we can indeed compute the vector d_t in (3.4) with only one sample of z per iteration. Through a completely different analysis from the ones in [13, 101], we show that the modified d_t is still a good gradient estimation (Lemma 2), which allows the establishment of the best known stochastic oracle complexity for our proposed algorithm.

Another issue of this scheme is that in (3.5) and (3.6), we need to calculate $\nabla^2 \tilde{F}(x_t(a); z_t(a))\cdot(x_t - x_{t-1})$ and $\nabla^2 \log p(x_t(a); z_t(a))(x_t - x_{t-1})$, where computation of Hessian is involved. When exact Hessian is not accessible, however, we can resort to an approximation by the difference of two gradients. Precisely, for any function $\psi : \mathbb{R}^d \to \mathbb{R}$, any vector $u \in \mathbb{R}^d$ with $\|u\| \leq D = \max_{x,y\in\mathcal{K}}\|x - y\|$, and some $\delta > 0$ small enough, we have

$$\phi(\delta; \psi) \triangleq \frac{\nabla\psi(x + \delta u) - \nabla\psi(x - \delta u)}{2\delta} \approx \nabla^2\psi(x)u.$$

If we assume that ψ is L_2-second-order smooth, *i.e.*, $\|\nabla^2\psi(x) - \nabla^2\psi(y))\| \leq L_2\|x - y\|,$ for all $x, y \in \mathbb{R}^d$, we can upper bound the approximation error quantitatively:

$$\|\nabla^2\psi(x)u - \phi(\delta; \psi)\| = \|\nabla^2\psi(x)u - \nabla^2\psi(\tilde{x})u)\| \leq D^2 L_2 \delta, \tag{3.7}$$

where $\tilde{x}$ is obtained by the mean-value theorem. In other words, the approximation error can be sufficiently small for proper δ. So we can estimate Δ_t by

$$\begin{aligned}\tilde{\Delta}_t = {} & \tilde{F}(x; z)[\nabla \log p(z; x)][\nabla \log p(z; x)]^\top u_t \\ & + \phi(\delta_t, \tilde{F}(x; z)) + [\nabla\tilde{F}(x; z)][\nabla \log p(z; x)]^\top u_t \\ & + \tilde{F}(x; z)\phi(\delta_t, \log p(z, x)) \\ & + [\nabla \log p(z; x)][\nabla\tilde{F}(x; z)]^\top u_t,\end{aligned} \tag{3.8}$$

where $u_t = x_t - x_{t-1}, x, z, \delta_t$ are chosen appropriately. We also note that since computation of gradient difference has a computational complexity of $\mathcal{O}(d)$, while that for Hessian is $\mathcal{O}(d^2)$, this approximation strategy can also help to accelerate the optimization process.

3.2.3 Variable Update

Once the gradient approximation d_t is computed, we can follow the update of conditional gradient methods for computing the iterate x_t. In this section, we introduce two different schemes for updating the iterates depending on the problem that we aim to solve.

For minimizing a general (non-)convex function using one sample stochastic FW, we update the iterates according to

$$x_{t+1} = x_t + \eta_t(v_t - x_t), \tag{3.9}$$

where $v_t = \arg\min_{v \in \mathcal{K}}\{v^\top d_t\}$. In this case, we find the direction that minimizes the inner product with the current gradient approximation d_t over the constraint set $\mathcal{K}$, and update the variable x_{t+1} by descending in the direction of $v_t - x_t$ with step size η_t.

For monotone DR-submodular maximization, the update rule is slightly different, and a stochastic variant of the continuous greedy method [114] can be used. Using the same stochastic estimator d_t as in the (non-)convex case, the update rule for DR-Submodular optimization is given by

$$x_{t+1} = x_t + \eta_t v_t, \tag{3.10}$$

where $v_t = \arg\max_{v \in \mathcal{K}}\{v^\top d_t\}, \eta_t = 1/T, T$ is the total number of iterations. Hence, if we start from the origin, after T steps the outcome will be a feasible point as it can be written as the average of T feasible points.

The description of our proposed `1-SFW` method for smooth (non-)convex minimization as well as monotone DR-submodular maximization is outlined in Algorithm 1.

Algorithm 1 One-Sample SFW (`1-SFW`)

Input: Step sizes $\rho_t \in (0,1), \eta_t \in (0,1)$, initial point $x_1 \in \mathcal{K}$, total number of iterations T
Output: x_{T+1} or x_o, where x_o is chosen from $\{x_1, x_2, \cdots, x_T\}$ uniformly at random

for $t = 1, 2, \ldots, T$ **do**
  if $t = 1$ **then**
    Sample a point z_1 according to $p(z_1, x_1)$
    Compute $d_1 = \nabla \tilde{F}(x_1; z_1)$
  else
    Choose a uniformly at random from $[0,1]$
    Compute $x_t(a) = a x_t + (1-a) x_{t-1}$
    Sample a point z_t according to $p(z; x_t(a))$
    Compute $\tilde{\Delta}_t$ either by $\tilde{\nabla}_t^2 = \tilde{\nabla}^2 F(x_t(a); z_t)$ based on (3.5) and $\tilde{\Delta}_t = \tilde{\nabla}_t^2 (x_t - x_{t-1})$ (Exact Hessian Option); or by Eq. (3.8) with $x = x_t(a), z = z_t$ (Gradient Difference Option)
    $d_t = (1-\rho_t)(d_{t-1} + \tilde{\Delta}_t) + \rho_t \nabla \tilde{F}(x_t, z_t)$
  end if
  (non-)convex minimization: Update x_{t+1} based on (3.9)
  DR-submodular maximization: Update x_{t+1} based on (3.10)
end for

3.3 Main Results

Before presenting the convergence results of our algorithm, we first state our assumptions on the constraint set $\mathcal{K}$, the stochastic function $\tilde{F}$, and the distribution $p(z; x)$.

Assumption 1. *The constraint set $\mathcal{K} \subseteq \mathbb{R}^d$ is compact with diameter $D = \max_{x,y \in \mathcal{K}} \|x - y\|$, and radius $R = \max_{x \in \mathcal{K}} \|x\|$.*

Assumption 2. *The stochastic function $\tilde{F}(x; z)$ has uniformly bounded function value,* i.e.*, $|\tilde{F}(x; z)| \leq B$ for all $x \in \mathcal{K}, z \in \mathcal{Z}$.*

Assumption 3. *The stochastic gradient $\nabla \tilde{F}$ has uniformly bound norm: $\|\nabla \tilde{F}(x; z)\| \leq G_{\tilde{F}}$,* for all *$x \in \mathcal{K}$,* for all *$z \in \mathcal{Z}$. The norm of the gradient of* $\log p$ *has bounded fourth-order moment: $\mathbb{E}_{z \sim p(z;x)} \|\nabla \log p(z; x)\|^4 \leq G_p^4$. We also define $G = \max\{G_{\tilde{F}}, G_p\}$.*

Assumption 4. *The stochastic Hessian $\nabla^2 \tilde{F}$ has uniformly bounded spectral norm: $\|\nabla^2 \tilde{F}(x; z)\| \leq L_{\tilde{F}}$,* for all *$x \in \mathcal{K}$,* for all *$z \in \mathcal{Z}$. The spectral norm of the Hessian of* $\log p$ *has bounded* *second-order moment:* $\mathbb{E}_{z\sim p(z;x)}\|\nabla^2 \log p(z;x)\|^2 \leq L_p^2$*. We also define* $L = \max\{L_{\tilde{F}}, L_p\}$.

We note that in Assumptions 2-4, we assume that the stochastic function $\tilde{F}$ has uniformly bounded function value, gradient norm, and second-order differential. We also note that all these assumptions are necessary, and not restrictive. We elaborate on the reasons as below:

- Assumption 1: The compactness of the feasible set has been assumed in all projection-free papers. It is indeed needed for the convergence of the linear optimization subroutine in the Frank-Wolfe method, otherwise, v_t in (3.9) can be unbounded.

- Assumptions 3 and 4 about $\tilde{F}$: Bounded gradient and Hessian of the stochastic function $\tilde{F}$ are the customary assumptions for all the variance reduction methods when we solve the problem over a compact set. The boundedness of the function values (Assumption 2) is a direct implication of bounded gradient and compact constraint set.

- Assumptions 3 and 4 about the distribution p: We emphasize these assumptions hold trivially for the oblivious setting (3.2), where p is not a function of the variable x. For the non-oblivious case (3.1), consider the reinforcement learning as an example where p is the distribution of a trajectory given the policy parameter x. It can be verified that for common Gaussian policy with bounded mean and variance, the smoothness of the parameterization of the policy (*e.g.*, neural network with smooth activation function) can imply Assumptions 3 and 4.

Now with these assumptions, we can establish an upper bound for the second-order moment of the spectral norm of the Hessian estimator $\tilde{\nabla}^2 F(x;z)$ in (3.5).

Lemma 1. *[Lemma 7.1 of [13]] Under Assumptions 2-4, for all* $x \in \mathcal{K}$*, we have*

$$\mathbb{E}_{z\sim p(z;x)}[\|\tilde{\nabla}^2 F(x;z)\|^2] \leq 4B^2G^4 + 16G^4 + 4L^2 + 4B^2L^2 \triangleq \bar{L}.$$

Note that the result in Lemma (1) also implies the $\bar{L}$-smoothness of F, since

$$\begin{aligned}\|\nabla^2 F(x)\|^2 &= \|\mathbb{E}_{z\sim p(z;x)}[\tilde{\nabla}^2 F(x;z)]\|^2 \\ &\leq \mathbb{E}_{z\sim p(z;x)}[\|\tilde{\nabla}^2 F(x;z)\|^2] \\ &\leq \bar{L}^2.\end{aligned}$$

In other words, the conditions in Assumptions 2-4 implicitly imply that the objective function F is $\bar{L}$-smooth.

To establish the convergence guarantees for our proposed `1-SFW` algorithm, the key step is to derive an upper bound on the errors of the estimated gradients. To do so, we prove the following lemma, which provides the required upper bounds in different settings of parameters.

Lemma 2. *Consider the gradient approximation d_t defined in* (3.4)*. Under Assumptions 1-4, if we run Algorithm 1 with Exact Hessian Option in Line 9, and with parameters* $\rho_t = (t-1)^{-\alpha}$ (for all $t \geq 2$)*, and* $\eta_t \leq t^{-\alpha}$ (for all $t \geq 1$ *and for some* $\alpha \in (0,1]$)*, then the gradient estimation d_t satisfies*

$$\mathbb{E}[\|\nabla F(x_t) - d_t\|^2] \leq Ct^{-\alpha}, \tag{3.11}$$

where $C = \max\left\{\frac{2(2G+D\bar{L})^2}{2-2^{-\alpha}-\alpha}, \left[\frac{2}{2-2^{-\alpha}-\alpha}\right]^4, [2D(\bar{L}+L)]^4\right\}$.

Lemma (2) shows that with an appropriate parameter setting, the gradient error converges to zero at a rate of $\mathcal{O}(t^{-\alpha})$. With this unifying upper bound, we can obtain the convergence rates of our algorithm for different kinds of objective functions.

If in the update of `1-SFW` we use the Gradient Difference Option in Line 9 of Algorithm 1 to estimate $\tilde{\Delta}_t$, as pointed out above, we need one further assumption on second-order smoothness of the functions $\tilde{F}$ and $\log p$.

Assumption 5. *The stochastic function $\tilde{F}$ is uniformly $L_{2,\tilde{F}}$- second-order smooth:* $\|\nabla^2 \tilde{F}(x;z) - \nabla^2 \tilde{F}(y;z)\| \leq L_{2,\tilde{F}}\|x-y\|,$ for all $x, y \in \mathcal{K}$, for all $z \in \mathcal{Z}$. *The log probability* $\log p(z;x)$ *is uniformly $L_{2,p}$-second-order smooth:* $\|\nabla^2 \log p(z;x) - \nabla^2 \log p(z;y)\| \leq L_{2,p}\|x - y\|,$ for all $x, y \in \mathcal{K}$, for all $z \in \mathcal{Z}$. *We also define* $L_2 = \max\{L_{2,\tilde{F}}, L_{2,p}\}$.

We note that under Assumption 5, the approximation bound in (3.7) holds for both $\tilde{F}$ and $\log p$. So for δ_t sufficiently small, the error introduced by the Hessian approximation can be ignored. Thus similar upper bound for errors of estimated gradients still holds.

Lemma 3. *Consider the gradient approximation d_t defined in* (3.4). *Under Assumptions 1-5, if we run Algorithm 1 with Gradient Difference Option in Line 9, and with parameters* $\rho_t = (t-1)^{-\alpha}, \delta_t = \frac{\sqrt{3}\eta_{t-1}\bar{L}}{DL_2(1+B)}$ (for all $t \geq 2$), *and* $\eta_t \leq t^{-\alpha}$ (for all $t \geq 1$ *and for some* $\alpha \in (0, 1]$)*, then the gradient estimation d_t satisfies*

$$\mathbb{E}[\|\nabla F(x_t) - d_t\|^2] \leq Ct^{-\alpha}, \tag{3.12}$$

where $C = \max\left\{\frac{8(D^2\bar{L}^2 + G^2 + GD\bar{L})}{2-2^{-\alpha}-\alpha}, \left(\frac{2}{2-2^{-\alpha}-\alpha}\right)^4, (4D(\bar{L}+L))^4\right\}$.

Lemma 3 shows that with Gradient Difference Optionin Line 9 of Algorithm 1, the error of estimated gradient has the same order of convergence rate as that with Exact Hessian Option. So in the following three subsections, we will present the theoretical results of our proposed `1-SFW` algorithm with Exact Hessian Option, for convex minimization, non-convex minimization, and monoton DR-submodular maximization, respectively. The results of Gradient Difference Option only differ in constant factors.

3.3.1 Convex Minimization

For convex minimization problems, to obtain an ϵ-suboptimal solution, Algorithm 1 only requires at most $\mathcal{O}(1/\epsilon^2)$ stochastic oracle queries, and $\mathcal{O}(1/\epsilon^2)$ linear optimization oracle calls. Or precisely, we have

Theorem 1 (Convex)**.** *Consider the* `1-SFW` *method outlined in Algorithm 1 with Exact Hessian Option in Line 9. Further, suppose the conditions in Assumptions 1-4 hold, and assume that F is convex on $\mathcal{K}$. If we set the algorithm parameters as $\rho_t = (t-1)^{-1}$ and $\eta_t = t^{-1}$, then the output $x_{T+1} \in \mathcal{K}$ is feasible and satisfies*

$$\mathbb{E}[F(x_{T+1}) - F(x^*)] \leq \frac{2\sqrt{C}D}{\sqrt{T}} + \frac{\bar{L}D^2(1+\ln T)}{2T},$$

where $C = \max\{4(2G + D\bar{L})^2, 256, [2D(\bar{L}+L)]^4\}$, and x^* is a minimizer of F on $\mathcal{K}$.

The result in Theorem 1 shows that the proposed one sample stochastic Frank-Wolfe method, in the convex setting, has an overall complexity of $\mathcal{O}(1/\epsilon^2)$ for finding an ϵ-suboptimal solution. Note that to prove this claim we used the result in Lemma 2 for the case where $\alpha = 1$, *i.e.*, the variance of gradient approximation converges to zero at a rate of $\mathcal{O}(1/t)$. We also highlight that `1-SFW` is *parameter-free*, as the learning rate η_t and the momentum parameter ρ_t do not depend on the parameters of the problem.

3.3.2 Non-Convex Minimization

For non-convex minimization problems, showing that the gradient norm approaches zero, *i.e.*, $\|\nabla F(x_t)\| \to 0$, implies convergence to a stationary point in the *unconstrained* setting. Thus, it is usually used as a measure for convergence. In the constrained setting, however, the norm of gradient is not a proper measure for defining stationarity and we instead use the Frank-Wolfe Gap [4, 9], which is defined by

$$\mathcal{G}(x) = \max_{v \in \mathcal{K}} \langle v - x, -\nabla F(x) \rangle.$$

We note that by definition, $\mathcal{G}(x) \geq 0$, for all $x \in \mathcal{K}$. If some point $x \in \mathcal{K}$ satisfies $\mathcal{G}(x) = 0$, then it is a first-order stationary point.

In the following theorem, we formally prove the number of iterations required for one sample stochastic FW to find an ϵ-first-order stationary point in expectation, *i.e.*, a point x that satisfies $\mathbb{E}[\mathcal{G}(x)] \leq \epsilon$.

Theorem 2 (Non-Convex). *Consider the* `1-SFW` *method outlined in Algorithm 1 with Exact Hessian Option in Line 9. Further, suppose the conditions in Assumptions 1-4 hold. If we set the algorithm parameters as $\rho_t = (t-1)^{-2/3}$, and $\eta_t = T^{-2/3}$, then the output $x_o \in \mathcal{K}$ is feasible and satisfies*

$$\mathbb{E}[\mathcal{G}(x_o)] \leq \frac{2B + 3\sqrt{C}D/2}{T^{1/3}} + \frac{\bar{L}D^2}{2T^{2/3}},$$

where $C = \max\left\{\frac{2(2G+D\bar{L})^2}{\frac{4}{3}-2^{-\frac{2}{3}}}, \left[\frac{2}{\frac{4}{3}-2^{-\frac{2}{3}}}\right]^4, [2D(\bar{L}+L)]^4\right\}.$

We remark that Theorem (2) shows that Algorithm 1 finds an ϵ-first-order stationary points after at most $\mathcal{O}(1/\epsilon^3)$ iterations, while uses exactly one stochastic gradient per iteration. Note that to obtain the best performance guarantee in Theorem (2), we used the result of Lemma 2 for the case where $\alpha = 2/3$, *i.e.*, the variance of gradient approximation converges to zero at a rate of $\mathcal{O}(T^{-2/3})$. Again, we highlight that `1-SFW` is a *parameter-free* algorithm.

3.3.3 Monotone DR-Submodular Maximization

In this subsection, we focus on the convergence properties of one-sample stochastic Frank-Wolfe or one-sample stochastic Continuous Greedy for solving a monotone continuous DR-submodular maximization problem.

Recall that for monotone continuous DR-submodular maximization, approximated solution within a factor of $(1 - e^{-1} + \epsilon)$ can not be obtained in polynomial time [57]. To achieve a $(1 - e^{-1})\mathrm{OPT} - \epsilon$ approximation guarantee, `1-SFW` requires at most $\mathcal{O}(1/\epsilon^2)$ stochastic oracle queries, and $\mathcal{O}(1/\epsilon^2)$ linear optimization oracle calls, which are the lower bounds of the complexity established in [13].

Theorem 3 (Submodular). *Consider the* `1-SFW` *method outlined in Algorithm 1 with Exact Hessian Option in Line 9 for maximizing DR-Submodular functions. Further, suppose the conditions in Assumptions 1-4 hold, and further assume that F is monotone and continuous DR-submodular on the positive orthant. If we set the algorithm parameters as $x_1 = 0, \rho_t = (t-1)^{-1}, \eta_t = T^{-1}$, then the output $x_{T+1} \in \mathcal{K}$ is feasible and satisfies*

$$\mathbb{E}[F(x_{T+1})] \geq (1 - e^{-1})F(x^*) - \frac{4R\sqrt{C}}{T^{1/2}} - \frac{\bar{L}R^2}{2T},$$

where $C = \max\{4(2G + R\bar{L})^2, 256, [2R(\bar{L} + L)]^4\}$.

Finally, we note that Algorithm 1 can also be used to solve stochastic discrete submodular maximization [101, 115]. Precisely, we can apply Algorithm 1 on the multilinear extension of the discrete submodular functions, and round the output to a feasible set by lossless rounding schemes like pipage rounding [116] and contention resolution method [117].

3.4 Oblivious Setting

In this section, we specifically study the oblivious problem introduced in (3.2) which is a special case of the non-oblivious problem defined in (3.1). In particular, we show that our proposed `1-SFW` method becomes significantly simpler and the corresponding theoretical results hold under less strict assumptions.

3.4.1 Algorithm

As we discussed in Section 3.2, a major challenge that we face for designing a variance reduced Frank-Wolfe method for the non-oblivious setting is computing an unbiased estimator of the gradient variation $\Delta_t = \nabla F(x_t) - \nabla F(x_{t-1})$. This is indeed not problematic in the oblivious setting, as in this case $z \sim p(z)$ is independent of x and

Algorithm 2 One-Sample SFW (Oblivious Setting)

Input: Step sizes $\rho_t \in (0,1), \eta_t \in (0,1)$, initial point $x_1 \in \mathcal{K}$, total number of iterations T
Output: x_{T+1} or x_o, where x_o is chosen from $\{x_1, x_2, \cdots, x_T\}$ uniformly at random

```
for t = 1, 2, ..., T do
    Sample a point z_t according to p(z)
    if t = 1 then
        Compute d_1 = ∇F̃(x_1; z_1)
    else
        Δ̃_t = ∇F̃(x_t; z_t) − ∇F̃(x_{t−1}; z_t)
        d_t = (1 − ρ_t)(d_{t−1} + Δ̃_t) + ρ_t ∇F̃(x_t, z_t)
    end if
    (non-)convex minimization: Update x_{t+1} based on (3.9)
    DR-submodular maximization: Update x_{t+1} based on (3.10)
end for
```

therefore $\nabla\tilde{F}(x_t; z) - \nabla\tilde{F}(x_{t-1}; z)$ is an unbiased estimator of the gradient variation $\Delta_t = \nabla F(x_t) - \nabla F(x_{t-1})$. Hence, in the oblivious setting, our proposed one sample FW uses the following gradient approximation

$$d_t = (1-\rho_t)(d_{t-1} + \tilde{\Delta}_t) + \rho_t \nabla\tilde{F}(x_t; z_t),$$

where $\tilde{\Delta}_t$ is given by

$$\tilde{\Delta}_t = \nabla\tilde{F}(x_t; z_t) - \nabla\tilde{F}(x_{t-1}; z_t).$$

The rest of the algorithm for updating the variable x_t is identical to the one for the non-oblivious setting. The description of our proposed algorithm for the oblivious setting is outlined in Algorithm 2.

Remark 1. *We note that by rewriting our proposed* `1-SFW` *method for the oblivious setting, we recover the variance reduction technique STORM [113] with different sets of parameters. In [113], however, the STORM algorithm was combined with SGD to solve* unconstrained *non-convex minimization problems, while our proposed* `1-SFW` *method solves convex minimization, non-convex minimization, and DR-submodular maximization in a* constrained *setting.*

3.4.2 Theoretical Results

In this subsection, we show that the variant of one sample stochastic FW for the oblivious setting (described in Algorithm 2) recovers the theoretical results for the non-oblivious setting with fewer assumptions. In particular, we only require the following condition for the stochastic functions $\tilde{F}$ to prove our main results.

Assumption 6. *The function $\tilde{F}$ has uniformly bound gradients,* i.e., for all $x \in \mathcal{K}$, for all $z \in \mathcal{Z}$,

$$\|\nabla \tilde{F}(x; z)\| \leq G.$$

Moreover, the function $\tilde{F}$ is uniformly L-smooth, i.e., for all $x, y \in \mathcal{K}$, for all $z \in \mathcal{Z}$,

$$\|\nabla \tilde{F}(x; z) - \nabla \tilde{F}(y; z)\| \leq L\|x - y\|.$$

We note that as direct corollaries of Theorems 1 to 3, Algorithm 2 achieves the same convergence rates, which is stated in Theorem 4 formally.

Theorem 4. *Consider the oblivious variant of `1-SFW` outlined in Algorithm 2, and assume that the conditions in Assumptions 1, 2 and 6 hold. Then we have*

1. *If F is convex on $\mathcal{K}$, and we set $\rho_t = (t-1)^{-1}$ and $\eta_t = t^{-1}$, then the output $x_{T+1} \in \mathcal{K}$ is feasible and satisfies*

$$\mathbb{E}[F(x_{T+1}) - F(x^*)] \leq \mathcal{O}(T^{-1/2}).$$

2. *If F is non-convex, and we set $\rho_t = (t-1)^{-2/3}$, and $\eta_t = T^{-2/3}$, then the output $x_o \in \mathcal{K}$ is feasible and satisfies*

$$\mathbb{E}[\mathcal{G}(x_o)] \leq \mathcal{O}(T^{-1/3}).$$

3. *If F is monotone DR-submodular on $\mathcal{K}$, and we set $x_1 = 0, \rho_t = (t-1)^{-1}$ and $\eta_t = T^{-1}$, then the output $x_{T+1} \in \mathcal{K}$ is feasible and satisfies*

$$\mathbb{E}[F(x_{T+1})] \geq (1 - e^{-1})F(x^*) - \mathcal{O}(T^{-1/2}).$$

Theorem 4 shows that the oblivious version of `1-SFW` requires at most $\mathcal{O}(1/\epsilon^2)$ stochastic oracle queries to find an ϵ-suboptimal solution for convex minimization, at most $\mathcal{O}(1/\epsilon^2)$ stochastic gradient evaluations to achieve a $(1-1/e)-\epsilon$ approximate solution for monotone DR-submodular maximization, and at most $\mathcal{O}(1/\epsilon^3)$ stochastic oracle queries to find an ϵ-first-order stationary point for non-convex minimization.

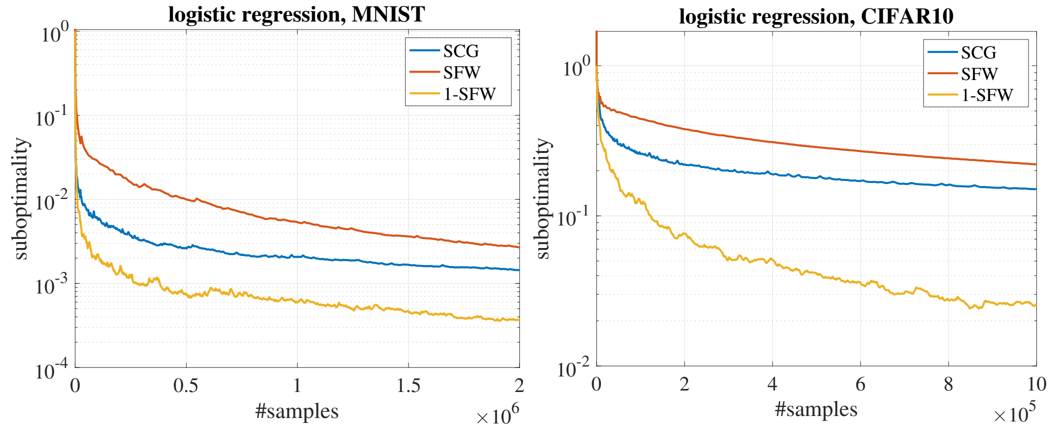
3.5 Experiments

In this section, we empirically validate the efficiency of the proposed `1-SFW` algorithm by comparing it with the baseline methods: Stochastic Frank-Wolfe (SFW) [7] and Stochastic Conditional Gradient (SCG) [8]. Note that SCG is the only existing provably convergent Frank-Wolfe variant that accepts a constant per-iteration mini-batch size (possibly 1). Denote the constant mini-batch size of `1-SFW` and SCG by m. The growing mini-batch size of SFW is set to $m \cdot t^2$, where t is the iteration count.

We study three types of problems, *i.e.*, ℓ_1-constrained logistic-regression (convex), robust low rank matrix recovery (non-convex), and maximization of multilinear extensions of monotone discrete submodular functions (DR-submodular).

3.5.1 Logistic Regression

In this task, we consider ℓ_1-constrained logistic regression problem. Concretely, denote each data point i by (a_i, y_i), where $a_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{1, \ldots, C\}$ is the



(a) MNIST (b) CIFAR10

Figure 3.1: Logistic Regression. (a) uses digit 2 and 4 in MNIST, (b) uses cat and dog in CIFAR10.

corresponding label. Our goal is to minimize the following loss

$$F(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i \mathbf{W}_c^T a_i)),$$

over the constraint $\mathcal{C} = \{\mathbf{W} \in \mathbb{R}^{d\times C} : \|\mathbf{W}\|_1 \leq r\}$ for some constant $r \in \mathbb{R}_+$, where $\|\mathbf{W}\|_1$ is the matrix ℓ_1 norm, *i.e.*, $\|\mathbf{W}\|_1 = \max_{1\leq j\leq C} \sum_{i=1}^{d} |[\mathbf{W}]_{ij}|$. We note that the loss function F is convex and smooth.

Two datasets are used in our experiments: MNIST (digit 2 and 4 as positive and negative class respectively) and CIFART10 (cat and dog as positive and negative class respectively). In terms of the parameter setting, we grid search the step size η_t for all three methods over the set $\{\min\{1, c/(t+1)^a\} | c \in \{0.1, 0.25, 0.5, 1.0, 2.0\}, a \in \{1, 2/3, 1/2\}\}$, set the mixing weights ρ_t of SCG and `1-SFW` to $1/(t+1)^{2/3}$, and set the constant mini-batch parameter $m = 16$. We report the results in Figure 3.1. We can see the advantage of `1-SFW` over its competitors.

3.5.2 Robust Low-Rank Matrix Recovery

LRMR plays a key role in solving many important learning tasks, such as collaborative filtering [118], dimensionality reduction [119], and multi-class learning [120]. The loss of LRMR is defined as

$$\min_{\mathbf{X}\in\mathbb{R}^{M\times N}} \quad \sum_{(i,j)\in\Omega} \psi(\mathbf{X}_{ij}-\mathbf{Y}_{ij}),$$
$$\text{subject to} \qquad \|\mathbf{X}\|_*\leq B,$$

where $\psi:\mathbb{R}\to\mathbb{R}$ is the potentially non-convex empirical loss function, $\mathbf{X}_{ij}$ is the (i,j)-th element of matrix $\mathbf{X}$, and Ω is the set of observed indices in target matrix $\mathbf{Y}\in\mathbb{R}^{M\times N}$. Here we focus on a robust version of LRMR with the loss ψ being:

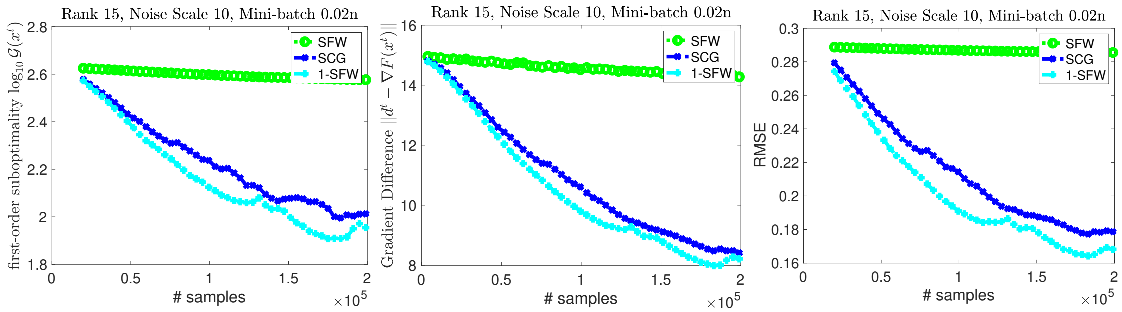
$$\psi(z;\sigma) = 1-\exp(-z^2/2\sigma), \tag{3.13}$$

where σ is a tunable parameter. Loss (3.13) is less sensitive to the discrepancy $\mathbf{X}_{ij}-\mathbf{Y}_{ij}$ compared to the common least square loss $\psi(z)=z^2/2$, and hence is robust to adversarial outliers [121].

In each trial, we first generate an underlying matrix $\mathbf{M}$ of size 200×200 and rank $\gamma=15$. The singular values of $\mathbf{M}$ are set as $2^{[\gamma]}/2^{\gamma}\times 50$ and hence $\|\mathbf{M}\|_*\leq C=100$, where $[\gamma]=\{1,\ldots,\gamma\}$. We then inject adversarial noise into $\mathbf{M}$ by (1) uniformly sampling 5% of the entries in $\mathbf{M}$ and (2) adding random noise uniformly sampled from $[-\rho,\rho]$ to each selected entry, where the noise level ρ equals 10. Denote $\hat{\mathbf{M}}$ as the matrix after noise injection. We uniformly sample 10% of the entries in $\hat{\mathbf{M}}$ to obtain the observations, *i.e.*, $\mathbf{Y}_{ij}$. Hence $|\Omega|$, the number of observation is $M\times N\times 10\%=4,000$.

In terms of algorithmic parameter setting, we set the mini-batch size m to $|\Omega|/20$. The number of epoch T is set to 50 for all cases, and the step size parameter η_t is set to $1/(T*|\Omega|/m)=1/1000$ in all cases for all methods.

We present the comparison of listed methods in Figure 3.2, where we observe that



(a) Frank-Wolfe gap (b) Gradient estimation acc. (c) Root Mean Square Error

Figure 3.2: Matrix Recovery. (a) compares the Frank-Wolfe gap, (b) compares the accuracy of gradient estimation, (c) compares the Root Mean Square Error (RMSE) between the prediction matrix and the underlying true matrix.

`1-SFW` has the best performance in terms of the Frank-Wolfe gap (Fig. 3.2a), gradient estimation accuracy (Fig. 3.2b), and the Root Mean Square Error (RMSE) between the prediction matrix and the underlying true matrix (Fig. 3.2c).

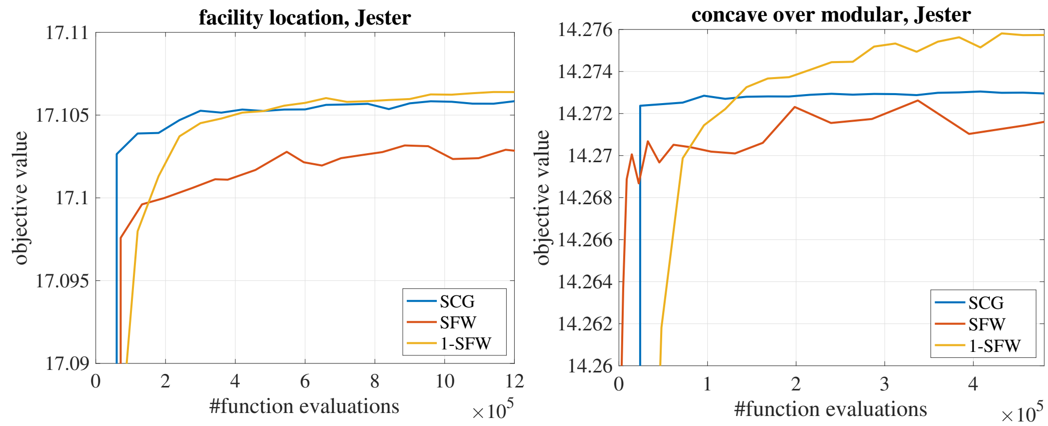
3.5.3 Discrete Monotone Submodular Maximization with Matroid Constraint

In this subsection, we consider the discrete monotone submodular maximization subject to a matroid constraint via the maximizing the corresponding multilinear extension. Let V be a finite set of d elements and $\mathcal{I}$ be a collection of its subsets. It is proved that to maximize a discrete monotone submodular function $f : 2^V \to \mathbb{R}_+$ subject to the matroid constraint $\mathcal{M} \stackrel{\text{def}}{=} \{V, \mathcal{I}\}$ is equivalent to maximize its multilinear extension, defined as

$$F(x) = \sum_{S \subset [d]} f(S) \prod_{j \in S} [x]_j \prod_{\ell \notin S} (1 - [x]_\ell), \tag{3.14}$$

subject to the constraint $x \in \mathcal{C}$, where $\mathcal{C}$ is the base polytope of $\mathcal{M}$. Further, it is known that F is monotone DR-submodular.

We now focus on a concrete recommendation problem which can be formulated as discrete monotone submodular maximization. We use $r(u, j)$ to denote user u's rating for



(a) Facility location utility

(b) Concave over modular utility

Figure 3.3: Submodular Maximization on Jester dataset. (a) uses the facility location utility and (b) uses the concave over modular utility.

item $j \in [d]$ and set $r(u, j) = 0$ if item j is not rated by user u. Our goal is to recommend a set of $k = 10$ items to all users such that they have the highest total rating. Two types of utility functions can be defined for such task: facility location

$$f(S) = \sum_u \max_{j \in S} r(u, j), \tag{3.15}$$

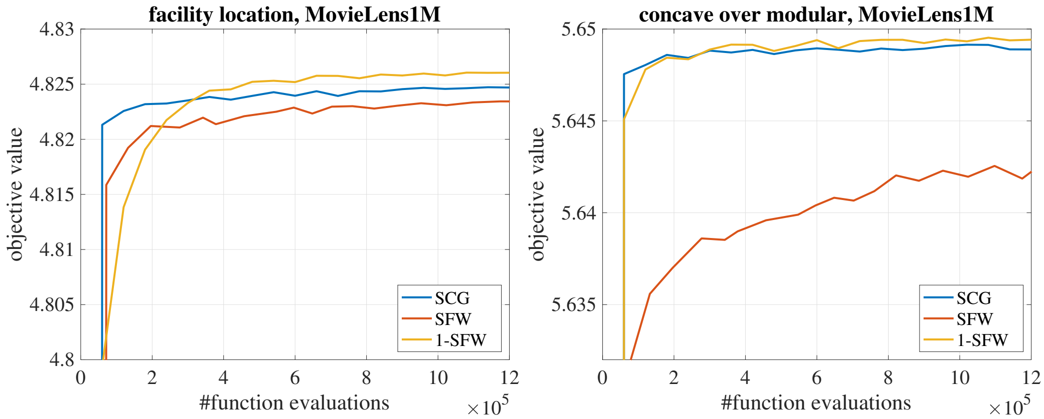
or concave over modular

$$f(S) = \sum_u (\sum_{j \in S} r(u, j))^{1/2}. \tag{3.16}$$

Here the matroid is $\{V, \mathcal{I} \overset{\text{def}}{=} \{S \subseteq V || S |= k\}\}$. Two datasets are used in this experiment, Jester 1[2] and movielens 1M[3] with the results presented in Figure 3.3 and Figure 3.4 respectively. We observe that `1-SFW` always achieves the highest utility after sufficient function evaluations.

[2]http://eigentaste.berkeley.edu/dataset/

[3]https://grouplens.org/datasets/movielens/



(a) Facility location utility

(b) Concave over modular utility

Figure 3.4: Submodular Maximization on Movielens dataset. (a) uses the facility location utility and (b) uses the concave over modular utility.

3.6 Conclusion

In this chapter, we studied the problem of solving constrained stochastic optimization programs using projection-free methods. We proposed the first stochastic variant of the Frank-Wolfe method, called `1-SFW`, that requires only one stochastic sample per iteration while achieving the best known complexity bounds for (non-)convex minimization and monotone DR-submodular maximization. In particular, we proved that `1-SFW` achieves the best known oracle complexity of $\mathcal{O}(1/\epsilon^2)$ for reaching an ϵ-suboptimal solution in the stochastic convex setting, and a $(1-1/e)\text{OPT}-\epsilon$ approximate solution for a stochastic monotone DR-submodular maximization problem. Moreover, in a non-convex setting, `1-SFW` finds an ϵ-first-order stationary point after at most $\mathcal{O}(1/\epsilon^3)$ iterations, achieving the best known overall complexity.

3.7 Proofs

3.7.1 Proof of Lemma 2

Proof. Let $A_t = \|\nabla F(x_t) - d_t\|^2$. By definition, we have

$$A_t = \|\nabla F(x_{t-1}) - d_{t-1} + \nabla F(x_t) - \nabla F(x_{t-1}) - (d_t - d_{t-1})\|^2.$$

Note that

$$d_t - d_{t-1} = -\rho_t d_{t-1} + \rho_t \nabla \tilde{F}(x_t, z_t) + (1-\rho_t)\tilde{\Delta}_t,$$

and define $\Delta_t = \nabla F(x_t) - \nabla F(x_{t-1})$, we have

$$\begin{aligned}
A_t &= \|\nabla F(x_{t-1}) - d_{t-1} + \Delta_t - (1-\rho_t)\tilde{\Delta}_t - \rho_t \nabla \tilde{F}(x_t, z_t) + \rho_t d_{t-1}\|^2 \\
&= \|\nabla F(x_{t-1}) - d_{t-1} + (1-\rho_t)(\Delta_t - \tilde{\Delta}_t) + \rho_t(\nabla F(x_t) - \nabla \tilde{F}(x_t, z_t) + \rho_t(d_{t-1} - \nabla F(x_{t-1})))\|^2 \\
&= \|(1-\rho_t)(\nabla F(x_{t-1}) - d_{t-1}) + (1-\rho_t)(\Delta_t - \tilde{\Delta}_t) + \rho_t(\nabla F(x_t) - \nabla \tilde{F}(x_t, z_t))\|^2.
\end{aligned}$$

Since $\tilde{\Delta}_t$ is an unbiased estimator of Δ_t, $\mathbb{E}[A_t]$ can be decomposed as

$$\begin{aligned}
\mathbb{E}[A_t] = \mathbb{E}\{&(1-\rho_t)^2\|\nabla F(x_{t-1}) - d_{t-1}\|^2 + (1-\rho_t)^2\|\Delta_t - \tilde{\Delta}_t\|^2 \\
&+ \rho_t^2\|\nabla F(x_t) - \nabla \tilde{F}(x_t, z_t)\|^2 \\
&+ 2\rho_t(1-\rho_t)\langle \nabla F(x_{t-1}) - d_{t-1}, \nabla F(x_t) - \nabla \tilde{F}(x_t, z_t)\rangle \\
&+ 2\rho_t(1-\rho_t)\langle \Delta_t - \tilde{\Delta}_t, \nabla F(x_t) - \nabla \tilde{F}(x_t, z_t)\rangle\}.
\end{aligned} \tag{3.17}$$

Then we turn to upper bound the items above. First, by Lemma 1, we have

$$\begin{aligned}\mathbb{E}[\|\tilde{\Delta}_t - \Delta_t\|^2] &= \mathbb{E}[\|\tilde{\nabla}_t^2(x_t - x_{t-1}) - (\nabla F(x_t) - \nabla F(x_{t-1}))]\|^2] \\ &\le \mathbb{E}[\|\tilde{\nabla}_t^2(x_t - x_{t-1})\|^2] \\ &= \mathbb{E}[\|\tilde{\nabla}_t^2(\eta_{t-1}(v_{t-1} - x_{t-1}))\|^2] \\ &\le \eta_{t-1}^2 D^2 \mathbb{E}[\|\tilde{\nabla}_t^2\|^2] \\ &\le \eta_{t-1}^2 D^2 \bar{L}^2.\end{aligned} \tag{3.18}$$

By Jensen's inequality, we have

$$\mathbb{E}[\|\tilde{\Delta}_t - \Delta_t\|] \le \sqrt{\mathbb{E}[\|\tilde{\Delta}_t - \Delta_t\|^2]} \le \eta_{t-1} D \bar{L},$$

and

$$\mathbb{E}[\|\nabla F(x_t) - d_t\|] = \sqrt{\mathbb{E}[\|\nabla F(x_t) - d_t\|^2]} = \sqrt{\mathbb{E}[A_t]}.$$

Note that z_t is sampled according to $p(z; x_t(a))$, where $x_t(a) = ax_t + (1-a)x_{t-1}$. Thus $\nabla \tilde{F}(x_t, z_t)$ is NOT an unbiased estimator of $\nabla F(x_t)$ when $a \neq 1$, which occurs with probability 1. However, we will show that $\nabla \tilde{F}(x_t, z_t)$ is still a good estimator. Let $\mathcal{F}_{t-1}$ be the σ-field generated by all the randomness before round t, then by Law of Total Expectation, we have

$$\begin{aligned}&\mathbb{E}[2\rho_t(1-\rho_t)\langle \nabla F(x_{t-1}) - d_{t-1}, \nabla F(x_t) - \nabla \tilde{F}(x_t, z_t)\rangle] \\ =&\mathbb{E}[\mathbb{E}[2\rho_t(1-\rho_t)\langle \nabla F(x_{t-1}) - d_{t-1}, \nabla F(x_t) - \nabla \tilde{F}(x_t, z_t)\rangle | \mathcal{F}_{t-1}, x_t(a)]] \\ =&\mathbb{E}[2\rho_t(1-\rho_t)\langle \nabla F(x_{t-1}) - d_{t-1}, \mathbb{E}[\nabla F(x_t) - \nabla \tilde{F}(x_t, z_t) | \mathcal{F}_{t-1}, x_t(a)]\rangle],\end{aligned} \tag{3.19}$$

where

$$\mathbb{E}[\nabla F(x_t) - \nabla \tilde{F}(x_t, z_t) | \mathcal{F}_{t-1}]\rangle] = \nabla F(x_t) - \nabla F(x_t(a)) + \nabla F(x_t(a)) - \mathbb{E}[\nabla \tilde{F}(x_t, z_t) | \mathcal{F}_{t-1}, x_t(a)].$$

By Lemma 1, F is $\bar{L}$-smooth, thus

$$\|\nabla F(x_t) - \nabla F(x_t(a))\| \leq \bar{L}\|x_t - x_t(a)\| = \bar{L}(1-a)\|\eta_{t-1}(v_{t-1} - x_{t-1})\| \leq \eta_{t-1} D\bar{L}.$$

We also have

$$\begin{aligned}
&\|\nabla F(x_t(a)) - \mathbb{E}[\nabla \tilde{F}(x_t, z_t)|\mathcal{F}_{t-1}, x_t(a)]\| \\
=& \|\int [\nabla \tilde{F}(x_t(a); z) - \nabla \tilde{F}(x_t; z)] p(z; x_t(a)) \mathrm{d}z\| \\
\leq& \int \|\nabla \tilde{F}(x_t(a); z) - \nabla \tilde{F}(x_t; z)\| p(z; x_t(a)) \mathrm{d}z \\
\leq& \int L\|x_t(a) - x_t\| p(z; x_t(a)) \mathrm{d}z \\
\leq& \eta_{t-1} DL,
\end{aligned}$$

where the second inequality holds because of Assumption 4. Combine the analysis above with Eq. (3.19), we have

$$\begin{aligned}
&\mathbb{E}[2\rho_t(1-\rho_t)\langle \nabla F(x_{t-1}) - d_{t-1}, \nabla F(x_t) - \nabla \tilde{F}(x_t, z_t)\rangle] \\
\leq& \mathbb{E}[2\rho_t(1-\rho_t)\|\nabla F(x_{t-1}) - d_{t-1}\| \cdot \|\mathbb{E}[\nabla F(x_t) - \nabla \tilde{F}(x_t, z_t)|\mathcal{F}_{t-1}]\|] \\
\leq& 2\rho_t(1-\rho_t)\mathbb{E}[\|\nabla F(x_{t-1}) - d_{t-1}\|] \cdot (\eta_{t-1} D\bar{L} + \eta_{t-1} DL) \\
\leq& 2\eta_{t-1}\rho_t(1-\rho_t)\sqrt{\mathbb{E}[A_{t-1}]} D(\bar{L} + L).
\end{aligned} \tag{3.20}$$

Finally, by Assumption 3, we have $\|\nabla F(x_t) - \nabla \tilde{F}(x_t, z_t)\| \leq 2G$. Thus

$$\rho_t^2 \|\nabla F(x_t) - \nabla \tilde{F}(x_t, z_t)\|^2 \leq 4\rho_t^2 G^2, \tag{3.21}$$

and

$$\begin{aligned}
&\mathbb{E}[2\rho_t(1-\rho_t)\langle \Delta_t - \tilde{\Delta}_t, \nabla F(x_t) - \nabla \tilde{F}(x_t, z_t)\rangle] \\
&\leq \mathbb{E}[2\rho_t(1-\rho_t)\|\Delta_t - \tilde{\Delta}_t\| \cdot \|\nabla F(x_t) - \nabla \tilde{F}(x_t, z_t)\|] \\
&\leq 4\eta_{t-1}\rho_t(1-\rho_t)GD\bar{L}.
\end{aligned} \tag{3.22}$$

Combine Eqs. (3.17), (3.18) and (3.20) to (3.22), we have

$$\begin{aligned}
\mathbb{E}[A_t] \leq{}& (1-\rho_t)^2\mathbb{E}[A_{t-1}] + (1-\rho_t)^2\eta_{t-1}^2 D^2\bar{L}^2 + \rho_t^2 4G^2 \\
&+ 2\eta_{t-1}\rho_t(1-\rho_t)\sqrt{\mathbb{E}[A_{t-1}]}D(\bar{L}+L) + 4\eta_{t-1}\rho_t(1-\rho_t)GD\bar{L}.
\end{aligned}$$

For the simplicity of analysis, we replace t by $t+1$, and have

$$\begin{aligned}
\mathbb{E}[A_{t+1}] \leq{}& (1-\rho_{t+1})^2\mathbb{E}[A_t] + (1-\rho_{t+1})^2\eta_t^2 D^2\bar{L}^2 + \rho_{t+1}^2 4G^2 \\
&+ 2\eta_t\rho_{t+1}(1-\rho_{t+1})\sqrt{\mathbb{E}[A_t]}D(\bar{L}+L) + 4\eta_t\rho_{t+1}(1-\rho_{t+1})GD\bar{L} \\
\leq{}& (1-\frac{1}{t^\alpha})^2\mathbb{E}[A_t] + \frac{D^2\bar{L}^2 + 4G^2 + 4GD\bar{L}}{t^{2\alpha}} + \frac{2D(\bar{L}+L)}{t^{2\alpha}}\sqrt{\mathbb{E}[A_t]}.
\end{aligned} \tag{3.23}$$

We claim that $\mathbb{E}[A_t] \leq Ct^{-\alpha}$, and prove it by induction. Before the proof, we first analyze one item in the definition of $C : \frac{2(2G+D\bar{L})^2}{2-2^{-\alpha}-\alpha}$. Define $h(\alpha) = 2 - 2^{-\alpha} - \alpha$. Since $h'(\alpha) = 2^{-\alpha}\ln(2) - 1 \leq 0$ for $\alpha \in (0,1]$, so $1 = h(0) \geq h(\alpha) \geq h(1) = 1/2 > 0$, for all $\alpha \in (0,1]$. As a result, $2 \leq \frac{2}{2-2^{-\alpha}-\alpha} \leq 4$.

When $t = 1$, we have

$$\mathbb{E}[A_1] = \mathbb{E}[\|\nabla F(x_1) - \nabla \tilde{F}(x_1; z_1)\|^2] \leq (2G)^2 \leq \frac{2(2G+D\bar{L})^2}{2-2^{-\alpha}-\alpha}/1 \leq C \cdot 1^{-\alpha}.$$

When $t = 2$, since $\rho_2 = 1$, we have

$$\mathbb{E}[A_2] = \mathbb{E}[\|\nabla \tilde{F}(x_2, z_2) - \nabla F(x_2)\|^2] \leq (2G)^2 \leq \frac{2(2G+D\bar{L})^2}{2-2^{-\alpha}-\alpha}/2 \leq C \cdot 2^{-\alpha}.$$

Now assume for $t \geq 2$, we have $\mathbb{E}[A_t] \leq Ct^{-\alpha}$, by Eq. (3.23) and the definition of C, we have

$$\begin{aligned}
\mathbb{E}[A_{t+1}] &\leq (1-\frac{1}{t^\alpha})^2 \cdot Ct^{-\alpha} + \frac{(2G+D\bar{L})^2}{t^{2\alpha}} + \frac{2D(\bar{L}+L)}{t^{(5/2)\alpha}}\sqrt{C} \\
&\leq Ct^{-\alpha} - 2Ct^{-2\alpha} + Ct^{-3\alpha} + \frac{(2-2^{-\alpha}-\alpha)C}{2t^{2\alpha}} + \frac{C^{3/4}}{t^{(5/2)\alpha}} \\
&\leq \frac{C}{t^\alpha} + \frac{-2C + Ct^{-\alpha} + (2-2^{-\alpha}-\alpha)C/2 + t^{-\alpha/2}C/C^{1/4}}{t^{2\alpha}} \\
&\leq \frac{C}{t^\alpha} + \frac{C[-2+2^{-\alpha}+(2-2^{-\alpha}-\alpha)/2+(2-2^{-\alpha}-\alpha)/2]}{t^{2\alpha}} \\
&\leq \frac{C}{t^\alpha} - \frac{\alpha C}{t^{2\alpha}}.
\end{aligned} \tag{3.24}$$

Define $g(t) = t^{-\alpha}$, then $g(t)$ is a convex function for $\alpha \in (0,1]$. Thus we have $g(t+1) - g(t) \geq g'(t)$, *i.e.*, $(t+1)^{-\alpha} - t^{-\alpha} \geq -\alpha t^{-(\alpha+1)}$. So we have

$$\frac{C}{t^\alpha} - \frac{\alpha C}{t^{2\alpha}} \leq C(t^{-\alpha} - \alpha t^{-(1+\alpha)}) \leq C(t+1)^{-\alpha}.$$

Combine with Eq. (3.24), we have $\mathbb{E}[A_{t+1}] \leq C(t+1)^{-\alpha}$. Thus by induction, we have $\mathbb{E}[A_t] \leq Ct^{-\alpha}$, for all $t \geq 1$. □

3.7.2 Proof of Lemma 3

The only difference with the proof of Lemma 2 is the bound for $\mathbb{E}\|\tilde{\Delta}_t - \Delta_t\|$. Specifically, we have

$$\begin{aligned}
\mathbb{E}[\|\tilde{\Delta}_t - \Delta_t\|^2] &= \mathbb{E}[\|\tilde{\Delta}_t - \tilde{\nabla}_t^2(x_t - x_{t-1}) + \tilde{\nabla}_t^2(x_t - x_{t-1}) - (\nabla F(x_t) - \nabla F(x_{t-1}))]\|^2] \\
&= \mathbb{E}[\|\tilde{\Delta}_t - \tilde{\nabla}_t^2(x_t - x_{t-1})\|^2] + \mathbb{E}[\|\tilde{\nabla}_t^2(x_t - x_{t-1}) - (\nabla F(x_t) - \nabla F(x_{t-1}))\|^2] \\
&\leq [D^2 L_2 \delta_t (1 + \tilde{F}(x_t(a), z_t))]^2 + \eta_{t-1}^2 D^2 \bar{L}^2 \\
&\leq (1+B)^2 L_2^2 D^4 \delta_t^2 + \eta_{t-1}^2 D^2 \bar{L}^2 \\
&\leq 4\eta_{t-1}^2 D^2 \bar{L}^2.
\end{aligned}$$

Then by the analysis same to the proof of Lemma 2, we have

$$\mathbb{E}[A_{t+1}] \leq (1-\frac{1}{t^\alpha})^2\mathbb{E}[A_t] + \frac{4(D^2\bar{L}^2+G^2+GD\bar{L})}{t^{2\alpha}} + \frac{4D(\bar{L}+L)}{t^{2\alpha}}\sqrt{\mathbb{E}[A_t]},$$

and thus $\mathbb{E}[A_{t+1}] \leq C(t+1)^{-\alpha}$, where $C = \max\left\{\frac{8(D^2\bar{L}^2+G^2+GD\bar{L})}{2-2^{-\alpha}-\alpha}, \left[\frac{2}{2-2^{-\alpha}-\alpha}\right]^4, [4D(\bar{L}+L)]^4\right\}$.

3.7.3 Proof of Theorem 1

First, since $x_{t+1} = (1-\eta_t)x_t + \eta_t v_t$ is a convex combination of x_t, v_t, and $x_1 \in \mathcal{K}, v_t \in \mathcal{K}$, for all t, we can prove $x_t \in \mathcal{K}$, for all t by induction. So $x_{T+1} \in \mathcal{K}$.

Then we present an auxiliary lemma.

Lemma 4 (Proof of Theorem 1 in [19]). *Under the condition of Theorem 1, in Algorithm 1, we have*

$$F(x_{t+1}) - F(x^*) \leq (1-\eta_t)(F(x_t) - F(x^*)) + \eta_t D\|\nabla F(x_t) - d_t\| + \frac{\bar{L}D^2\eta_t^2}{2}.$$

By Jensen's inequality and Lemma 2 with $\alpha = 1$, we have

$$\mathbb{E}[\|\nabla F(x_t) - d_t\|] \leq \sqrt{\mathbb{E}[\|\nabla F(x_t) - d_t\|^2]} \leq \frac{\sqrt{C}}{\sqrt{t}},$$

where $C=\max\{4(2G+D\bar{L})^2, 256, [2D(\bar{L}+L)]^4\}$. Then by Lemma 4, we have

$$
\begin{aligned}
&\mathbb{E}[F(x_{T+1})-F(x^*)] \\
&\leq (1-\eta_T)\mathbb{E}[F(x_T)-F(x^*)]+\eta_T D\mathbb{E}[\|\nabla F(x_T)-d_T\|]+\frac{\bar{L}D^2\eta_T^2}{2} \\
&= \prod_{i=1}^{T}(1-\eta_i)\mathbb{E}[F(x_1)-F(x^*)]+D\sum_{k=1}^{T}\eta_k\mathbb{E}[\|\nabla F(x_k)-d_k\|]\prod_{i=k+1}^{T}(1-\eta_i) \\
&\qquad+\frac{\bar{L}D^2}{2}\sum_{k=1}^{T}\eta_k^2\prod_{i=k+1}^{T}(1-\eta_i) \\
&\leq 0+D\sum_{k=1}^{T}k^{-1}\frac{\sqrt{C}}{\sqrt{k}}\prod_{i=k+1}^{T}\frac{i-1}{i}+\frac{\bar{L}D^2}{2}\sum_{k=1}^{T}k^{-2}\prod_{i=k+1}^{T}\frac{i-1}{i} \\
&= \frac{\sqrt{C}D}{T}\sum_{k=1}^{T}\frac{1}{\sqrt{k}}+\frac{\bar{L}D^2}{2T}\sum_{k=1}^{T}k^{-1}.
\end{aligned}
\tag{3.25}
$$

Since

$$\sum_{k=1}^{T}\frac{1}{\sqrt{k}}\leq\int_0^T x^{-1/2}\mathrm{d}x=2\sqrt{T},$$

and

$$\sum_{k=1}^{T}k^{-1}\leq 1+\int_1^T x^{-1}\mathrm{d}x=1+\ln T,$$

by Eq. (3.25), we have

$$\mathbb{E}[F(x_{T+1})-F(x^*)]\leq\frac{2\sqrt{C}D}{\sqrt{T}}+\frac{\bar{L}D^2}{2T}(1+\ln T).$$

3.7.4 Proof of Theorem 2

First, since $x_{t+1}=(1-\eta_t)x_t+\eta_t v_t$ is a convex combination of x_t, v_t, and $x_1\in\mathcal{K}, v_t\in\mathcal{K}$, for all t, we can prove $x_t\in\mathcal{K}$, for all t by induction. So $x_o\in\mathcal{K}$.

Note that if we define $v_t'=\arg\min_{v\in\mathcal{K}}\langle v,\nabla F(x_t)\rangle$, then $\mathcal{G}(x_t)=\langle v_t'-x_t,-\nabla F(x_t)\rangle=-\langle v_t'-x_t,\nabla F(x_t)\rangle$. So we have

$$
\begin{aligned}
F(x_{t+1}) &\overset{(a)}{\leq} F(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{\bar{L}}{2}\|x_{t+1} - x_t\|^2 \\
&= F(x_t) + \langle \nabla F(x_t), \eta_t(v_t - x_t) \rangle + \frac{\bar{L}}{2}\|\eta_t(v_t - x_t)\|^2 \\
&\overset{(b)}{\leq} F(x_t) + \eta_t\langle \nabla F(x_t), v_t - x_t \rangle + \frac{\bar{L}\eta_t^2 D^2}{2} \\
&= F(x_t) + \eta_t\langle d_t, v_t - x_t \rangle + \eta_t\langle \nabla F(x_t) - d_t, v_t - x_t \rangle + \frac{\bar{L}\eta_t^2 D^2}{2} \\
&\overset{(c)}{\leq} F(x_t) + \eta_t\langle d_t, v_t' - x_t \rangle + \eta_t\langle \nabla F(x_t) - d_t, v_t - x_t \rangle + \frac{\bar{L}\eta_t^2 D^2}{2} \\
&= F(x_t) + \eta_t\langle \nabla F(x_t), v_t' - x_t \rangle + \eta_t\langle d_t - \nabla F(x_t), v_t' - x_t \rangle \\
&\qquad + \eta_t\langle \nabla F(x_t) - d_t, v_t - x_t \rangle + \frac{\bar{L}\eta_t^2 D^2}{2} \\
&= F(x_t) - \eta_t\mathcal{G}(x_t) + \eta_t\langle \nabla F(x_t) - d_t, v_t - v_t' \rangle + \frac{\bar{L}\eta_t^2 D^2}{2} \\
&\overset{(d)}{\leq} F(x_t) - \eta_t\mathcal{G}(x_t) + \eta_t\|\nabla F(x_t) - d_t\|\|v_t - v_t'\| + \frac{\bar{L}\eta_t^2 D^2}{2} \\
&\overset{(e)}{\leq} F(x_t) - \eta_t\mathcal{G}(x_t) + \eta_t D\|\nabla F(x_t) - d_t\| + \frac{\bar{L}\eta_t^2 D^2}{2},
\end{aligned}
$$

where we used the fact that F is $\bar{L}$-smooth in inequality (a). Inequalities (b), (e) hold because of Assumption 1. Inequality (c) is due to the optimality of v_t, and in (d), we applied the Cauchy-Schwarz inequality.

Rearrange the inequality above, we have

$$
\eta_t\mathcal{G}(x_t) \leq F(x_t) - F(x_{t+1}) + \eta_t D\|\nabla F(x_t) - d_t\| + \frac{\bar{L}\eta_t^2 D^2}{2}. \tag{3.26}
$$

Apply Eq. (3.26) recursively for $t = 1, 2, \cdots, T$, and take expectations, we attain the following inequality:

$$
\sum_{t=1}^{T} \eta_t \mathbb{E}[\mathcal{G}(x_t)] \leq F(x_1) - F(x_{T+1}) + D\sum_{t=1}^{T}\eta_t\mathbb{E}[\|\nabla F(x_t) - d_t\|] + \frac{\bar{L}D^2}{2}\sum_{t=1}^{T}\eta_t^2.
$$

By Jensen's inequality and Lemma 2 with $\alpha = 2/3$, we have

$$\mathbb{E}[\|\nabla F(x_t) - d_t\|] \leq \sqrt{\mathbb{E}[\|\nabla F(x_t) - d_t\|^2]} \leq \frac{\sqrt{C}}{t^{1/3}},$$

where $C = \max\{\frac{2(2G+D\bar{L})^2}{4/3-2^{-2/3}}, \left(\frac{2}{4/3-2^{-2/3}}\right)^4, [2D(\bar{L}+L)]^4\}$. Since $\eta_t = T^{-2/3}$, we have

$$\begin{aligned}
\mathbb{E}[\mathcal{G}(x_o)] &= \frac{\sum_{t=1}^{T} \mathbb{E}[\mathcal{G}(x_t)]}{T} \\
&\leq \frac{1}{T \cdot T^{-2/3}}[F(x_1) - F(x_{T+1}) + D\sum_{t=1}^{T} T^{-2/3}\frac{\sqrt{C}}{t^{1/3}} + \frac{\bar{L}D^2}{2}\sum_{t=1}^{T} T^{-4/3}] \\
&\leq \frac{1}{T^{1/3}}[2B + D\sqrt{C}T^{-2/3}\frac{3}{2}T^{2/3} + \frac{\bar{L}D^2}{2T^{1/3}}] \\
&= \frac{2B + 3\sqrt{C}D/2}{T^{1/3}} + \frac{\bar{L}D^2}{2T^{2/3}},
\end{aligned}$$

where the second inequality holds because $\sum_{t=1}^{T} t^{-1/3} \leq \int_0^T x^{-1/3}\mathrm{d}x = \frac{3}{2}T^{2/3}$.

3.7.5 Proof of Theorem 3

First, since $x_{t+1} = x_t + \eta_t v_t = x_t + T^{-1}v_t$, we have $x_{T+1} = \frac{\sum_{t=1}^{T} v_t}{T} \in \mathcal{K}$. Also, because now $\|x_{t+1} - x_t\| = \|\eta_t v_t\| \leq \eta_t R$, (rather than $\eta_t D$), Lemma 2 holds with new constant $C = \max\{\frac{2(2G+R\bar{L})^2}{2-2^{-\alpha}-\alpha}, \left(\frac{2}{2-2^{-\alpha}-\alpha}\right)^4, [2R(\bar{L}+L)]^4\}$. Since $\alpha = 1$, we have $C = \max\{4(2G+R\bar{L})^2, 256, [2R(\bar{L}+L)]^4\}$. Then by Jensen's inequality, we have

$$\mathbb{E}[\|\nabla F(x_t) - d_t\|] \leq \sqrt{\mathbb{E}[\|\nabla F(x_t) - d_t\|^2]} \leq \frac{\sqrt{C}}{\sqrt{t}}.$$

We observe that

$$\begin{aligned}
F(x_{t+1}) &\overset{(a)}{\geq} F(x_t) + \langle \nabla F(x_t), x_{t+1} - x_t \rangle - \frac{\bar{L}}{2} \|x_{t+1} - x_t\| \\
&= F(x_t) + \frac{1}{T} \langle \nabla F(x_t), v_t \rangle - \frac{\bar{L}}{2T^2} \|v_t\| \\
&\overset{(b)}{\geq} F(x_t) + \frac{1}{T} \langle d_t, v_t \rangle + \frac{1}{T} \langle \nabla F(x_t) - d_t, v_t \rangle - \frac{\bar{L}R^2}{2T^2} \\
&\overset{(c)}{\geq} F(x_t) + \frac{1}{T} \langle d_t, x^* \rangle + \frac{1}{T} \langle \nabla F(x_t) - d_t, v_t \rangle - \frac{\bar{L}R^2}{2T^2} \\
&= F(x_t) + \frac{1}{T} \langle \nabla F(x_t), x^* \rangle + \frac{1}{T} \langle \nabla F(x_t) - d_t, v_t - x^* \rangle - \frac{\bar{L}R^2}{2T^2} \\
&\overset{(d)}{\geq} F(x_t) + \frac{F(x^*) - F(x_t)}{T} - \frac{1}{T} \langle \nabla F(x_t) - d_t, -v_t + x^* \rangle - \frac{\bar{L}R^2}{2T^2} \\
&\overset{(e)}{\geq} F(x_t) + \frac{F(x^*) - F(x_t)}{T} - \frac{1}{T} \|\nabla F(x_t) - d_t\| \cdot \|-v_t + x^*\| - \frac{\bar{L}R^2}{2T^2} \\
&\overset{(f)}{\geq} F(x_t) + \frac{F(x^*) - F(x_t)}{T} - \frac{1}{T} 2R \|\nabla F(x_t) - d_t\| - \frac{\bar{L}R^2}{2T^2},
\end{aligned} \tag{3.27}$$

where inequality (a) holds because of the $\bar{L}$-smoothness of F, inequalities $(b), (e)$ comes from Assumption 1. We used the optimality of v_t in inequality (c), and applied the Cauchy-Schwarz inequality in(e). Inequality (d) is a little involved, since F is monotone and concave in positive directions, we have

$$\begin{aligned}
F(x^*) - F(x_t) &\leq F(x^* \vee x_t) - F(x_t) \\
&\leq \langle \nabla F(x_t), x^* \vee x_t - x_t \rangle \\
&= \langle \nabla F(x_t), (x^* - x_t) \vee 0 \rangle \\
&\leq \langle \nabla F(x_t), x^* \rangle.
\end{aligned}$$

Taking expectations on both sides of Eq. (3.27),

$$\mathbb{E}[F(x_{t+1})] \geq \mathbb{E}[F(x_t)] + \frac{F(x^*) - \mathbb{E}[F(x_t)]}{T} - \frac{2R}{T} \frac{\sqrt{C}}{\sqrt{t}} - \frac{\bar{L}R^2}{2T^2}.$$

Or

$$F(x^*) - \mathbb{E}[F(x_{t+1})] \le (1 - \frac{1}{T})[F(x^*) - \mathbb{E}[F(x_t)]] + \frac{2R}{T}\frac{\sqrt{C}}{\sqrt{t}} + \frac{\bar{L}R^2}{2T^2}.$$

Apply the inequality above recursively for $t = 1, 2, \cdots, T$, we have

$$\begin{aligned} F(x^*) - \mathbb{E}[F(x_{T+1})] &\le (1 - \frac{1}{T})^T[F(x^*) - F(x_1)] + \frac{2R\sqrt{C}}{T}\sum_{t=1}^{T} t^{-1/2} + \frac{\bar{L}R^2}{2T} \\ &\le e^{-1}F(x^*) + \frac{4R\sqrt{C}}{T^{1/2}} + \frac{\bar{L}R^2}{2T}, \end{aligned}$$

where the second inequality holds since $\sum_{t=1}^{T} t^{-1/2} \le \int_0^T x^{-1/2}\mathrm{d}x = 2T^{1/2}$. Thus we have

$$\mathbb{E}[F(x_{T+1})] \ge (1 - e^{-1})F(x^*) - \frac{4R\sqrt{C}}{T^{1/2}} - \frac{\bar{L}R^2}{2T}.$$

Chapter 4

Quantized Frank-Wolfe

4.1 Introduction

In this chapter[1], we study the application of Frank-Wolfe methods to large-scale problems. To be precise, we develop `Quantized Frank-Wolfe` (QFW), a general communication-efficient distributed FW framework for both convex and non-convex objective functions. We study the performance of QFW in two widely recognized settings: 1) stochastic optimization and 2) finite-sum optimization.

To be more specific, let $\mathcal{K} \subseteq \mathbb{R}^d$ be the constraint set. In *constrained stochastic optimization* the goal is to solve

$$\min_{x \in \mathcal{K}} f(x) \ := \ \min_{x \in \mathcal{K}} \mathbb{E}_{z \sim P}[\tilde{f}(x, z)], \tag{4.1}$$

where $x \in \mathbb{R}^d$ is the optimization variable, $z \in \mathbb{R}^q$ is a random variable drawn from a probability distribution P, which determines the choice of a stochastic function $\tilde{f} : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}$. For *constrained finite-sum optimization*, we further assume that P is a uniform distribution over $[N] = \{1, 2, \cdots, N\}$ and the goal is to solve a special case of

[1]This chapter is based on our work in [2].

problem (4.1), namely,

$$\min_{x\in\mathcal{K}} f(x) \;:=\; \min_{x\in\mathcal{K}} \frac{1}{N}\sum_{j=1}^{N} f_j(x).$$

In parallel settings, we suppose that we have a computing system consisting of a master node and M workers, and each worker maintains a local copy of x. At every iteration of the stochastic case, each worker has access to independent stochastic gradients of f; whereas in the finite-sum case, we assume $N = Mn$, thus the objective function can be decomposed as $f(x) = \frac{1}{Mn}\sum_{m\in[M],j\in[n]} f_{m,j}(x)$, and each worker m has access to the exact gradients of n component functions $f_{m,j}(x)$ for all $j \in [n]$.

This way the task of computing gradients is divided among the workers. The master node aggregates local gradients from the workers, and sends the aggregated gradients back to them so that each worker can update the model (*i.e.*, their own iterate) locally. Thus, by transmitting quantized gradients, we can reduce the communication complexity (*i.e.*, number of transmitted bits) significantly. The workflow diagram of the proposed `Quantized Frank-Wolfe` scheme is summarized in Figure 4.1. We should highlight that there is a trade-off between gradient quantization and information flow. Intuitively, more intensive quantization reduces the communication cost, but also loses more information, which may decelerate the convergence rate.

Our contributions: In this chapter, we propose a novel distributed projection-free framework that handles quantization for constrained convex and non-convex optimization problems in finite-sum and stochastic cases. It is well-known that unlike projected gradient-based methods, FW methods may diverge when fed with stochastic gradient [7, 8]. Indeed, a similar issue arises in a distributed setting where nodes exchange *quantized gradients* which are noisy estimates of the gradients. By incorporating appropriate variance reduction techniques, we show that with quantized gradients, we can obtain a provably convergent method which preserves the convergence rates of the state-of-the-art vanilla centralized methods in all the considered cases [12, 13, 18, 19]. We believe our work presents the first

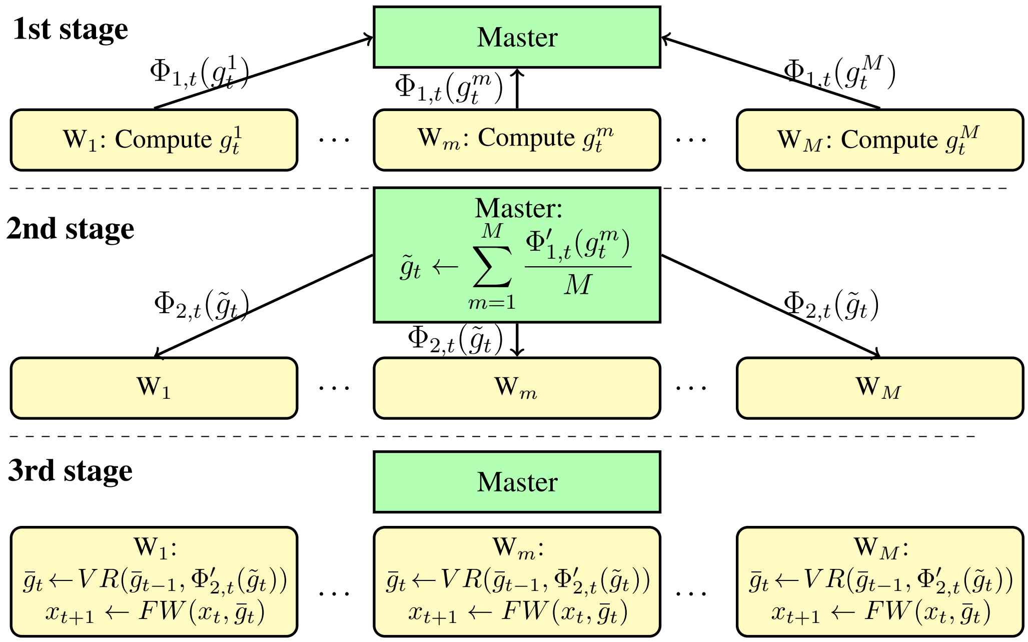


Figure 4.1: Stages of our general `Quantized Frank-Wolfe` scheme at time t. In the first stage, each worker m computes its local gradient information g_t^m and sends the quantized version $\Phi_{1,t}(g_t^m)$ to the master node. In the second stage, master computes the average of decoded received signals $\Phi'_{1,t}(g_t^m)$, *i.e.*, $\tilde{g}_t \leftarrow (1/M)\sum_{m=1}^{M} \Phi'_{1,t}(g_t^m)$ and then sends its quantized version $\Phi_{2,t}(\tilde{g}_t)$ to the workers. Note that the two quantization schemes $\Phi_{1,t}, \Phi_{2,t}$ depend on t and can be different from each other. In the third stage, workers use the decoded gradient information computed by all workers $\Phi'_{2,t}(\tilde{g}_t)$ and their previous gradient estimation $\bar{g}_{t-1}$ to update their new gradient estimation $\bar{g}_t$ via a variance reduction (VR) scheme. Once the variance reduced gradient approximation $\bar{g}_t$ is evaluated, workers compute the new variable x_{t+1} by following the update of Frank-Wolfe (FW).

quantized, distributed, and projection-free method. Our theoretical results for `Quantized Frank-Wolfe` (QFW) are summarized in Table 4.1, where the SFO complexity is the required number of stochastic gradients in stochastic case, and the IFO complexity is the number of exact gradients for component functions in finite-sum case. For the convex case, the complexity indicates the number of gradients to achieve an ϵ-suboptimal solution; while in the non-convex case, it refers to the number of gradients to find a first-order ϵ-stationary point. We note that since the M workers compute the gradients simultaneously, the time to obtain gradients is proportional to the SFO/IFO complexity *per worker*. So we report

Table 4.1: SFO/IFO Complexity per worker in different settings (M is the number of workers).

Setting	Function	SFO/IFO Complexity
Finite-sum	Convex	$\mathcal{O}\left(\frac{N\ln(1/\epsilon)+1/\epsilon^2}{M}\right)$
Finite-sum	Non-convex	$\mathcal{O}\left(\frac{\sqrt{N}}{\epsilon^2\sqrt{M}}\right)$
Stochastic	Convex	$\mathcal{O}\left(\frac{1}{M\epsilon^2}\right)$
Stochastic	Non-convex	$\mathcal{O}\left(\frac{1}{\epsilon^3\sqrt{M}}\right)$

the SFO/IFO complexity per worker, as in many other works on parallel optimization (*e.g.*, Sign-SGD [35]). The results in Table 4.1 show that more workers can decrease the SFO/IFO complexity per worker effectively, and thus accelerate the optimization procedure. All the proofs in this chapter are provided in Section 4.7.

4.2 Gradient Quantization Schemes

In most distributed optimization algorithms, the task of computing gradients is divided among the workers, and the master node uses parts of gradients at the workers to update the model (iterate) directly or sends the aggregated gradients to the worker so that each of them can update the model (iterate) locally. Therefore, the information that workers need to send to the master is the elements of the objective function gradient. Thus, by transmitting quantized gradients, we can reduce the communication bits effectively. In this section, we introduce a quantization scheme called s-`Partition Encoding Scheme` and explain how this scheme reduces the overall cost of exchanging gradients. Consider the gradient vector $g \in \mathbb{R}^d$ and let g_i be the i-th coordinate of the gradient. The s-`Partition Encoding Scheme` encodes g_i into an element from the set $\{\pm 1, \pm\frac{s-1}{s}, \cdots, \pm\frac{1}{s}, 0\}$

in a random way. To do so, we first compute the ratio $|g_i|/\|g\|_\infty$ and find the indicator $l_i \in \{0, 1, \cdots, s-1\}$ such that $|g_i|/\|g\|_\infty \in [l_i/s, (l_i+1)/s]$. Then we define the random variable b_i as

$$b_i = \begin{cases} l_i/s, & \text{w.p. } 1 - \frac{|g_i|}{\|g\|_\infty}s + l_i, \\ (l_i+1)/s, & \text{w.p. } \frac{|g_i|}{\|g\|_\infty}s - l_i. \end{cases}$$

Finally, instead of transmitting g_i, we send $\text{sign}(g_i) \cdot b_i$, alongside the norm $\|g\|_\infty$. It can be verified that $\mathbb{E}[b_i|g] = |g_i|/\|g\|_\infty$. So we define the corresponding decoding scheme as $\phi'(g_i) = \text{sign}(g_i) b_i \|g\|_\infty$ to ensure that $\phi'(g_i)$ is an unbiased estimator of g_i. We note that the encoding/decoding schemes in Fig. 4.1 are denoted as capital Φ/Φ', indicating that they can be any general schemes. The proposed s-`Partition Encoding Scheme` is denoted by ϕ/ϕ'. We also note that this quantization scheme is similar to the Stochastic Quantization method in [34], except that we use ℓ_∞-norm while they adopt the ℓ_2-norm. In the s-`Partition Encoding Scheme`, for each coordinate i, we need 1 bit to transmit $\text{sign}(g_i)$. Moreover, since $b_i \in \{0, 1/s, \ldots, (s-1)/s, 1\}$, we need $z = \log_2(s+1)$ bits to send b_i. Finally, we need 32 bits to transmit $\|g\|_\infty$. Hence, the total number of communicated bits is $32 + d(z+1)$. Here, by "bits" we mean the number of 0's and 1's transmitted.

One major advantage of the s-`Partition Encoding Scheme` is that by tuning the partition parameter s or the corresponding assigned bits z, we can smoothly control the trade-off between gradient quantization and information loss, which helps distributed algorithms to attain their best performance. We proceed to characterize the variance of the s-`Partition Encoding Scheme`.

Lemma 5. *The variance of* s-`Partition Encoding Scheme` ϕ *for any* $g \in \mathbb{R}^d$ *is bounded by*

$$\text{var}\left[\phi'(g)|g\right] \leq \frac{d}{s^2}\|g\|_\infty^2.$$

Lemma 5 demonstrates the trade-off between the error of quantization and the communication cost for s-`Partition Encoding Scheme`. In a nutshell, for larger choices of s, the variance is smaller, which in turn results in higher communication cost. If we set $s = 1$, we obtain the `Sign Encoding Scheme`, which requires communicating the encoded scalars $\text{sign}(g_i)b_i \in \{\pm 1, 0\}$ and the norm $\|g\|_\infty$. Since $z = \log_2(s+1) = 1$, the overall communicated bits for each worker are $32 + 2d$ per round. We characterize its variance in Lemma 6.

Lemma 6. *The variance of* `Sign Encoding Scheme` *is given by*

$$\text{var}\left[\phi'(g)|g\right] = \|g\|_1 \|g\|_\infty - \|g\|_2^2.$$

Remark 2. *For the probability distribution of the random variable b_i, instead of $\|g\|_\infty$, we can use other norms $\|g\|_p$ (where $p \geq 1$). But it can be verified that the ℓ_∞-norm leads to the smallest variance for* `Sign Encoding Scheme`*. That is also the reason why we do not use ℓ_2-norm as in [34].*

4.3 Convex Minimization

In this section, we analyze the convex minimization problem in both finite-sum and stochastic settings. Note that even in the setting without quantization, if we use stochastic gradients in the update of FW, it might diverge [7, 8]. So appropriate variance reduction techniques are needed for communicating quantized gradients. Nguyen et al. [71, 72, 122] developed the StochAstic Recursive grAdient algoritHm (SARAH), a stochastic recursive gradient update framework. Fang et al. [73] proposed Stochastic Path-Integrated Differential Estimator (SPIDER) technique, a variant of SARAH, for centralized unconstrained optimization. Recently, Hassani et al. [13], Yurtsever et al. [19], Shen et al. [123] proposed the SPIDER variants of FW method for both convex and non-convex optimization problems. Similar variance reduction idea was also combined with SGD to solve non-convex finite-sum problems in [70]. In this chapter, we generalize SPIDER to the constrained and distributed settings.

We first consider the case where no quantization is performed. Let $\{p_i\} \in \mathbb{N}^+$ be a sequence of period parameters. At the beginning of each period i, namely, $t = \sum_{j=1}^{i-1} p_j + 1$, each worker m samples $S_{i,1}$ component functions in finite-sum case, or stochastic functions in stochastic case, which are denoted as $\mathcal{S}_{i,1}^m$. We define the local average gradient on set $\mathcal{S}_{i,1}^m$ as $g_{i,1}^m \triangleq \nabla f_{\mathcal{S}_{i,1}^m}(x_t) = \frac{1}{S_{i,1}} \sum_{j\in\mathcal{S}_{i,1}^m} \nabla f_j(x_t)$. Then each worker m computes the average of all these local gradients $g_{i,1}^m$ and sends it to the master. Then, master node calculates the average of the M received signals and broadcasts it to all workers. Then, the workers update their gradient estimation $\bar{g}_t$ as the averaged signal $\bar{g}_t = \frac{1}{M}\sum_{m=1}^M g_{i,1}^m$.

Note $\bar{g}_t$ is identical for all the workers. In the rest of that period, *i.e.*, $t = \sum_{j=1}^{i-1} p_j + k$, where $2 \le k \le p_i$, each worker m samples a set of local functions, denoted as $\mathcal{S}_{i,k}^m$, of size $S_{i,k}$ uniformly at random, and computes the difference of averages of these gradients

$$g_{i,k}^m \triangleq \nabla f_{\mathcal{S}_{i,k}^m}(x_t) - \nabla f_{\mathcal{S}_{i,k}^m}(x_{t-1}),$$

and sends it to master. Then master node calculates the average of the M signals and broadcasts it to all the workers. The workers update their gradient estimation g_t as

$$\bar{g}_t = \bar{g}_{t-1} + \frac{1}{M}\sum_{m=1}^M g_{i,k}^m.$$

So $\bar{g}_t$ is still identical for all the workers. In order to incorporate quantization, each worker simply pushes the quantized version of the average gradients. Then the master decodes the quantizations, encodes the average of decoded signals in a quantized fashion, and broadcasts the quantization. Finally, each worker decodes the quantized signal and updates x_t locally. To be more specific, in the quantized setting, in each iteration t such that $t = \sum_{j=1}^{i-1} p_j + k$ where $1 \le k \le p_i$, each worker m sends the quantized version of its

Algorithm 3 `Quantized Frank-Wolfe` (QFW)

Input: constraint set $\mathcal{K}$, total iteration number T, No. of workers M, period parameters $\{p_i\}$, sample sizes $\{S_{i,k}\}$, learning rate η_t, initial point $x_1 \in \mathcal{K}$

Output: x_{T+1} or x_o, where x_o is chosen from $\{x_1, x_2, \cdots, x_T\}$ uniformly at random

1: **for** $t = 1, 2, \ldots, T$ **do**

2: Set $x_{i,k} \leftarrow x_t$, where $t = \sum_{j=1}^{i-1} p_j + k, 1 \leq k \leq p_i$

3: Each worker m computes local gradient $g_{i,k}^m$ by $g_{i,1}^m = \nabla f_{\mathcal{S}_{i,1}^m}(x_{i,k}) = \nabla f_{\mathcal{S}_{i,1}^m}(x_t)$ for $k = 1$, or $g_{i,k}^m \triangleq \nabla f_{\mathcal{S}_{i,k}^m}(x_{i,k}) - \nabla f_{\mathcal{S}_{i,k}^m}(x_{i,k-1}) = \nabla f_{\mathcal{S}_{i,k}^m}(x_t) - \nabla f_{\mathcal{S}_{i,k}^m}(x_{t-1})$ for $k \geq 2$

4: Each worker m encodes $g_{i,k}^m$ as $\Phi_{1,i,k}(g_{i,k}^m)$ and pushes it to the master

5: Master decodes $\Phi_{1,i,k}(g_{i,k}^m)$ as $\Phi'_{1,i,k}(g_{i,k}^m)$, and computes $\tilde{g}_{i,k} \leftarrow \frac{1}{M}\sum_{m=1}^{M} \Phi'_{1,i,k}(g_{i,k}^m)$

6: Master encodes $\tilde{g}_{i,k}$ as $\Phi_{2,i,k}(\tilde{g}_{i,k})$, and broadcasts it to all workers

7: Workers decode $\Phi_{2,i,k}(\tilde{g}_{i,k})$ as $\Phi'_{2,i,k}(\tilde{g}_{i,k})$

8: **if** $k = 1$ **then**

9: Workers update $\bar{g}_{i,k} \leftarrow \Phi'_{2,i,k}(\tilde{g}_{i,k})$

10: **else**

11: Workers update $\bar{g}_{i,k} \leftarrow \Phi'_{2,i,k}(\tilde{g}_{i,k}) + \bar{g}_{i,k-1}$

12: **end if**

13: Each worker updates $x_{t+1} \leftarrow x_t + \eta_t(v_t - x_t) = x_{i,k} + \eta_{i,k}(v_{i,k} - x_{i,k})$ where $v_{i,k} \leftarrow \arg\min_{v \in \mathcal{K}} \langle v, \bar{g}_{i,k} \rangle$

14: **end for**

local gradient information $\Phi_{1,t}(g_{i,k}^m)$ to the master. Once master collects all the quantized information, it decodes them, *i.e.*, finds $\{\Phi'_{1,t}(g_{i,k}^m)\}_{m=1}^M$, computes their average $\tilde{g}_t$, and sends its quantized version $\Phi_{2,t}(\tilde{g}_t)$ to all workers. Then, all the workers decode the received quantized signal and use it as their new gradient approximation $\bar{g}_t$ and update their variable according to the update of Frank-Wolfe, *i.e.*,

$$x_{t+1} = x_t + \eta_t(v_t - x_t),$$

where $v_t \leftarrow \arg\min_{v \in \mathcal{K}} \langle v, \bar{g}_t \rangle$. The description of our proposed `Quantized Frank-Wolfe` (QFW) method is shown in Fig. 4.1 and outlined in Algorithm 3.

Remark 3. *The model update (Line 13 in Algorithm 3) should be performed at each worker. Since all the linear programming problems (to obtain v_t) are solved simultaneously, the* *total running time is the same with that where the model updating is performed in the master node. However, additional variance would be introduced if the master node updates the model and broadcasts it in quantized manners. Thus the master-updating method lacks theoretical justification regarding convergence, and we adopt the worker-updating approach.*

4.3.1 Finite-Sum Setting

Now we proceed to establish the convergence properties of our proposed QFW in the finite-sum setting. Recall that we assume that there are N functions and M workers in total, and each worker m has access to $n = N/M$ functions $f_{m,j}$ for $j \in [n]$. We first make two assumptions on the constraint set and component functions. Let $\|\cdot\|$ denote the ℓ_2 norm in Euclidean space through out the chapter.

Assumption 7. *The constraint set $\mathcal{K}$ is convex and compact, with diameter $D = \sup_{x,y\in\mathcal{K}}\|x-y\|$.*

Assumption 8. *The functions $f_{m,i}$ are convex, L-smooth on $\mathcal{K}$, and satisfy that $\|\nabla f_{m,i}(x)\|_\infty \leq G_\infty$,* for all $m \in [M], i \in [n], x, y \in \mathcal{K}$.

Theorem 5 (Finite-Sum Convex)**.** *Consider QFW outlined in Algorithm 3. Under Assumptions 7 and 8, if we set $p_i = 2^{i-1}, \mathcal{S}_{i,1}^m = \{f_{m,j} : j \in [n]\}$* (i.e.*, each worker m samples all its n component functions), $S_{i,k} = p_i/M = 2^{i-1}/M,$* for all $i \geq 1, k \geq 2,$ *and $\eta_{i,k} = 2/(p_i + k) = 2/(2^{i-1} + k)$, and use the $s_{1,i,1} = (\sqrt{\frac{dp_i^2}{M}})$-`Partition Encoding Scheme`, $s_{2,i,1} = (\sqrt{dp_i^2})$-`Partition Encoding Scheme` for $k = 1$, and $s_{1,i,k} = (\sqrt{\frac{dp_i}{M}})$-`Partition Encoding Scheme`, $s_{2,i,k} = (\sqrt{dp_i})$-`Partition Encoding Scheme` for $k \geq 2$ as $\Phi_{1,i,k}$ and $\Phi_{2,i,k}$ in Algorithm 3, then the output $x_{T+1} \in \mathcal{K}$ satisfies*

$$\mathbb{E}[f(x_{T+1})] - f(x^*) \leq \frac{4D\sqrt{2(G_\infty^2 + 6L^2D^2)} + 2LD^2}{T},$$

where x^* is a minimizer of f on $\mathcal{K}$.

Corollary 4.3.1. *To obtain an ϵ-suboptimal solution, we need to run the QFW method for at most $\mathcal{O}(1/\epsilon)$ iterations. The IFO complexity per worker in this case is $\mathcal{O}(\frac{N\ln(1/\epsilon)+1/\epsilon^2}{M})$.*

Corollary 4.3.1 shows IFO complexity per worker is linear in $1/M$, which implicates that increasing the number of workers M will decrease the IFO complexity per worker effectively, thus accelerate the optimization procedure. Also, our numerical experiments in Section 4.5 showed that our proposed method requires significantly fewer bits than the unquantized version to achieve a specific accuracy.

4.3.2 Stochastic Setting

QFW can also be applied to the stochastic case. Recall that in the stochastic setting we assume that the objective function is $f(x) = \mathbb{E}_{z\sim P}[\tilde{f}(x,z)]$ and each worker has access to independent samples $\tilde{f}(x,z)$. Before proving the convergence properties of QFW for the stochastic setting, we first make a standard assumption on $\tilde{f}(x,z)$.

Assumption 9. *The stochastic function $\tilde{f}(x,z)$ is convex, L-smooth on $\mathcal{K}$. The gradient $\nabla\tilde{f}(x,z)$ is an unbiased estimate of $\nabla f(x)$ with bounded variance σ^2, and satisfies that $\|\nabla\tilde{f}(x,z)\|_\infty \leq G_\infty$,* for all $x\in\mathcal{K}, z\in\mathbb{R}^q$.

Theorem 6 (Stochastic Convex)**.** *Consider QFW outlined in Algorithm 3. Under Assumptions 7 and 9, if we set $p_i = 2^{i-1}, S_{i,1} = \frac{\sigma^2 p_i^2}{ML^2D^2}$, $S_{i,k} = p_i/M = 2^{i-1}/M$,* for all *$i \geq 1, k\geq 2$, and $\eta_{i,k} = 2/(p_i+k) = 2/(2^{i-1}+k)$, and use the $s_{1,i,1} = (\sqrt{\frac{dp_i^2}{M}})$-`Partition Encoding Scheme`, $s_{2,i,1} = (\sqrt{dp_i^2})$-`Partition Encoding Scheme` for $k=1$, and $s_{1,i,k} = (\sqrt{\frac{dp_i}{M}})$-`Partition Encoding Scheme`, $s_{2,i,k} = (\sqrt{dp_i})$-`Partition Encoding Scheme` for $k \geq 2$ as $\Phi_{1,i,k}$ and $\Phi_{2,i,k}$ in Algorithm 3, then the output $x_{T+1}\in\mathcal{K}$ satisfies*

$$\mathbb{E}[f(x_{T+1})] - f(x^*) \leq \frac{4D\sqrt{13L^2D^2+2G_\infty^2}+2LD^2}{T},$$

where x^* is a minimizer of f on $\mathcal{K}$.

Corollary 4.3.2. *To obtain an ϵ-suboptimal solution, we need to run the QFW method outlined in Algorithm 3 for at most $\mathcal{O}(1/\epsilon)$ iterations. The SFO complexity per worker in this case is $\mathcal{O}(1/(M\epsilon^2))$.*

Corollary 4.3.2 shows SFO complexity per worker is linear in $1/M$, which implies a speed-up for distributed settings. It also shows that the dependency of QFW's complexity on ϵ for convex settings is optimal.

Remark 4. *In theory, the partitioning levels of quantization do depend on the number of iterations. Thus more transmission bits are required over the optimization procedure. But it will not render our QFW method communication-expensive. We set the partitioning levels conservatively to achieve the theoretical guarantees. However, as shown in the experiments (Section 4.5), much smaller quantization levels (which are actually constants) are usually preferred in practice.*

4.4 Non-Convex Optimization

With slightly different parameters, QFW can be applied to non-convex settings as well. In *unconstrained* non-convex optimization problems, the gradient norm $\|\nabla f\|$ is usually a good measure of convergence as $\|\nabla f\| \to 0$ implies convergence to a stationary point. However, in the constrained setting we study the Frank-Wolfe Gap [4, 9] defined as

$$\mathcal{G}(x) = \max_{v \in \mathcal{K}} \langle v - x, -\nabla f(x) \rangle.$$

For constrained optimization problem, if a point x satisfies $\mathcal{G}(x) = 0$, then it is a first-order stationary point. Also, by definition, we have $\mathcal{G}(x) \geq 0,$ for all $x \in \mathcal{K}$. We first analyze the finite-sum setting and then the more general stochastic setting.

4.4.1 Finite-Sum Setting

To extend our results to the non-convex setting we first assume that the following condition is satisfied.

Assumption 10. *The component functions $f_{m,i}$ are L-smooth on $\mathcal{K}$ and uniformly bounded,* i.e., $\sup_{x\in\mathcal{K}}|f_{m,i}(x)|\leq M_0$. *Further,* $\|\nabla f_{m,i}(x)\|_\infty\leq G_\infty,$ for all $m \in [M], i \in [n], x, y \in \mathcal{K}$.

Theorem 7 (Finite-Sum Non-Convex)**.** *Under Assumptions 7 and 10, if we set $p_i = \sqrt{n}$, $\mathcal{S}_{i,1}^m = \{f_{m,j} : j \in [n]\}$ (*i.e., *each worker m samples all its n component functions), $S_{i,k} = \sqrt{n}/M$* for all $i \geq 1, k \geq 2, \eta_t = T^{-1/2}$ for all t, *and use the $s_{1,i,1} = (\sqrt{\frac{Td}{M}})$-`Partition Encoding Scheme`, $s_{2,i,1} = (\sqrt{Td})$-`Partition Encoding Scheme` for $k = 1$, and $s_{1,i,k} = (\frac{d^{1/2}n^{1/4}}{\sqrt{M}})$-`Partition Encoding Scheme`, $s_{2,i,k} = (d^{1/2}n^{1/4})$-`Partition Encoding Scheme` for $k \geq 2$ as $\Phi_{1,i,k}$ and $\Phi_{2,i,k}$ in Algorithm 3, then the output $x_o \in \mathcal{K}$ satisfies*

$$\mathbb{E}[\mathcal{G}(x_o)] \leq \frac{2M_0 + D\sqrt{3L^2D^2 + 2G_\infty^2} + \frac{LD^2}{2}}{\sqrt{T}}.$$

Corollary 4.4.1. *Algorithm 3 finds an ϵ-first-order stationary point after at most $\mathcal{O}(1/\epsilon^2)$ iterations. The IFO complexity per worker is $\mathcal{O}(\sqrt{N}/(\epsilon^2\sqrt{M}))$.*

Corollary 4.4.1 shows IFO complexity per worker is linear in $1/\sqrt{M}$, implicating that increasing the number of workers M will decrease the IFO complexity per worker effectively, thus accelerate the optimization procedure.

4.4.2 Stochastic Setting

For non-convex objective function, the stochastic optimization problem in (4.1) can be solved approximately by the QFW method described in Algorithm 3. Specifically, the

Algorithm 4 `Stochastic Non-Convex Quantized Frank-Wolfe` (SNC-QFW)

Input: constraint set $\mathcal{K}$, iteration number T, No. of workers M, initial point $x_1 \in \mathcal{K}$

1: Obtain T independent samples of z_i, and define finite-sum $\hat{f}(x) = \frac{1}{T}\sum_{i=1}^{T} \tilde{f}(x, z_i)$

2: Apply Algorithm 3 on $\hat{f}$ with $N = T$ and all other parameters being the same as in Theorem 7

Output: x_o, where x_o is chosen from $\{x_1, x_2, \cdots, x_T\}$ uniformly at random

objective function $f(x) = \mathbb{E}_{z\sim P}[\tilde{f}(x, z)]$ can be approximated by a finite-sum problems with B samples where the samples $\{z_1, \ldots, z_B\}$ are independently drawn according to the probability distribution P. Thus we define the surrogate function $\hat{f}$

$$\hat{f}(x) = \frac{1}{B}\sum_{i=1}^{B} \tilde{f}(x, z_i),$$

as the finite-sum approximation of $f(x)$. As a result, we can apply QFW on $\hat{f}$, thus optimize f approximately. The algorithm is outlined in Algorithm 4.

In the non-convex setting, if we further assume that $\tilde{f}(x, z)$ is G-Lipschitz for all $z \in \mathbb{R}^q$, then we have the following lemma:

Lemma 7 (Theorem 5 of [10])**.** *If we define $\hat{\mathcal{G}}(x) = \max_{v\in\mathcal{K}}\langle v - x, -\nabla\hat{f}(x)\rangle$, then $\mathbb{E}[\mathcal{G}(x) - \hat{\mathcal{G}}(x)] \leq \frac{GD}{\sqrt{B}}$. Recall that D is the diameter of $\mathcal{K}$ as defined in Assumption 7, $\mathcal{G}(x) = \max_{v\in\mathcal{K}}\langle v - x, -\nabla f(x)\rangle$. Thus for the output x_o, we have*

$$\mathbb{E}[\mathcal{G}(x_o)] \leq \mathbb{E}[\hat{\mathcal{G}}(x_o)] + \frac{GD}{\sqrt{B}}.$$

Baesd on Theorem 7 and Lemma 7, we have the following theoretical guarantee for stochastic non-convex minimization.

Theorem 8 (Stochastic Non-Convex)**.** *Assuming that for all $z \in \mathbb{R}^q$, $\tilde{f}(x, z)$ is G-Lipschitz, L-smooth, and satisfies $|\tilde{f}(x, z)| \leq M_0$ for all $x \in \mathcal{K}$. If we obtain T independent samples of z_i, and apply Algorithm 3 on $\hat{f}(x) = \frac{1}{T}\sum_{i=1}^{T} \tilde{f}(x, z_i)$ with $N = T, n = T/M$, and* *all the other parameters set the same as in Theorem 7, then after T iterations, the output $x_o \in \mathcal{K}$ satisfies*

$$\mathbb{E}[\mathcal{G}(x_o)] \leq \frac{2M_0 + D\sqrt{3L^2D^2 + 2G^2} + \frac{LD^2}{2}}{\sqrt{T}} + \frac{GD}{\sqrt{T}}.$$

We note that the algorithm finds an ϵ-first-order stationary point with at most $\mathcal{O}(1/\epsilon^2)$ rounds. The SFO complexity per worker is $\mathcal{O}(\sqrt{N}/(\sqrt{M}\epsilon^2)) = \mathcal{O}(\frac{1}{\epsilon^3\sqrt{M}})$. Thus the SFO complexity per worker is linear in $1/\sqrt{M}$, which implicates that increasing the number of workers M will decrease the SFO complexity per worker.

4.5 Experiments

We evaluate the performance of algorithms by visualizing their loss $f(x_t)$ vs. the number of transmitted bits. The experiments were performed on 20 Intel Xeon E5-2660 cores and thus the number of workers is 20. For each curve in the figures below, we ran at least 50 repeated experiments, and the height of shaded regions represents two standard deviations.

In our first setup, we consider a multinomial logistic regression problem. Consider the dataset $\{(x_i, y_i)\}_{i=1}^N \subseteq \mathbb{R}^d \times \{1, \dots, C\}$ with N samples that have C different labels. We aim to find a model w to classify these sample points under the condition that the solution has a small ℓ_1-norm. Therefore, we aim to solve the following convex problem

$$\begin{aligned} \min_w f(w) &:= -\sum_{i=1}^{N}\sum_{c=1}^{C} 1\{y_i = c\} \log \frac{\exp(w_c^\top x_i)}{\sum_{j=1}^{C} \exp(w_j^\top x_i)}, \\ \text{subject to} &\quad \|w\|_1 \leq 1. \end{aligned} \tag{4.2}$$

In our experiments, we use the MNIST dataset and assume that each worker stores 3000 images. Therefore, the overall number of samples in the training set is $N = 60000$.

In our second setup, our goal is to minimize the loss of a three-layer neural network under some conditions on the norm of the solution. Before stating the problem precisely, let us define the log-loss function as $h(y,p) \triangleq -\sum_{c=1}^{C} 1\{y=c\} \log p_c$ for $y \in \{1,\ldots,C\}$ and a C-dimensional probability vector $p := (p_1, \cdots, p_C)$. We aim to solve the following non-convex problem

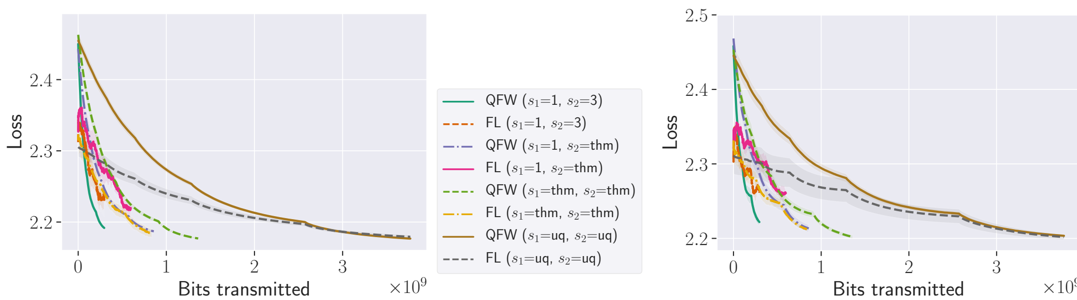
$$\min_{W_1,W_2} \sum_{i=1}^{N} h(y_i, \phi(W_2 \,\mathrm{relu}(W_1 x_i + b) + b_2)),$$

$$\text{subject to } \|W_i\|_1 \leq a_1, \|b_i\|_1 \leq a_2, \tag{4.3}$$

where $\mathrm{relu}(x) \triangleq \max\{0, x\}$ is the ReLU function and ϕ is the softmax function. The imposed ℓ_1 constraint on the weights leads to a sparse network. We further remark that Frank-Wolfe methods are suitable for training a neural network subject to an ℓ_1 constraint as they are equivalent to a dropout regularization [21]. In our setup, the size of matrices W_1 and W_2 are 784×50 and 50×10, respectively, and the constraints parameters are $a_1 = a_2 = 5$.

For all of the considered settings, we vary the quantization level, use the s_1-partition encoding scheme when workers send encoded tensors to the master and use the s_2-partition encoding scheme when the master broadcasts encoded tensors to the workers ($s_i = uq$ indicates FW without quantization and $s_i = thm$ indicates QFW with the quantization level recommended by our theorems, where $i = 1, 2$). We also propose the federated learning approach `FL`, an effective heuristic based on QFW, where each worker performs its local Frank-Wolfe update autonomously without communicating with each other and synchronizes the model only at the end of each round. This method may not enjoy the strong theoretical guarantees of QFW and we observe in our experiments that it is even prone to divergence. In stochastic minimization, each worker samples 1000 images uniformly at random and without replacement.

In Fig. 4.2, we observe the performance of `FL`, FW without quantization, and different



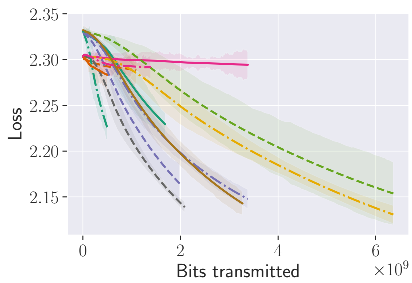
(a) Stochastic convex minimization

(b) Finite-sum convex minimization

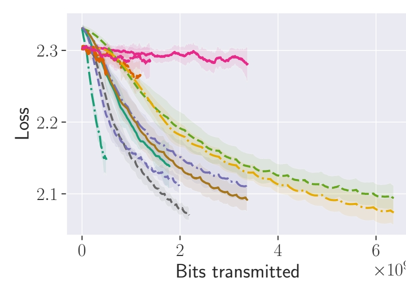
Figure 4.2: Comparison in terms of the loss versus the number of transmitted bits for a multinomial logistic regression problem. The best performance belongs to QFW with `Partition Encoding Scheme` ($s_1 = 1, s_2 = 3$), and FW without quantization has the worst performance.

variants of QFW for solving the multinomial logistic regression problem in (4.2). The stochastic minimization is presented in Fig. 4.2a and the finite-sum minimization is shown in Fig. 4.2b. We observe that QFW with `Partition Encoding Scheme` ($s_1 = 1, s_2 = 3$) has the best performance in terms of the amount of transmitted bits. Specifically, QFW with `Partition Encoding Scheme` ($s_1 = 1, s_2 = 3$) requires 3×10^8 bits to hit the lowest loss in Figs. 4.2a and 4.2b, while `FL` with the same level of quantization only achieves a suboptimal loss (approximately 2.24) with the same amount of communication. Furthermore, FW without quantization requires more than 3.8×10^9 bits to reach the same error, *i.e.*, quantization reduces communication load by at least an order of magnitude.

Fig. 4.3 demonstrates the performance of `FL`, FW without quantization, and different variants of QFW for solving the three-layer neural network in (4.3). Again we show the stochastic minimization on the left (Fig. 4.3a) and the finite-sum minimization on the right (Fig. 4.3b). We observe four divergent curves of the federated learning method `FL` ($s_1 = 3, s_2 = 1$; $s_1 = thm, s_2 = 1$; $s_1 = 1, s_2 = thm$; and $s_1 = 1, s_2 = uq$), while all QFW curves converge. This observation is in accordance with the fact that `FL` has no theoretical guarantee, in contrast to the proposed QFW method. FW without quantization consumes approximately 6.5×10^9 bits to achieve the lowest loss. Its amount



(a) Stochastic non-convex minimization



(b) Finite-sum non-convex minimization

Figure 4.3: Comparison of algorithms in terms of the loss versus the number of transmitted bits for a three-layer neural network. FW without quantization ($s = uq$) significantly underperforms the quantized FW methods. We observe four divergent curves of the federated learning method FL ($s_1 = 3, s_2 = 1$; $s_1 = thm, s_2 = 1$; $s_1 = 1, s_2 = thm$; and $s_1 = 1, s_2 = uq$).

of communication is twice that of QFW with the quantization levels recommended by Theorems 7 and 8.

In both convex and non-convex setups, the theoretically guaranteed quantization levels recommended by our theorems may be conservative. In fact, a `Partition Encoding Scheme` with partitions much fewer than our theorems recommend achieves a similar loss level and saves even more communication bits. For example, QFW with $s_1 = 1, s_2 = 3$ and $s_1 = 1, s_2 = thm$ exhibits a higher communication efficiency than QFW with $s_1 = thm, s_2 = thm$ in Figs. 4.2a and 4.2b.

4.6 Conclusion

In this chapter, we developed `Quantized Frank-Wolfe` (QFW), the first general-purpose projection-free and communication-efficient framework for constrained optimization. Along with proposing various quantization schemes, QFW can address both convex and non-convex optimization settings in stochastic and finite-sum cases. We provided theoretical guarantees on the convergence rate of QFW and validated its efficiency empirically on training multinomial logistic regression and neural networks. Our theoretical results highlighted the importance of variance reduction techniques to stabalize FW and achieve a sweet trade-off between the communication complexity and convergence rate in distributed settings. We also note that it might be possible to design simpler Quantized FW methods based on the new developments [1].

4.7 Proofs

4.7.1 Proof of Lemma 5

Proof. For any given vector $g \in \mathbb{R}^d$, the ratio $|g_i|/\|g\|_\infty$ lies in an interval of the form $[l_i/s, (l_i+1)/s]$ where $l_i \in \{0, 1, \ldots, s-1\}$. Hence, for that specific l_i, the following inequalities

$$\frac{l_i}{s} \leq \frac{|g_i|}{\|g\|_\infty} \leq \frac{l_i+1}{s} \tag{4.4}$$

are satisfied. Moreover, based on the probability distribution of b_i we know that

$$\frac{l_i}{s} \leq b_i \leq \frac{l_i+1}{s}. \tag{4.5}$$

Therefore, based on the inequalities in (4.4) and (4.5) we can write

$$-\frac{1}{s} \leq \frac{|g_i|}{\|g\|_\infty} - b_i \leq \frac{1}{s}. \tag{4.6}$$

Hence, we can show that the variance of s-`Partition Encoding Scheme` is upper bounded by

$$\begin{aligned}\text{var}\,[\phi'(g)|g] &= \mathbb{E}[\|\phi'(g) - g\|^2|g] \\ &= \sum_{i=1}^{d} \mathbb{E}[(g_i - \text{sign}(g_i)b_i\|g\|_\infty)^2|g] \\ &= \sum_{i=1}^{d} \mathbb{E}[(|g_i| - b_i\|g\|_\infty)^2|g] \\ &= \sum_{i=1}^{d} \|g\|_\infty^2 \mathbb{E}\left[\left(\frac{|g_i|}{\|g\|_\infty} - b_i\right)^2 \mid g\right] \\ &\le \frac{d}{s^2}\|g\|_\infty^2,\end{aligned}$$

where the inequality holds due to (4.6). □

4.7.2 Proof of Theorem 5 and Corollary 4.3.1

The key to the proofs of Theorem 5 is to upper bound the difference between the true gradient $\nabla f(x_t) = \nabla f(x_{i,k})$ and the estimated gradient $\bar{g}_{i,k}$. Intuitively, if the error is small enough, then we can approximate $\nabla f(x_{i,k})$ by $\bar{g}_{i,k}$. Thus the algorithm fed with the estimated gradient $\bar{g}_{i,k}$ will still converge.

So we first address the bound of $\|\nabla f(x_{i,k}) - \bar{g}_{i,k}\|$, which is resolved in the following lemma.

Lemma 8. *Under the condition of Theorem 5, we have*

$$\mathbb{E}[\|\nabla f(x_{i,k}) - \bar{g}_{i,k}\|^2] \le \frac{2(G_\infty^2 + 2L^2D^2 + 4L_\infty^2 D^2)}{p_i^2}.$$

Proof. We first define a few auxiliary variables. On each worker m, we define the average function of its n component functions as $f^{(m)}(x) = \frac{\sum_{j=1}^{n} f_{m,j}(x)}{n}$, so $f(x) = \frac{\sum_{m=1}^{M} f^{(m)}(x)}{M}$.

We also define

$$g_{i,k}^{(m)} = \begin{cases} g_{i,k}^{m} & k = 1, \\ g_{i,k-1}^{(m)} + g_{i,k}^{m} = \sum_{j=1}^{k} g_{i,j}^{m} & k \geq 2, \end{cases}$$

where $g_{i,k}^{m}$ is defined in Algorithm 3. Then $g_{i,k}^{(m)}$ is an unbiased estimator of $\nabla f^{(m)}(x_{i,k})$. We define the average of $g_{i,k}^{(m)}$ as

$$g_{i,k} = \frac{\sum_{m=1}^{M} g_{i,k}^{(m)}}{M}.$$

We also define $\mathcal{F}_{i,k}$ to be the σ-field generated by all the randomness before round (i, k), *i.e,* round $t = \sum_{j=1}^{i-1} p_j + k$. We note that given $\mathcal{F}_{i,k}$, $x_{i,k}$ is actually determined, and we can verify that $\mathbb{E}[g_{i,k}|\mathcal{F}_{i,k}] = \nabla f(x_{i,k})$, and $\mathbb{E}[\bar{g}_{i,k}|\mathcal{F}_{i,k}, g_{i,k}] = g_{i,k}$, for all (i, k). Here, with abuse of notation, $\mathbb{E}[\cdot|g_{i,k}]$ is the conditional expectation given not only the value of $g_{i,k}$, but also the sampled gradients $\nabla f_{m,j}(x_{i,k}), \nabla f_{m,j}(x_{i,k-1})$(if defined) for all $j \in \mathcal{S}_{i,k}^{m}, m \in [M]$.

Then by law of total expectation, we have

$$\begin{aligned} \mathbb{E}[\|\nabla f(x_{i,k}) - \bar{g}_{i,k}\|^2] &= \mathbb{E}[\mathbb{E}[\|\nabla f(x_{i,k}) - \bar{g}_{i,k}\|^2|\mathcal{F}_{i,k}]] \\ &= \mathbb{E}[\mathbb{E}[\|\nabla f(x_{i,k}) - g_{i,k} + g_{i,k} - \bar{g}_{i,k}\|^2|\mathcal{F}_{i,k}]] \\ &= \mathbb{E}[\mathbb{E}[\|\nabla f(x_{i,k}) - g_{i,k}\|^2|\mathcal{F}_{t-1}]] + \mathbb{E}[\mathbb{E}[\|g_{i,k} - \bar{g}_{i,k}\|^2|\mathcal{F}_{i,k}]] \\ &\quad + 2\mathbb{E}[\mathbb{E}[\langle \nabla f(x_{i,k}) - g_{i,k}, g_{i,k} - \bar{g}_{i,k}\rangle|\mathcal{F}_{i,k}]] \\ &= \mathbb{E}[\|\nabla f(x_{i,k}) - g_{i,k}\|^2] + \mathbb{E}[\|g_{i,k} - \bar{g}_{i,k}\|^2], \end{aligned} \tag{4.7}$$

where the last equation holds since

$$\begin{aligned} \mathbb{E}[\langle \nabla f(x_{i,k}) - g_{i,k}, g_{i,k} - \bar{g}_{i,k}\rangle|\mathcal{F}_{i,k}] &= \mathbb{E}[\mathbb{E}[\langle \nabla f(x_{i,k}) - g_{i,k}, g_{i,k} - \bar{g}_{i,k}\rangle|\mathcal{F}_{i,k}, g_{i,k}]|\mathcal{F}_{i,k}] \\ &= \mathbb{E}[\langle \nabla f(x_{i,k}) - g_{i,k}, \mathbb{E}[g_{i,k} - \bar{g}_{i,k}|\mathcal{F}_{i,k}, g_{i,k}]\rangle|\mathcal{F}_{i,k}] \\ &= 0. \end{aligned}$$

Now we turn to bound $\mathbb{E}[\|\nabla f(x_{i,k}) - g_{i,k}\|^2]$. In fact, we have

$$\begin{aligned}\mathbb{E}[\|\nabla f(x_{i,k}) - g_{i,k}\|^2] &= \mathbb{E}[\|\frac{\sum_{m=1}^{M} \nabla f^{(m)}(x_{i,k})}{M} - \frac{\sum_{m=1}^{M} g_{i,k}^{(m)}}{M}\|^2] \\ &= \frac{\sum_{m=1}^{M} \mathbb{E}[\|\nabla f^{(m)}(x_{i,k}) - g_{i,k}^{(m)}\|^2]}{M^2}.\end{aligned} \tag{4.8}$$

For $k \geq 2$, we have

$$\begin{aligned}&\mathbb{E}[\|\nabla f^{(m)}(x_{i,k}) - g_{i,k}^{(m)}\|^2] \\ =& \mathbb{E}[\mathbb{E}[\|[\nabla f^{(m)}(x_{i,k}) - \nabla f^{(m)}(x_{i,k-1})] - g_{i,k}^{m}\|^2|\mathcal{F}_{i,k}]] + \mathbb{E}[\mathbb{E}[\|\nabla f^{(m)}(x_{i,k-1}) - g_{i,k-1}^{(m)}\|^2|\mathcal{F}_{i,k}]] \\ =& \mathbb{E}[\text{var}\left[g_{i,k}^{m}|\mathcal{F}_{i,k}\right]] + \mathbb{E}[\|\nabla f^{(m)}(x_{i,k-1}) - g_{i,k-1}^{(m)}\|^2] \\ =& \mathbb{E}[\text{var}\left[\frac{\sum_{j\in\mathcal{S}_{i,k}^{m}} \nabla f_j(x_{i,k}) - \nabla f_j(x_{i,k-1})}{S_{i,k}}|\mathcal{F}_{i,k}\right]] + \mathbb{E}[\|\nabla f^{(m)}(x_{i,k-1}) - g_{i,k-1}^{(m)}\|^2] \\ =& \mathbb{E}[\frac{\sum_{j\in\mathcal{S}_{i,k}^{m}} \text{var}\left[\nabla f_j(x_{i,k}) - \nabla f_j(x_{i,k-1})|\mathcal{F}_{i,k}\right]}{[S_{i,k}]^2}] + \mathbb{E}[\|\nabla f^{(m)}(x_{i,k-1}) - g_{i,k-1}^{(m)}\|^2] \\ \leq& \mathbb{E}[\frac{\sum_{j\in\mathcal{S}_{i,k}^{m}} \mathbb{E}[\|\nabla f_j(x_{i,k}) - \nabla f_j(x_{i,k-1})\|^2|\mathcal{F}_{i,k}]}{[S_{i,k}]^2})] + \mathbb{E}[\|\nabla f^{(m)}(x_{i,k-1}) - g_{i,k-1}^{(m)}\|^2] \\ \leq& \frac{1}{S_{i,k}}(LD\eta_{i,k-1})^2 + \mathbb{E}[\|\nabla f(x_{i,k-1}) - g_{i,k-1}\|^2] \\ =& \frac{L^2D^2\eta_{i,k-1}^2}{S_{i,k}} + \mathbb{E}[\|\nabla f^{(m)}(x_{i,k-1}) - g_{i,k-1}^{(m)}\|^2].\end{aligned}$$

For $k = 1$, we have $g_{i,1}^{(m)} = \nabla f^{(m)}(x_{i,1})$. So

$$\mathbb{E}[\|\nabla f^{(m)}(x_{i,k}) - g_{i,k}^{(m)}\|^2] \leq L^2D^2 \sum_{j=2}^{k} \frac{\eta_{i,j-1}^2}{S_{i,j}} = \frac{L^2D^2M}{p_i} \sum_{j=2}^{k} \eta_{i,j-1}^2.$$

Since

$$\sum_{j=2}^{k} \eta_{i,j-1}^2 = \sum_{j=2}^{k} \frac{4}{(p_i + j - 1)^2} \leq \sum_{j=2}^{k} \frac{4}{p_i^2} \leq \frac{4}{p_i},$$

we have

$$\mathbb{E}[\|\nabla f^{(m)}(x_{i,k}) - g_{i,k}^{(m)}\|^2] \leq \frac{4ML^2D^2}{p_i^2}.$$

Combine with Eq. (4.8), we have

$$\mathbb{E}[\|\nabla f(x_{i,k}) - g_{i,k}\|^2] \leq \frac{M \cdot 4ML^2D^2}{M^2 \cdot p_i^2} = \frac{4L^2D^2}{p_i^2}. \tag{4.9}$$

Now we only need to bound $\mathbb{E}[\|g_{i,k} - \bar{g}_{i,k}\|^2]$. For $k \geq 2$, we have

$$\begin{aligned}
&\mathbb{E}[\|g_{i,k} - \bar{g}_{i,k}\|^2] \\
&= \mathbb{E}[\mathbb{E}[\|\frac{\sum_{m=1}^M g_{i,k}^m}{M} + g_{i,k-1} - \phi'_{2,i,k}(\tilde{g}_{i,k}) - \bar{g}_{i,k-1}\|^2 | \mathcal{F}_{i,k}, g_{i,k}]] \\
&= \mathbb{E}[\mathbb{E}[\|\frac{\sum_{m=1}^M g_{i,k}^m}{M} - \phi'_{2,i,k}(\tilde{g}_{i,k})\|^2 | \mathcal{F}_{i,k}, g_{i,k}]] + \mathbb{E}[\|g_{i,k-1} - \bar{g}_{i,k-1}\|^2] \\
&\quad + 2\mathbb{E}[\mathbb{E}[\frac{\sum_{m=1}^M g_{i,k}^m}{M} - \phi'_{2,i,k}(\tilde{g}_{i,k}), g_{i,k-1} - \bar{g}_{i,k-1}\rangle | \mathcal{F}_{i,k}, g_{i,k-1}]].
\end{aligned}$$

Moreover

$$\begin{aligned}
\mathbb{E}[\phi'_{2,i,k}(\tilde{g}_{i,k}) | \mathcal{F}_{i,k}, g_{i,k}] &= \mathbb{E}[\tilde{g}_{i,k} | \mathcal{F}_{i,k}, g_{i,k}] \\
&= \mathbb{E}[\sum_{m=1}^M \phi'_{1,i,k}(g_{i,k}^m)/M | \mathcal{F}_{i,k}, g_{i,k}] \\
&= \frac{\sum_{m=1}^M g_{i,k}^m}{M},
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{E}[\mathbb{E}[\|\frac{\sum_{m=1}^M g_{i,k}^m}{M} - \phi'_{2,i,k}(\tilde{g}_{i,k})\|^2 | \mathcal{F}_{i,k}, g_{i,k}]] \\
&= \mathbb{E}[\mathbb{E}[\|\frac{\sum_{m=1}^M g_{i,k}^m}{M} - \tilde{g}_{i,k} + \tilde{g}_{i,k} - \phi'_{2,i,k}(\tilde{g}_{i,k})\|^2 | \mathcal{F}_{i,k}, g_{i,k}]] \\
&= \mathbb{E}[\mathbb{E}[\|\frac{\sum_{m=1}^M g_{i,k}^m}{M} - \sum_{m=1}^M \phi'_{1,i,k}(g_{i,k}^m)/M\|^2 | \mathcal{F}_{i,k}, g_{i,k}]] + \mathbb{E}[\mathbb{E}[\|\tilde{g}_{i,k} - \phi'_{2,i,k}(\tilde{g}_{i,k})\|^2 | \mathcal{F}_{i,k}, g_{i,k}, \tilde{g}_{i,k}]] \\
&\leq \frac{1}{M} \frac{d}{s_{1,i,k}^2} (\eta_{i,k-1} LD)^2 + \frac{d}{s_{2,i,k}^2} (\eta_{i,k-1} LD)^2 \\
&= \frac{\eta_{i,k-1}^2 dL^2D^2}{Ms_{1,i,k}^2} + \frac{\eta_{i,k-1}^2 dL^2D^2}{s_{2,i,k}^2},
\end{aligned}$$

where in the inequality, we apply Lemma 5 with $\|g_{i,k}^m\|_\infty = \|\nabla f_{\mathcal{S}_{i,k}^m}(x_{i,k}) - \nabla f_{\mathcal{S}_{i,k}^m}(x_{i,k-1})\|_\infty \leq \|\nabla f_{\mathcal{S}_{i,k}^m}(x_{i,k}) - \nabla f_{\mathcal{S}_{i,k}^m}(x_{i,k-1})\|_2 \leq \eta_{i,k-1}LD$ and $\|\tilde{g}_{i,k}\|_\infty = \|\sum_{m=1}^M \phi'_{1,i,k}(g_{i,k}^m)/M\|_\infty \leq \eta_{i,k-1}LD$. Now for $k \geq 2$ we have,

$$\mathbb{E}[\|g_{i,k} - \bar{g}_{i,k}\|^2] \leq \frac{\eta_{i,k-1}^2 dL^2D^2}{Ms_{1,i,k}^2} + \frac{\eta_{i,k-1}^2 dL^2D^2}{s_{2,i,k}^2} + \mathbb{E}[\|g_{i,k-1} - \bar{g}_{i,k-1}\|^2].$$

If $k = 1$, we have

$$\begin{aligned}
\mathbb{E}[\|g_{i,k} - \bar{g}_{i,k}\|^2] &= \mathbb{E}[\|\nabla f(x_{i,k}) - \tilde{g}_{i,k} + \tilde{g}_{i,k} - \phi'_{2,i,k}(\tilde{g}_{i,k})\|^2] \\
&= \mathbb{E}[\mathbb{E}[\|\nabla f(x_{i,k}) - \frac{\sum_{m=1}^M \phi'_{1,i,k}(\nabla f^{(m)}(x_{i,k}))}{M}\|^2 | \mathcal{F}_{i,k}, g_{i,k}]] \\
&\quad + \mathbb{E}[\mathbb{E}[\|\tilde{g}_{i,k} - \phi'_{2,i,k}(\tilde{g}_{i,k})\|^2 | \mathcal{F}_{i,k}, g_{i,k}, \tilde{g}_{i,k}]] \\
&\leq \frac{1}{M^2}\mathbb{E}[\sum_{m=1}^M \mathbb{E}[\|\nabla f^{(m)}(x_{i,k}) - \phi'_{1,t}(\nabla f^{(m)}(x_{i,k}))\|^2 | \mathcal{F}_{i,k}, g_{i,k}]] + \frac{d}{s_{2,i,k}^2}G_\infty^2 \\
&\leq \frac{dG_\infty^2}{Ms_{1,i,k}^2} + \frac{dG_\infty^2}{s_{2,i,k}^2},
\end{aligned}$$

where in the inequality, we apply Lemma 5 with $\|\nabla f^{(m)}(x_k)\|_\infty \leq G_\infty$ and $\|\tilde{g}_{i,k}\|_\infty = \|\frac{\sum_{m=1}^M \phi'_{1,i,k}(\nabla f^{(m)}(x_k))}{M}\|_\infty \leq G_\infty$. Then we have

$$\begin{aligned}
\mathbb{E}[\|g_{i,k} - \bar{g}_{i,k}\|^2] &\leq \sum_{j=2}^k \frac{\eta_{i,j-1}^2 dL^2D^2}{Ms_{1,i,j}^2} + \sum_{j=2}^k \frac{\eta_{i,j-1}^2 dL^2D^2}{s_{2,i,j}^2} + \frac{dG_\infty^2}{Ms_{1,i,1}^2} + \frac{dG_\infty^2}{s_{2,i,1}^2} \\
&\leq \frac{dL^2D^2}{Ms_{1,i}^2}\sum_{j=2}^k \eta_{i,j-1}^2 + \frac{dL^2D^2}{s_{2,i}^2}\sum_{j=2}^k \eta_{i,j-1}^2 + \frac{dG_\infty^2}{Ms_{1,i,1}^2} + \frac{dG_\infty^2}{s_{2,i,1}^2} \\
&\leq \frac{dL^2D^2}{M\frac{p_i d}{M}}\frac{4}{p_i} + \frac{dL^2D^2}{p_i d}\frac{4}{p_i} + \frac{dG_\infty^2}{M\frac{dp_i^2}{M}} + \frac{dG_\infty^2}{dp_i^2} \\
&= \frac{2G_\infty^2 + 8L^2D^2}{p_i^2}.
\end{aligned} \tag{4.10}$$

Now combine Eqs. (4.7), (4.9) and (4.10), we have

$$\mathbb{E}[\|\nabla f(x_{i,k}) - \bar{g}_{i,k}\|^2] \leq \frac{2(G_\infty^2 + 6L^2D^2)}{p_i^2} \triangleq \frac{C_1^2}{p_i^2}.$$

□

Now we turn to prove Theorem 5. First, since $x_{t+1} = (1-\eta_{i,k})x_t + \eta_{i,k}v_{i,k}$ is a convex combination of $x_t, v_{i,k}$, and $x_1 \in \mathcal{K}, v_{i,k} \in \mathcal{K},$ for all t, we can prove $x_t \in \mathcal{K},$ for all t by induction. So $x_{T+1} \in \mathcal{K}$. Then we need the following lemma.

Lemma 9 (Proof of Theorem 1 in [19])**.** *Consider Algorithm 3, under the conditions of Theorem 5, we have*

$$\mathbb{E}[f_{i,k+1}] - f(x^*) \leq (1-\eta_{i,k})(\mathbb{E}[f(x_{i,k}) - f(x^*)]) + \eta_{i,k}D\mathbb{E}[\|\nabla f(x_i,k) - \bar{g}_{i,k}\|] + \eta_{i,k}^2\frac{LD^2}{2}.$$

Moreover, by analyzing the telescopic sum of the inequality over (i,k), we have

$$\mathbb{E}[f(x_{i,k+1})] - f(x^*) \leq \sum_{(\tau,j)} \left(\eta_{\tau,j}D\mathbb{E}[\|\nabla f(x_{\tau,j}) - \bar{g}_{\tau,j}\|] + \eta_{\tau,j}^2\frac{LD^2}{2}\right)\frac{(p_\tau+j-2)(p_\tau+j-1)}{(p_i+k-1)(p_i+k)}.$$

By Lemma 8 and Jensen's inequality, we have

$$\mathbb{E}[\|\nabla f(x_i,k) - \bar{g}_{i,k}\|] \leq \sqrt{\mathbb{E}[\|\nabla f(x_i,k) - \bar{g}_{i,k}\|^2]} \leq \frac{C_1}{p_i}.$$

So

$$
\begin{aligned}
&\sum_{(\tau,j)} \eta_{\tau,j} D\mathbb{E}[\|\nabla f(x_i,k) - \bar{g}_{i,k}\|] \frac{(p_\tau + j - 2)(p_\tau + j - 1)}{(p_i + k - 1)(p_i + k)} \\
\leq &\sum_{(\tau,j)} \frac{2}{p_\tau + j} D \frac{C_1}{p_\tau} \frac{(p_\tau + j - 2)(p_\tau + j - 1)}{(p_i + k - 1)(p_i + k)} \\
\leq &\frac{4C_1 D}{(p_i + k - 1)(p_i + k)} \sum_{(\tau,j)} 1 \\
\leq &\frac{4C_1 D}{p_i + k}.
\end{aligned}
$$

We also have

$$
\begin{aligned}
\sum_{(\tau,j)} \eta_{\tau,j}^2 \frac{LD^2}{2} \frac{(p_\tau + j - 2)(p_\tau + j - 1)}{(p_i + k - 1)(p_i + k)} &= \sum_{(\tau,j)} \frac{4}{(p_\tau + j)^2} \frac{LD^2}{2} \frac{(p_\tau + j - 2)(p_\tau + j - 1)}{(p_i + k - 1)(p_i + k)} \\
&\leq \frac{2LD^2}{(p_i + k - 1)(p_i + k)} \sum_{(\tau,j)} 1 \\
&\leq \frac{2LD^2}{p_i + k}.
\end{aligned}
$$

Thus by Lemma 9, we have

$$
\mathbb{E}[f(x_{i,k+1})] - f(x^*) \leq \frac{4C_1 D + 2LD^2}{p_i + k}.
$$

By definition, $x_{i,k+1} = x_t$, where $t = \sum_{j=1}^{i-1} p_j + k + 1 = p_i + k$. When $t = T$, we have

$$
\mathbb{E}[f(x_T)] - f(x^*) \leq \frac{4C_1 D + 2LD^2}{T}.
$$

Therefore, to obtain an ϵ-suboptimal solution, we need $\mathcal{O}(1/\epsilon)$ iterations. Let $T =$

$\sum_{j=1}^{I-1} p_i + K = p_I + K - 1$, then $I \leq \log_2(T) + 1$, and thus IFO complexity per worker is

$$\begin{aligned} IFO &\leq \sum_{i=1}^{I}(n + \sum_{j=2}^{p_i} S_{i,k}) \\ &\leq \sum_{i=1}^{I}(n + \frac{2^{2(i-1)}}{M}) \\ &\leq nI + 2^{2I}/M \\ &\leq [\log_2(T) + 1]N/M + 4T^2/M \\ &= \mathcal{O}(\frac{N\ln(1/\epsilon) + 1/\epsilon^2}{M}). \end{aligned}$$

4.7.3 Proof of Theorem 6 and Corollary 4.3.2

The proof is quite similar to that of Theorem 5.

We first need to upper bound $\mathbb{E}[\|\nabla f(x_{i,k}) - \bar{g}_{i,k}\|^2]$. Eqs. (4.7) and (4.10) still hold. Similarly, we also have for $k \geq 2$,

$$\begin{aligned} \mathbb{E}[\|f(x_{i,k}) - g_{i,k}\|^2] &\leq \frac{L^2 D^2 \eta_{i,k-1}^2}{MS_{i,k}} + \mathbb{E}[\|f(x_{i,k-1}) - g_{i,k-1}\|^2 \\ &= \frac{L^2 D^2 \eta_{i,k-1}^2}{p_i} + \mathbb{E}[\|f(x_{i,k-1}) - g_{i,k-1}\|^2. \end{aligned}$$

For $k = 1$,

$$\mathbb{E}[\|f(x_{i,k}) - g_{i,k}\|^2] \leq \frac{\sigma^2}{MS_{i,1}} = \frac{\sigma^2}{M\frac{\sigma^2 p_i^2}{ML^2D^2}} = \frac{L^2D^2}{p_i^2}.$$

So

$$\mathbb{E}[\|f(x_{i,k}) - g_{i,k}\|^2] \leq \frac{L^2D^2}{p_i^2} + \frac{L^2D^2}{p_i}\sum_{j=2}^{k}\eta_{i,j-1}^2 \leq \frac{L^2D^2}{p_i^2} + \frac{4L^2D^2}{p_i^2} = \frac{5L^2D^2}{p_i^2}. \quad (4.11)$$

Combine Eqs. (4.7), (4.10) and (4.11), we have

$$\mathbb{E}[\|f(x_{i,k}) - \bar{g}_{i,k}\|^2] \leq \frac{13L^2D^2 + 2G_\infty^2}{p_i^2} \triangleq \frac{C_2^2}{p_i^2}.$$

Applying Lemma 9, we have

$$\mathbb{E}[f(x_{i,k+1})] - f(x^*) \leq \frac{4C_2D + 2LD^2}{p_i + k}.$$

By definition, $x_{i,k+1} = x_t$, where $t = \sum_{j=1}^{i-1} p_j + k + 1 = p_i + k$. When $t = T$, we have

$$\mathbb{E}[f(x_T)] - f(x^*) \leq \frac{4C_2D + 2LD^2}{T}.$$

Therefore, to obtain an ϵ-suboptimal solution, we need $\mathcal{O}(1/\epsilon)$ iterations. Let $T = \sum_{j=1}^{I-1} p_i + K = p_I + K - 1$, then $I \leq \log_2(T) + 1$, and thus SFO complexity per worker is

$$\begin{aligned} SFO &\leq \sum_{i=1}^{I}(\frac{\sigma^2 p_i^2}{ML^2D^2} + \sum_{j=2}^{p_i} S_{i,k}) \\ &\leq \sum_{i=1}^{I}(\frac{\sigma^2 2^{2(i-1)}}{ML^2D^2} + \frac{2^{2(i-1)}}{M}) \\ &\leq \frac{2^{2I}}{M}(\frac{\sigma^2}{L^2D^2} + 1) \\ &\leq \frac{4T^2}{M}(\frac{\sigma^2}{L^2D^2} + 1) \\ &= \mathcal{O}(1/(M\epsilon^2)). \end{aligned}$$

4.7.4 Proof of Theorem 7 and Corollary 4.4.1

First, since $x_{t+1} = (1 - \eta_t)x_t + \eta_t v_t$ is a convex combination of x_t, v_t, and $x_1 \in \mathcal{K}, v_t \in \mathcal{K},$ for all t, we can prove $x_t \in \mathcal{K},$ for all t by induction. So $x_o \in \mathcal{K}$.

Then we turn to upper bound $\mathbb{E}[\|\nabla f(x_{i,k}) - \bar{g}_{i,k}\|^2]$. Eq. (4.7) still holds. Similarly, we also have for $k \geq 2$,

$$\begin{aligned}\mathbb{E}[\|f^{(m)}(x_{i,k}) - g_{i,k}^{(m)}\|^2] &\leq \frac{L^2D^2\eta_{i,k-1}^2}{S_{i,k}} + \mathbb{E}[\|f^{(m)}(x_{i,k-1}) - g_{i,k-1}^{(m)}\|^2 \\ &= \frac{L^2D^2T^{-1}}{\frac{\sqrt{n}}{M}} + \mathbb{E}[\|f^{(m)}(x_{i,k-1}) - g_{i,k-1}^{(m)}\|^2 \\ &= \frac{ML^2D^2}{\sqrt{n}T} + \mathbb{E}[\|f^{(m)}(x_{i,k-1}) - g_{i,k-1}^{(m)}\|^2.\end{aligned}$$

For $k = 1$, we have $g_{i,1}^{(m)} = \nabla f^{(m)}(x_{i,1})$. So

$$\mathbb{E}[\|\nabla f^{(m)}(x_{i,k}) - g_{i,k}^{(m)}\|^2] \leq \frac{ML^2D^2}{\sqrt{n}T}(k-1) \leq \frac{ML^2D^2}{\sqrt{n}T}p_i = \frac{ML^2D^2}{T}.$$

By Eq. (4.8),

$$\mathbb{E}[\nabla f(x_{i,k}) - g_{i,k}\|^2] \leq \frac{M\frac{ML^2D^2}{T}}{M^2} = \frac{L^2D^2}{T}. \tag{4.12}$$

We also have

$$\begin{aligned}\mathbb{E}[\|g_{i,k} - \bar{g}_{i,k}\|^2] &\leq \sum_{j=2}^{k}\frac{\eta_{i,j-1}^2 dL^2D^2}{Ms_{1,i,j}^2} + \sum_{j=2}^{k}\frac{\eta_{i,j-1}^2 dL^2D^2}{s_{2,i,j}^2} + \frac{dG_\infty^2}{Ms_{1,i,1}^2} + \frac{dG_\infty^2}{s_{2,i,1}^2} \\ &\leq \frac{p_i dL^2D^2}{TM\frac{d\sqrt{n}}{M}} + \frac{p_i dL^2D^2}{Td\sqrt{n}} + \frac{dG_\infty^2}{M\frac{Td}{M}} + \frac{dG_\infty^2}{dT} \\ &= \frac{2(L^2D^2 + G_\infty^2)}{T}.\end{aligned} \tag{4.13}$$

Combine Eqs. (4.7), (4.12) and (4.13)

$$\mathbb{E}[\|\nabla f(x_{i,k}) - \bar{g}_{i,k}\|^2] \leq \frac{3L^2D^2 + 2G_\infty^2}{T}.$$

By Assumption 10, f is also a bounded (potentially) non-convex function on $\mathcal{K}$ with L-Lipschitz continuous gradient. Specifically, we have $\sup_{x\in\mathcal{K}}|f(x)|\leq M_0$. Note that if we define $v_t' = \arg\min_{v\in\mathcal{K}}\langle v, \nabla f(x_t)\rangle$, then $\mathcal{G}(x_t) = \langle v_t' - x_t, -\nabla f(x_t)\rangle = -\langle v_t' - x_t, \nabla f(x_t)\rangle$. So we have

$$
\begin{aligned}
f(x_{t+1}) &\overset{(a)}{\leq} f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{L}{2}\|x_{t+1} - x_t\|^2 \\
&= f(x_t) + \langle \nabla f(x_t), \eta_t(v_t - x_t) \rangle + \frac{L}{2}\|\eta_t(v_t - x_t)\|^2 \\
&\overset{(b)}{\leq} f(x_t) + \eta_t \langle \nabla f(x_t), v_t - x_t \rangle + \frac{L\eta_t^2 D^2}{2} \\
&= f(x_t) + \eta_t \langle \bar{g}_t, v_t - x_t \rangle + \eta_t \langle \nabla f(x_t) - \bar{g}_t, v_t - x_t \rangle + \frac{L\eta_t^2 D^2}{2} \\
&\overset{(c)}{\leq} f(x_t) + \eta_t \langle \bar{g}_t, v_t' - x_t \rangle + \eta_t \langle \nabla f(x_t) - \bar{g}_t, v_t - x_t \rangle + \frac{L\eta_t^2 D^2}{2} \\
&= f(x_t) + \eta_t \langle \nabla f(x_t), v_t' - x_t \rangle + \eta_t \langle \bar{g}_t - \nabla f(x_t), v_t' - x_t \rangle \\
&\qquad + \eta_t \langle \nabla f(x_t) - \bar{g}_t, v_t - x_t \rangle + \frac{L\eta_t^2 D^2}{2} \\
&= f(x_t) - \eta_t \mathcal{G}(x_t) + \eta_t \langle \nabla f(x_t) - \bar{g}_t, v_t - v_t' \rangle + \frac{L\eta_t^2 D^2}{2} \\
&\overset{(d)}{\leq} f(x_t) - \eta_t \mathcal{G}(x_t) + \eta_t \|\nabla f(x_t) - \bar{g}_t\| \|v_t - v_t'\| + \frac{L\eta_t^2 D^2}{2} \\
&\overset{(e)}{\leq} f(x_t) - \eta_t \mathcal{G}(x_t) + \eta_t D \|\nabla f(x_t) - \bar{g}_t\| + \frac{L\eta_t^2 D^2}{2},
\end{aligned}
$$

where we used the assumption that f has L-Lipschitz continuous gradient in inequality (a). Inequalities (b), (e) hold because of Assumption 7. Inequality (c) is due to the optimality of v_t, and in (d), we applied the Cauchy-Schwarz inequality.

Rearrange the inequality above, we have

$$
\eta_t \mathcal{G}(x_t) \leq f(x_t) - f(x_{t+1}) + \eta_t D \|\nabla f(x_t) - \bar{g}_t\| + \frac{L\eta_t^2 D^2}{2}. \tag{4.14}
$$

Apply Eq. (4.14) recursively for $t = 1, 2, \cdots, T$, and take expectations, we attain the following inequality:

$$
\sum_{t=1}^{T} \eta_t \mathbb{E}[\mathcal{G}(x_t)] \leq f(x_1) - f(x_{T+1}) + D \sum_{t=1}^{T} \eta_t \mathbb{E}[\|\nabla f(x_t) - \bar{g}_t\|] + \frac{LD^2}{2} \sum_{t=1}^{T} \eta_t^2.
$$

Since we have $\mathbb{E}[\|\nabla f(x_{i,k}) - \bar{g}_{i,k}\|^2] \leq \frac{3L^2D^2+2G_\infty^2}{T} \triangleq \frac{c^2}{T}$, we have

$$\mathbb{E}[\|\nabla f(x_t) - \bar{g}_t\|] \leq \sqrt{\mathbb{E}[\|\nabla f(x_t) - \bar{g}_t\|^2]} \leq \frac{c}{\sqrt{T}}.$$

With $\eta_t = T^{-1/2}$, we then have

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{E}[\mathcal{G}(x_t)] &\leq \sqrt{T}[f(x_1) - f(x_{T+1})] + D\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(x_t) - \bar{g}_t\|] + \sqrt{T}\frac{LD^2}{2}T(T^{-1/2})^2 \\ &\leq 2M_0\sqrt{T} + DT\frac{c}{\sqrt{T}} + \frac{LD^2}{2}\sqrt{T} \\ &= (2M_0 + cD + \frac{LD^2}{2})\sqrt{T}.\end{aligned}$$

So

$$\mathbb{E}[\mathcal{G}(x_o)] = \frac{\sum_{t=1}^{T}\mathbb{E}[\mathcal{G}(x_t)]}{T} \leq \frac{2M_0 + cD + \frac{LD^2}{2}}{\sqrt{T}}.$$

Therefore, in order to find an ϵ-first-order stationary points, we need at most $\mathcal{O}(1/\epsilon^2)$ iterations. The IFO complexity per worker is $[n + 2(p-1)S_{i,k}] \cdot \frac{T}{p} = \mathcal{O}(\sqrt{n}/\epsilon^2) = \mathcal{O}(\sqrt{N}/(\epsilon^2\sqrt{M}))$. The average communication bits per round is $\frac{1}{p}\{M[32 + d(z_{1,i,1}+1) + (p-1)(32+d(z_{1,i,k}+1))] + [32+d(z_{2,i,1}+1)+(p-1)(32+d(z_{2,i,k}+1))]\} = (32+d)(M+1) + \frac{Md}{\sqrt{n}}\log_2(\sqrt{\frac{Td}{M}}+1) + Md\log_2(\frac{d^{1/2}n^{1/4}}{\sqrt{M}}+1) + \frac{d}{\sqrt{n}}\log_2(\sqrt{TD}+1) + d\log_2(d^{1/2}n^{1/4}+1)$.

Chapter 5

Black-Box Submodular Maximization

5.1 Introduction

The focus of this chapter[1] is the *constrained* continuous DR-submodular maximization over a bounded convex body. We aim to design an algorithm that uses only zeroth-order information while avoiding expensive projection operations. Note that one way the optimization methods can deal with constraints is to apply the projection oracle once the proposed iterates land outside the feasibility region. However, computing the projection in many constrained settings is computationally prohibitive (*e.g.*, projection over bounded trace norm matrices, flow polytope, matroid polytope, rotation matrices). In such scenarios, projection-free algorithms, *a.k.a.*, Frank-Wolfe [14], replace the projection with a linear program. Indeed, our proposed algorithm combines efficiently the zeroth-order information with solving a series of linear programs to ensure convergence to a near-optimal solution.

Motivation: Computing the gradient of a continuous DR-submodular function has been shown to be computationally prohibitive (or even intractable) in many applications. For example, the objective function of influence maximization is defined via specific stochastic processes [82, 83] and computing/estimating the gradient of the mutliliear

[1]This chapter is based on our work in [3].

extension would require a relatively high computational complexity. In the problem of D-optimal experimental design , the gradient of the objective function involves inversion of a potentially large matrix [61]. Moreover, when one attacks a submodular recommender model, only black-box information is available and the service provider is unlikely to provide additional first-order information (this is known as the black-box adversarial attack model) [124].

There has been very recent progress on developing zeroth-order methods for constrained optimization problems in convex and non-convex settings [53, 125]. Such methods typically assume the objective function is defined on the whole $\mathbb{R}^d$ so that they can sample points from a proper distribution defined on $\mathbb{R}^d$. For DR-submodular functions, this assumption might be unrealistic, since many DR-submodular functions might be only defined on a subset of $\mathbb{R}^d$, *e.g.*, the multi-linear extension [114], a canonical example of DR-submodular functions, is only defined on a unit cube. Moreover, they can only guarantee to reach a first-order stationary point. However, Hassani et al. [58] showed that for a monotone DR-submodular function, the stationary points can only guarantee $1/2$ approximation to the optimum. Therefore, if a state-of-the-art zeroth-order non-convex algorithm is used for maximizing a monotone DR-submodular function, it is likely to terminate at a suboptimal stationary point whose approximation ratio is only $1/2$.

Our contributions: In this chapter, we propose a derivative-free and projection-free algorithm `Black-Box Continuous Greedy` (BCG), that maximizes a monotone continuous DR-submodular function over a bounded convex body $\mathcal{K} \subseteq \mathbb{R}^d$. We consider three scenarios:

(1) In the deterministic setting, where function evaluations can be obtained exactly, BCG achieves the tight $[(1-1/e)OPT-\epsilon]$ approximation guarantee with $\mathcal{O}(d/\epsilon^3)$ function evaluations.

(2) In the stochastic setting, where function evaluations are noisy, BCG achieves the tight $[(1-1/e)OPT-\epsilon]$ approximation guarantee with $\mathcal{O}(d^3/\epsilon^5)$ function evaluations.

Table 5.1: Number of function queries in different settings, where D_1 is the diameter of $\mathcal{K}$.

Function	Function Queries
continuous DR-submodular	$\mathcal{O}(\max\{G, LD_1\}^3 \cdot \frac{d}{\epsilon^3})$ [Theorem 9]
stochastic continuous DR-submodular	$\mathcal{O}(\max\{G, LD_1\}^3 \cdot \frac{d^3}{\epsilon^5})$ [Theorem 10]
discrete submodular	$\mathcal{O}(\frac{d^5}{\epsilon^5})$ [Theorem 11]

(3) In the discrete setting, `Discrete Black-Box Greedy` (DBG), the discrete version of BCG, achieves the tight $[(1 - 1/e)OPT - \epsilon]$ approximation guarantee with $\mathcal{O}(d^5/\epsilon^5)$ function evaluations.

All the theoretical results are summarized in Table 5.1.

We would like to note that in the discrete setting, due to the conservative upper bounds for the Lipschitz and smooth parameters of general multilinear extensions, and the variance of the gradient estimators subject to noisy function evaluations, the required number of function queries in theory is larger than the best known result, $\mathcal{O}(d^{5/2}/\epsilon^3)$ in [8, 59]. However, our experiments (in Section 5.5) show that empirically, our proposed algorithm often requires significantly fewer function evaluations and less running time, while achieving a practically similar utility.

Novelty of our work: All the previous results in constrained DR-submodular maximization assume access to (stochastic) gradients. In this work, we address a harder problem, *i.e.*, we provide the first rigorous analysis when only (stochastic) function values can be obtained. More specifically, with the smoothing trick [126], one can construct an unbiased gradient estimator via function queries. However, this estimator has a large $\mathcal{O}(d^2/\delta^2)$ variance which may cause FW-type methods to diverge. To overcome this issue, we build on the momentum method proposed by Mokhtari et al. [59] in which they assumed access to the *first-order* information.

Given a point x, the smoothed version of F at x is defined as $\mathbb{E}_{v \sim B^d}[F(x + \delta v)]$. If x is close to the boundary of the domain $\mathcal{X}$, $(x + \delta v)$ may fall outside of $\mathcal{X}$, leaving the smoothed function undefined for many instances of DR-submodular functions (*e.g.*, the multilinear extension is only defined over the unit cube). Thus the vanilla smoothing trick will not work. To this end, we transform the domain $\mathcal{X}$ and constraint set $\mathcal{K}$ in a proper way and run our zeroth-order method on the transformed constraint set $\mathcal{K}'$. Importantly, we retrieve the same convergence rate of $\mathcal{O}(T^{-1/3})$ as in [59] with a minimum number of function queries in different settings (continuous, stochastic continuous, discrete).

We further note that by using more recent variance reduction techniques [1], one might be able to reduce the required number of function evaluations. All the proofs in this chapter are provided in Section 5.7.

5.2 Smoothing Trick

For a function F defined on $\mathbb{R}^d$, its δ-smoothed version is given as

$$\tilde{F}_\delta(x) \triangleq \mathbb{E}_{v\sim B^d}[F(x+\delta v)],$$

where v is chosen uniformly at random from the d-dimensional unit ball B^d. In words, the function $\tilde{F}_\delta$ at any point x is obtained by "averaging" F over a ball of radius δ around x. In the sequel, we omit the subscript δ for the sake of simplicity and use $\tilde{F}$ instead of $\tilde{F}_\delta$.

Lemma 10 below shows that under the Lipschitz assumption for F, the smoothed version $\tilde{F}$ is a good approximation of F, and also inherits the key structural properties of F (such as monotonicity and submodularity). Thus one can (approximately) optimize F via optimizing $\tilde{F}$.

Lemma 10. *If F is monotone continuous DR-submodular and G-Lipschitz continuous on $\mathbb{R}^d$, then so is $\tilde{F}$ and*

$$|\tilde{F}(x) - F(x)| \leq \delta G.$$

An important property of $\tilde{F}$ is that one can obtain an unbiased estimation for its gradient $\nabla\tilde{F}$ by a single query of F. This property plays a key role in our proposed derivative-free algorithms.

Lemma 11 (Lemma 6.5 in [127]). *Given a function F on $\mathbb{R}^d$, if we choose u uniformly at random from the $(d-1)$-dimensional unit sphere S^{d-1}, then we have*

$$\nabla \tilde{F}(x) = \mathbb{E}_{u \sim S^{d-1}} \left[\frac{d}{\delta} F(x + \delta u) u \right].$$

5.3 DR-Submodular Maximization

In this chapter, we mainly focus on the constrained optimization problem:

$$\max_{x \in \mathcal{K}} F(x),$$

where F is a monotone continuous DR-submodular function on $\mathbb{R}^d$, and the constraint set $\mathcal{K} \subseteq \mathcal{X} \subseteq \mathbb{R}^d$ is convex and compact.

For *first-order* monotone DR-submodular maximization, one can use `Continuous Greedy` [57, 116], a variant of Frank-Wolfe Algorithm [4, 5, 14], to achieve the $[(1 - 1/e)OPT - \epsilon]$ approximation guarantee. At iteration t, the FW variant first maximizes the linearization of the objective function F:

$$v_t = \arg\max_{v \in \mathcal{K}} \langle v, \nabla F(x_t) \rangle.$$

Then the current point x_t moves in the direction of v_t with a step size $\gamma_t \in (0, 1]$:

$$x_{t+1} = x_t + \gamma_t v_t.$$

Hence, by solving linear optimization problems, the iterates are updated without resorting to the projection oracle.

Here we introduce our main algorithm `Black-Box Continuous Greedy` which assumes access only to function values (*i.e.*, zeroth-order information). This algorithm is partially based on the idea of `Continuous Greedy`. The basic idea is to utilize the function evaluations of F at carefully selected points to obtain unbiased estimations of the gradient of the smoothed version, $\nabla \tilde{F}$. By extending `Continuous Greedy` to the derivative-free setting and using recently proposed variance reduction techniques, we can then optimize $\tilde{F}$ near-optimally. Finally, by Lemma 10 we show that the obtained optimizer also provides a good solution for F.

Recall that continuous DR-submodular functions are defined on a box $\mathcal{X} = \Pi_{i=1}^{n} \mathcal{X}_i$. To simplify the exposition, we can assume, without loss of generality, that the objective function F is defined on $\mathcal{X} \triangleq \prod_{i=1}^{d} [0, a_i]$ [56]. Moreover, we note that since $\tilde{F} = \mathbb{E}_{v \sim B^d}[F(x + \delta v)]$, for x close to $\partial \mathcal{X}$ (the boundary of $\mathcal{X}$), the point $x + \delta v$ may fall outside of $\mathcal{X}$, leaving the function $\tilde{F}$ undefined.

To circumvent this issue, we shrink the domain $\mathcal{X}$ by δ. Precisely, the shrunk domain is defined as

$$\mathcal{X}'_\delta = \{x \in \mathcal{X} | d(x, \partial \mathcal{X}) \geq \delta\}.$$

Since we assume $\mathcal{X} = \prod_{i=1}^{d} [0, a_i]$, the shrunk domain is $\mathcal{X}'_\delta = \prod_{i=1}^{d} [\delta, a_i - \delta]$. Then for all $x \in \mathcal{X}'_\delta$, we have $x + \delta v \in \mathcal{X}$. So $\tilde{F}$ is well-defined on $\mathcal{X}'_\delta$. By Lemma 10, the optimum of $\tilde{F}$ on the shrunk domain $\mathcal{X}'_\delta$ will be close to that on the original domain $\mathcal{X}$, if δ is small enough. Therefore, we can first optimize $\tilde{F}$ on $\mathcal{X}'_\delta$, then approximately optimize $\tilde{F}$ (and thus F) on $\mathcal{X}$. For simplicity of analysis, we also translate the shrunk domain $\mathcal{X}'_\delta$ by $-\delta$, and denote it as $\mathcal{X}_\delta = \prod_{i=1}^{d} [0, a_i - 2\delta]$.

Besides the domain $\mathcal{X}$, we also need to consider the transformation on constraint set $\mathcal{K}$. Intuitively, if there is no translation, we should consider the intersection of $\mathcal{K}$ and the shrunk domain $\mathcal{X}'_\delta$. But since we translate $\mathcal{X}'_\delta$ by $-\delta$, the same transformation should be performed on $\mathcal{K}$. Thus, we define the transformed constraint set as the translated intersection (by $-\delta$)

of $\mathcal{X}'_\delta$ and $\mathcal{K}$:

$$\mathcal{K}' \triangleq (\mathcal{X}'_\delta \cap \mathcal{K}) - \delta\mathbf{1} = \mathcal{X}_\delta \cap (\mathcal{K} - \delta\mathbf{1}).$$

It is well known that the FW Algorithm is sensitive to the accuracy of gradient, and may have arbitrarily poor performance with stochastic gradients [7, 8]. Thus we incorporate two methods of variance reduction into our proposed algorithm `Black-Box Continuous Greedy` which correspond to Step 5 and Step 6 in Algorithm 5, respectively. First, instead of the one-point gradient estimation in Lemma 11, we adopt the two-point estimator of $\nabla\tilde{F}(x)$ [128, 129]:

$$\frac{d}{2\delta}(F(x+\delta u) - F(x-\delta u))u, \tag{5.1}$$

where u is chosen uniformly at random from the unit sphere S^{d-1}.We note that (5.1) is an unbiased gradient estimator with less variance w.r.t. the one-point estimator. We also average over a mini-batch of B_t independently sampled two-point estimators for further variance reduction. The second variance-reduction technique is the momentum method used in [59] to estimate the gradient by a vector $\bar{g}_t$ which is updated at each iteration as follows:

$$\bar{g}_t = (1-\rho_t)\bar{g}_{t-1} + \rho_t g_t.$$

Here ρ_t is a given step size, $\bar{g}_0$ is initialized as an all zero vector $\mathbf{0}$, and g_t is an unbiased estimate of the gradient at iterate x_t. As $\bar{g}_t$ is a weighted average of previous gradient approximation $\bar{g}_{t-1}$ and the newly updated stochastic gradient g_t, it has a lower variance compared with g_t. Although $\bar{g}_t$ is not an unbiased estimation of the true gradient, the error of it will approach zero as time proceeds. The detailed description of `Black-Box Continuous Greedy` is provided in Algorithm 5.

Theorem 9. *For a monotone continuous DR-submodular function F, which is also G-Lipschitz continuous and L-smooth on a convex and compact constraint set $\mathcal{K}$, if we set*

Algorithm 5 `Black-Box Continuous Greedy`

Input: constraint set $\mathcal{K}$, iteration number T, radius δ, step size ρ_t, batch size B_t

Output: $x_{T+1} + \delta \mathbf{1}$

1: $x_1 \leftarrow \mathbf{0}$, $\bar{g}_0 \leftarrow \mathbf{0}$

2: **for** $t = 1, \cdots, T$ **do**

3: Sample $u_{t,1}, \ldots, u_{t,B_t}$ i.i.d. from S^{d-1}

4: For $i = 1$ to B_t, let $y_{t,i}^{+} \leftarrow \delta\mathbf{1} + x_t + \delta u_{t,i}, y_{t,i}^{-} \leftarrow \delta\mathbf{1} + x_t - \delta u_{t,i}$ and evaluate $F(y_{t,i}^{+}), F(y_{t,i}^{-})$

5: $g_t \leftarrow \frac{1}{B_t} \sum_{i=1}^{B_t} \frac{d}{2\delta}[F(y_{t,i}^{+}) - F(y_{t,i}^{-})] u_{t,i}$

6: $\bar{g}_t \leftarrow (1-\rho_t)\bar{g}_{t-1} + \rho_t g_t$

7: $v_t \leftarrow \arg\max_{v \in \mathcal{K}'} \langle v, \bar{g}_t \rangle$

8: $x_{t+1} \leftarrow x_t + \frac{v_t}{T}$

9: **end for**

10: Output $x_{T+1} + \delta\mathbf{1}$

$\rho_t = 2/(t+3)^{2/3}$ *in Algorithm 5, then we have*

$$(1-1/e)F(x^*) - \mathbb{E}[F(x_{T+1} + \delta\mathbf{1})] \leq \frac{3D_1 Q^{1/2}}{T^{1/3}} + \frac{LD_1^2}{2T} + \delta G(1 + (\sqrt{d}+1)(1-1/e)),$$

where $Q = \max\{4^{2/3}G^2, 4cdG^2/B_t + 6L^2D_1^2\}$, c *is a constant,* $D_1 = \operatorname{diam}(\mathcal{K}')$*, and* x^* *is the global maximizer of* F *on* $\mathcal{K}$*.*

Remark 5. *By setting* $T = \mathcal{O}(1/\epsilon^3)$*,* $B_t = d$*, and* $\delta = \epsilon/\sqrt{d}$*, the error term is guaranteed to be at most* $\mathcal{O}(\epsilon)$*. Also, the total number of function evaluations is at most* $\mathcal{O}(d/\epsilon^3)$*.*

We can also extend Algorithm 5 to the stochastic case in which we obtain information about F only through its noisy function evaluations $\hat{F}(x) = F(x) + \xi$, where ξ is stochastic zero-mean noise. In particular, in Step 4 of Algorithm 5, we obtain independent stochastic function evaluations $\hat{F}(y_{t,i}^{+})$ and $\hat{F}(y_{t,i}^{-})$, instead of the exact function values $F(y_{t,i}^{+})$ and $F(y_{t,i}^{-})$. For unbiased function evaluation oracles with uniformly bounded variance, we have the following theorem.

Theorem 10. *Under the condition of Theorem 9, if we further assume that for all* x*,*

$\mathbb{E}[\hat{F}(x)] = F(x)$ *and* $\mathbb{E}[|\hat{F}(x) - F(x)|^2] \leq \sigma_0^2$*, then we have*

$$(1 - 1/e)F(x^*) - \mathbb{E}[F(x_{T+1} + \delta\mathbf{1})] \leq \frac{3D_1Q^{1/2}}{T^{1/3}} + \frac{LD_1^2}{2T} + \delta G(1 + (\sqrt{d} + 1)(1 - 1/e)),$$

where $D_1 = \mathrm{diam}(\mathcal{K}'), Q = \max\{4^{2/3}G^2, 6L^2D_1^2 + (4cdG^2 + 2d^2\sigma_0^2/\delta^2)/B_t\}, c$ *is a constant, and* x^* *is the global maximizer of* F *on* $\mathcal{K}$*.*

Remark 6. *By setting* $T = \mathcal{O}(1/\epsilon^3)$*,* $B_t = d^3/\epsilon^2$*, and* $\delta = \epsilon/\sqrt{d}$*, the error term is at most* $\mathcal{O}(\epsilon)$*. The total number of evaluations is at most* $\mathcal{O}(d^3/\epsilon^5)$*.*

5.4 Discrete Submodular Maximization

In this section, we describe how `Black-Box Continuous Greedy` can be used to solve a discrete submodular maximization problem with a general matroid constraint, *i.e.*, $\max_{S \in \mathcal{I}} f(S)$, where f is a monotone submodular set function and $\mathcal{I}$ is the matroid constraint.

In combinatorics, the matroid is an analogue to the notion of linear independence in linear algebra. Precisely, consider a ground set Ω and a family of subsets of Ω denoted as $\mathcal{I}$. We say the pair $(\Omega, \mathcal{I})$ is a matroid[2] if

1. $\emptyset \in \mathcal{I}$.

2. For each $A \in \mathcal{I}$, if $A' \subseteq A$, then $A' \in \mathcal{I}$.

3. If $A \in \mathcal{I}, B \in \mathcal{I}, |A| > |B|$, then $\exists x \in A \setminus B$, such that $\{x\} \cup B \in \mathcal{I}$.

For any monotone submodular set function $f : 2^\Omega \to \mathbb{R}_{\geq 0}$, its multilinear extension

[2]For a detailed review of matroid theory, interested readers refer to [130].

$F : [0,1]^d \to \mathbb{R}_{\geq 0}$, defined as

$$F(x) = \sum_{S \subseteq \Omega} f(S) \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j),$$

is monotone and DR-submodular [116]. Here, $d = |\Omega|$ is the size of the ground set Ω. Equivalently, we have $F(x) = \mathbb{E}_{S \sim x}[f(S)]$, where $S \sim x$ means that the each element $i \in \Omega$ is included in S with probability x_i independently.

It can be shown that in lieu of solving the discrete optimization problem one can solve the continuous optimization problem $\max_{x \in \mathcal{K}} F(x)$, where $\mathcal{K} = \text{conv}\{1_I : I \in \mathcal{I}\}$ is the matroid polytope [116]. This equivalence is obtained by showing that (i) the optimal values of the two problems are the same, and (ii) for any fractional vector $x \in \mathcal{K}$ we can deploy efficient, lossless rounding procedures that produce a set $S \in \mathcal{I}$ such that $\mathbb{E}[f(S)] \geq F(x)$ (*e.g.*, pipage rounding [116, 131] and contention resolution [117]). So we can view $\tilde{F}$ as the underlying function that we intend to optimize, and invoke `Black-Box Continuous Greedy`. As a result, we want that F is G-Lipschitz and L-smooth as in Theorem 9. The following lemma shows these properties are satisfied automatically if f is bounded.

Lemma 12. *For a submodular set function f defined on Ω with $\sup_{X \subseteq \Omega} |f(X)| \leq M$, its multilinear extension F is $2M\sqrt{d}$-Lipschitz and $4M\sqrt{d(d-1)}$-smooth.*

We note that the bounds for Lipschitz and smoothness parameters actually depend on the norms that we consider. However, different norms are equivalent up to a factor that may depend on the dimension. If we consider another norm, some dimension factors may be absorbed into the norm. Therefore, we only study Euclidean norm in Lemma 12.

We further note that computing the exact value of F is difficult as it requires evaluating f over all the subsets $S \in \Omega$. However, one can construct an unbiased estimate for the value $F(x)$ by simply sampling a random set $S \sim x$ and returning $f(S)$ as the estimate. We present our algorithm in detail in Algorithm 6, where we have $\mathcal{X} = [0,1]^d$, since F is

Algorithm 6 `Discrete Black-Box Greedy`

Input: matroid constraint $\mathcal{I}$, transformed constraint set $\mathcal{K}' = \mathcal{X}_\delta \cap (\mathcal{K} - \delta\mathbf{1})$ where $\mathcal{K} = \text{conv}\{1_I : I \in \mathcal{I}\}$, number of iterations T, radius δ, step size ρ_t, batch size B_t, sample size $S_{t,i}$

Output: X_{T+1}

1: $x_1 \leftarrow \mathbf{0}$, $\bar{g}_0 \leftarrow \mathbf{0}$,
2: **for** $t = 1, \cdots, T$ **do**
3: Sample $u_{t,1}, \ldots, u_{t,B_t}$ i.i.d. from S^{d-1}
4: For $i = 1$ to B_t, let $y^+_{t,i} \leftarrow \delta\mathbf{1} + x_t + \delta u_{t,i}, y^-_{t,i} \leftarrow \delta\mathbf{1} + x_t - \delta u_{t,i}$, independently sample subsets $Y^+_{t,i}$ and $Y^-_{t,i}$ for $S_{t,i}$ times according to $y^+_{t,i}, y^-_{t,i}$, get sampled subsets $Y^+_{t,i,j}, Y^-_{t,i,j}$, for all $j \in [S_{t,i}]$, evaluate the function values $f(Y^+_{t,i,j}), f(Y^-_{t,i,j})$, for all $j \in [S_{t,i}]$, and calculate the averages $\bar{f}^+_{t,i} \leftarrow \frac{\sum_{j=1}^{S_{t,i}} f(Y^+_{t,i,j})}{S_{t,i}}, \bar{f}^-_{t,i} \leftarrow \frac{\sum_{j=1}^{S_{t,i}} f(Y^-_{t,i,j})}{S_{t,i}}$
5: $g_t \leftarrow \frac{1}{B_t}\sum_{i=1}^{B_t} \frac{d}{2\delta}(\bar{f}^+_{t,i} - \bar{f}^-_{t,i})u_{t,i}$
6: $\bar{g}_t \leftarrow (1-\rho_t)\bar{g}_{t-1} + \rho_t g_t$
7: $v_t \leftarrow \arg\max_{v\in\mathcal{K}'}\langle v, \bar{g}_t\rangle$
8: $x_{t+1} \leftarrow x_t + \frac{v_t}{T}$
9: **end for**
10: Output $X_{T+1} = \text{round}(x_{T+1} + \delta\mathbf{1})$

defined on $[0,1]^d$, and thus $\mathcal{X}_\delta = [0, 1-2\delta]^d$. We state the theoretical result formally in Theorem 11.

Theorem 11. *For a monotone submodular set function f with $\sup_{X\subseteq\Omega}|f(X)|\leq M$, if we set $\rho_t = 2/(t+3)^{2/3}, S_{t,i} = l$ in Algorithm 6, then we have*

$$\begin{aligned}&(1-1/e)f(X^*) - \mathbb{E}[f(X_{T+1})]\\ \leq\ &\frac{3D_1Q^{1/2}}{T^{1/3}} + \frac{2M\sqrt{d(d-1)}D_1^2}{T} + 2M\delta\sqrt{d}(1+(\sqrt{d}+1)(1-1/e)),\end{aligned}$$

where $D_1 = \text{diam}(\mathcal{K}')$, $Q = \max\{\frac{2d^2M^2(\frac{1}{l\delta^2}+8c)}{B_t} + 96d(d-1)M^2D_1^2, 4^{5/3}dM^2\}, c$ is a constant, X^* is the global maximizer of f under matroid constraint $\mathcal{I}$.

Remark 7. *By setting $T = \mathcal{O}(d^3/\epsilon^3)$, $B_t = 1, l = d^2/\epsilon^2$, and $\delta = \epsilon/d$, the error term is at most $\mathcal{O}(\epsilon)$. The total number of evaluations is at most $\mathcal{O}(d^5/\epsilon^5)$.*

We note that in Algorithm 6, $\bar{f}^+_{t,i}$ is the unbiased estimation of $F(y^+_{t,i})$, and the same holds for $\bar{f}^-_{t,i}$ and $F(y^-_{t,i})$. As a result, we can analyze the algorithm under the framework of stochastic continuous submodular maximization. By applying Theorem 10, Lemma 12, and the facts $\mathbb{E}[|\bar{f}_{t,i}^{+} - F(y_{t,i}^{+})|^2] \leq M^2/S_{t,i}, \mathbb{E}[|\bar{f}_{t,i}^{-} - F(y_{t,i}^{-})|^2] \leq M^2/S_{t,i}$ directly, we can also attain Theorem 11.

5.5 Experiments

In this section, we will compare `Black-Box Continuous Greedy` (BCG) and `Discrete Black-Box Greedy` (DBG) with the following baselines:

(1) `Zeroth-Order Gradient Ascent` (ZGA) is the projected gradient ascent algorithm equipped with the same two-point gradient estimator as BCG uses. Therefore, it is a *zeroth-order* projected algorithm.

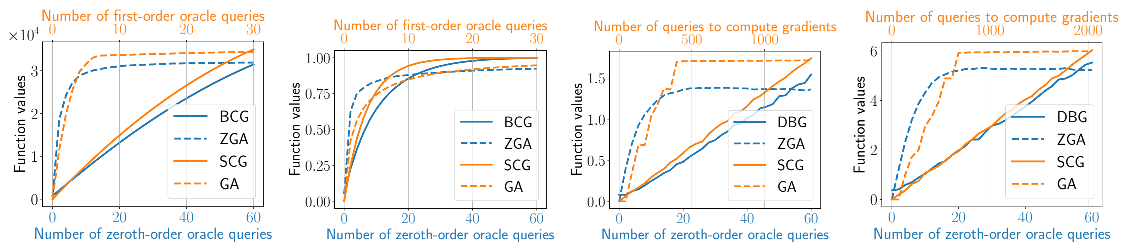
(2) `Stochastic Continuous Greedy` (SCG) is the state-of-the-art *first-order* algorithm for maximizing continuous DR-submodular functions [8, 59]. Note that it is a projection-free algorithm.

(3) `Gradient Ascent` (GA) is the *first-order* projected gradient ascent algorithm [58].

The stopping criterion for the algorithms is whenever a given number of iterations is achieved. Moreover, the batch sizes $S_{t,i}$ in Algorithm 5 and B_t in Algorithm 6 are both 1. Therefore, in the experiments, DBG uses 1 query per iteration while SCG uses $\mathcal{O}(d)$ queries.

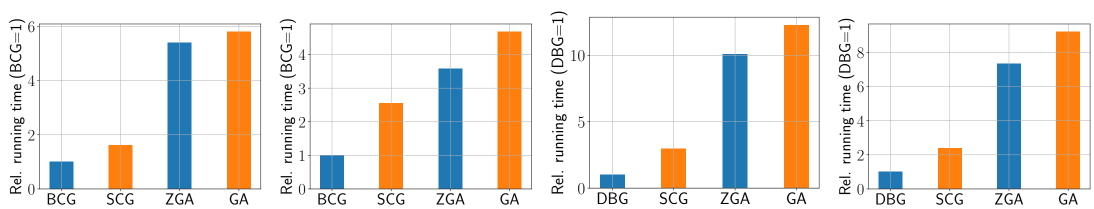
We perform four sets of experiments which are described in detail in the following. The first two sets of experiments are maximization of continuous DR-submodular functions, which `Black-Box Continuous Greedy` is designed to solve. The last two are submodular set maximization problems. We will apply `Discrete Black-Box Greedy` to solve these problems. The function values at different rounds and the execution times are presented in Figs. 5.1 and 5.2. The first-order algorithms (SCG and GA) are marked in orange, and the zeroth-order algorithms are marked in blue.

Non-convex/non-concave Quadratic Programming (NQP): In this set of experi-



(a) NQP (b) Topic summarization (c) Active set selection (d) Influence maximization

Figure 5.1: Function value vs. number of oracle queries. Note that every chart has dual horizontal axes. Orange lines use the orange horizontal axes above while blue lines use the blue ones below.



(a) NQP (b) Topic summarization (c) Active set selection (d) Influence maximization

Figure 5.2: Relative running time normalized with respect to BCG (for continuous DR-submodular maximization in the first two sets of experiments) and DBG (for submodular set maximization in the last two sets of experiments).

ments, we apply our proposed algorithm and the baselines to the problem of non-convex/non-concave quadratic programming. The objective function is of the form $F(x) = \frac{1}{2}x^\top Hx + b^\top x$, where x is a 100-dimensional vector, H is a 100-by-100 matrix, and every component of H is an i.i.d. random variable whose distribution is equal to that of the negated absolute value of a standard normal distribution. The constraints are $\sum_{i=1}^{30} x_i \leq 30$, $\sum_{i=31}^{60} x_i \leq 20$, and $\sum_{i=61}^{100} x_i \leq 20$. To guarantee that the gradient is non-negative, we set $b_t = -H^\top \mathbf{1}$. One can observe from Fig. 5.1a that the function value that BCG attains is only slightly lower than that of the first-order algorithm SCG. The final function value that BCG attains is similar to that of ZGA.

Topic Summarization: Next, we consider the topic summarization problem [107, 132], which is to maximize the probabilistic coverage of selected articles on news topics. Each news article is characterized by its topic distribution, which is obtained by applying latent

Dirichlet allocation to the corpus of Reuters-21578, Distribution 1.0. The number of topics is set to 10. We will choose from 120 news articles. The probabilistic coverage of a subset of news articles (denoted by X) is defined by $f(X) = \frac{1}{10}\sum_{j=1}^{10}[1 - \prod_{a\in X}(1 - p_a(j))]$, where $p_a(\cdot)$ is the topic distribution of article a. The multilinear extension function of f is $F(x) = \frac{1}{10}\sum_{j=1}^{10}[1 - \prod_{a\in\Omega}(1 - p_a(j)x_a)]$, where $x \in [0,1]^{120}$ [133]. The constraint is $\sum_{i=1}^{40} x_i \leq 25$, $\sum_{i=41}^{80} x_i \leq 30$, $\sum_{i=81}^{120} x_i \leq 35$. It can be observed from Fig. 5.1b that the proposed BCG algorithm achieves the same function value as the first-ordered algorithm SCG and outperforms the other two. As shown in Fig. 5.2a, BCG is the most efficient method. The two projection-free algorithms BCG and SCG run faster than the projected methods ZGA and GA. We will elaborate on the running time later in this section.

Active Set Selection: We study the active set selection problem that arises in Gaussian process regression [134]. We use the *Parkinsons Telemonitoring* dataset, which is composed of biomedical voice measurements from people with early-stage Parkinson's disease [135]. Let $X \in \mathbb{R}^{n\times d}$ denote the data matrix. Each row $X[i,:]$ is a voice recording while each column $X[:,j]$ denotes an attribute. The covariance matrix Σ is defined by $\Sigma_{ij} = \exp(-\|X[:,i] - X[:,j]\|^2)/h^2$, where h is set to 0.75. The objective function of the active set selection problem is defined by $f(S) = \log\det(I + \Sigma_{S,S})$, where $S \subseteq [d]$ and $\Sigma_{S,S}$ is the principal submatrix indexed by S. The total number of 22 attributes are partitioned into 5 disjoint subsets with sizes 4, 4, 4, 5 and 5, respectively. The problem is subject to a partition matroid requiring that at most one attribute should be active within each subset. Since this is a submodular set maximization problem, in order to evaluate the gradient (*i.e.*, obtain an unbiased estimate of gradient) required by first-order algorithms SCG and GA, it needs $2d$ function value queries. To be precise, the i-th component of gradient is $\mathbb{E}_{S\sim x}[f(S \cup \{i\}) - f(S)]$ and requires two function value queries. It can be observed from Fig. 5.1c that DBG outperforms the other zeroth-order algorithm ZGA. Although its performance is slightly worse than the two first-order algorithms SCG and GA, it require significantly less number of function value queries than the other two first-order methods (as discussed above).

Influence Maximization: In the influence maximization problem, we assume that every node in the network is able to influence all of its one-hop neighbors. The objective of influence maximization is to select a subset of nodes in the network, called the seed set (and denoted by S), so that the total number of influenced nodes, including the seed nodes, is maximized. We choose the social network of Zachary's karate club [136] in this study. The subjects in this social network are partitioned into three disjoint groups, whose sizes are 10, 14, and 10 respectively. The chosen seed nodes should be subject to a partition matroid; *i.e.*, we will select at most two subjects from each of the three groups. Note that this problem is also a submodular set maximization problem. Similar to the situation in the active set selection problem, first-order algorithms need function value queries to obtain an unbiased estimate of gradient. We can observe from Fig. 5.1d that DBG attains a better influence coverage than the other zeroth-order algorithm ZGA. Again, even though SCG and GA achieve a slightly better coverage, due to their first-order nature, they require a significantly larger number of function value queries.

Running Time The running times of the our proposed algorithms and the baselines are presented in Fig. 5.2 for the above-mentioned experimental set-ups. There are two main conclusions. First, the two projection-based algorithms (ZGA and GA) require significantly higher time complexity compared to the projection-free algorithms (BCG, DBG, and SCG), as the projection-based algorithms require solving quadratic optimization problems whereas projection-free ones require solving linear optimization problems which can be solved more efficiently. Second, when we compare first-order and zeroth-order algorithms, we can observe that zeroth-order algorithms (BCG, DBG, and ZGA) run faster than their first-order counterparts (SCG and GA).

Summary The above experiment results show the following major advantages of our method over the baselines including SCG and ZGA.

1. BCG/DBG is at least twice faster than SCG and ZGA in all tasks in terms of running time (Figs. 5.2a to 5.2d).

2. DBG requires remarkably fewer function evaluations in the discrete setting (Figs. 5.1c and 5.1d).

3. In addition to saving function evaluations, BCG/DBG achieves an objective function value comparable to that of the first-order baselines SCG and GA.

Furthermore, we note that the number of first-order queries required by SCG is only half the number required by BCG. However, as is shown in Figs. 5.2a and 5.2b, BCG runs significantly faster than SCG since a zeroth-order evaluation is faster than a first-order one.

In the topic summarization task (Fig. 5.1b), BCG exhibits a similar performance to that of the first-order baselines SCG and GA, in terms of the attained objective function value. In the other three tasks, BCG/DBG runs notably faster while achieving an only slightly inferior function value. Therefore, BCG/DBG is particularly preferable in a large-scale machine learning task and an application where the total number of function evaluations or the running time is subject to a budget.

5.6 Conclusion

In this chapter, we presented `Black-Box Continuous Greedy`, a derivative-free and projection-free algorithm for maximizing a monotone and continuous DR-submodular function subject to a general convex body constraint. We showed that `Black-Box Continuous Greedy` achieves the tight $[(1-1/e)OPT-\epsilon]$ approximation guarantee with $\mathcal{O}(d/\epsilon^3)$ function evaluations. We then extended the algorithm to the stochastic continuous setting and the discrete submodular maximization problem. Our experiments on both synthetic and real data validated the performance of our proposed algorithms. In particular, we observed that `Black-Box Continuous Greedy` practically achieves the same utility as `Continuous Greedy` while being way more efficient in terms of number of function evaluations.

5.7 Proofs

5.7.1 Proof of Lemma 10

Proof. Using the assumption that F is G-Lipschitz continuous, we have

$$\begin{aligned}|\tilde{F}(x) - \tilde{F}(y)| &= |\mathbb{E}_{v\sim B^d}[F(x+\delta v) - F(y+\delta v)]| \\ &\leq \mathbb{E}_{v\sim B^d}[|F(x+\delta v) - F(y+\delta v)|] \\ &\leq \mathbb{E}_{v\sim B^d}[G\|(x+\delta v) - (y+\delta v)\|] \\ &= G\|x-y\|,\end{aligned}$$

and

$$\begin{aligned}|\tilde{F}(x) - F(x)| &= |\mathbb{E}_{v\sim B^d}[F(x+\delta v) - F(x)]| \\ &\leq \mathbb{E}_{v\sim B^d}[|F(x+\delta v) - F(x)|] \\ &\leq \mathbb{E}_{v\sim B^d}[G\delta\|v\|] \\ &\leq \delta G.\end{aligned}$$

If F is G-Lipschitz continuous and monotone continuous DR-submodular, then F is differentiable. For all $x \leq y$, we also have

$$\nabla F(x) \geq \nabla F(y),$$

and

$$F(x) \leq F(y).$$

By definition of $\tilde{F}$, we have $\tilde{F}$ is differentiable and for for all $x \leq y$,

$$\begin{aligned}
\nabla \tilde{F}(x) - \nabla \tilde{F}(y) &= \nabla \mathbb{E}_{v \sim B^d}[F(x + \delta v)] - \nabla \mathbb{E}_{v \sim B^d}[F(y + \delta v)] \\
&= \mathbb{E}_{v \sim B^d}[\nabla F(x + \delta v) - \nabla F(y + \delta v)] \\
&\geq \mathbb{E}_{v \sim B^d}[0] \\
&= 0,
\end{aligned}$$

and

$$\begin{aligned}
\tilde{F}(x) - \tilde{F}(y) &= \mathbb{E}_{v \sim B^d}[F(x + \delta v)] - \mathbb{E}_{v \sim B^d}[F(y + \delta v)] \\
&= \mathbb{E}_{v \sim B^d}[F(x + \delta v) - F(y + \delta v)] \\
&\leq \mathbb{E}_{v \sim B^d}[0] \\
&= 0,
\end{aligned}$$

i.e., $\nabla \tilde{F}(x) \geq \nabla \tilde{F}(y), \tilde{F}(x) \leq \tilde{F}(y)$. So $\tilde{F}$ is also a monotone continuous DR-submodular function. □

5.7.2 Proof of Theorem 9

In order to prove Theorem 9, we need the following variance reduction lemmas [129, 137], where the second one is a slight improvement of Lemma 2 in [59] and Lemma 5 in [8].

Lemma 13 (Lemma 10 of [129]). *It holds that*

$$\mathbb{E}_{u\sim S^{d-1}}[\frac{d}{2\delta}(F(z+\delta u)-F(z-\delta u))u|z]=\nabla\tilde{F}(z),$$

$$\mathbb{E}_{u\sim S^{d-1}}[\|\frac{d}{2\delta}(F(z+\delta u)-F(z-\delta u))u-\nabla\tilde{F}(z)\|^2|z]\leq cdG^2,$$

where c is a constant.

Lemma 14 (Theorem 3 of [137]). *Let $\{a_t\}_{t=0}^T$ be a sequence of points in $\mathbb{R}^n$ such that $\|a_t-a_{t-1}\|\leq G_0/(t+s)$ for all $1\leq t\leq T$ with fixed constants $G_0\geq 0$ and $s\geq 3$. Let $\{\tilde{a}_t\}_{t=1}^T$ be a sequence of random variables such that $\mathbb{E}[\tilde{a}_t|\mathcal{F}_{t-1}]=a_t$ and $\mathbb{E}[\|\tilde{a}_t-a_t\|^2|\mathcal{F}_{t-1}]\leq\sigma^2$ for every $t\geq 0$, where $\mathcal{F}_{t-1}$ is the σ-field generated by $\{\tilde{a}_i\}_{i=1}^t$ and $\mathcal{F}_0=\varnothing$. Let $\{d_t\}_{t=0}^T$ be a sequence of random variables where d_0 is fixed and subsequent d_t are obtained by the recurrence*

$$d_t=(1-\rho_t)d_{t-1}+\rho_t\tilde{a}_t$$

with $\rho_t=\frac{2}{(t+s)^{2/3}}$. Then, we have

$$\mathbb{E}[\|a_t-d_t\|^2]\leq\frac{Q}{(t+s+1)^{2/3}},$$

where $Q\triangleq\max\{\|a_0-d_0\|^2(s+1)^{2/3},4\sigma^2+3G_0^2/2\}$.

Now we turn to prove Theorem 9.

Proof of Theorem 9. First of all, we note that technically we need the iteration number $T\geq 4$, which always holds in practical applications.

Then we show that for all $t=1,\dots,T+1$, $x_t\in\mathcal{X}_\delta$. By the definition of x_t, we have $x_t=\sum_{i=1}^{t-1}\frac{v_i}{T}$. Since v_t's are non-negative vectors, we know that x_t's are also non-negative vectors and that $0=x_1\leq x_2\leq\ldots\leq x_{T+1}$. It suffices to show that $x_{T+1}\in\mathcal{X}_\delta$. Since x_{T+1} is a convex combination of $v_1,\dots,v_T$ and v_t's are in $\mathcal{X}_\delta$, we conclude that $x_{T+1}\in\mathcal{X}_\delta$.

In addition, since v_t's are also in $\mathcal{K} - \delta\mathbf{1}$, x_{T+1} is also in $\mathcal{K} - \delta\mathbf{1}$. Therefore our final choice $x_{T+1} + \delta\mathbf{1}$ resides in the constraint $\mathcal{K}$.

Let $z_t \triangleq x_t + \delta\mathbf{1}$ and the shrunk domain (without translation) $\mathcal{X}'_\delta \triangleq \mathcal{X}_\delta + \delta\mathbf{1} = \prod_{i=1}^d [\delta, a_i - \delta] \subseteq \mathcal{X}$. By Jensen's inequality and the fact F has L-Lipschitz continuous gradients, we have

$$\|\nabla\tilde{F}(x) - \nabla\tilde{F}(y)\| \leq L\|x - y\|.$$

Thus,

$$\begin{aligned}
\tilde{F}(z_{t+1}) - \tilde{F}(z_t) &= \tilde{F}(z_t + \frac{v_t}{T}) - \tilde{F}(z_t) \\
&\geq \frac{1}{T}\nabla\tilde{F}(z_t)^\top v_t - \frac{L}{2T^2}\|v_t\|^2 \\
&\geq \frac{1}{T}\nabla\tilde{F}(z_t)^\top v_t - \frac{L}{2T^2}D_1^2 \\
&= \frac{1}{T}\left(\bar{g}_t^\top v_t + (\nabla\tilde{F}(z_t) - \bar{g}_t)^\top v_t\right) - \frac{L}{2T^2}D_1^2.
\end{aligned} \tag{5.2}$$

Let $x_\delta^* \triangleq \arg\max_{x \in \mathcal{X}'_\delta \cap \mathcal{K}} \tilde{F}(x)$. Since $x_\delta^*, z_t \in \mathcal{X}'_\delta$, we have $v_t^* \triangleq (x_\delta^* - z_t) \vee 0 \in \mathcal{X}_\delta$. We know $z_t + v_t^* = x_\delta^* \vee z_t \in \mathcal{X}'_\delta$ and

$$v_t^* + \delta\mathbf{1} = (x_\delta^* - x_t) \vee \delta\mathbf{1} \leq x_\delta^*.$$

Since we assume that F is monotone continuous DR-submodular, by Lemma 10, $\tilde{F}$ is also monotone continuous DR-submodular. As a result, $\tilde{F}$ is concave along non-negative directions, and $\nabla\tilde{F}$ is entry-wise non-negative. Thus we have

$$\begin{aligned}
\tilde{F}(z_t + v_t^*) - \tilde{F}(z_t) &\leq \nabla\tilde{F}(z_t)^\top v_t^* \\
&\leq \nabla\tilde{F}(z_t)^\top (x_\delta^* - \delta\mathbf{1}).
\end{aligned}$$

Since $x_\delta^* - \delta\mathbf{1} \in \mathcal{K}'$, we deduce

$$\begin{aligned}
\bar{g}_t^\top v_t &\geq \bar{g}_t^\top (x_\delta^* - \delta\mathbf{1}) \\
&= \nabla\tilde{F}(z_t)^\top (x_\delta^* - \delta\mathbf{1}) + (\bar{g}_t - \nabla\tilde{F}(z_t))^\top (x_\delta^* - \delta\mathbf{1}) \\
&\geq \tilde{F}(z_t + v_t^*) - \tilde{F}(z_t) + (\bar{g}_t - \nabla\tilde{F}(z_t))^\top (x_\delta^* - \delta\mathbf{1}) \\
&\geq \tilde{F}(x_\delta^*) - \tilde{F}(z_t) + (\bar{g}_t - \nabla\tilde{F}(z_t))^\top (x_\delta^* - \delta\mathbf{1}).
\end{aligned}$$

Therefore, we obtain

$$\bar{g}_t^\top v_t + (\nabla\tilde{F}(z_t) - \bar{g}_t)^\top v_t \geq \tilde{F}(x_\delta^*) - \tilde{F}(z_t) + (\nabla\tilde{F}(z_t) - \bar{g}_t)^\top (v_t - (x_\delta^* - \delta\mathbf{1})). \quad (5.3)$$

By plugging Eq. (5.3) into Eq. (5.2), after re-arrangement of the terms, we obtain

$$h_{t+1} \leq (1 - \frac{1}{T})h_t + \frac{1}{T}(\nabla\tilde{F}(z_t) - \bar{g}_t)^\top ((x_\delta^* - \delta\mathbf{1}) - v_t) + \frac{L}{2T^2}D_1^2, \quad (5.4)$$

where $h_t \triangleq \tilde{F}(x_\delta^*) - \tilde{F}(z_t)$. Next we derive an upper bound for $(\nabla\tilde{F}(z_t) - \bar{g}_t)^\top ((x_\delta^* - \delta\mathbf{1}) - v_t)$. By Young's inequality, it can be deduced that for any $\beta_t > 0$,

$$\begin{aligned}
(\nabla\tilde{F}(z_t) - \bar{g}_t)^\top ((x_\delta^* - \delta\mathbf{1}) - v_t) &\leq \frac{\beta_t}{2}\|\nabla\tilde{F}(z_t) - \bar{g}_t\|^2 + \frac{1}{2\beta_t}\|(x_\delta^* - \delta\mathbf{1}) - v_t\|^2 \\
&\leq \frac{\beta_t}{2}\|\nabla\tilde{F}(z_t) - \bar{g}_t\|^2 + \frac{1}{2\beta_t}D_1^2.
\end{aligned} \quad (5.5)$$

Now let $\mathcal{F}_1 \triangleq \varnothing$ and $\mathcal{F}_t$ be the σ-field generate by $\{\bar{g}_1, \ldots, \bar{g}_{t-1}\}$, then by Lemma 13, we have

$$\mathbb{E}[\frac{d}{2\delta}(F(y_{t,i}^+) - F(y_{t,i}^-))u_{t,i}|\mathcal{F}_{t-1}] = \nabla\tilde{F}(z_t),$$

and

$$\mathbb{E}[\|\frac{d}{2\delta}(F(y_{t,i}^+) - F(y_{t,i}^-))u_{t,i} - \nabla\tilde{F}(z_t)\|^2|\mathcal{F}_{t-1}] \leq cdG^2.$$

Therefore,

$$\mathbb{E}[g_t|\mathcal{F}_{t-1}] = \mathbb{E}[\frac{1}{B_t}\sum_{i=1}^{B_t}\frac{d}{2\delta}(F(y_{t,i}^+) - F(y_{t,i}^-))u_{t,i}|\mathcal{F}_{t-1}]$$
$$= \nabla\tilde{F}(z_t),$$

and

$$\mathbb{E}[\|g_t - \nabla\tilde{F}(z_t)\|^2|\mathcal{F}_{t-1}] = \frac{1}{B_t^2}\sum_{i=1}^{B_t}\mathbb{E}[\|\frac{d}{2\delta}(F(y_{t,i}^+) - F(y_{t,i}^-))u_{t,i} - \nabla\tilde{F}(z_t)\|^2|\mathcal{F}_{t-1}]$$
$$\leq \frac{cdG^2}{B_t}. \tag{5.6}$$

By Jensen's inequality and the assumption F is L-smooth, we have

$$\|\nabla\tilde{F}(z_t) - \nabla\tilde{F}(z_{t-1})\| \leq L\frac{D_1}{T} \leq \frac{2LD_1}{t+3}.$$

Then by Lemma 14 with $s = 3, d_t = \bar{g}_t,$ for all $t \geq 0, \tilde{a}_t = g_t, a_t = \nabla\tilde{F}(z_t),$ for all $t \geq 1, a_0 = \nabla\tilde{F}(z_1), G_0 = 2LD_1$, we have

$$\mathbb{E}[\|\nabla\tilde{F}(z_t) - \bar{g}_t\|^2] \leq \frac{Q}{(t+4)^{2/3}}, \tag{5.7}$$

where $Q \triangleq \max\{\|\nabla\tilde{F}(x_1 + \delta\mathbf{1})\|^2 4^{2/3}, \frac{4cdG^2}{B_t} + 6L^2D_1^2\}$. Note that by Lemma 10, we have $\|\nabla\tilde{F}(x)\| \leq G$, thus we can re-define $Q = \max\{4^{2/3}G^2, \frac{4cdG^2}{B_t} + 6L^2D_1^2\}$.

Using Eqs. (5.4), (5.5) and (5.7) and taking expectation, we obtain

$$\mathbb{E}[h_{t+1}] \leq (1 - \frac{1}{T})\mathbb{E}[h_t] + \frac{1}{T}\left(\frac{\beta_t}{2}\cdot\frac{Q}{(t+4)^{2/3}} + \frac{D_1^2}{2\beta_t}\right) + \frac{L}{2T^2}D_1^2$$
$$\leq (1 - \frac{1}{T})\mathbb{E}[h_t] + \frac{D_1Q^{1/2}}{T(t+4)^{1/3}} + \frac{L}{2T^2}D_1^2,$$

where we set $\beta_t = \frac{D_1(t+4)^{1/3}}{Q^{1/2}}$. Using the above inequality recursively, we have

$$\begin{aligned}
\mathbb{E}[h_{T+1}] &\leq (1-\frac{1}{T})^T(\tilde{F}(x_\delta^*) - \tilde{F}(\delta\mathbf{1})) + \sum_{t=1}^{T} \frac{D_1 Q^{1/2}}{T(t+4)^{1/3}} + \frac{L}{2T}D_1^2 \\
&\leq e^{-1}(\tilde{F}(x_\delta^*) - \tilde{F}(\delta\mathbf{1})) + \frac{D_1 Q^{1/2}}{T}\int_0^T \frac{\mathrm{d}x}{(x+4)^{1/3}} + \frac{L}{2T}D_1^2 \\
&\leq e^{-1}(\tilde{F}(x_\delta^*) - \tilde{F}(\delta\mathbf{1})) + \frac{D_1 Q^{1/2}}{T}\frac{3}{2}(T+4)^{2/3} + \frac{L}{2T}D_1^2 \\
&\leq e^{-1}(\tilde{F}(x_\delta^*) - \tilde{F}(\delta\mathbf{1})) + \frac{D_1 Q^{1/2}}{T}\frac{3}{2}(2T)^{2/3} + \frac{L}{2T}D_1^2 \\
&\leq e^{-1}(\tilde{F}(x_\delta^*) - \tilde{F}(\delta\mathbf{1})) + \frac{3D_1 Q^{1/2}}{T^{1/3}} + \frac{LD_1^2}{2T}.
\end{aligned}$$

By re-arranging the terms, we conclude

$$\begin{aligned}
(1-\frac{1}{e})\tilde{F}(x_\delta^*) - \mathbb{E}[\tilde{F}(z_{T+1})] \leq & -e^{-1}\tilde{F}(\delta\mathbf{1}) + \frac{3D_1 Q^{1/2}}{T^{1/3}} + \frac{LD_1^2}{2T} \\
\leq & \frac{3D_1 Q^{1/2}}{T^{1/3}} + \frac{LD_1^2}{2T},
\end{aligned}$$

where the second inequality holds since the image of F is in $\mathbb{R}_+$.

By Lemma 10, we have $\tilde{F}(z_{T+1}) \leq F(z_{T+1}) + \delta G$ and

$$\tilde{F}(x_\delta^*) \geq \tilde{F}(x^*) - \delta G\sqrt{d} \geq F(x^*) - \delta G(\sqrt{d}+1).$$

Therefore,

$$(1-\frac{1}{e})F(x^*) - \mathbb{E}[F(z_{T+1})] \leq \frac{3D_1 Q^{1/2}}{T^{1/3}} + \frac{LD_1^2}{2T} + \delta G(1 + (\sqrt{d}+1)(1-\frac{1}{e})).$$

□

5.7.3 Proof of Theorem 10

Proof. By the unbiasedness of $\hat{F}$ and Lemma 13, we have

$$\begin{aligned}
\mathbb{E}[\frac{d}{2\delta}(\hat{F}(y_{t,i}^{+}) - \hat{F}(y_{t,i}^{-}))u_{t,i}|\mathcal{F}_{t-1}] &= \mathbb{E}[\mathbb{E}[\frac{d}{2\delta}(\hat{F}(y_{t,i}^{+}) - \hat{F}(y_{t,i}^{-}))u_{t,i}|\mathcal{F}_{t-1}, u_{t,i}]|\mathcal{F}_{t-1}] \\
&= \mathbb{E}[\frac{d}{2\delta}(F(y_{t,i}^{+}) - F(y_{t,i}^{-}))u_{t,i}|\mathcal{F}_{t-1}] \\
&= \nabla\tilde{F}(z_t),
\end{aligned}$$

where $z_t = x_t + \delta\mathbf{1}$, and

$$\begin{aligned}
&\mathbb{E}[\|\frac{d}{2\delta}(\hat{F}(y_{t,i}^{+}) - \hat{F}(y_{t,i}^{-}))u_{t,i} - \nabla\tilde{F}(z_t)\|^2|\mathcal{F}_{t-1}] \\
=& \mathbb{E}[\mathbb{E}[\|\frac{d}{2\delta}(F(y_{t,i}^{+}) - F(y_{t,i}^{-}))u_{t,i} - \nabla\tilde{F}(z_t) \\
&+ \frac{d}{2\delta}(\hat{F}(y_{t,i}^{+}) - F(y_{t,i}^{+}))u_{t,i} \\
&- \frac{d}{2\delta}(\hat{F}(y_{t,i}^{-}) - F(y_{t,i}^{-}))u_{t,i}\|^2|\mathcal{F}_{t-1}, u_{t,i}]|\mathcal{F}_{t-1}] \\
=& \mathbb{E}[\mathbb{E}[\|\frac{d}{2\delta}(F(y_{t,i}^{+}) - F(y_{t,i}^{-}))u_{t,i} - \nabla\tilde{F}(z_t)\|^2|\mathcal{F}_{t-1}, u_{t,i}]|\mathcal{F}_{t-1}] \\
&+ \mathbb{E}[\mathbb{E}[\|\frac{d}{2\delta}(\hat{F}(y_{t,i}^{+}) - F(y_{t,i}^{+}))u_{t,i}\|^2|\mathcal{F}_{t-1}, u_{t,i}]|\mathcal{F}_{t-1}] \\
&+ \mathbb{E}[\mathbb{E}[\|\frac{d}{2\delta}(\hat{F}(y_{t,i}^{-}) - F(y_{t,i}^{-}))u_{t,i}\|^2|\mathcal{F}_{t-1}, u_{t,i}]|\mathcal{F}_{t-1}] \\
\leq& \mathbb{E}[\|\frac{d}{2\delta}(F(y_{t,i}^{+}) - F(y_{t,i}^{-}))u_{t,i} - \nabla\tilde{F}(z_t)\|^2|\mathcal{F}_{t-1}] \\
&+ \frac{d^2}{4\delta^2}\mathbb{E}[\mathbb{E}[|\hat{F}(y_{t,i}^{+}) - F(y_{t,i}^{+})|^2 \cdot \|u_{t,i}\|^2|\mathcal{F}_{t-1}, u_{t,i}]|\mathcal{F}_{t-1}] \\
&+ \frac{d^2}{4\delta^2}\mathbb{E}[\mathbb{E}[|\hat{F}(y_{t,i}^{-}) - F(y_{t,i}^{-})|^2 \cdot \|u_{t,i}\|^2|\mathcal{F}_{t-1}, u_{t,i}]|\mathcal{F}_{t-1}] \\
\leq& cdG^2 + \frac{d^2}{4\delta^2}\sigma_0^2 + \frac{d^2}{4\delta^2}\sigma_0^2 \\
=& cdG^2 + \frac{d^2}{2\delta^2}\sigma_0^2.
\end{aligned}$$

Then we have

$$\mathbb{E}[g_t|\mathcal{F}_{t-1}] = \mathbb{E}[\frac{1}{B_t}\sum_{i=1}^{B_t}\frac{d}{2\delta}(\hat{F}(y_{t,i}^{+}) - \hat{F}(y_{t,i}^{-}))u_{t,i}|\mathcal{F}_{t-1}]$$
$$= \nabla\tilde{F}(z_t),$$

and

$$\mathbb{E}[\|g_t - \nabla\tilde{F}(z_t)\|^2|\mathcal{F}_{t-1}] = \frac{1}{B_t^2}\sum_{i=1}^{B_t}\mathbb{E}[\|\frac{d}{2\delta}(\hat{F}(y_{t,i}^{+}) - \hat{F}(y_{t,i}^{-}))u_{t,i} - \nabla\tilde{F}(z_t)\|^2|\mathcal{F}_{t-1}]$$
$$\leq \frac{cdG^2 + \frac{d^2}{2\delta^2}\sigma_0^2}{B_t}.$$

Similar to the proof of Theorem 9, we have

$$\mathbb{E}[\|\nabla\tilde{F}(z_t) - \bar{g}_t\|^2] \leq \frac{Q}{(t+4)^{2/3}},$$

where $Q = \max\{4^{2/3}G^2, 6L^2D_1^2 + \frac{4cdG^2+2d^2\sigma_0^2/\delta^2}{B_t}\}$. Thus we conclude

$$(1-\frac{1}{e})F(x^*) - \mathbb{E}[F(z_{T+1})] \leq \frac{3D_1Q^{1/2}}{T^{1/3}} + \frac{LD_1^2}{2T} + \delta G(1 + (\sqrt{d}+1)(1-\frac{1}{e})).$$

□

5.7.4 Proof of Lemma 12

Proof. Recall that $F(x) = \mathbb{E}_{X\sim x}[f(X)] = \sum_{S\subseteq\Omega} f(S)\prod_{i\in S} x_i \prod_{j\notin S}(1-x_j)$, then for any fixed $i \in [d]$, where $d = |\Omega|$, we have

$$
\begin{aligned}
|\frac{\partial F(x)}{\partial x_i}| &= |\sum_{\substack{S\subseteq\Omega\\ i\in S}} f(S)\prod_{\substack{j\in S\\ j\neq i}} x_j \prod_{\substack{k\notin S\\ k\neq i}}(1-x_k) - \sum_{\substack{S\subseteq\Omega\\ i\notin S}} f(S)\prod_{\substack{j\in S\\ j\neq i}} x_j \prod_{\substack{k\notin S\\ k\neq i}}(1-x_k)| \\
&\leq M[\sum_{\substack{S\subseteq\Omega\\ i\in S}} \prod_{\substack{j\in S\\ j\neq i}} x_j \prod_{\substack{k\notin S\\ k\neq i}}(1-x_k) + \sum_{\substack{S\subseteq\Omega\\ i\notin S}} \prod_{\substack{j\in S\\ j\neq i}} x_j \prod_{\substack{k\notin S\\ k\neq i}}(1-x_k)] \\
&= 2M.
\end{aligned}
$$

So we have

$$\|\nabla F(x)\| \leq 2M\sqrt{d}.$$

Then F is $2M\sqrt{d}$-Lipschitz.

Now we turn to prove that F has Lipschitz continuous gradients. Thanks to the multilinearity, we have

$$\frac{\partial F}{\partial x_i} = F(x|x_i = 1) - F(x|x_i = 0).$$

Since

$$F(x|x_i = 1) = \sum_{\substack{S\subseteq\Omega\\ i\in S}} f(S)\prod_{\substack{j\in S\\ j\neq i}} x_j \prod_{\substack{k\notin S\\ k\neq i}}(1-x_k),$$

we have

$$\frac{\partial F(x|x_i = 1)}{\partial x_i} = 0,$$

and for any fixed $j \neq i$,

$$
\begin{aligned}
|\frac{\partial F(x|x_i=1)}{\partial x_j}| &= |\sum_{\substack{S\subseteq\Omega\\ i,j\in S}} f(S) \prod_{\substack{l\in S\\ l\notin\{i,j\}}} x_l \prod_{\substack{k\notin S\\ k\notin\{i,j\}}} (1-x_k) - \sum_{\substack{S\subseteq\Omega\\ i\in S, j\notin S}} f(S) \prod_{\substack{l\in S\\ l\notin\{i,j\}}} x_l \prod_{\substack{k\notin S\\ k\notin\{i,j\}}} (1-x_k)| \\
&\leq M[\sum_{\substack{S\subseteq\Omega\\ i,j\in S}} \prod_{\substack{l\in S\\ l\notin\{i,j\}}} x_l \prod_{\substack{k\notin S\\ k\notin\{i,j\}}} (1-x_k) + \sum_{\substack{S\subseteq\Omega\\ i\in S, j\notin S}} \prod_{\substack{l\in S\\ l\notin\{i,j\}}} x_l \prod_{\substack{k\notin S\\ k\notin\{i,j\}}} (1-x_k)] \\
&= 2M.
\end{aligned}
$$

Similarly, we have $\frac{\partial F(x|x_i=0)}{\partial x_i} = 0$, and $|\frac{\partial F(x|x_i=0)}{\partial x_j}| \leq 2M$ for $j \neq i$. So we conclude that

$$
|\frac{\partial^2 F}{\partial x_j \partial x_i}| \leq \begin{cases} 0, & \text{if } j = i, \\ 4M, & \text{if } j \neq i. \end{cases}
$$

Then $\|\nabla \frac{\partial F}{\partial x_i}\| \leq 4M\sqrt{d-1}$, *i.e.*, $\frac{\partial F}{\partial x_i}$ is $4M\sqrt{d-1}$-Lipschitz.

Then we deduce that

$$
\begin{aligned}
\|\nabla F(z_1) - \nabla F(z_2)\| &= \left[\sum_{i=1}^{d} \left(\frac{\partial F(z_1)}{\partial x_i} - \frac{\partial F(z_2)}{\partial x_i}\right)^2\right]^{1/2} \\
&\leq \left[\sum_{i=1}^{d} (4M\sqrt{d-1})^2 \|z_1 - z_2\|^2\right]^{1/2} \\
&= \sqrt{\sum_{i=1}^{d} (4M\sqrt{d-1})^2} \cdot \|z_1 - z_2\| \\
&= 4M\sqrt{d(d-1)}\|z_1 - z_2\|.
\end{aligned}
$$

So F is $4M\sqrt{d(d-1)}$-smooth. □

5.7.5 Proof of Theorem 11

Proof. Recall that we define $z_t = x_t + \delta\mathbf{1}$. Then we have

$$\begin{aligned}
\mathbb{E}[\|g_t - \nabla\tilde{F}(z_t)\|^2|\mathcal{F}_{t-1}] &= \frac{1}{B_t^2}\sum_{i=1}^{B_t}\mathbb{E}[\|\frac{d}{2\delta}(\bar{f}_{t,i}^{+} - \bar{f}_{t,i}^{-})u_{t,i} - \nabla\tilde{F}(z_t)\|^2|\mathcal{F}_{t-1}] \\
&= \frac{1}{B_t^2}\sum_{i=1}^{B_t}\mathbb{E}[\|[\frac{d}{2\delta}(F(y_{t,i}^{+}) - F(y_{t,i}^{-}))u_{t,i} - \nabla\tilde{F}(z_t)] \\
&\quad + \frac{d}{2\delta}[\bar{f}_{t,i}^{+} - F(y_{t,i}^{+})]u_{t,i} - \frac{d}{2\delta}[\bar{f}_{t,i}^{-} - F(y_{t,i}^{-})]u_{t,i}\|^2|\mathcal{F}_{t-1}] \\
&= \frac{1}{B_t^2}\sum_{i=1}^{B_t}\mathbb{E}[\|[\frac{d}{2\delta}(F(y_{t,i}^{+}) - F(y_{t,i}^{-}))u_{t,i} - \nabla\tilde{F}(z_t)]\|^2|\mathcal{F}_{t-1}] \\
&\quad + \frac{1}{B_t^2}\sum_{i=1}^{B_t}\mathbb{E}[|\frac{d}{2\delta}[\bar{f}_{t,i}^{+} - F(y_{t,i}^{+})]|^2|\mathcal{F}_{t-1}] \\
&\quad + \frac{1}{B_t^2}\sum_{i=1}^{B_t}\mathbb{E}[|\frac{d}{2\delta}[\bar{f}_{t,i}^{-} - F(y_{t,i}^{-})]|^2|\mathcal{F}_{t-1}],
\end{aligned}$$

where we used the independence of $\bar{f}_{t,i}^{\pm}$ and the facts that $\mathbb{E}[\bar{f}_{t,i}^{\pm}] = F(y_{t,i}^{\pm}), \mathbb{E}[\frac{d}{2\delta}(F(y_{t,i}^{+}) - F(y_{t,i}^{-}))u_{t,i}] = \nabla\tilde{F}(z_t)$.

Then same to Eq. (5.6) and by Lemma 12, the first item is no more than $\frac{4cd^2M^2}{B_t}$. To upper bound the last two items, we have for every $i \in [B_t]$,

$$\begin{aligned}
\mathbb{E}[|\frac{d}{2\delta}[\bar{f}_{t,i}^{+} - F(y_{t,i}^{+})]|^2|\mathcal{F}_{t-1}] &= \frac{d^2}{4\delta^2}\mathbb{E}[[\sum_{j=1}^{l}[f(Y_{t,i,j}^{+}) - F(y_{t,i}^{+})]/l]^2|\mathcal{F}_{t-1}] \\
&\le \frac{d^2}{4\delta^2}\cdot l \cdot \frac{M^2}{l^2} \\
&= \frac{d^2M^2}{4l\delta^2}.
\end{aligned}$$

Similarly, we have

$$\mathbb{E}[|\frac{d}{2\delta}[\bar{f}_{t,i}^{-} - F(y_{t,i}^{-})]|^2|\mathcal{F}_{t-1}] \le \frac{d^2M^2}{4l\delta^2}.$$

As a result, we have

$$\begin{aligned}\mathbb{E}[\|g_t - \nabla \tilde{F}(z_t)\|^2|\mathcal{F}_{t-1}] &\le \frac{4cd^2M^2}{B_t} + \frac{1}{B_t^2}\cdot B_t \cdot \frac{d^2M^2}{4l\delta^2} + \frac{1}{B_t^2}\cdot B_t\cdot\frac{d^2M^2}{4l\delta^2}\\ &= \frac{4cd^2M^2}{B_t} + \frac{d^2M^2}{2B_tl\delta^2}.\end{aligned}$$

Then same to the proof for Theorem 9, we have

$$\begin{aligned}&(1-\frac{1}{e})F(x^*) - \mathbb{E}[F(z_{T+1})]\\ \le& \frac{3D_1Q^{1/2}}{T^{1/3}} + \frac{2M\sqrt{d(d-1)}D_1^2}{T} + 2M\delta\sqrt{d}(1+(\sqrt{d}+1)(1-\frac{1}{e})),\end{aligned} \tag{5.8}$$

where $D_1 \triangleq \mathrm{diam}(\mathcal{K}')$, $Q = \max\{4^{5/3}dM^2, \frac{2d^2M^2(8c+\frac{1}{l\delta^2})}{B_t} + 96d(d-1)M^2D_1^2\}$, x^* is the global maximizer of F on $\mathcal{K}$.

Note that since the rounding scheme is lossless, we have

$$(1-\frac{1}{e})f(X^*) - \mathbb{E}[f(X_{T+1})] \le (1-\frac{1}{e})F(x^*) - \mathbb{E}[F(z_{T+1})]. \tag{5.9}$$

Combine Eqs. (5.8) and (5.9), we have

$$\begin{aligned}&(1-\frac{1}{e})f(X^*) - \mathbb{E}[f(X_{T+1})]\\ \le& \frac{3D_1Q^{1/2}}{T^{1/3}} + \frac{2M\sqrt{d(d-1)}D_1^2}{T} + 2M\delta\sqrt{d}(1+(\sqrt{d}+1)(1-\frac{1}{e})).\end{aligned}$$

□

Bibliography

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

[2] Mingrui Zhang, Lin Chen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Quantized frank-wolfe: Faster optimization, lower communication, and projection free. In *International Conference on Artificial Intelligence and Statistics*, pages 3696–3706. PMLR, 2020.

[3] Lin Chen, Mingrui Zhang, Hamed Hassani, and Amin Karbasi. Black box submodular maximization: Discrete and continuous settings. In *International Conference on Artificial Intelligence and Statistics*, pages 1058–1070. PMLR, 2020.

[4] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML*, pages 427–435, 2013.

[5] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504, 2015.

[6] Dan Garber and Elad Hazan. Faster rates for the frank-wolfe method over strongly-convex sets. In *ICML*, volume 15, pages 541–549, 2015.

[7] Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. In *ICML*, pages 1263–1271, 2016.

[8] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *arXiv preprint arXiv:1804.09554*, 2018.

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

[10] Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic frank-wolfe methods for nonconvex optimization. In *2016 54th Annual Allerton Conference*

on Communication, Control, and Computing (Allerton), pages 1244–1251. IEEE, 2016.

[11] Aryan Mokhtari, Asuman Ozdaglar, and Ali Jadbabaie. Escaping saddle points in constrained optimization. In *Advances in Neural Information Processing Systems*, pages 3629–3639, 2018.

[12] Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. One sample stochastic frank-wolfe. *arXiv preprint arXiv:1910.04322*, 2019.

[13] Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Zebang Shen. Stochastic continuous greedy++: When upper and lower bounds match. In *Advances in Neural Information Processing Systems*, pages 13066–13076, 2019.

[14] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics (NRL)*, 3(1-2):95–110, 1956.

[15] Elad Hazan and Satyen Kale. Projection-free online learning. In *ICML*, pages 1843–1850, 2012.

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

[17] Gábor Braun, Sebastian Pokutta, and Daniel Zink. Lazifying conditional gradient algorithms. In *International conference on machine learning*, pages 566–575. PMLR, 2017.

[18] Zebang Shen, Cong Fang, Peilin Zhao, Junzhou Huang, and Hui Qian. Complexities in projection-free stochastic non-convex minimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2868–2876, 2019.

[19] Alp Yurtsever, Suvrit Sra, and Volkan Cevher. Conditional gradient methods via stochastic path-integrated differential estimator. In *International Conference on Machine Learning*, pages 7282–7291, 2019.

[20] Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *arXiv preprint arXiv:1812.04529*, 2018.

[21] Sathya N Ravi, Tuan Dinh, Vishnu Sai Rao Lokhande, and Vikas Singh. Constrained deep learning using conditional gradient and applications in computer vision. *arXiv preprint arXiv:1803.06453*, 2018.

[22] Patrick Schramowski, Christian Bauckhage, and Kristian Kersting. Neural conditional gradients. *arXiv preprint arXiv:1803.04300*, 2018.

[23] Leonard Berrada, Andrew Zisserman, and M Pawan Kumar. Deep frank-wolfe for neural network optimization. *arXiv preprint arXiv:1811.07591*, 2018.

[24] Wei Ping, Qiang Liu, and Alexander T Ihler. Learning infinite rbms with frank-wolfe. In *Advances in Neural Information Processing Systems*, pages 3063–3071, 2016.

[25] Wenpeng Zhang, Peilin Zhao, Wenwu Zhu, Steven CH Hoi, and Tong Zhang. Projection-free distributed online learning in networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 4054–4062. JMLR. org, 2017.

[26] Wenjie Zheng, Aurélien Bellet, and Patrick Gallinari. A distributed frank–wolfe framework for learning low-rank matrices with the trace norm. *Machine Learning*, 107(8-10):1457–1475, 2018.

[27] Yu-Xiang Wang, Veeranjaneyulu Sadhanala, Wei Dai, Willie Neiswanger, Suvrit Sra, and Eric Xing. Parallel and distributed block-coordinate frank-wolfe algorithms. In *International Conference on Machine Learning*, pages 1548–1557, 2016.

[28] Trishul M Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *OSDI*, volume 14, pages 571–582, 2014.

[29] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[30] Nikko Strom. Scalable distributed dnn training using commodity gpu cloud computing. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[31] Christopher M De Sa, Ce Zhang, Kunle Olukotun, and Christopher Ré. Taming the wild: A unified analysis of hogwild-style algorithms. In *Advances in neural information processing systems*, pages 2674–2682, 2015.

[32] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[33] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *Advances in neural information processing systems*, pages 1509–1519, 2017.

[34] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.

[35] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd: compressed optimisation for non-convex problems. *arXiv preprint arXiv:1802.04434*, 2018.

[36] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, pages 4452–4463, 2018.

[37] Aurélien Bellet, Yingyu Liang, Alireza Bagheri Garakani, Maria-Florina Balcan, and Fei Sha. A distributed frank-wolfe algorithm for communication-efficient sparse learning. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 478–486. SIAM, 2015.

[38] Jean Lafond, Hoi-To Wai, and Eric Moulines. D-fw: Communication efficient distributed algorithms for high-dimensional sparse optimization. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4144–4148. IEEE, 2016.

[39] Charles Audet and Warren Hare. *Derivative-free and blackbox optimization*. Springer, 2017.

[40] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*, volume 8. Siam, 2009.

[41] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

[42] Luis Miguel Rios and Nikolaos V Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

[43] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[44] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2): 1–305, 2008.

[45] Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. Learning structured prediction models: A large margin approach. In *Proceedings of the 22nd international conference on Machine learning*, pages 896–903. ACM, 2005.

[46] Artem Sokolov, Julia Kreutzer, Stefan Riezler, and Christopher Lo. Stochastic structured prediction under bandit feedback. In *Advances in Neural Information Processing Systems*, pages 1489–1497, 2016.

[47] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[48] Chris Thornton, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 847–855. ACM, 2013.

[49] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

[50] Andrew Ilyas, Logan Engstrom, Anish Athalye, Jessy Lin, Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning,{ICML} 2018*, 2018.

[51] Yining Wang, Simon Du, Sivaraman Balakrishnan, and Aarti Singh. Stochastic zeroth-order optimization in high dimensions. *arXiv preprint arXiv:1710.10551*, 2017.

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

[53] Anit Kumar Sahu, Manzil Zaheer, and Soummya Kar. Towards gradient free and projection free stochastic optimization. *arXiv preprint arXiv:1810.03233*, 2018.

[54] Francis Bach. Submodular functions: from discrete to continuous domains. *Mathematical Programming*, pages 1–41, 2016.

[55] Matthew Staib and Stefanie Jegelka. Robust budget allocation via continuous submodular functions. In *ICML*, pages 3230–3240, 2017.

[56] An Bian, Kfir Levy, Andreas Krause, and Joachim M Buhmann. Continuous dr-submodular maximization: Structure and algorithms. In *Advances in Neural Information Processing Systems*, pages 486–496, 2017.

[57] Andrew An Bian, Baharan Mirzasoleiman, Joachim M. Buhmann, and Andreas Krause. Guaranteed non-convex optimization: Submodular maximization over continuous domains. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 111–120. PMLR, 2017. URL `http://proceedings.mlr.press/v54/bian17a.html`.

[58] Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. *arXiv preprint arXiv:1708.03949*, 2017.

[59] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Conditional gradient method for stochastic submodular maximization: Closing the gap. In *AISTATS*, pages 1886–1895, 2018.

[60] Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

[61] Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization. In *International Conference on Artificial Intelligence and Statistics*, pages 1896–1905, 2018.

[62] Reza Eghbali and Maryam Fazel. Designing smoothing functions for improved worst-case competitive ratio in online optimization. In *NIPS*, pages 3287–3295, 2016.

[63] An Bian, Joachim M Buhmann, and Andreas Krause. Optimal dr-submodular maximization and applications to provable mean field inference. *arXiv preprint arXiv:1805.07482*, 2018.

[64] Zebang Shen, Alejandro Ribeiro, Hamed Hassani, Hui Qian, and Chao Mi. Hessian aided policy gradient. In *International Conference on Machine Learning*, pages 5729–5738, 2019.

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

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

[67] Lin Chen, Mingrui Zhang, and Amin Karbasi. Projection-free bandit convex optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2047–2056. PMLR, 2019.

[68] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323, 2013.

[69] Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pages 699–707, 2016.

[70] Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduced gradient descent for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 3921–3932, 2018.

[71] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2613–2621. JMLR. org, 2017.

[72] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Stochastic recursive gradient algorithm for nonconvex optimization. *arXiv preprint arXiv:1705.07261*, 2017.

[73] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 687–697, 2018.

[74] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[75] Stefanie Jegelka and Jeff Bilmes. Submodularity beyond submodular energies: coupling edges in graph cuts. 2011.

[76] Stefanie Jegelka and Jeff A Bilmes. Approximation bounds for inference using cooperative cuts. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 577–584, 2011.

[77] Hui Lin and Jeff Bilmes. Word alignment via submodular maximization over matroids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 170–175. Association for Computational Linguistics, 2011.

[78] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *ACL*, pages 510–520, 2011.

[79] Sebastian Tschiatschek, Rishabh K Iyer, Haochen Wei, and Jeff A Bilmes. Learning mixtures of submodular functions for image collection summarization. In *Advances in neural information processing systems*, pages 1413–1421, 2014.

[80] Lin Chen, Moran Feldman, and Amin Karbasi. Weakly submodular maximization beyond cardinality constraints: Does randomization help greedy? In *ICML*, pages 804–813, 2018.

[81] Lin Chen, Andreas Krause, and Amin Karbasi. Interactive submodular bandit. In *NeurIPS*, pages 141–152, 2017.

[82] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, pages 137–146. ACM, 2003.

[83] Manuel Gomez Rodriguez and Bernhard Schölkopf. Influence maximization in continuous time diffusion networks. *arXiv preprint arXiv:1205.1682*, 2012.

[84] Yuanxing Zhang, Yichong Bai, Lin Chen, Kaigui Bian, and Xiaoming Li. Influence maximization in messenger-based social networks. In *GLOBECOM*, pages 1–6. IEEE, 2016.

[85] Mohammadhossein Bateni, Lin Chen, Hossein Esfandiari, Thomas Fu, Vahab Mirrokni, and Afshin Rostamizadeh. Categorical feature compression via submodular optimization. In *International Conference on Machine Learning*, pages 515–523, 2019.

[86] Lin Chen, Forrest W Crawford, and Amin Karbasi. Submodular variational inference for network reconstruction. In *UAI*, 2017.

[87] Andrew Guillory and Jeff Bilmes. Interactive submodular set cover. *arXiv preprint arXiv:1002.3345*, 2010.

[88] Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *JAIR*, 42:427–486, 2011.

[89] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pages 1954–1963, 2015.

[90] Adish Singla, Ilija Bogunovic, Gábor Bartók, Amin Karbasi, and Andreas Krause. Near-optimally teaching the crowd to classify. In *ICML*, pages 154–162, 2014.

[91] Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975*, 2011.

[92] Mehraveh Salehi, Amin Karbasi, Dustin Scheinost, and R Todd Constable. A submodular approach to create individualized parcellations of the human brain. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 478–485. Springer, 2017.

[93] Francis R Bach. Structured sparsity-inducing norms through submodular functions. In *Advances in Neural Information Processing Systems*, pages 118–126, 2010.

[94] Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012.

[95] Eric Balkanski and Yaron Singer. Mechanisms for fair attribution. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 529–546. ACM, 2015.

[96] L Elisa Celis, Amit Deshpande, Tarun Kathuria, and Nisheeth K Vishnoi. How to be fair and diverse? *arXiv preprint arXiv:1610.07183*, 2016.

[97] Bastian Steudel, Dominik Janzing, and Bernhard Schölkopf. Causal markov condition for submodular information measures. *arXiv preprint arXiv:1002.4020*, 2010.

[98] Yuxun Zhou and Costas J Spanos. Causal meets submodular: Subset selection with directed information. In *Advances in Neural Information Processing Systems*, pages 2649–2657, 2016.

[99] Yatao Bian, Joachim Buhmann, and Andreas Krause. Optimal continuous dr-submodular maximization and applications to provable mean field inference. In *International Conference on Machine Learning*, pages 644–653, 2019.

[100] Francis Bach. Submodular functions: from discrete to continous domains. *arXiv preprint arXiv:1511.00394*, 2015.

[101] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Conditional gradient method for stochastic submodular maximization: Closing the gap. In *AISTATS*, pages 1886–1895, 2018.

[102] Rad Niazadeh, Tim Roughgarden, and Joshua Wang. Optimal algorithms for continuous non-monotone submodular and dr-submodular maximization. In *Advances in Neural Information Processing Systems*, pages 9594–9604, 2018.

[103] Lin Chen, Moran Feldman, and Amin Karbasi. Unconstrained submodular maximization with constant adaptive complexity. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 102–113, 2019.

[104] Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *NIPS*, pages 1577–1584, 2009.

[105] Daniel Golovin, Andreas Krause, and Matthew Streeter. Online submodular maximization under a matroid constraint with application to learning assignments. Technical report, arXiv, 2014.

[106] Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and Shanmugavelayutham Muthukrishnan. Adaptive submodular maximization in bandit setting. In *NIPS*, pages 2697–2705. Citeseer, 2013.

[107] Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In *NIPS*, pages 2483–2491, 2011.

[108] Baosheng Yu, Meng Fang, and Dacheng Tao. Linear submodular bandits with a knapsack constraint. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[109] Mingrui Zhang, Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization: From full-information to bandit feedback. In *Advances in Neural Information Processing Systems*, pages 9206–9217, 2019.

[110] Simon S Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1049–1058. JMLR. org, 2017.

[111] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[112] Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli. Stochastic variance-reduced policy gradient. *arXiv preprint arXiv:1806.05618*, 2018.

[113] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *arXiv preprint arXiv:1905.10018*, 2019.

[114] Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, pages 67–74. ACM, 2008.

[115] Mohammad Karimi, Mario Lucic, Hamed Hassani, and Andreas Krause. Stochastic submodular maximization: The case of coverage functions. In *NIPS*, page to appear, 2017.

[116] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[117] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.

[118] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[119] Kilian Q Weinberger and Lawrence K Saul. Unsupervised learning of image manifolds by semidefinite programming. *International journal of computer vision*, 70(1): 77–90, 2006.

[120] Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *Advances in neural information processing systems*, pages 2301–2309, 2013.

[121] Chao Qu, Yan Li, and Huan Xu. Non-convex conditional gradient sliding. *arXiv preprint arXiv:1708.04783*, 2017.

[122] Lam M Nguyen, Marten van Dijk, Dzung T Phan, Phuong Ha Nguyen, Tsui-Wei Weng, and Jayant R Kalagnanam. Optimal finite-sum smooth non-convex optimization with sarah. *arXiv preprint arXiv:1901.07648*, 2019.

[123] Zebang Shen, Cong Fang, Peilin Zhao, Junzhou Huang, and Hui Qian. Complexities in projection-free stochastic non-convex minimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2868–2876, 2019.

[124] Qi Lei, Lingfei Wu, Pin-Yu Chen, Alexandros Dimakis, Inderjit Dhillon, and Michael Witbrock. Discrete adversarial attacks and submodular optimization with applications to text classification. *Systems and Machine Learning (SysML)*, 2019.

[125] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[126] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *SODA*, pages 385–394, 2005.

[127] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[128] Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pages 28–40. Citeseer, 2010.

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

[130] James G Oxley. *Matroid theory*, volume 3. Oxford University Press, USA, 2006.

[131] Alexander A Ageev and Maxim I Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004.

[132] Khalid El-Arini, Gaurav Veda, Dafna Shahaf, and Carlos Guestrin. Turning down the noise in the blogosphere. In *SIGKDD*, pages 289–298. ACM, 2009.

[133] Rishabh Iyer, Stefanie Jegelka, and Jeff Bilmes. Monotone closure of relaxed constraints in submodular optimization: Connections between minimization and maximization. In *Uncertainty in Artificial Intelligence (UAI)*, Quebic City, Quebec Canada, July 2014. AUAI.

[134] Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *NIPS*, pages 2049–2057, 2013.

[135] Athanasios Tsanas, Max A Little, Patrick E McSharry, and Lorraine O Ramig. Enhanced classical dysphonia measures and sparse regression for telemonitoring of parkinson's disease progression. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 594–597. IEEE, 2010.

[136] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

[137] Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In *ICML*, pages 813–822, 10–15 Jul 2018.