Online Decision Making: DR-Submodular Objectives and Stochastic Linear Constraints

Prasanna Sanjay Raut

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2020

Committee:

Maryam Fazel

Ashis Banerjee

Steven L. Brunton

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Online Decision Making:
DR-Submodular Objectives
and
Stochastic Linear Constraints

Prasanna Sanjay Raut

Chair of the Supervisory Committee:
Maryam Fazel
Department of Electrical and Computer Engineering

In this thesis, we consider online continuous DR-submodular maximization with linear stochastic long-term constraints. Compared to the prior work on online submodular maximization [10], our setting introduces the extra complication of stochastic linear constraint functions generated at each round and are independent and identically distributed (i.i.d). To be precise, at step $t \in \{1, \ldots, T\}$, a DR-submodular utility function $f_t(\cdot)$ and a constraint vector p_t, i.i.d. generated from an unknown distribution with mean p, are revealed after committing to an action x_t and we aim to maximize the overall utility while the expected cumulative resource consumption $\sum_{t=1}^{T} \langle p, x_t \rangle$ is below a fixed budget B_T. Stochastic long-term constraints arise naturally in applications where there is a limited budget or resource available and resource consumption at each step is governed by stochastically time-varying environments. We propose the Online Lagrangian Frank-Wolfe (OLFW) algorithm to solve this class of online problems. We analyze the performance of the OLFW algorithm and we obtain sub-linear regret bounds as well as sub-linear cumulative constraint violation bounds, both in expectation and with high probability.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Maryam Fazel. In my first two quarters, I got the opportunity to take both her classes which served as an excellent introduction to optimization theory and was my main motivation to work in this area for my thesis. She provided me the freedom and support to explore different projects and find something that I enjoyed. Her breadth of knowledge and deep mathematical understanding have been inspirational. I feel very fortunate to have had Maryam as my advisor and teacher.

I am extremely grateful to Omid Sadeghi for mentoring me. His mathematical brilliance, intuition and ability to translate ideas into research papers has made our collaboration a truly humbling experience for me. I am proud to be his co-author on my first paper. I am thankful to my committee members Ashis and Steve for providing valuable feedback on my thesis presentation.

I also would like to extend gratitude to my teachers from undergrad and high-school, especially Indranath Sengupta, M. Prakash and Smita Potnis, who set the highest standards for me in learning mathematics and influenced the way I approach problem solving. I am also grateful for my undergraduate research experiences with Harish P.M. at IIT Gandhinagar and Ashis at UW, the lessons learnt then were much appreciated during my current research.

I was fortunate to have many friends and peers at UW — Abhay, Ekta, Hari, Karthik, Rahul and Siva, who made my time here very memorable. Special thanks to my roommates Chinmay, Gaurav and Ravi for being a family away from home. I also wish to thank Yogesh Chukewad who helped me get funding for a number of quarters.

Finally, I would like to thank my parents and my sister for their unconditional support, without which I would not have been successful.

DEDICATION

In memory of my grandfather

Chandrakant Shankarrao Raut,

whose love and affection I'll forever miss.

Chapter 1
INTRODUCTION

In this thesis, we study a new class of online allocation problems with long-term resource constraints where the utility functions are DR-submodular (and not necessarily concave) and the constraint functions are linear with coefficient vectors drawn i.i.d. from some unknown underlying distribution. The problem has been extensively studied in the convex setting [33, 31, 23, 19]; furthermore, [28] considered a similar framework under the assumption that the linear constraint functions are chosen adversarially. However, [28] do not provide any high probability bounds for the regret and constraint violation with random i.i.d. linear constraints, and their expected constraint violation bound is worse than ours as well. We provide the *first* sub-linear bounds for the regret and total budget violation that hold in expectation as well as with high probability. The thesis is outlined as below:

- Chapter 1 includes some definitions and mathematical notations used throughout the thesis.

- Chapter 2 introduces the Online Convex Optimization framework and discusses the state-of-art methods for dealing with time-varying constraints in the online setting.

- Chapter 3 contains definitions and properties of *submodular* and *DR-submodular* functions which allow for their efficient optimization.

- In Chapter 4, in order to illustrate the generality of our framework, we have listed a number of interesting applications that could be cast into our setting.

- In Chapter 5, *a*) We propose the Online Lagrangian Frank-Wolfe (OLFW) algorithm

for this class of online continuous DR-submodular maximization problems with stochastic cumulative constraints. The OLFW algorithm is inspired by the quadratic penalty method in constrained optimization literature [24] and it generalizes a Frank-Wolfe variant proposed by [10] for solving online continuous DR-submodular maximization problems to take into account the additional stochastically time-varying linear constraints. Note that this extension is not straightforward and the choice of the penalty function and the update rule for the dual variable are crucial in obtaining bounds for the total budget violation as well as the regret. *b*) We analyze the performance of the OLFW algorithm with high probability and in expectation in Section 5.2.2 and Section 5.2.3 respectively, and we establish the first sub-linear *expected* and *high probability* bounds on both the *regret* and *total budget violation* of the algorithm.

- In Chapter 6, we demonstrate the effectiveness of our proposed algorithm on simulated and real-world problem instances, and compare the performance of the OLFW algorithm with several baseline algorithms. We discuss connections to different papers in literature and conclude the thesis.

1.1 Notation

$[T]$ is used to denote the set $\{1, 2, \ldots, T\}$. For $u \in \mathbb{R}$, we define $[u]_+ := \max\{u, 0\}$. For a vector $x \in \mathbb{R}^n$, we use x_i to denote the i-th entry of x. The inner product of two vectors $x, y \in \mathbb{R}^n$ is denoted by either $\langle x, y\rangle$ or $x^T y$. Also, for two vectors $x, y \in \mathbb{R}^n$, $x \preceq y$ implies that $x_i \leq y_i \ \forall i \in [n]$. A function $f : \mathbb{R}^n \to \mathbb{R}$ is called monotone if for all x, y such that $x \preceq y$, $f(x) \leq f(y)$ holds. For a vector $x \in \mathbb{R}^n$, we use $\|x\|$ to denote the Euclidean norm of x. The unit ball of the Euclidean norm is denoted by $\mathcal{B}$, i.e., $\mathcal{B} = \{x \in \mathbb{R}^n \mid \|x\| \leq 1\}$. For a convex set $\mathcal{X}$, we will use $\mathcal{P}_{\mathcal{X}}(y) = \arg\min_{x\in\mathcal{X}} \|x - y\|$ to denote the projection of y onto set $\mathcal{X}$. The Fenchel conjugate of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as $f^*(y) = \sup_x (x^T y - f(x))$.

Chapter 2

ONLINE CONVEX OPTIMIZATION

The Online Convex Optimization (OCO) problem has been extensively studied in the literature [13, 29, 34, 25]. In this problem, a sequence of arbitrary convex cost functions $\{f_t(\cdot)\}_{t=1}^T$ are revealed one by one by "nature" and at each round $t \in [T]$, the decision maker chooses an action $x_t \in \mathcal{X}$, where $\mathcal{X}$ is the fixed domain set, before the corresponding function $f_t(\cdot)$ is revealed. Throughout the thesis, we assume that at each round $t \in [T]$ when "nature" reveals $f_t(\cdot)$ — this entails that there is an oracle access available to $\nabla f_t(\cdot)$. The goal is to minimize the regret defined as [34]

$$R_T := \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x).$$

In other words, regret characterizes the difference between the overall cost incurred by the decision maker and that of a fixed benchmark action which has access to all the cost functions $\{f_t\}_{t=1}^T$. In general, the decision maker aims to attain sub-linear regret i.e. $R_T \leq o(T)$ so that $\frac{R_T}{T} \to 0$ as T becomes large. This implies that the average cost incurred per round $\frac{\sum_{t=1}^T f_t(x_t)}{T}$ converges to that of the fixed benchmark action $\frac{\sum_{t=1}^T f_t(x^*)}{T}$.

2.1 Basics

In this section we look at the Online Gradient Descent (OGD) algorithm for OCO. This algorithm uses ideas which are common to most online optimization algorithms, even outside the framework of OCO. Thus it is useful to briefly consider the analysis of OGD in obtaining upper bounds on the regret.

Theorem 1. *[34] Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact set with diameter $D := \max_{x,y \in \mathcal{X}} \|x - y\|$. Let $\{f_t(\cdot)\}_{t=1}^T$ be a arbitrary sequence of convex loss functions, differentiable on int $\mathcal{X}$. Pick any*

Algorithm 2 Online Gradient Descent

Input: $\mathcal{X}$ is the constraint set, $x_1 \in \mathcal{X}$, step sizes $\{\mu_t\}_{t=1}^T$.

Output: $\{x_t : 1 \leq t \leq T\}$.

for $t = 1$ **to** T **do**

Play x_t and incur loss $f_t(x_t)$.

Update $x_{t+1} = \Pi_{\mathcal{X}}(x_t - \mu_t \nabla f_t(x_t))$.

end for

$x_1 \in \mathcal{X}$ and choose the remaining x_t's according to OGD. Also, assume $\mu_{t+1} \leq \mu_t$. With $x^* \in \arg\min_{x \in \mathcal{X}} \sum_{t=1}^T f_t(x)$, the following regret bound holds:

$$\sum_{t=1}^T (f_t(x_t) - f_t(x^*)) \leq \frac{D^2}{2\mu_T} + \sum_{t=1}^T \frac{\mu_t}{2} \|\nabla f_t(x_t)\|^2. \tag{2.1}$$

Proof. By convexity, we have that:

$$f_t(x_t) - f_t(x^*) \leq \langle \nabla f_t(x_t), x_t - x^* \rangle. \tag{2.2}$$

We now upper bound $\langle \nabla f_t(x_t), x_t - x^* \rangle$ using the update rule for x_{t+1},

$$\|x_{t+1} - x^*\|^2 = \|\Pi_{\mathcal{X}}(x_t - \mu_t \nabla f_t(x_t)) - x^*\|^2 \leq \|x_t - \mu_t \nabla f_t(x_t) - x^*\|^2. \tag{2.3}$$

The inequality above follows from the fact that projection on a convex set in Euclidean spaces never increases the distance from points inside the set e.g. see Proposition 2.11 in [25]. Thus, we have:

$$\begin{aligned} \|x_{t+1} - x^*\|^2 &\leq \|x_t - \mu_t \nabla f_t(x_t) - x^*\| \\ &= \|x_t - x^*\|^2 + \mu_t^2 \|\nabla f_t(x_t)\|^2 - 2\mu_t \langle \nabla f_t(x_t), x_t - x^* \rangle. \end{aligned}$$

Rearranging we get that:

$$\langle \nabla f_t(x_t), x_t - x^* \rangle \leq \frac{1}{2\mu_t} (\|x_{t+1} - x^*\|^2 - \|x_t - x^*\|^2) + \frac{\mu_t}{2} \|\nabla f_t(x_t)\|^2. \tag{2.4}$$

Combining (2.2) and (2.4) and summing over $t = 1$ to T, we have:

$$\begin{aligned}
\sum_{t=1}^{T}(f_t(x_t) - f_t(x^*)) &\leq \sum_{t=1}^{T}\left(\frac{1}{2\mu_t}(\|x_{t+1} - x^*\|^2 - \|x_t - x^*\|^2)\right) + \sum_{t=1}^{T}\frac{\mu_t}{2}\|\nabla f_t(x_t)\|^2 \\
&\leq \frac{1}{2\mu_1}\|x_1 - x^*\|^2 + \sum_{t=1}^{T-1}\left(\frac{1}{2\mu_{t+1}} - \frac{1}{2\mu_t}\right)\|x_{t+1} - x^*\|^2 \\
&+ \sum_{t=1}^{T}\frac{\mu_t}{2}\|\nabla f_t(x_t)\|^2 \\
&\leq \frac{1}{2\mu_1}D^2 + D^2\sum_{t=1}^{T-1}\left(\frac{1}{2\mu_{t+1}} - \frac{1}{2\mu_t}\right) + \sum_{t=1}^{T}\frac{\mu_t}{2}\|\nabla f_t(x_t)\|^2 \\
\implies \sum_{t=1}^{T}(f_t(x_t) - f_t(x^*)) &\leq \frac{D^2}{2\mu_T} + \sum_{t=1}^{T}\frac{\mu_t}{2}\|\nabla f_t(x_t)\|^2.
\end{aligned}$$

■

For convex functions f on bounded sets, it is known that f is L-Lipschitz for some L e.g. Corollary 13.3.3 in [27]. Assume that we have a constant L such that $\forall x \in \text{int}\,\mathcal{X}$ and $\forall t \in [T]$, $\|\nabla f_t(x)\| \leq L$. The regret bound can then be written as: $\sum_{t=1}^{T}(f_t(x_t) - f_t(x^*)) \leq \frac{D^2}{2\mu_T} + L^2\sum_{t=1}^{T}\frac{\mu_t}{2}$.

Usually, there are two different types of step sizes, one might be more suitable depending on the specific problem or application:

1. $\mu_t = \mu$ is a constant — In this case, the regret bound becomes $R_T \leq \frac{D^2}{2\mu} + \frac{L^2T\mu}{2}$. By minimizing the right hand side, the optimal step-size is $\mu^* = \frac{D}{L\sqrt{T}}$ and the final regret bound is $R_T \leq DL\sqrt{T}$. Note that the step-size depends on the time horizon T, and if it is unknown before the start of the algorithm, a standard technique called "The Doubling Trick" (e.g. Section 2.3.1 in [29])can be applied. For rounds $t = 2^m, \ldots, 2^{m+1} - 1$, use $\mu = \frac{D}{L\sqrt{2^m}}$ where $m = 0, 1, 2, \ldots$. The regret bound is only slightly worse: $R_T \leq \frac{\sqrt{2}}{\sqrt{2}-1}\frac{D}{L\sqrt{T}}$.

2. μ_t is time-varying — Here, a possible choice of time-steps is $\forall t \in [T]$, $\mu_t := \frac{D}{L\sqrt{t}}$. The final regret bound becomes $R_T \leq \frac{3}{2}DL\sqrt{T}$.

There might be scenarios where the Lipschitz constant L is not readily available, or computationally inefficient to calculate. In such cases, the obvious choice of $\mu_t = \frac{D}{\sqrt{\sum_{\tau=1}^{t} \|\nabla f_t(x_\tau)\|^2}}$ proves to be a good choice and gives a sub-linear regret bound $R_T \leq 2DL\sqrt{T}$.

2.2 Online Budgeted Optimization

In many applications, however, in addition to maximizing the total reward (minimizing the overall cost), there are restrictions on the sequence of decisions made by the learner that need to be satisfied on average [2, 4, 1, 26]. Therefore, it may be beneficial to sacrifice some of the reward to meet other desired goals or restrictions over the time horizon. Such long-term constraints arise naturally in applications with limited budget (resource) availability [5, 6, 11].

As an illustrative example, consider the online ad allocation problem for an advertiser. At each round $t \in [T]$, the advertiser should choose her investment on ads to be placed on n different websites. Beyond the immediate goal of maximizing the overall impressions of the ads, the advertiser needs to balance her total investment against an allotted budget on a daily, monthly or yearly basis [5]. However, the cost of ad placement in each round depends on the number of clicks the ads receive, so they are not known ahead of time. Therefore, the advertiser needs to strike the right balance between the total reward and budget used. These problems can be written as the following optimization problem:

$$\begin{aligned} &\min_{x_1, x_2, \ldots, x_T} \sum_{t=1}^{T} f_t(x_t) \\ &\text{subject to } \sum_{t=1}^{T} g_t(x_t) \leq 0. \end{aligned} \tag{2.5}$$

Depending on the specific problem scenario, $g_t(\cdot)$'s can be adversarial, stochastic or even constant. In each of these scenarios, there are multiple ways to choose the feasibility set for the benchmark action x^*, when defining the regret. We now look at two main approaches used to tackle online budgeted optimization problems. These approaches are closely related but lead to very different styles of analyses.

2.2.1 Gradient methods based on the Lagrangian

The offline optimization problem corresponding to (2.5) can be written as:

$$\min_{x} \max_{\lambda} \sum_{t=1}^{T} f_t(x) + \lambda \sum_{t=1}^{T} g_t(x). \tag{2.6}$$

Thus, the per round Lagrangian function is $\mathcal{L}_t(x, \lambda) = f_t(x) + \lambda g_t(x)$. In literature, a regularizer (which is usually the quadratic penalty) on λ is added to the Lagrangian in order to avoid the gradients becoming too large and we write the regularized Lagrangian as:

$$\mathcal{L}_t(x, \lambda) = f_t(x) + \lambda g_t(x) - \frac{\delta\mu}{2}\lambda^2. \tag{2.7}$$

We can then write the typical algorithm as follows: This algorithm with minor modifica-

Algorithm 3 General Algorithm for Online Budgeted Optimization

Input: $\mathcal{X}$ is the constraint set, $x_1 \in \mathcal{X}$, $\lambda_1 = 0$ step sizes $\{\mu_t\}_{t=1}^{T}$ and constant δ.

Output: $\{x_t : 1 \leq t \leq T\}$.

for $t = 1$ **to** T **do**

  Play x_t and incur loss $f_t(x_t)$ and use budget $g_t(x_t)$.

  Update $x_{t+1} = \Pi_{\mathcal{X}}(x_t - \mu_t \nabla_x \mathcal{L}_t(x_t, \lambda_t))$.

  Update $\lambda_{t+1} = \Pi_{\mathbb{R}^+}(\lambda_t + \mu_t \nabla_\lambda \mathcal{L}_t(x_t, \lambda_t))$.

end for

tions is used in [20, 16, 33] for the setting where g is constant and known. [20] and [16] provide sub-linear bounds on the regret and constraint violation (i.e. $\sum_{t=1}^{T} g(x_t)$) whereas [33] provides stronger guarantees on the constraint violation by providing sub-linear bounds on $\sum_{t=1}^{T}([g(x_t)]_+)^2$.

2.2.2 Gradient descent with virtual queues

In (2.7), the Lagrangian function can be seen as way to handle a trade-off between the losses $f_t(x)$ and the budget $g_t(x)$ using the dual variable λ. Works such as [32, 23, 19] use a

different approach to handle this trade-off. They consider the following per-round regularized Lagrangian-like function:

$$\mathcal{L}_t(x, Q(t)) = V f_t(x) + Q(t) g_t(x) + \alpha \|x - x_{t-1}\|^2. \tag{2.8}$$

- The parameter V handles the trade-off between the loss and constraint violation.
- $Q(t)$ is the virtual queue which plays the role of Lagrangian dual variable.
- $\|x - x_{t-1}\|^2$ is the regularization on x to avoid large fluctuations in the decisions, and α is the strength of the regularization.

The algorithm for this framework takes the following form[19]: Intuitively, each round t the

Algorithm 4 Online Budgeted Optimization with Virtual Queues

Input: $\mathcal{X}$ is the constraint set, $x_0 \in \mathcal{X}$, $Q(1) = 0$, parameters V and α.

Output: $\{x_t : 1 \leq t \leq T\}$.

for $t = 1$ **to** T **do**

Play $x_t = \Pi_{\mathcal{X}}\big(x_{t-1} - \frac{V \nabla f_{t-1}(x_{t-1}) + Q(t-1) \nabla g_{t-1}(x_{t-1})}{2\alpha}\big)$.

Update $Q(t) = [Q(t-1) + \underbrace{g_{t-1}(x_{t-1}) + \langle \nabla g_{t-1}(x_{t-1}), x_t - x_{t-1} \rangle}_{\widehat{g}_t(x_t)}]_+$.

Receive $f_t(\cdot)$ and $g_t(\cdot)$.

end for

algorithm takes an action to minimize the estimate of the Lagrangian function (2.8) before the actual information for it is received. Thus, x_t minimizes $V \widehat{f}_t(x) + Q(t) \widehat{g}_t(x) + \alpha \|x - x_{t-1}\|^2$, where $\widehat{f}_t(x) = f_{t-1}(x_{t-1}) + \langle \nabla f_{t-1}(x_{t-1}), x - x_{t-1} \rangle$ is an estimate of $f_t(\cdot)$ using the information from the previous round. [19] is an excellent resource to understand further the analysis of the algorithm. Due to the impossibility result of [22] — that simultaneous sub-linear bounds on regret and constraint violation are not possible when both the functions and constraints are chosen by the adversary, additional assumptions are required to obtain

reasonable performance. Both [23] and [19] restrict the power of the benchmark action x^* used to define regret. [23] choose x^* such that $g_t(x^*) < 0$ for each round t in the adversarial setting or $\mathbb{E}[g_t(x^*)] < 0$ in the stochastic model. On the other hand, [19] introduce what they call a K-*benchmark* that requires x^* to satisfy the overall constraint over all K-length windows i.e. $\sum_{\tau=i}^{i+K-1} g_\tau(x^*) \leq 0$ for $i \in [T-K]$.

Chapter 3
DR-SUBMODULAR FUNCTIONS

Up to this point, we have only talked about problems where the objective function is convex. There are many applications which require solving a non-convex optimization problem e.g. dictionary learning, clustering and training of deep neural networks are some applications in machine learning. Finding the global optima of a general non-convex function is NP-hard and thus, most of the methods focus on finding a local optima usually with no guarantees on the quality of the obtained solution. Submodular functions are a general class of non-convex functions which have efficient algorithms for their optimization along with strong guarantees. As these functions naturally capture the "diminishing returns" property they find a number of applications in economics, operations research, and machine learning. Submodular functions were more commonly used in discrete settings whereas we consider their continuous counterpart *DR-submodular* functions for online optimization problems in this thesis.

3.1 Submodular functions and DR-submodularity

There are a number of equivalent definitions for submodular. We mention two of the most common ones below. Let S be any set, then the set of all subsets of S is denoted by 2^S.

Definition 1. *(Submodular set function) $f : 2^S \to \mathbb{R}$ is a submodular function iff $\forall A, B \subset S$ s.t. $A \subset B$ and $\forall j$ s.t. $j \in S \backslash B$ the following holds:*

$$f(A \cup \{j\}) - f(A) \geq f(B \cup \{j\}) - f(B). \tag{3.1}$$

Equivalently,

Definition 2. *(Submodular set function) $f : 2^S \to \mathbb{R}$ is a submodular function iff $\forall A, B \subset S$ the following holds:*

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B). \tag{3.2}$$

Assume that the cardinality of the set S is n i.e $|S| = n$. Let $A \subset S$ and $\chi_A \in \{0,1\}^n$ be the *characteristic* vector of A, which is defined using some arbitrary ordering of the elements of S — $\chi(a) = 1$ if $a \in A$, otherwise 0. Thus, we can write the definition above and equation (3.2) using the characteristic vector representation of sets as: $f : \{0,1\}^n \to \mathbb{R}$ is a submodular function iff $\forall x, y \in \{0,1\}^n$ the following holds: $f(x) + f(y) \geq f(x \vee y) + f(x \wedge y)$, where $\vee$ represents the coordinate-wise maximum binary operator and $\wedge$ is the coordinate-wise minimum binary operator. As these operations don't assume anything specific about the set $\{0,1\}^n$, we can write the same definition for other sets e.g. [3, 7].

Definition 3. *(Submodular continuous function) Let $\mathcal{X} = \Pi_{i=1}^n \mathcal{X}_i \subset \mathbb{R}^n$ s.t. $\mathcal{X}_i$ is compact subset of $\mathbb{R}$ for each $i \in [n]$. $f : \mathcal{X} \to \mathbb{R}$ is a continuous submodular function iff $\forall x, y \in \mathcal{X}$ the following holds:*

$$f(x) + f(y) \geq f(x \vee y) + f(x \wedge y). \tag{3.3}$$

We now define continuous DR-submodular functions:

Definition 4. *$f : \mathcal{X} \to \mathbb{R}$ satisfies the "diminishing returns" (DR) property iff $\forall a \preceq b \in \mathcal{X}$ $\forall i \in [n]$, $\forall k \in \mathbb{R}_+$ s.t. $k\chi_i + a$ and $k\chi_i + b$ are still in $\mathcal{X}$ the following holds:*

$$f(k\chi_i + a) - f(a) \geq f(k\chi_i + b) - f(b), \tag{3.4}$$

and $f(\cdot)$ which satisfies the DR property is called a *DR-submodular* function. If the given function is differentiable, then the DR property is equivalent to $\nabla f(\cdot)$ being order-reversing:

Proposition 1. *$f : \mathcal{X} \to \mathbb{R}$ is DR-submodular iff $\forall x, y \in \mathcal{X}, x \succeq y \implies \nabla f(x) \preceq \nabla f(y)$.*

Proof. ($\implies$) If f is DR-submodular and $x \succeq y$, then for each χ_i for $i \in [n]$ dividing (3.4) by k on both sides of the inequality and taking limits as $k \to 0$, we get that $\nabla f(x) \preceq \nabla f(y)$.

($\Longleftarrow$) Now assume that $a \preceq b$ and $\nabla f(b) \succeq \nabla f(a)$. Define $h(t) := f(a + t\chi_i) - f(b + t\chi_i)$. Thus, $h^{'}(t) = t\langle \nabla f(a + t\chi_i) - \nabla f(b + t\chi_i), \chi_i \rangle$. Due to the order-reversing nature of $\nabla f(\cdot)$, we have that $h^{'}(t) \geq 0$. Thus, $h(k) \geq h(0)$ i.e. $f(k\chi_i + a) - f(k\chi_i + b) \geq f(a) - f(b)$ or rearranging we get $f(k\chi_i + a) - f(a) \geq f(k\chi_i + b) - f(b)$. ■

If f is twice differentiable, the DR property is equivalent to the Hessian matrix being element-wise non-positive. Note that for $n = 1$, the DR property is equivalent to concavity. However, for $n > 1$, concavity corresponds to negative semidefiniteness of the Hessian matrix (which is not equivalent to the Hessian matrix being element-wise non-positive). DR-submodular functions are also known as "smooth submodular" in the submodularity literature (e.g., see [30]). Bian et al. [7] showed that a DR-submodular function f is concave along any non-negative and any non-positive direction; that is, if $t \geq 0$ and $v \in \mathbb{R}^n$ satisfies $v \succeq 0$ or $v \preceq 0$, we have

$$f(x + tv) \leq f(x) + t\langle \nabla f(x), v \rangle. \tag{3.5}$$

Remark. Note that (3.5) is the key structural property of DR-submodular functions, analogous to the convexity property ($\forall t \in \mathbb{R}$ and $v \in \mathbb{R}^n$ $f(x + tv) \geq f(x) + t\langle \nabla f(x), v \rangle$) for convex functions which allows for efficient optimization e.g. Algorithm 1 in [7].

3.2 Examples of non-concave DR-submodular functions

Multilinear extension of discrete submodular set functions. A discrete function $F : \{0, 1\}^V \to \mathbb{R}$ is submodular if for all $j \in V$ and for all sets $A \subseteq B \subseteq V \setminus \{j\}$, the following holds:

$$F(A \cup \{j\}) - F(A) \geq F(B \cup \{j\}) - F(B).$$

The multilinear extension $f : [0, 1]^V \to \mathbb{R}$ of F is defined as [8]

$$f(x) = \sum_{S \subset V} F(S) \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j) = \mathbb{E}_{S \sim x}[F(S)].$$

Multilinear extensions are extensively used for maximizing their corresponding discrete submodular set functions and are known to be a special case of non-concave DR-submodular

functions. The Hessian matrix of this class of functions has non-positive off-diagonal entries and all its diagonal entries are zero. It has been shown that multilinear extensions can be efficiently computed for a large class of submodular set functions, for example, weighted matroid rank function, set cover function, probabilistic coverage function and graph cut function (see [15] for more examples and details).

Indefinite quadratic functions. Consider the quadratic function $f(x) = \frac{1}{2}x^T Hx + h^T x + c$. If the matrix H is element-wise non-positive, f is a DR-submodular function.
More generally, if $h_i : \mathbb{R} \to \mathbb{R}$ is concave for all $i \in [n]$ and $\theta_{ij} \leq 0 \ \forall i \neq j$, the following function $f : \mathbb{R}_+^n \to \mathbb{R}$ is DR-submodular:

$$f(x) = \sum_{i=1}^{n} h_i(x_i) + \sum_{i,j:i\neq j} \theta_{ij} x_i x_j.$$

Log-determinant function. Let the function $f : [0,1]^n \to \mathbb{R}$ be defined as

$$f(x) = \log\det\big(\mathrm{diag}(x)(L - I) + I\big),$$

where $L \succeq 0$ is a positive semidefinite matrix and $\mathrm{diag}(x)$ denotes a diagonal matrix with vector x on its diagonal. This function is used as the objective function in Determinantal Point Processes (DPPs). It was proved in [12] that f is a DR-submodular function.

Chapter 4

MOTIVATING APPLICATIONS

In this chapter, in order to illustrate the generality of our framework, we have listed a number of interesting applications that could be cast into our setting.

4.1 Online ad allocation[5]

Consider the following online ad placement problem: At round $t \in [T]$, an advertiser should choose an investment vector $x_t \in R_+^n$ over n different websites where i-th entry of x_t denotes the amount that the advertiser is willing to pay per each click on the ad on the i-th website (i.e., cost per click). In other words, each website has different tiers of ads and choosing x_t corresponds to ordering a certain type of ad. The aggregate cost of investment is determined when the number of clicks the ad receives is revealed. Namely, the cost of such an investment is characterized by p_t where the i-th entry of the vector p_t is the number of clicks the ad on the i-th website receives. The stochastic nature of the number of visitors of these n websites validates our choice of stochastic linear constraint functions. The advertiser needs to balance her total investment against an allotted long-term budget B_T. At round $t \in [T]$, the advertiser's utility function $f_t(x_t)$ is a monotone DR-submodular function with respect to the vector of investments and this function quantifies the overall impressions of the ads. DR-submodularity of the utility function characterizes the diminishing returns property of the impressions. In other words, making an ad more visible will attract proportionally fewer extra viewers because each website shares a portion of its visitors with other websites.

4.2 Online task assignment in crowdsourcing markets.

In this problem, there exists a requester with a limited budget B_T that submit jobs and benefits from them being completed. There are n types of jobs available to be assigned to workers arriving online. At each step $t \in [T]$, a worker arrives and the requester has to assign a bundle $x_t \in \mathcal{X} = \{x \in \mathbb{R}_+^n : 0 \preceq x \preceq 1\}$ of the jobs to the worker. The worker has a cost $[p_t]_i$ $\forall i \in [n]$ for performing one unit of the i-th job where $[p_t]_i$ denotes the i-th entry of vector p_t. The workers' evaluation of the cost of performing each of these n jobs is governed by the fluctuations of the wages in the job market and is stochastic in nature. The rewards obtained by the requester from this job assignment is a DR-submodular function $f_t(x_t)$. The DR property of the utility function captures the diminishing returns of assigning more jobs to the worker, i.e., as the number of assigned jobs to the worker increases, she has less time to devote to each fixed job $i \in [n]$ and therefore, the reward (quality of the completed task) obtained from the worker performing one unit of job i decreases. In other words, if $x \preceq y$, $\nabla_i f(x) \geq \nabla_i f(y)$ $\forall i \in [n]$ holds. The goal is to maximize the overall rewards obtained by the requester while the budget constraint is not violated as well. Note that if the jobs are indivisible, for all $t \in [T]$, the utility function f_t corresponds to the multilinear extension of the monotone submodular set function $F_t : 2^n \to \mathbb{R}$ and using the lossless pipage rounding technique of [8], we allocate an integral bundle of jobs to the workers at each step.

4.3 Online welfare maximization with production cost[14]

In this problem, there are n types of products for sale that may be produced on demand using a fixed limited budget B_T. At each step $t \in [T]$, an agent (customer) arrives online and the algorithm has to assign a bundle $x_t \in \mathcal{X} = \{x \in \mathbb{R}_+^n : 0 \preceq x \preceq 1\}$ of products to the agent. Producing each unit of product $i \in [n]$ costs an unknown amount $[p_t]_i$ and the production cost of the item may change over time $\{1, \ldots, T\}$ because of the stochastic fluctuations of the prices of ingredients. The agent has an unknown private DR-submodular valuation function f_t over the items where the DR property characterizes the diversity of the

assigned bundle. Therefore, the utility obtained by assigning the bundle x_t equals $f_t(x_t)$. The goal is to maximize the overall valuation of the agents while satisfying the budget constraint. Note that if the products are indivisible, for all $t \in [T]$, the utility function f_t corresponds to the multilinear extension of the monotone submodular set function $F_t : 2^n \to \mathbb{R}$ and using the lossless pipage rounding technique of [8], we allocate an integral bundle of products to the agents at each step.

Chapter 5

MAIN CONTRIBUTIONS

5.1 Problem statement

Consider the following overall offline optimization problem:

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{t=1}^{T} f_t(x_t) \\ \text{subject to} \quad & x_t \in \mathcal{X}, \ \forall t \in [T] \\ & \textstyle\sum_{t=1}^{T} \langle p, x_t \rangle \leq B_T. \end{aligned} \tag{5.1}$$

The online setup is as follows: At each round $t \in [T]$, the algorithm chooses an action $x_t \in \mathcal{X}$, where $\mathcal{X} \subset \mathbb{R}_+^n$ is a fixed, known set. Upon committing to this action, the utility function $f_t : \mathcal{X} \to \mathbb{R}_+$ and a random i.i.d. sample $p_t \sim \mathcal{D}(p, \Sigma)$ are revealed and the algorithm receives a reward of $f_t(x_t)$ while using $\langle p, x_t \rangle$ of its fixed total allotted budget B_T. The overall goal is to maximize the total obtained reward while satisfying the budget constraint asymptotically (i.e., $\sum_{t=1}^{T} \langle p, x_t \rangle - B_T$ being sub-linear in T). [21] considered a similar setup and performance metric for the special case of linear utility functions.

Note that our proposed algorithm can handle multiple linear constraints as well, and similar regret and constraint violation bounds can be derived. However, for ease of notation, we focus on the case with only one linear constraint.

We make the following assumptions about our problem framework:

A1. The domain $\mathcal{X} \subset \mathbb{R}_+^n$ is a closed, bounded, convex set containing the origin, i.e., $0 \in \mathcal{X}$. We denote the diameter of $\mathcal{X}$ with R; i.e., $R := \max_{x,y \in \mathcal{X}} \|y - x\|$.

A2. For all $t \in [T]$, the utility function $f_t(\cdot)$ is normalized (i.e., $f_t(0) = 0$), monotone, DR-submodular, β_f-Lipschitz and L-smooth. In other words, for all $x, y \in \mathcal{X}$ and $u \in \mathbb{R}^n$

where $u \succeq 0$ or $u \preceq 0$, the following holds:

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

A3. For all $t \in [T]$, $p_t \in \mathbb{R}_+^n$ is i.i.d. generated from the distribution $\mathcal{D}$ with bounded support $\beta_p \mathcal{B} \cap \mathbb{R}_+^n$, mean $p \succeq 0$ and covariance matrix Σ, i.e., $p_t \sim \mathcal{D}(p, \Sigma)$.
Let $\beta = \max\{\beta_f, \beta_p\}$. Under the above assumptions, we have:

$$F := \max_{t \in [T]} \max_{x,y \in \mathcal{X}} |f_t(x) - f_t(y)| \leq \beta R < \infty$$
$$G := \max_{p' \sim \mathcal{D}(p,\Sigma)} \max_{x \in \mathcal{X}} |\langle p', x \rangle - \frac{B_T}{T}| \leq \beta R - \frac{B_T}{T} < \infty.$$

5.1.1 Performance metric

We characterize the performance of our proposed algorithm through bounding the notions of regret and cumulative constraint violation which are defined below:

Definition 5. *The* $(1 - \frac{1}{e})$*-regret is defined as:*

$$R_T = (1 - \frac{1}{e}) \max_{x \in \mathcal{X}^*} \sum_{t=1}^{T} f_t(x) - \sum_{t=1}^{T} f_t(x_t),$$

where:

$$\mathcal{X}^* = \{x \in \mathcal{X} : \sum_{t=1}^{T} \langle p, x \rangle \leq B_T\} = \{x \in \mathcal{X} : \langle p, x \rangle \leq \frac{B_T}{T}\}.$$

The regret metric R_T quantifies the difference of the reward obtained by the algorithm and the $(1 - \frac{1}{e})$-approximation of the reward of the best fixed benchmark action that has access to all the utility functions f_t $\forall t \in [T]$, the *mean* p of the linear constraint functions, and satisfies the cumulative budget constraint. Note that $1 - \frac{1}{e}$ is the optimal approximation ratio for offline continuous DR-submodular maximization; in other words, even if all the online input were available beforehand, we could only obtain a $(1 - \frac{1}{e})$ fraction of the maximum reward in polynomial time. The $(1 - \frac{1}{e})$-regret is commonly used in the online submodular maximization literature (e.g., [10]).

Paper	Cost (utility)	Constraint	Window size	R_T	C_T
[33]	convex	convex (fixed)	—	$\mathcal{O}(\sqrt{T})$	$\mathcal{O}(T^{\frac{3}{4}})$
[31]	convex	convex (stochastic)	—	$\mathcal{O}(\sqrt{T})$	$\mathcal{O}(\sqrt{T})$
[23]	convex	convex (adversarial)	1	$\mathcal{O}(\sqrt{T})$	$\mathcal{O}(\sqrt{T})$
[19]$^{(a)}$	convex	convex (adversarial)	W	$\mathcal{O}(\sqrt{T}+\frac{WT}{V})$	$\mathcal{O}(\sqrt{VT})$
[28]	DR-submodular	linear (adversarial)	W	$\mathcal{O}(\sqrt{WT})$	$\mathcal{O}(W^{\frac{1}{4}}T^{\frac{3}{4}})$

Table 5.1: State of the art results for online problems with cumulative constraints in various settings. Note that in (a), $V \in (W, T)$ is a tunable parameter.

Definition 6. *The cumulative constraint violation is defined as follows:*

$$C_T = \sum_{t=1}^{T} \langle p, x_t \rangle - B_T.$$

Note that since p_t $\forall t \in [T]$ is i.i.d. drawn from the distribution $\mathcal{D}$ with mean p, our cumulative constraint violation metric C_T is defined with respect to the true underlying fixed linear constraint p (as opposed to p_t).

5.1.2 Related work

Consider the following general framework of online problems with long-term constraints: At round $t \in [T]$, the player chooses $x_t \in \mathcal{X}$. Then, cost (utility) function $f_t : \mathcal{X} \to \mathbb{R}$ ($\mathcal{X}$ is a fixed convex set) and constraint function $g_t : \mathcal{X} \to \mathbb{R}$ are revealed, the player incurs a loss (obtains a reward) of $f_t(x_t)$ and uses the amount $g_t(x_t)$ of her budget (with the long-term constraint $\sum_{t=1}^{T} g_t(x_t) \leq 0$). This problem has been extensively studied under various assumptions where the cost (utility) functions are adversarially chosen and are assumed to be linear, convex or DR-submodular and the constraint functions are linear or convex and are either fixed (i.e., $g_t(\cdot) = g(\cdot)$ $\forall t \in [T]$), stochastic and i.i.d drawn from some unknown distribution, or adversarial. For the setting with adversarial utility and constraint functions, [22] provided

a simple counterexample to show that regardless of the decisions of the algorithm, it is impossible to guarantee sub-linear regret against the benchmark action while the overall budget violation is sub-linear. Therefore, prior works in this setting have further restricted the fixed comparator action to be chosen from $\mathcal{X}_W = \{x \in \mathcal{X} : \sum_{\tau=t}^{t+W-1} g_\tau(x) \leq 0,\ 1 \leq t \leq T-W+1\}$. In other words, in addition to merely satisfying the overall cumulative constraint (which corresponds to the $W = T$ case), the benchmark action is required to satisfy the budget constraint proportionally over any window of length W. On the other hand, for fixed or stochastic constraint functions, sub-linear regret and constraint violation bounds have been derived in the literature. A summary of the state of the art results for online problems with long-term constraints is provided in Table 1.

5.2 Online Lagrangian Frank-Wolfe (OLFW) algorithm

In this section, we first introduce our proposed algorithm, namely the Online Lagrangian Frank-Wolfe (OLFW) algorithm, in Section 5.2.1 and subsequently, we analyze the performance of the algorithm with high probability and in expectation in Section 5.2.2 and Section 5.2.3 respectively. All missing proofs are provided in the appendix.

5.2.1 Algorithm

The Online Lagrangian Frank-Wolfe (OLFW) algorithm is presented in Algorithm 5. First, note that for all $t \in [T]$, $x_t = \frac{1}{K}\sum_{k=1}^{K} v_t^{(k)}$ is the average of vectors in the convex domain $\mathcal{X}$ and hence, $x_t \in \mathcal{X}$. The intuition for using K online maximization subroutines to update x_t is the Frank-Wolfe variant proposed in [7] to obtain the optimal approximation guarantee of $1 - \frac{1}{e}$ for solving the offline DR-submodular maximization problem without the additional linear constraints. To be more precise, consider the first iteration $t = 1$ of our online setting (ignoring the linear cumulative constraints) and the corresponding DR-submodular utility function $f_1(\cdot)$ arriving at this step. Note that f_1 is not revealed until the algorithm commits to an action $x_1 \in \mathcal{X}$. If we were in the offline setting, we could use the mentioned Frank-Wolfe variant of [7], run it for K iterations and maximize f_1 over $\mathcal{X}$. Starting from $x_1^{(1)} = 0$, for all

$k \in [K]$, we would find a vector $v_1^{(k)}$ that maximizes $\langle x, \nabla f_1(x_1^{(k)})\rangle$ over $x \in \mathcal{X}$, perform the update $x_1^{(k+1)} = x_1^{(k)} + \frac{1}{K} v_1^{(k)}$ and derive $x_1 = x_1^{(K+1)}$ as the output. However, in the online setting, the utility function f_1 is not available before committing to the action x_1. Therefore, for each $k \in [K]$, we instead use a separate instance of a no-regret online linear maximization algorithm to obtain $v_1^{(k)}$. We repeat the same process for the subsequent utility functions $\{f_t\}_{t>1}$. This intuition was first provided in [10] and they managed to obtain an $\mathcal{O}(\sqrt{T})$ regret bound for the unconstrained online monotone submodular maximization problem.

Our choice of Lagrangian function is inspired by the quadratic penalty method in constrained optimization [24]. The penalized formulation of the overall optimization problem (5.1) with quadratic penalty function could be written as follows:

$$\begin{aligned} \max_{x_t} \quad & \sum_{t=1}^{T} f_t(x_t) - \frac{1}{2\delta\mu}\Big(\sum_{t=1}^{T}\langle p, x_t\rangle - B_T\Big)^2 \\ \text{subject to} \quad & x_t \in \mathcal{X} \ \forall t \in [T]. \end{aligned}$$

Considering that the Fenchel conjugate of the function $h(\cdot) = \frac{1}{2\delta\mu}(\cdot)^2$ is $h^*(\cdot) = \frac{\delta\mu}{2}(\cdot)^2$, we can write the above problem in the following equivalent form:

$$\begin{aligned} \max_{x_t}\min_{\lambda} \quad & \sum_{t=1}^{T} f_t(x_t) - \lambda\Big(\sum_{t=1}^{T}\langle p, x_t\rangle - B_T\Big) + \frac{\delta\mu}{2}\lambda^2 \\ \text{subject to} \quad & x_t \in \mathcal{X} \ \forall t \in [T]. \end{aligned}$$

Therefore, the corresponding Lagrangian function at round $t \in [T]$ is $\mathcal{L}_t(x, \lambda) = f_t(x) - \lambda(\langle p, x\rangle - \frac{B_T}{T}) + \frac{\delta\mu}{2}\lambda^2$. However, p is unknown to the online algorithm. Therefore, we alternatively use $\widehat{p}_t := \frac{1}{t-1}\sum_{s=1}^{t-1} p_s$ instead of p in the Lagrangian function. Note that $\widehat{p}_t$ is the empirical estimation of p at round t.

Remark. Here, δ is a parameter to balance the trade-off between the cumulative utility and total contraint violation and, its optimal value is determined through the analysis. Also, μ is the step size used in the algorithm.

We first provide a lemma which is central to obtaining the regret and constraint violation bounds both in expectation and with high probability.

Algorithm 5 Online Lagrangian Frank-Wolfe (OLFW)

Input: $\mathcal{X}$ is the constraint set, T is the horizon, $\mu > 0$, $\delta > 0$, $\{\gamma_t\}_{t=1}^T$ and K.

Output: $\{x_t : 1 \leq t \leq T\}$.

Initialize K instances $\{\mathcal{E}_k\}_{k\in[K]}$ of Online Gradient Ascent with step size μ for online maximization of linear functions over $\mathcal{X}$.

for $t = 1$ **to** T **do**

  $x_t^{(1)} = 0$.

  for $k = 1$ **to** K **do**

    Let $v_t^{(k)}$ be the output of oracle $\mathcal{E}_k$ from round $t-1$.

    $x_t^{(k+1)} = x_t^{(k)} + \frac{1}{K} v_t^{(k)}$.

  end for

  Set $x_t = x_t^{(K+1)}$.

  Let $\widehat{p}_t := \frac{1}{t-1}\sum_{s=1}^{t-1} p_s$ for $t > 1$.

  Let

$$\widetilde{g}_t(\cdot) = \begin{cases} \langle \widehat{p}_t, \cdot \rangle - \frac{B_T}{T} & \text{expectation analysis (I)} \\ \langle \widehat{p}_t, \cdot \rangle - \frac{B_T}{T} - \gamma_t & \text{high probability analysis (II)} \end{cases}.$$

  Set $\lambda_t = \frac{[\widetilde{g}_t(x_t)]_+}{\delta\mu}$ for $t > 1$ and 0 otherwise.

  Play x_t and observe the Lagrangian function $\mathcal{L}_t(x_t, \lambda_t) = f_t(x_t) - \lambda_t \widetilde{g}_t(x_t) + \frac{\delta\mu}{2}\lambda_t^2$

  for $k = 1$ **to** K **do**

    Feedback $\langle v_t^{(k)}, \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t)\rangle$ as the payoff to be received by $\mathcal{E}_k$.

  end for

end for

Lemma 6. *Let $x \in \mathcal{X}$ be a fixed vector. In the OLFW algorithm, set $\delta = \beta^2$. We then have:*

$$\sum_{t=1}^{T} \big((1-\frac{1}{e})f_t(x) - f_t(x_t)\big) \leq \frac{LR^2T}{2K} + \frac{R^2}{\mu} + \beta^2\mu T + \sum_{t=1}^{T} \lambda_t \widetilde{g}_t(x). \tag{5.2}$$

5.2.2 Performance analysis with high probability

In order to analyze the performance of the OLFW algorithm with high probability, the following lemmas detailing the concentration inequalities for the stochastic linear constraints are used.

Lemma 7. *[18] For all $t \in [T]$, the following holds:*

$$\mathbb{P}\{\|p_t - p\|_2 \geq \zeta\} \leq 2e^{-\frac{\zeta^2}{2\sigma^2}} \quad \forall \zeta \in \mathbb{R}.$$

Lemma 8. *For $t = 2, 3, \ldots, T$, the following holds with probability at least $1 - \frac{\epsilon}{T}$:*

$$\|\widehat{p}_t - p\| \leq c'\sigma\sqrt{\frac{\log(\frac{2nT}{\epsilon})}{t-1}}.$$

Lemma 9. *The following holds with probability at least $1 - \epsilon$:*

$$\sum_{t=2}^{T} \|\widehat{p}_t - p\| \leq C\sigma\sqrt{T\log\left(\frac{2nT}{\epsilon}\right)}.$$

Lemma 10. *Let $x \in \mathcal{X}$ be fixed. Define $\widehat{g}_t(x) := \langle \widehat{p}_t, x\rangle - \frac{B_T}{T}$ and $g(x) := \langle p, x\rangle - \frac{B_T}{T}$. For a fixed $t \in \{2, 3, \ldots, T\}$ and $\{\gamma_t := \sqrt{\frac{2G^2\log(\frac{2T}{\epsilon})}{t}}\}_{t=2}^{T}$, $|\widehat{g}_t(x) - g(x)| \leq \gamma_t$ holds with probability at least $1 - \frac{\epsilon}{T}$.*

Now, we have all the machinery to obtain the high probability performance bounds of the OLFW algorithm.

Theorem 2. ***(High probability regret bound)*** *Let $\epsilon \in (0, 1)$ be given. Set $\mu = \frac{R}{\beta\sqrt{T}}$, $K = \sqrt{T}$, $\delta = \beta^2$ and $\{\gamma_t\}_{t=2}^{T}$ be chosen according to Lemma 10. Then, the OLFW algorithm with update (II) for $\widetilde{g}_t(\cdot)$ obtains the following regret bound with probability at least $1 - \epsilon$.*

$$R_T \leq \left(\frac{LR^2}{2} + 2R\beta\right)\sqrt{T}.$$

Proof. We begin from Lemma 6. Substitute the benchmark $x = x^*$ as the fixed vector in (5.2) and the constants as given in the hypothesis. We get: $R_T \leq (\frac{LR^2}{2} + 2R\beta)\sqrt{T} + \sum_{t=1}^{T} \lambda_t \widetilde{g}_t(x^*)$. Now let us bound $\sum_{t=1}^{T} \lambda_t \widetilde{g}_t(x^*)$. From Lemma 10, we have that with probability at least $1 - \frac{\epsilon}{T}$, $\widehat{g}_t(x^*) - \gamma_t \leq g(x^*)$ holds, i.e., $\widetilde{g}_t(x^*) \leq g(x^*)$. Also, $g(x^*) \leq 0$ holds according to the definition of the benchmark action. Therefore, we have: $\widetilde{g}_t(x^*) \leq 0$. As $\lambda_t \geq 0$, $\lambda_t \widetilde{g}_t(x^*) \leq 0$ holds. Now, taking union bound over all $t \in [T]$, we have with probability at least $1 - \epsilon$ that $\sum_{t=1}^{T} \lambda_t \widetilde{g}_t(x^*) \leq 0$. The result follows immediately. ■

We will use the following lemma to get performance bounds for the constraint violation.

Lemma 11. *Let $\{\gamma_t\}_{t=2}^{T}$ be defined as in Lemma 10, then the following holds.*

$$C_T \leq \sum_{t=1}^{T} [\widetilde{g}_t(x_t)]_+ + R \sum_{t=1}^{T} \|\widehat{p}_t - p\| + \sum_{t=2}^{T} \gamma_t,$$

where $\widetilde{g}_t(\cdot)$ is derived using update (II).

Theorem 3. ***(High probability constraint violation bound)*** *Let $\epsilon \in (0, 1)$ be given. Set $\mu = \frac{R}{\beta\sqrt{T}}$, $K = \sqrt{T}$, $\delta = \beta^2$ and $\{\gamma_t\}_{t=2}^{T}$ be chosen according to Lemma 10. Then the following holds with probability at least $1 - \epsilon$ for the OLFW algorithm with update rule (II).*

$$C_T \leq \sqrt{2G^2 T \log\left(\frac{2T}{\epsilon}\right)} + CR\sigma\sqrt{T \log\left(\frac{2nT}{\epsilon}\right)} + \frac{T}{B_T} R\beta F\sqrt{T} + \frac{TR\beta}{B_T}\left(\frac{LR^2}{2} + 2R\beta\right) + R\beta.$$

So, we obtain $\tilde{\mathcal{O}}(\sqrt{T})$ constraint violation bound with high probability.

Proof. We begin with Lemma 6 again but now substitute $x = 0$ as the fixed vector in (5.2).

$$\frac{B_T}{T} \sum_{t=1}^{T} \lambda_t + \sum_{t=1}^{T} \lambda_t \gamma_t \leq \underbrace{\sum_{t=1}^{T} f_t(x_t)}_{\leq FT} + \frac{LR^2 T}{2K} + \frac{R^2}{\mu} + \beta^2 \mu T. \tag{5.3}$$

Rearranging and substituting the values of input parameters as given in the hypothesis, we get:

$$\sum_{t=1}^{T}[\widetilde{g}_t(x_t)]_+ + \frac{T\delta\mu}{B_T}\sum_{t=1}^{T}\lambda_t\gamma_t \leq \frac{T}{B_T}R\beta F\sqrt{T} + \frac{TR\beta}{B_T}\left(\frac{LR^2}{2} + 2R\beta\right).$$

Both terms in the left hand side of the above equation are positive. Thus, we can drop the second term. We have:

$$\sum_{t=1}^{T}[\widetilde{g}_t(x_t)]_+ \leq \frac{T}{B_T}R\beta F\sqrt{T} + \frac{TR\beta}{B_T}\left(\frac{LR^2}{2} + 2R\beta\right).$$

Combining Lemma 11 and the equation above, we obtain:

$$C_T \leq \frac{T}{B_T}R\beta F\sqrt{T} + \frac{TR\beta}{B_T}\left(\frac{LR^2}{2} + 2R\beta\right) + R\sum_{t=1}^{T}\|\widehat{p}_t - p\| + \sum_{t=2}^{T}\gamma_t.$$

Therefore, we can conclude:

$$C_T \leq \frac{T}{B_T}R\beta F\sqrt{T} + \frac{TR\beta}{B_T}\left(\frac{LR^2}{2} + 2R\beta\right) + \sqrt{2G^2T\log\left(\frac{2T}{\epsilon}\right)} + \underbrace{R\sum_{t=1}^{T}\|\widehat{p}_t - p\|}_{\text{(A)}},$$

where the last inequality follows from summing γ_t's. Now, Lemma 9 tells us that (A) $\leq R\beta + CR\sigma\sqrt{T\log(\frac{2nT}{\epsilon})}$ holds with probability at least $1-\epsilon$. Thus, we get the result. ■

Theorem 2 and Theorem 3 are indeed the first high probability bounds obtained for the online DR-submodular maximization problem with stochastic cumulative constraints. Note that the $\mathcal{O}(\sqrt{T})$ regret bound obtained in Theorem 2 is known to be optimal.

5.2.3 Performance analysis in expectation

We first provide a simple lemma that will be used throughout the analysis in expectation.

Lemma 12. *For $t > 1$, we have:*

$$\mathbb{E}\|\widehat{p}_t - p\|^2 = \frac{Tr(\Sigma)}{t-1},$$

where $Tr(\Sigma)$ denotes the trace of the covariance matrix Σ.

Now, we present the main performance bounds in expectation, namely the expected regret bound and the expected cumulative constraint violation bound. In the Appendix, we have also considered the case where we only have access to unbiased stochastic gradient estimates of the utility functions $\{f_t\}_{t=1}^T$ and exact gradient computation is not possible. For this setting, we modify the OLFW algorithm through incorporating the variance reduction technique introduced by [9] and we obtain similar regret and constraint violation bounds in expectation for the modified algorithm.

Theorem 4. *(**Expected regret bound**) The regret bound of the OLFW algorithm with update rule (I) is the following:*

$$\mathbb{E}[R_T] \leq \tilde{\mathcal{O}}(T^{\frac{3}{4}}).$$

Proof. We first observe from (5.3) that $\sum_{t=1}^T \lambda_t \leq \mathcal{O}(T)$. Now, substitute $x = x^*$, the benchmark, in (5.2) and take expectation on both sides to obtain:

$$\begin{aligned}
\mathbb{E}[R_T] &\leq \frac{LR^2T}{2K} + \frac{R^2}{\mu} + \beta^2\mu T + \mathbb{E}[\sum_{t=1}^T \lambda_t(\widehat{g}_t(x^*) - g(x^*))] + \sum_{t=1}^T \lambda_t g(x^*) \\
&\leq \frac{LR^2T}{2K} + \frac{R^2}{\mu} + \beta^2\mu T + \underbrace{\mathbb{E}\left[\sum_{t=1}^T \lambda_t(\widehat{g}_t(x^*) - g(x^*))\right]}_{\text{(B)}}.
\end{aligned}$$

Now we bound (B) as follows:

$$\begin{aligned}
(\mathrm{B}) &= \sum_{t=1}^{T} \lambda_t(\widehat{g}_t(x^*) - g(x^*)) \\
&\leq \sqrt{\sum_{t=1}^{T} \lambda_t^2} \sqrt{\sum_{t=1}^{T} (\widehat{g}_t(x^*) - g(x^*))^2} \\
&= \sqrt{\|\lambda\|^2} \sqrt{\sum_{t=1}^{T} (\langle \widehat{p}_t - p, x^* \rangle)^2} \\
&\leq \|\lambda\| \sqrt{\sum_{t=1}^{T} \|\widehat{p}_t - p\|^2 R^2} \\
&= R\|\lambda\| \sqrt{\sum_{t=1}^{T} \|\widehat{p}_t - p\|^2}.
\end{aligned}$$

Both the inequalities above are obtained using Cauchy-Schwarz inequality, where $\lambda := [\lambda_1, \lambda_2, \ldots, \lambda_T]^T$.

Using the Cauchy-Schwarz inequality again, we have $\|\lambda\| \leq \sqrt{\|\lambda\|_1 \|\lambda\|_\infty}$. Thus, we obtain $\|\lambda\| \leq \sqrt{(\sum_{t=1}^{T} \lambda_t)(\frac{G}{\delta\mu})} \leq \mathcal{O}(T^{3/4})$. Therefore, the following holds:

$$\|\lambda\| \leq \mathcal{O}(T^{3/4}). \tag{5.4}$$

Using Jensen's inequality, we have

$$\mathbb{E}\sqrt{\sum_{t=1}^{T} \|\widehat{p}_t - p\|^2} \leq \sqrt{\sum_{t=1}^{T} \mathbb{E}\|\widehat{p}_t - p\|^2}.$$

We can use Lemma 12 and write:

$$\mathbb{E}\sqrt{\sum_{t=1}^{T} \|\widehat{p}_t - p\|^2} \leq \sqrt{Tr(\Sigma)\log(T)}. \tag{5.5}$$

Thus, combining (5.4) and (5.5), we obtain:

$$\mathbb{E}[(\mathrm{B})] \leq \mathcal{O}(T^{3/4}\sqrt{Tr(\Sigma)\log(T)}).$$

The result thus follows. ■

Remark. The main challenge in bounding R_T in expectation is the fact that in our algorithm, the choice of λ_t is dependent on $\widehat{p}_t$ and thus, we cannot use

$$\mathbb{E}[\lambda_t \widehat{g}_t(x^*)] = \mathbb{E}[\lambda_t \mathbb{E}[\widehat{g}_t(x^*)]] = \mathbb{E}[\lambda_t g(x^*)] \leq 0$$

and this term is indeed the dominating term in the regret bound. However, as we saw earlier, we do not encounter this problem in the high probability setting due to subtracting γ_t from all the constraint functions and using the concentration inequalities, and thus we were able to obtain $\mathcal{O}(\sqrt{T})$ high probability regret bound.

Theorem 5. *(**Expected cumulative constraint violation bound**) For the OLFW algorithm with update rule (I), we have:*

$$\mathbb{E}[C_T] \leq \frac{T}{B_T} R\beta F\sqrt{T} + R\sqrt{Tr(\Sigma)}\sqrt{T} + \frac{TR\beta}{B_T}\left(\frac{LR^2}{2} + 2R\beta\right) + R\beta.$$

Therefore, $\mathbb{E}[C_T] \leq \tilde{\mathcal{O}}(\sqrt{T})$ holds.

Proof. Similar to the proof of Theorem 2, we begin by setting $x = 0$ to be the fixed vector in (2). We obtain

$$\sum_{t=1}^{T} [\widetilde{g}_t(x_t)]_+ \leq \frac{T}{B_T} R\beta F\sqrt{T} + \frac{TR\beta}{B_T}\left(\frac{LR^2}{2} + 2R\beta\right).$$

Now, we lower bound the left-hand side following the idea of the proof for Lemma 4. Thus, we obtain

$$C_T \leq \frac{T}{B_T} R\beta F\sqrt{T} + \frac{TR\beta}{B_T}\left(\frac{LR^2}{2} + 2R\beta\right) + \underbrace{R\sum_{t=1}^{T} \|\widehat{p}_t - p\|}_{\text{(C)}}.$$

In order to bound (C), we take expectation on both sides to obtain:

$$\begin{aligned}
\mathbb{E}[(\mathrm{C})] &= R\mathbb{E}[\sum_{t=2}^{T} \|\widehat{p}_t - p\|] \\
&= R\sum_{t=2}^{T} \mathbb{E}\|\widehat{p}_t - p\| \\
&= R\sum_{t=2}^{T} \mathbb{E}\sqrt{\|\widehat{p}_t - p\|^2} \\
&\leq R\sum_{t=2}^{T} \sqrt{\mathbb{E}\|\widehat{p}_t - p\|^2},
\end{aligned}$$

where the last inequality is due to Jensen's inequality.

Therefore, we have:

$$\begin{aligned}
\mathbb{E}[(\mathrm{C})] &\leq R\sum_{t=2}^{T} \frac{\sqrt{Tr(\Sigma)}}{\sqrt{t-1}} \\
&\leq R\sqrt{Tr(\Sigma)}\sqrt{T}.
\end{aligned}$$

Combining the inequalities, we obtain the result. Theorem 4 and Theorem 5 provide the first sub-linear expectation bounds on the regret and cumulative constraint violation of the online DR-submodular maximization problem with stochastic cumulative constraints.

5.2.4 Modified Online Lagrangian Frank-Wolfe Algorithm

For the setting where only unbiased stochastic gradient estimates of the utility functions $\{f_t\}_{t=1}^{T}$ with bounded variance σ^2 are available, inspired by the variance reduction technique introduced by [9], we propose the Modified Online Lagrangian Frank-Wolfe (MOLFW) algorithm in Algorithm 2. Compared to the OLFW algorithm, the MOLFW algorithm uses $\langle v_t^{(k)}, d_t^{(k)}\rangle$ (instead of $\langle v_t^{(k)}, \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t)\rangle$) as the payoff to be received by $\mathcal{E}_k$ $\forall k \in [K]$ at round $t \in [T]$. The analysis of this algorithm is similar to that of OLFW and the corresponding expected regret and constraint violation bounds can be obtained.

Algorithm 6 Modified Online Lagrangian Frank-Wolfe (MOLFW)

Input: $\mathcal{X}$ is the constraint set, T is the horizon, $\mu > 0$, $\delta > 0$, $\{\gamma_t\}_{t=1}^T$, $\{\rho_k\}_{k=1}^K$ and K.

Output: $\{x_t : 1 \leq t \leq T\}$.

Initialize K instances $\{\mathcal{E}_k\}_{k\in[K]}$ of Online Gradient Ascent with step size μ for online maximization of linear functions over $\mathcal{X}$.

for $t = 1$ **to** T **do**

  $x_t^{(1)} = 0$.

  for $k = 1$ **to** K **do**

    Let $v_t^{(k)}$ be the output of oracle $\mathcal{E}_k$ from round $t-1$.

    $x_t^{(k+1)} = x_t^{(k)} + \frac{1}{K} v_t^{(k)}$.

  end for

  Set $x_t = x_t^{(K+1)}$.

  Let $\widehat{p}_t := \frac{1}{t-1}\sum_{s=1}^{t-1} p_s$ for $t > 1$.

  Let

$$\widetilde{g}_t(\cdot) = \begin{cases} \langle \widehat{p}_t, \cdot \rangle - \frac{B_T}{T} & \text{expectation analysis (I)} \\ \langle \widehat{p}_t, \cdot \rangle - \frac{B_T}{T} - \gamma_t & \text{high probability analysis (II)} \end{cases}$$

  Set $\lambda_t = \frac{[\widetilde{g}_t(x_t)]_+}{\delta\mu}$ for $t > 1$ and 0 otherwise.

  Play x_t and observe the Lagrangian function $\mathcal{L}_t(x_t, \lambda_t) = f_t(x_t) - \lambda_t \widetilde{g}_t(x_t) + \frac{\delta\mu}{2}\lambda_t^2$

  $d_t^{(0)} = -\lambda_t \widehat{p}_t$.

  for $k = 1$ **to** K **do**

    $d_t^{(k)} = (1-\rho_k) d_t^{(k-1)} + \rho_k \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t)$.

    Feedback $\langle v_t^{(k)}, d_t^{(k)} \rangle$ as the payoff to be received by $\mathcal{E}_k$.

  end for

end for

Chapter 6

NUMERICAL EXPERIMENTS AND CONCLUSION

6.1 Numerical Results

In this section, we present the results of numerical experiments conducted over simulated and real-world datasets.

6.1.1 Joke Recommendation

We look at the problem of DR-submodular function maximization over the *Jester* dataset[1]. We consider a fraction of the dataset where there are 100 jokes and user ratings from 10000 users are available for these jokes. The ratings take values in $[-10, 10]$, we re-scale them to be in $[0, 10]$. Let $R_{u,j}$ be the rating of user u for joke j. As some of the user ratings are missing in the dataset, we set such ratings to be 5. In the online setting, a user arrives and we have to recommend at most $M = 15$ jokes to her. The utility function for each round $t \in [T]$ is of the form $f_t(x) = \sum_{i=1}^{100} R^t_{u_t,i} x_i + \sum_{i,j:i\neq j} \theta_{ij} x_i x_j$, where u_t is the user being served in the current round. $\{\theta_{ij}\}_{i\neq j}$ are chosen such that the function is monotone. These DR-submodular utility functions capture the overall impression of the displayed jokes on the user. There is a limited total time (denoted by $B_T = 1.5T$) available to recommend the jokes to the users. For all $i \in [n], p_i$ denotes the average time it takes to read joke i. As some jokes are relatively longer, we do not want the user to spend more time on jokes which do not lead to larger utility. The linear budget functions are chosen randomly with entries uniformly drawn from $[0.03, 0.35]$. We compare the performance of our algorithm against the following strategies:

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

- *Uniform*: At every round, we assign 15 randomly chosen jokes to the user.

- *Greedy*: We deploy an exploration-exploitation strategy where with probability 0.1, we randomly assign 15 jokes and with probability 0.9, we present the top 15 jokes based on the ratings observed so far.

- *Meta-FW* [10]: This corresponds to solving the unconstrained DR-submodular maximization problem (i.e., ignoring the budget constraints).

- *Budget-Cautious*: At each round, we assign 15 jokes which have the lowest average budget consumption observed so far.

The results are presented in Figure 6.1. As it can be seen in the plots, our OLFW algorithm obtains a reasonable utility while approximately satisfying the budget constraint as well.

6.1.2 Indefinite quadratic functions.

We choose $\mathcal{X} = \{x \in \mathbb{R}^2 : 0 \preceq x \preceq 1\}$ and for each $t \in [T]$, we generate quadratic functions of the form $f_t(x) = \frac{1}{2}x^T H_t x + h_t^T x$ where $H_t \in \mathbb{R}^{2\times 2}$ is a random matrix whose entries are chosen uniformly from $[-1, 0]$. We let $h_t = -H_t^T \mathbf{1}$ to ensure the monotonocity of the objective. We let $T = 1000$. At each round, we randomly generate linear budget functions whose entries are chosen uniformly from $[0.5, 2.5]$ and the mean vector is $p = [1, 2]^T$. Also, we set the total budget to be $B_T = 2T$. We run the OLFW algorithm 10 times and take the respective averages for the cumulative utility and total remaining budget. We vary δ, the parameter of the penalty function, in the range $[0.1, 1000]$ and plot the trade-off curve $\big($i.e., $\sum_{t=1}^{1000} f_t(x_t)$ versus $B_T - \sum_{t=1}^{1000} \langle p, x_t \rangle\big)$ for 100 chosen values of δ in Figure 6.2. In this example, our choice of δ in the OLFW algorithm, highlighted in the plot, achieves the highest possible cumulative utility while satisfying the total budget constraint.

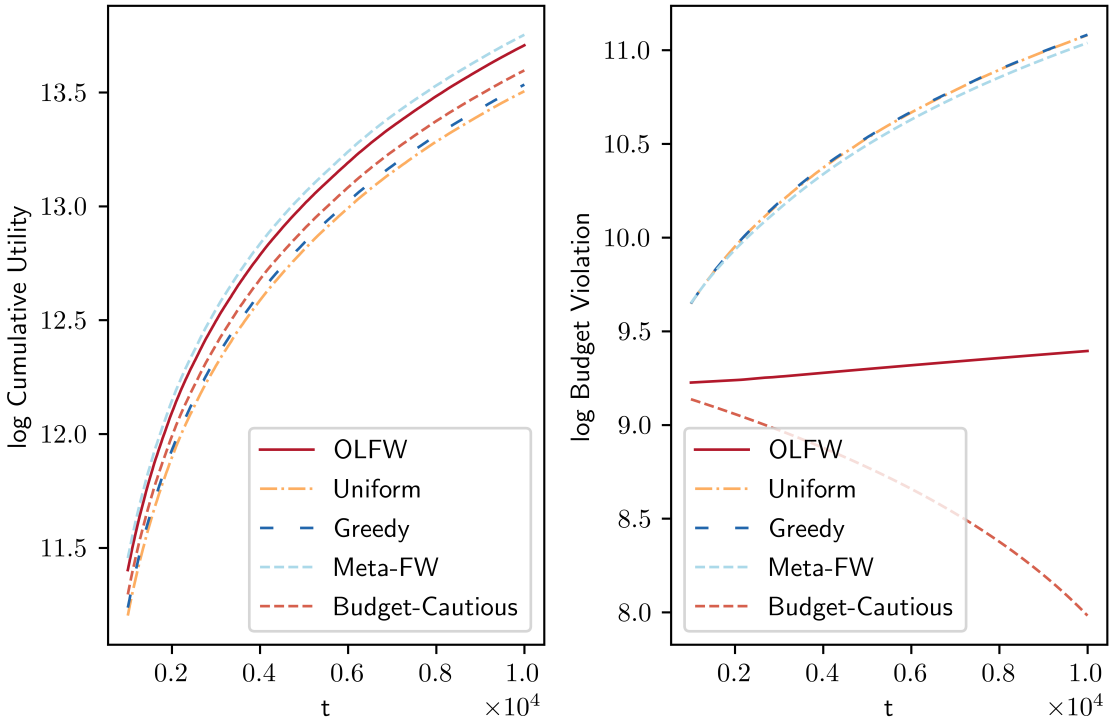

Figure 6.1: Comparison of the overall utility $\sum_{t=1}^{T} f_t(x_t)$ and cumulative budget violation $\sum_{t=1}^{T} \langle p, x_t \rangle - B_T$ for the Jester dataset.

6.1.3 Log-determinant functions.

We choose $\mathcal{X} = \{x \in \mathbb{R}^{10} : 0 \preceq x \preceq 1\}$ and for each $t \in [T]$, we generate log-determinant functions of the form $f_t(x) = \log\det\big(\mathrm{diag}(x)(L_t - I) + I\big)$, where each L_t is a random positive definite matrix with eigenvalues falling in the range $[2, 3]$. The choice of eigenvalues ensures the monotonocity of the function. Let $T = 4900$. At each round, we generate linear budget functions whose entries are chosen uniformly from the range $[0.3, 5.7]$. We run the OLFW algorithm for different choices of the step size μ and plot the cumulative utility and the total budget violation in Figure 6.3. Our OLFW algorithm chooses μ such that the overall utility and cumulative budget consumption are balanced.

Remark. It is interesting to note that in the OLFW algorithm, the online linear max-

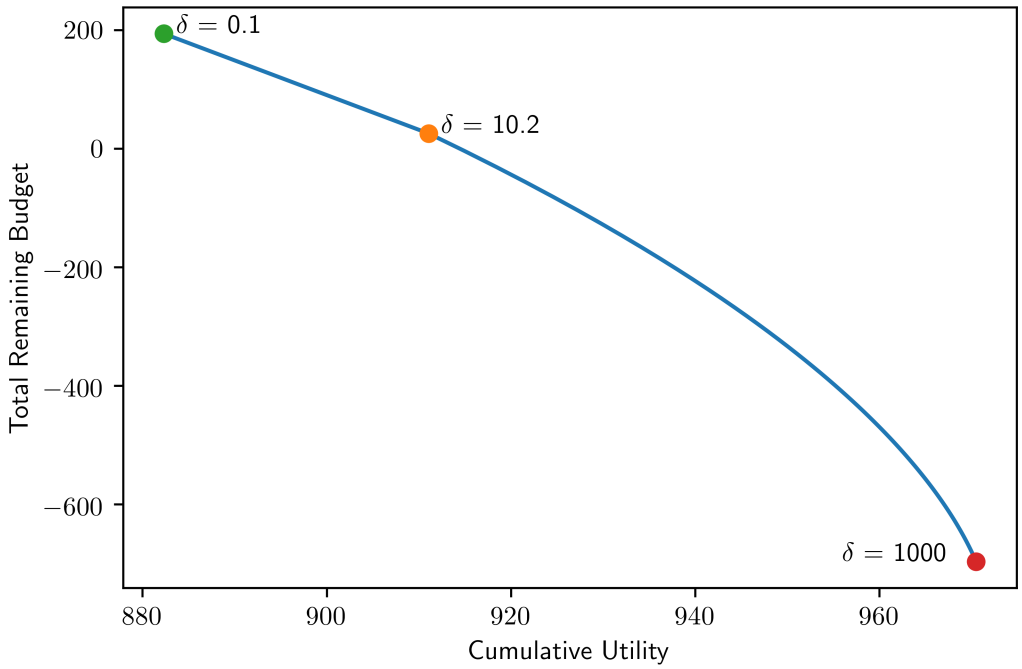

Figure 6.2: Trade-off between the overall utility $\sum_{t=1}^{T} f_t(x_t)$ and the total remaining budget $B_T - \sum_{t=1}^{T} \langle p, x_t \rangle$ of quadratic functions for different choices of parameter δ. $\delta = 10.2$ is our choice of the penalty parameter.

imization oracles are independent of each other and thus we can run them in parallel and improve the time complexity of the algorithm.

6.2 Relation with Previous Works

- Yuan and Lamperski [33] studied a similar problem in the convex setting where all the constraint functions are deterministic and given offline and they obtained $\mathcal{O}(\sqrt{T})$ regret bound and $\mathcal{O}(T^{\frac{3}{4}})$ constraint violation bound. On the other hand, applying our OLFW algorithm with update rule (I) to the online DR-submodular maximization problem subject to deterministic linear constraints, we obtain $\mathcal{O}(\sqrt{T})$ regret and

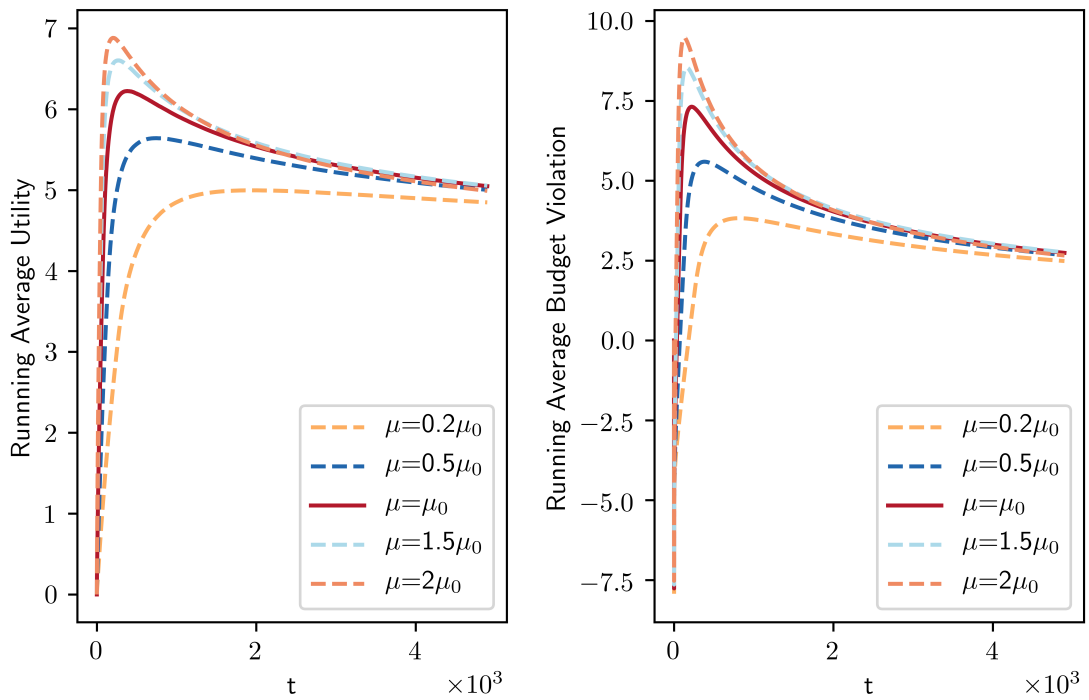

Figure 6.3: Running average of cumulative utility $\frac{1}{t}\sum_{\tau=1}^{t} f_\tau(x_\tau)$ and running average of budget violation $\frac{1}{t}\sum_{\tau=1}^{t}(\langle p, x_\tau\rangle - \frac{B_T}{T})$ for different choices of the step size μ, where $\mu_0 := \frac{R}{\beta\sqrt{T}}$ is our choice of step size in the OLFW algorithm.

constraint violation bounds simultaneously. Note that in this setting, $\Sigma = 0$ and thus, the dominating $\mathcal{O}(T^{\frac{3}{4}})$ term in the expected regret bound of the OLFW algorithm vanishes.

- Jenatton et al. [17] considered a related problem with concave utility functions and linear constraint functions where at each round, the reward functions arrive before committing to an action (i.e., the 1-lookahead setting) and long-term constraints are penalized through a penalty function in the objective. They obtained sub-linear bounds for the dynamic regret in their setting. On the contrary, in our framework, we deal

with the extra complication that the utility function at each step is DR-submodular (and generally non-concave) and it is revealed after committing to an action.

- Mahdavi et al. [21] considered the exact same framework as ours in the special case where the utility functions are linear and they obtained $\mathcal{O}(\sqrt{T})$ regret bound and $\mathcal{O}(T^{\frac{3}{4}})$ constraint violation bound in both expectation and high probability settings. Note that our OLFW algorithm obtains improved $\mathcal{O}(\sqrt{T})$ constraint violation bounds in expectation and with high probability for the more general setting of DR-submodular utility functions.

- Sadeghi and Fazel [28] considered the online DR-submodular maximization problem subject to linear constraints in the adversarial setting where the constraint functions are chosen arbitrarily. Their proposed OSPHG algorithm in the setting with window length $W = 1$ could be adapted to obtain $\mathcal{O}(\sqrt{T})$ regret and $\mathcal{O}(T^{\frac{3}{4}})$ constraint violation bounds in expectation. However, the OSPHG algorithm fails to provide *any* bounds for the high probability setting. On the other hand, our algorithm uses the current estimate of the cost vector to exploit the stochastic nature of the constraints. Furthermore, through using the update rule (II) in the OLFW algorithm, we are able to guarantee sub-linear bounds in the high probability setting as well.

6.3 Conclusion and future work

In this work, we studied online continuous DR-submodular maximization with stochastic linear cumulative constraints. We proposed the Online Lagrangian Frank-Wolfe (OLFW) algorithm to solve this problem and we obtained the first sub-linear bounds, both in expectation and with high probability, for the regret and constraint violation of this algorithm. The current work could be further extended in a number of interesting directions. First, it is yet to be seen whether the online DR-submodular maximization setting could handle general, stochastic or adversarial, convex long-term constraints. Furthermore, it is interesting to see whether it is possible to improve the expected regret bound to match the $\mathcal{O}(\sqrt{T})$ high

probability regret bound. Finally, studying this problem under bandit feedback (as opposed to the full information setting considered in this thesis) is left to future work.

BIBLIOGRAPHY

[1] Shipra Agrawal and Nikhil R. Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, page 989–1006, New York, NY, USA, 2014. Association for Computing Machinery.

[2] Shipra Agrawal and Nikhil R. Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, page 1405–1424, USA, 2015. Society for Industrial and Applied Mathematics.

[3] Francis Bach. Learning with submodular functions: A convex optimization perspective. *arXiv preprint arXiv:1111.6453*, 2011.

[4] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *J. ACM*, 65(3), March 2018.

[5] Santiago R. Balseiro and Yonatan Gur. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science*, 65(9):3952–3968, 2019.

[6] Omar Besbes and Assaf Zeevi. Blind network revenue management. *Operations research*, 60(6):1537–1550, 2012.

[7] Andrew An Bian, Baharan Mirzasoleiman, Joachim Buhmann, and Andreas Krause. Guaranteed Non-convex Optimization: Submodular Maximization over Continuous Domains. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Ma-*

chine Learning Research, pages 111–120, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

[8] 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, January 2011.

[9] Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. volume 80 of *Proceedings of Machine Learning Research*, pages 814–823, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[10] Lin Chen, Hamed Hassani, and Amin Karbasi. Online Continuous Submodular Maximization. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1896–1905, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

[11] Kris Johnson Ferreira, David Simchi-Levi, and He Wang. Online network revenue management using thompson sampling. *Operations research*, 66(6):1586–1602, 2018.

[12] Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. Near-Optimal MAP Inference for Determinantal Point Processes. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2735–2743. Curran Associates, Inc., 2012.

[13] Elad Hazan. Introduction to online convex optimization. *Found. Trends Optim.*, 2(3–4):157–325, August 2016.

[14] Zhiyi Huang and Anthony Kim. Welfare maximization with production costs: A primal dual approach. *Games and Economic Behavior*, 118:648 – 667, 2019.

[15] Rishabh Iyer, Stefanie Jegelka, and Jeff Bilmes. Monotone closure of relaxed constraints in submodular optimization: Connections between minimization and maximization. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, page 360–369, Arlington, Virginia, USA, 2014. AUAI Press.

[16] Rodolphe Jenatton, Jim Huang, and Cedric Archambeau. Adaptive algorithms for online convex optimization with long-term constraints. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 402–411, New York, New York, USA, 20–22 Jun 2016. PMLR.

[17] Rodolphe Jenatton, Jim Huang, Dominik Csiba, and Cedric Archambeau. Online optimization and regret guarantees for non-additive long-term constraints. *arXiv:1602.05394 [cs, math, stat]*, June 2016. ZSCC: 0000010.

[18] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M. Kakade, and Michael I. Jordan. A Short Note on Concentration Inequalities for Random Vectors with SubGaussian Norm. *arXiv:1902.03736 [cs, math, stat]*, February 2019. ZSCC: 0000006.

[19] Nikolaos Liakopoulos, Apostolos Destounis, Georgios Paschos, Thrasyvoulos Spyropoulos, and Panayotis Mertikopoulos. Cautious regret minimization: Online optimization with long-term budget constraints. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3944–3952, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[20] Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. Trading regret for efficiency: Online convex optimization with long term constraints. *Journal of Machine Learning Research*, 13(81):2503–2528, 2012.

[21] Mehrdad Mahdavi, Tianbao Yang, and Rong Jin. Efficient Constrained Regret Minimization. *arXiv:1205.2265 [cs]*, October 2012. ZSCC: 0000002.

[22] Shie Mannor, John N Tsitsiklis, and Jia Yuan Yu. Online learning with sample path constraints. *Journal of Machine Learning Research*, 10(Mar):569–590, 2009.

[23] Michael J. Neely and Hao Yu. Online Convex Optimization with Time-Varying Constraints. *arXiv:1702.04783 [math]*, February 2017.

[24] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 1999. ZSCC: 0000956.

[25] Francesco Orabona. A modern introduction to online learning, 2019.

[26] Adrian Rivera, He Wang, and Huan Xu. The online saddle point problem and online convex optimization with knapsacks, 2018.

[27] R Tyrrell Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1970.

[28] Omid Sadeghi and Maryam Fazel. Online continuous dr-submodular maximization with long-term budget constraints. volume 108 of *Proceedings of Machine Learning Research*, pages 4410–4419, Online, 26–28 Aug 2020. PMLR.

[29] Shai Shalev-Shwartz. Online learning and online convex optimization. *Found. Trends Mach. Learn.*, 4(2):107–194, February 2012.

[30] Jan Vondrak. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the Fourtieth Annual ACM Symposium on Theory of Computing - STOC 08*, page 67, Victoria, British Columbia, Canada, 2008. ACM Press.

[31] Xiaohan Wei, Hao Yu, and Michael J. Neely. Online Primal-Dual Mirror Descent under Stochastic Constraints. *arXiv:1908.00305 [math]*, August 2019.

[32] Hao Yu, Michael Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 1428–1438. Curran Associates, Inc., 2017.

[33] Jianjun Yuan and Andrew Lamperski. Online convex optimization for cumulative constraints. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6137–6146. Curran Associates, Inc., 2018.

[34] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

Appendix A

PROOF OF LEMMA 6

Fix $k \in [K]$. Using L-smoothness of the function $\mathcal{L}_t$, we have:

$$\begin{aligned}
\mathcal{L}_t(x_t^{(k+1)}, \lambda_t) &\geq \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} \rangle - \frac{L}{2K^2}\|v_t^{(k)}\|_2^2 \\
&\overset{\text{(a)}}{\geq} \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} \rangle - \frac{LR^2}{2K^2} \\
&= \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} - x \rangle + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), x \rangle - \frac{LR^2}{2K^2} \\
&= \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} - x \rangle + \frac{1}{K}\langle \nabla f_t(x_t^{(k)}), x \rangle \\
&- \frac{1}{K}\lambda_t \langle \nabla \widehat{g}_t(x_t^{(k)}), x \rangle - \frac{LR^2}{2K^2} \\
&\overset{\text{(b)}}{=} \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} - x \rangle + \frac{1}{K}\langle \nabla f_t(x_t^{(k)}), x \rangle \\
&- \frac{1}{K}\lambda_t \widehat{g}_t(x) - \frac{1}{K}\lambda_t \frac{B_T}{T} - \frac{LR^2}{2K^2},
\end{aligned}$$

where (a) is due to the assumption that $\text{diam}(\mathcal{X}) \leq R$. Note that in order to obtain (b), we have used linearity of the budget functions for all $t \in [T]$ to write $\langle \nabla \widehat{g}_t(x_t^{(k)}), x \rangle = \langle \widehat{p}_t, x \rangle = \widehat{g}_t(x) + \frac{B_T}{T}$. More general assumptions such as convexity would not be enough for the proof to go through.

Considering that $f_t(x)$ is monotone DR-submodular for all $t \in [T]$, we can write:

$$\begin{aligned}
f_t(x) - f_t(x_t^{(k)}) &\overset{\text{(c)}}{\leq} f_t(x \vee x_t^{(k)}) - f_t(x_t^{(k)}) \\
&\overset{\text{(d)}}{\leq} \langle \nabla f_t(x_t^{(k)}), (x \vee x_t^{(k)}) - x_t^{(k)} \rangle \\
&= \langle \nabla f_t(x_t^{(k)}), (x - x_t^{(k)}) \vee 0 \rangle \\
&\overset{\text{(e)}}{\leq} \langle \nabla f_t(x_t^{(k)}), x \rangle,
\end{aligned}$$

where for $a, b \in \mathbb{R}^n$, $a \vee b$ denotes the entry-wise maximum of vectors a and b, (c) and (e) are due to monotonocity of f_t and (d) uses concavity of f_t along non-negative directions. Therefore, we conclude:

$$\mathcal{L}_t(x_t^{(k+1)}, \lambda_t) \geq \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} - x\rangle + \frac{1}{K}\big(f_t(x) - f_t(x_t^{(k)})\big) - \frac{1}{K}\lambda_t \widehat{g}_t(x) - \frac{1}{K}\lambda_t \frac{B_T}{T} - \frac{LR^2}{2K^2}.$$

Equivalently, we can write:

$$\begin{aligned} f_t(x) - f_t(x_t^{(k+1)}) &\leq (1 - \frac{1}{K})\big(f_t(x) - f_t(x_t^{(k)})\big) - \lambda_t\big(\widehat{g}_t(x_t^{(k+1)}) - \widehat{g}_t(x_t^{(k)})\big) + \frac{1}{K}\lambda_t \widehat{g}_t(x) \\ &+ \frac{1}{K}\lambda_t \frac{B_T}{T} + \frac{LR^2}{2K^2} + \frac{1}{K}\langle \nabla \mathcal{L}_t(x_t^{(k)}, \lambda_t), x - v_t^{(k)}\rangle \\ &= (1 - \frac{1}{K})\big(f_t(x) - f_t(x_t^{(k)})\big) + \frac{1}{K}\big[\lambda_t \frac{B_T}{T} - \lambda_t \langle \widehat{p}_t, v_t^{(k)}\rangle \\ &+ \lambda_t \widehat{g}_t(x) + \frac{LR^2}{2K} + \langle \nabla \mathcal{L}_t(x_t^{(k)}, \lambda_t), x - v_t^{(k)}\rangle\big]. \end{aligned}$$

Taking the sum over $t \in \{1, \ldots, T\}$, we obtain:

$$\begin{aligned} \sum_{t=1}^{T} \big(f_t(x) - f_t(x_t^{(k+1)})\big) &\leq (1 - \frac{1}{K}) \sum_{t=1}^{T} \big(f_t(x) - f_t(x_t^{(k)})\big) \\ &+ \frac{1}{K}\sum_{t=1}^{T}\big[- \lambda_t \langle \widehat{p}_t, v_t^{(k)}\rangle + \lambda_t \widehat{g}_t(x) \\ &+ \lambda_t \frac{B_T}{T} + \frac{LR^2}{2K} + \langle \nabla \mathcal{L}_t(x_t^{(k)}, \lambda_t), x - v_t^{(k)}\rangle\big]. \end{aligned} \tag{A.1}$$

Applying inequality $(A.1)$ recursively for all $k \in \{1, \ldots, K\}$, we obtain:

$$\begin{aligned} \sum_{t=1}^{T} \big(f_t(x) - f_t(\underbrace{x_t^{(K+1)}}_{=x_t})\big) &\leq \Pi_{k=0}^{K-1}(1 - \frac{1}{K}) \sum_{t=1}^{T} \big(f_t(x) - f_t(x_t^{(0)})\big) \\ &+ \sum_{k=0}^{K-1} \frac{1}{K} \Pi_{j=k+1}^{K-1}(1 - \frac{1}{K}) \sum_{t=1}^{T}\big[- \lambda_t \langle \widehat{p}_t, v_t^{(k)}\rangle \\ &+ \lambda_t \widehat{g}_t(x) + \lambda_t \frac{B_T}{T} + \frac{LR^2}{2K} \\ &+ \langle \nabla \mathcal{L}_t(x_t^{(k)}, \lambda_t), x - v_t^{(k)}\rangle\big]. \end{aligned} \tag{A.2}$$

Using the regret bound of Online Gradient Ascent instance $\mathcal{E}_k$ $\forall k \in [K]$, the following holds:

$$\begin{aligned}
\sum_{t=1}^{T}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), x - v_t^{(k)}\rangle &= \sum_{t=1}^{T}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), x\rangle - \sum_{t=1}^{T}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)}\rangle \\
&\leq \frac{R^2}{\mu} + \frac{\mu}{2}\sum_{t=1}^{T}\|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t)\|^2 \\
&= \frac{R^2}{\mu} + \frac{\mu}{2}\sum_{t=1}^{T}\|\nabla_x f_t(x_t^{(k)}) - \lambda_t \widehat{p}_t\|^2 \\
&\overset{\text{(a)}}{\leq} \frac{R^2}{\mu} + \frac{\mu}{2}\sum_{t=1}^{T}\big(2\|\nabla_x f_t(x_t^{(k)})\|^2 + 2\lambda_t^2\|\widehat{p}_t\|^2\big) \\
&\overset{\text{(b)}}{\leq} \frac{R^2}{\mu} + \beta^2\mu T + \beta^2\mu\sum_{t=1}^{T}\lambda_t^2,
\end{aligned}$$

where (a) uses the inequality $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ $\forall a, b \in \mathbb{R}^n$ and (b) is due to β-Lipschitzness of functions f_t, g_t for all $t \in [T]$.

Using the inequality $(1 - \frac{1}{K})^K \leq \frac{1}{e}$ in $(A.2)$, we have:

$$\begin{aligned}
\sum_{t=1}^{T}\big(f_t(x) - f_t(x_t)\big) &\leq \frac{1}{e}\sum_{t=1}^{T}\big(f_t(x) - f_t(x_t^{(0)})\big) \\
&+ \sum_{t=1}^{T}\sum_{k=0}^{K-1}\frac{1}{K}\big[-\lambda_t\langle \widehat{p}_t, v_t^{(k)}\rangle + \lambda_t \widehat{g}_t(x) \\
&+ \lambda_t\frac{B_T}{T} + \frac{LR^2}{2K} + \langle \nabla \mathcal{L}_t(x_t^{(k)}, \lambda_t), x - v_t^{(k)}\rangle\big] \\
&= \frac{1}{e}\sum_{t=1}^{T}\big(f_t(x_t^*) - \underbrace{f_t(0)}_{=0}\big) \\
&+ \sum_{t=1}^{T}\big[-\lambda_t\widehat{g}_t(x_t) - \lambda_t\frac{B_T}{T} + \lambda_t\widehat{g}_t(x) \\
&+ \lambda_t\frac{B_T}{T} + \frac{LR^2}{2K} + \sum_{k=0}^{K-1}\frac{1}{K}\langle \nabla \mathcal{L}_t(x_t^{(k)}, \lambda_t), x - v_t^{(k)}\rangle\big]. \qquad (A.3)
\end{aligned}$$

Rearranging the terms in $(A.3)$, we obtain:

$$\begin{aligned}\sum_{t=1}^{T}\big((1-\frac{1}{e})f_t(x)-f_t(x_t)\big)+\sum_{t=1}^{T}\lambda_t\widehat{g}_t(x_t) &\leq \frac{LR^2T}{2K}+\sum_{t=1}^{T}\lambda_t\widehat{g}_t(x)\\ &+\sum_{k=0}^{K-1}\frac{1}{K}\sum_{t=1}^{T}\langle\nabla\mathcal{L}_t(x_t^{(k)},\lambda_t),x-v_t^{(k)}\rangle\\ &\leq \frac{LR^2T}{2K}+\sum_{t=1}^{T}\lambda_t\widehat{g}_t(x)+\frac{R^2}{\mu}+\beta^2\mu T\\ &+\beta^2\mu\sum_{t=1}^{T}\lambda_t^2. \end{aligned}\tag{A.4}$$

Now, subtract $\sum_{t=1}^{T}\lambda_t\gamma_t$ from each side of the inequality $(A.4)$. Thus, obtain that:

$$\begin{aligned}\sum_{t=1}^{T}\big((1-\frac{1}{e})f_t(x)-f_t(x_t)\big) &\leq \frac{LR^2T}{2K}+\frac{R^2}{\mu}+\beta^2\mu T\\ &+\sum_{t=1}^{T}\lambda_t\widetilde{g}_t(x)+\big[\beta^2\mu\sum_{t=1}^{T}\lambda_t^2-\sum_{t=1}^{T}\lambda_t\widetilde{g}_t(x_t)\big].\end{aligned}$$

Setting $\delta=\beta^2$ and using the update rule of the algorithm for λ_t $\forall t\in[T]$, we have:

$$\begin{aligned}\beta^2\mu\sum_{t=1}^{T}\lambda_t^2-\sum_{t=1}^{T}\lambda_t\widetilde{g}_t(x_t) &= \frac{\beta^2\mu}{\delta^2\mu^2}\sum_{t=1}^{T}[\widetilde{g}_t(x_t)]_+^2-\frac{1}{\delta\mu}\sum_{t=1}^{T}[\widetilde{g}_t(x_t)]_+\widetilde{g}_t(x_t)\\ &\overset{\text{(a)}}{\leq}\frac{1}{\delta\mu}\sum_{t=1}^{T}\big(\widetilde{g}_t^2(x_t)-\widetilde{g}_t(x_t)\widetilde{g}_t(x_t)\big)\\ &\leq 0,\end{aligned}$$

where we have used $[\widetilde{g}_t(x_t)]_+\geq\widetilde{g}_t(x_t)$ and $|[\widetilde{g}_t(x_t)]_+|\leq\widetilde{g}_t(x_t)$ to obtain (a). Thus, we obtain the result.

Appendix B

PROOFS OF PROBABILITY LEMMAS

B.1 Proof of Lemma 7

From assumption A3, we have that $\|p_t\| \leq \beta$. Thus, Lemma 1 of [18] holds with norm sub-gaussian parameter $\sigma = c\beta$ for some universal constant c. The result follows immediately.

B.2 Proof of Lemma 8

Note that $\|\widehat{p}_t - p\| = \frac{1}{t-1}\|\sum_{s=1}^{t-1}(p_s - p)\|$ and since vector $p_s - p$ satisfies the result of Lemma 7, we can apply Corollary 7 of [18] to the random vectors $\{p_s - p\}_{s=1}^{t-1}$ and obtain:

$$\begin{aligned}\|\sum_{s=1}^{t-1}(p_s - p)\| &\leq c'\sqrt{\sum_{s=1}^{t-1}\sigma^2 \log\left(\frac{2nT}{\epsilon}\right)} \\ &= \sqrt{t-1}c'\sigma\sqrt{\log\left(\frac{2nT}{\epsilon}\right)}.\end{aligned}$$

Combining the above equations, we get the desired result.

B.3 Proof of Lemma 9

Consider the events $\mathcal{E}_t := \{\|\widehat{p}_t - p\| \leq c'\sigma\sqrt{\log(\frac{2nT}{\epsilon})/(t-1)}\}$ for $t \in [T]/\{1\}$ and $\mathcal{E} := \{\sum_{t=2}^{T}\|\widehat{p}_t - p\| \leq C\sigma\sqrt{T\log(2nT/\epsilon)}\}$. Then, apply union bound to $\bigcup_{t=2}^{T}\mathcal{E}_t^c$ with the obser-

vation that $\bigcap_{t=2}^{T} \mathcal{E}_t \subset \mathcal{E}$. Explicitly,

$$\mathbb{P}(\cup_{t=2}^{T}\mathcal{E}_t) \leq \sum_{t=2}^{T} \mathbb{P}(\mathcal{E}_t^c),$$
$$1 - \mathbb{P}(\cap_{t=2}^{T}\mathcal{E}_t) \leq \sum_{t=2}^{T}(1 - \mathbb{P}(\mathcal{E}_t)),$$
$$1 - \sum_{t=2}^{T}(1 - \mathbb{P}(\mathcal{E}_t)) \leq \mathbb{P}(\cap_{t=2}^{T}\mathcal{E}_t) \leq \mathbb{P}(\mathcal{E}).$$

Using the result of Lemma 8, we obtain the result.

B.4 Proof of Lemma 10

First, note that $\mathbb{E}[\widehat{g}_t(x)] = \mathbb{E}[\langle \widehat{p}_t, x\rangle - \frac{B_T}{T}] = \langle p, x\rangle - \frac{B_T}{T} = g(x)$. If $y_t = g_t(x)$ is a random variable, then by assumption, $y_t \in [-G, G]$ holds for each t, i.e., y_t is a bounded random variable. Therefore we can apply Hoeffding's inequality and get $\mathbb{P}\{|\widehat{g}_t(x) - g(x)| > \gamma_t\} \leq 2\exp(-\frac{t\gamma_t^2}{2G^2})$. Substituting the value of γ_t in the right hand side, we get that $\mathbb{P}\{|\widehat{g}_t(x) - g(x)| > \gamma_t\} \leq \frac{\epsilon}{T}$. The result follows immediately.

B.5 Proof of Lemma 12

We can write:

$$\begin{aligned}
\mathbb{E}\|\widehat{p}_t - p\|^2 &= \mathbb{E}(\widehat{p}_t - p)^T(\widehat{p}_t - p) \\
&= \mathbb{E}[Tr\big((\widehat{p}_t - p)^T(\widehat{p}_t - p)\big)] \\
&= \mathbb{E}[Tr\big((\widehat{p}_t - p)(\widehat{p}_t - p)^T\big)] \\
&= Tr\big(\mathbb{E}[(\widehat{p}_t - p)(\widehat{p}_t - p)^T]\big) \\
&= Tr(Cov(\widehat{p}_t)) \\
&= Tr\big(\frac{\Sigma}{t-1}\big) \\
&= \frac{Tr(\Sigma)}{t-1}.
\end{aligned}$$

Appendix C

MODIFIED ONLINE LAGRANGIAN FRANK-WOLFE (MOLFW) ALGORITHM

For the setting where only unbiased stochastic gradient estimates of the utility functions $\{f_t\}_{t=1}^T$ with bounded variance σ^2 are available, inspired by the variance reduction technique introduced by [9], we propose the Modified Online Lagrangian Frank-Wolfe (MOLFW) algorithm in Algorithm 2. Compared to the OLFW algorithm, the MOLFW algorithm uses $\langle v_t^{(k)}, d_t^{(k)} \rangle$ (instead of $\langle v_t^{(k)}, \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) \rangle$) as the payoff to be received by $\mathcal{E}_k$ $\forall k \in [K]$ at round $t \in [T]$. We analyze the performance of the MOLFW algorithm below.
Fix $k \in [K]$. Using L-smoothness of the function $\mathcal{L}_t$, we have:

$$
\begin{aligned}
\mathcal{L}_t(x_t^{(k+1)}, \lambda_t) &\geq \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} \rangle - \frac{L}{2K^2}\|v_t^{(k)}\|_2^2 \\
&\overset{(a)}{\geq} \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} \rangle - \frac{LR^2}{2K^2} \\
&= \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}, v_t^{(k)} - x \rangle \\
&+ \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), x \rangle + \frac{1}{K}\langle d_t^{(k)}, v_t^{(k)} - x \rangle - \frac{LR^2}{2K^2} \\
&= \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}, v_t^{(k)} - x \rangle \\
&+ \frac{1}{K}\langle \nabla f_t(x_t^{(k)}), x \rangle - \frac{1}{K}\lambda_t \langle \widehat{p}_t, x \rangle + \frac{1}{K}\langle d_t^{(k)}, v_t^{(k)} - x \rangle - \frac{LR^2}{2K^2},
\end{aligned}
$$

where (a) is due to the assumption that $\text{diam}(\mathcal{X}) \leq R$. Considering that $f_t(x)$ is monotone DR-submodular for all $t \in [T]$, we can write:

$$\begin{aligned} f_t(x) - f_t(x_t^{(k)}) &\overset{\text{(b)}}{\leq} f_t(x \vee x_t^{(k)}) - f_t(x_t^{(k)}) \\ &\overset{\text{(c)}}{\leq} \langle \nabla f_t(x_t^{(k)}), (x \vee x_t^{(k)}) - x_t^{(k)} \rangle \\ &= \langle \nabla f_t(x_t^{(k)}), (x - x_t^{(k)}) \vee 0 \rangle \\ &\overset{\text{(d)}}{\leq} \langle \nabla f_t(x_t^{(k)}), x \rangle, \end{aligned}$$

where for $a, b \in \mathbb{R}^n$, $a \vee b$ denotes the entry-wise maximum of vectors a and b, (b) and (d) are due to monotonocity of f_t and (c) uses concavity of f_t along non-negative directions. Therefore, we conclude:

$$\begin{aligned} \mathcal{L}_t(x_t^{(k+1)}, \lambda_t) \geq \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\big(&\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}, v_t^{(k)} - x \rangle \\ &+ f_t(x) - f_t(x_t^{(k)}) - \lambda_t \widehat{g}_t(x) - \lambda_t \frac{B_T}{T} + \langle d_t^{(k)}, v_t^{(k)} - x \rangle\big) - \frac{LR^2}{2K^2}. \end{aligned}$$

Using the Young's inequality, we have:

$$\begin{aligned} \langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}, v_t^{(k)} - x \rangle &\geq -\frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 - \frac{\beta_k}{2} \|v_t^{(k)} - x\|^2 \\ &\geq -\frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 - \frac{R^2 \beta_k}{2}. \end{aligned}$$

Therefore, we can equivalently write:

$$\begin{aligned} f_t(x) - f_t(x_t^{(k+1)}) \leq\ & (1 - \frac{1}{K})\big(f_t(x) - f_t(x_t^{(k)})\big) - \lambda_t\big(\widehat{g}_t(x_t^{(k+1)}) - \widehat{g}_t(x_t^{(k)})\big) + \frac{1}{K}\lambda_t \widehat{g}_t(x) \\ &+ \frac{1}{K}\lambda_t \frac{B_T}{T} + \frac{LR^2}{2K^2} + \frac{1}{K}\langle d_t^{(k)}, x - v_t^{(k)} \rangle + \frac{1}{K}\frac{R^2 \beta_k}{2} \\ &+ \frac{1}{K}\frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 \\ =\ & (1 - \frac{1}{K})\big(f_t(x) - f_t(x_t^{(k)})\big) + \frac{1}{K}\big[\lambda_t \frac{B_T}{T} - \lambda_t \langle \widehat{p}_t, v_t^{(k)} \rangle + \lambda_t \widehat{g}_t(x) \\ &+ \frac{LR^2}{2K} + \langle d_t^{(k)}, x - v_t^{(k)} \rangle + \frac{R^2 \beta_k}{2} + \frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2\big]. \end{aligned}$$

Taking the sum over $t \in \{1, \dots, T\}$, we obtain:

$$\begin{aligned}\sum_{t=1}^{T} \big(f_t(x) - f_t(x_t^{(k+1)})\big) \leq (1 - \frac{1}{K}) \sum_{t=1}^{T} \big(f_t(x) - f_t(x_t^{(k)})\big) \\ + \frac{1}{K} \sum_{t=1}^{T} \big[- \lambda_t \langle \widehat{p}_t, v_t^{(k)} \rangle + \lambda_t \widehat{g}_t(x) + \lambda_t \frac{B_T}{T} + \frac{LR^2}{2K} \\ + \langle d_t^{(k)}, x - v_t^{(k)} \rangle + \frac{R^2 \beta_k}{2} + \frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 \big]. \qquad \text{(C.1)}\end{aligned}$$

Applying inequality $(C.1)$ recursively for all $k \in \{1, \dots, K\}$, we obtain:

$$\begin{aligned}\sum_{t=1}^{T} \big(f_t(x) - f_t(\underbrace{x_t^{(K+1)}}_{=x_t})\big) \leq \Pi_{k=0}^{K-1}(1 - \frac{1}{K}) \sum_{t=1}^{T} \big(f_t(x) - f_t(x_t^{(0)})\big) \\ + \sum_{k=0}^{K-1} \frac{1}{K} \Pi_{j=k+1}^{K-1}(1 - \frac{1}{K}) \sum_{t=1}^{T} \big[- \lambda_t \langle \widehat{p}_t, v_t^{(k)} \rangle + \lambda_t \widehat{g}_t(x) + \lambda_t \frac{B_T}{T} \\ + \frac{LR^2}{2K} + \langle d_t^{(k)}, x - v_t^{(k)} \rangle + \frac{R^2 \beta_k}{2} + \frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 \big]. \qquad \text{(C.2)}\end{aligned}$$

Using the regret bound of Online Gradient Ascent instance $\mathcal{E}_k$ $\forall k \in [K]$, the following holds:

$$\begin{aligned}\sum_{t=1}^{T} \langle d_t^{(k)}, x - v_t^{(k)} \rangle &= \sum_{t=1}^{T} \langle d_t^{(k)}, x \rangle - \sum_{t=1}^{T} \langle d_t^{(k)}, v_t^{(k)} \rangle \\ &\leq \frac{R^2}{\mu} + \frac{\mu}{2} \sum_{t=1}^{T} \|d_t^{(k)}\|^2 \\ &\leq \frac{R^2}{\mu} + \mu \sum_{t=1}^{T} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 + \mu \sum_{t=1}^{T} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t)\|^2 \\ &\overset{\text{(a)}}{\leq} \frac{R^2}{\mu} + \mu \sum_{t=1}^{T} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 + \mu \sum_{t=1}^{T} \big(2\|\nabla f_t(x_t^{(k)})\|^2 + 2\lambda_t^2 \|\widehat{p}_t\|^2\big) \\ &\overset{\text{(b)}}{\leq} \frac{R^2}{\mu} + 2\beta^2 \mu T + 2\beta^2 \mu \sum_{t=1}^{T} \lambda_t^2 + \mu \sum_{t=1}^{T} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2,\end{aligned}$$

where (a) uses the inequality $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ $\forall a, b \in \mathbb{R}^n$ and (b) is due to β-Lipschitzness of functions f_t, g_t for all $t \in [T]$.

Using the inequality $(1-\frac{1}{K})^K \leq \frac{1}{e}$ in $(C.2)$, we have:

$$\begin{aligned}\sum_{t=1}^{T}\big(f_t(x)-f_t(x_t)\big) &\leq \frac{1}{e}\sum_{t=1}^{T}\big(f_t(x)-f_t(x_t^{(0)})\big)+\sum_{t=1}^{T}\sum_{k=0}^{K-1}\frac{1}{K}\big[-\lambda_t\langle\widehat{p}_t,v_t^{(k)}\rangle+\lambda_t\widehat{g}_t(x)\\ &+\lambda_t\frac{B_T}{T}+\frac{LR^2}{2K}+\langle d_t^{(k)},x-v_t^{(k)}\rangle+\frac{R^2\beta_k}{2}+\frac{1}{2\beta_k}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\big]\\ &=\frac{1}{e}\sum_{t=1}^{T}\big(f_t(x)-\underbrace{f_t(0)}_{=0}\big)+\sum_{t=1}^{T}\big[-\lambda_t\widehat{g}_t(x_t)-\lambda_t\frac{B_T}{T}+\lambda_t\widehat{g}_t(x)+\lambda_t\frac{B_T}{T}\\ &+\frac{LR^2}{2K}+\sum_{k=0}^{K-1}\frac{1}{K}\big(\langle d_t^{(k)},x-v_t^{(k)}\rangle+\frac{R^2\beta_k}{2}+\frac{1}{2\beta_k}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\big)\big].\end{aligned} \tag{C.3}$$

Rearranging the terms in $(C.3)$, we obtain:

$$\begin{aligned}\sum_{t=1}^{T}\big((1-\frac{1}{e})f_t(x)-f_t(x_t)\big)+\sum_{t=1}^{T}\lambda_t\widehat{g}_t(x_t) &\leq \frac{LR^2T}{2K}+\sum_{t=1}^{T}\lambda_t\widehat{g}_t(x)+\sum_{k=0}^{K-1}\frac{1}{K}\big(\langle d_t^{(k)},x-v_t^{(k)}\rangle\\ &+\frac{R^2\beta_k}{2}+\frac{1}{2\beta_k}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\big)\\ &\leq \frac{LR^2T}{2K}+\sum_{t=1}^{T}\lambda_t\widehat{g}_t(x)+\frac{R^2}{\mu}+2\beta^2\mu T\\ &+\sum_{t=1}^{T}\sum_{k=0}^{K-1}\frac{1}{K}\big(\frac{R^2\beta_k}{2}+\frac{1}{2\beta_k}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\\ &+\mu\sum_{t=1}^{T}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\big)+2\beta^2\mu\sum_{t=1}^{T}\lambda_t^2.\end{aligned} \tag{C.4}$$

Now, subtract $\sum_{t=1}^{T}\lambda_t\gamma_t$ from each side of the inequality $(C.4)$. Thus, we obtain:

$$\begin{aligned}\sum_{t=1}^{T}\big((1-\frac{1}{e})f_t(x)-f_t(x_t)\big) &\leq \sum_{t=1}^{T}\lambda_t\widetilde{g}_t(x)+\frac{R^2}{\mu}+2\beta^2\mu T+\big[2\beta^2\mu\sum_{t=1}^{T}\lambda_t^2-\sum_{t=1}^{T}\lambda_t\widetilde{g}_t(x_t)\big]\\ &+\sum_{t=1}^{T}\sum_{k=0}^{K-1}\frac{1}{K}\big(\frac{R^2\beta_k}{2}+\frac{1}{2\beta_k}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\\ &+\mu\sum_{t=1}^{T}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\big)+\frac{LR^2T}{2K}.\end{aligned}$$

Setting $\delta = 2\beta^2$ and using the update rule of the algorithm for λ_t $\forall t \in [T]$, we have:

$$\begin{aligned}
2\beta^2\mu \sum_{t=1}^{T} \lambda_t^2 - \sum_{t=1}^{T} \lambda_t \widetilde{g}_t(x_t) &= \frac{2\beta^2\mu}{\delta^2\mu^2} \sum_{t=1}^{T} [\widetilde{g}_t(x_t)]_+^2 - \frac{1}{\delta\mu} \sum_{t=1}^{T} [\widetilde{g}_t(x_t)]_+ \widetilde{g}_t(x_t) \\
&\overset{(a)}{\leq} \frac{1}{\delta\mu} \sum_{t=1}^{T} \left(\widetilde{g}_t^2(x_t) - \widetilde{g}_t(x_t)\widetilde{g}_t(x_t)\right) \\
&\leq 0,
\end{aligned}$$

where we have used $[\widetilde{g}_t(x_t)]_+ \geq \widetilde{g}_t(x_t)$ and $|[\widetilde{g}_t(x_t)]_+| \leq \widetilde{g}_t(x_t)$ to obtain (a).
Using Theorem 3 of [9], we have:

$$\mathbb{E}\big[\|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2\big] \leq \frac{Q}{(k+4)^{2/3}} \ \forall t \in [T], k \in [K],$$

where $Q = \max\{\max_{t\in[T]} \|\nabla f_t(0)\|^2 4^{2/3}, 4\sigma^2 + 6L^2R^2\}$. Therefore, taking expectation of both sides, setting $\beta_k = \frac{Q^{1/2}}{R(k+4)^{1/3}}$ and simplifying the result, we obtain:

$$\mathbb{E}[R_T] \leq \frac{R^2}{\mu} + 2\beta^2\mu T + \frac{LR^2T}{2K} + \sum_{t=1}^{T} \mathbb{E}[\lambda_t \widetilde{g}_t(x)] + \frac{3Q\mu T}{K^{2/3}} + \frac{3RQ^{1/2}T}{2K^{1/3}}.$$

Therefore, if we set $K = \mathcal{O}(T^{3/2})$, we can analyze the expected regret and constraint violation of the MOLFW algorithm in a similar fashion to Theorem 3 and Theorem 4, and we obtain similar performance bounds.