上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

THESIS OF BACHELOR

论文题目：不同场景下的多臂赌博机问题

学生姓名：	胡丕河
学生学号：	515030910542
专　　业：	计算机科学与技术（IEEE 试点班）
指导教师：	黄隆波教授、高晓沨教授
学院(系)：	电子信息与电气工程学院

B

Submitted in total fulfillment of the requirements for the degree of
Bachelor in Computer Science and Technology

Multi-armed Bandits in Versatile Settings

Pihe Hu

Advisors

Prof. Longbo Huang
Institute for Interdisciplinary
Information Sciences
Tsinghua University
Beijing, P.R.China

Prof. Xiaofeng Gao
Department of Computer Science
and Engineering
Shanghai Jiao Tong University
Shanghai, P.R.China

上海交通大学
学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：_____

日　期：_____年_____月_____日

上海交通大学
学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

保　密 □，在 _____ 年解密后适用本授权书。

不保密 □。

（请在以上方框内打 ✓）

学位论文作者签名：_____　　指导教师签名：_____

日　期：_____年 ____月 ____日　　日　期：_____年 ____月 ____日

不同场景下的多臂赌博机问题

摘　要

在本文中，我们研究了不同条件下的多臂赌博机问题。准确而言，我们考虑多臂赌博机（bandit）问题和赌博机凸优化。赌博机凸优化被视为泛化版本的多臂赌博机问题，其动作集是连续域而不是离散集。特别地，我们关注具有复杂反馈形式的多臂赌博机问题和赌博机凸优化，例如延迟和匿名反馈这种复杂的反馈形式。在整篇论文中，我们的工作可以归纳为三个部分。首先，我们对多臂赌博机和赌博机凸优化的领域进行了广泛的文献调研。然后，我们提出了一种最优算法，该算法能够让在某种复杂反馈下的赌博机凸优化的目标函数达到最小损失（regret）上界。最后，我们进行了大量的实验来验证算法的有效性和正确性。

首先，我们对多臂赌博机问题和赌博机凸优化问题领域进行了广泛的文献调查，其中包括最基本的、不带任何变种的多臂赌博机问题和赌博机凸优化的研究进展，以及它们在复杂反馈条件下变式问题的研究进展。其一，我们调研了解决随机（stochastic）赌博机和敌对（adversarial）赌博机问题的基本结果。对于具有复杂反馈的多臂赌博机问题，一般的想法是在某一区间内连续出发特定动作，并观察由此单类动作引起的反馈，以此解耦复杂的复合反馈情况。其二，至于赌博机凸优化问题，它可以被视为多臂赌博机问题和在线凸优化的混合问题，这比任一原始问题都困难得多。解决赌博机凸优化问题算法的一般思想是通过梯度估计将基于梯度的在线凸优化算法变换为基于梯度的赌博机凸优化算法。特别地，我们可以在不计算梯度的情况下执行梯度下降的策略。也就是说，我们将使用基于随机性的梯度估计器。

其次，我们研究了延迟参数 d 未知情况下收到延迟和匿名反馈的赌博机凸优化问题。我们率先提出了一个通用的算法框架，可以应用于在这个复杂反馈下的赌博机凸优化问题。该框架的基本思想很简单，即首先执行延迟估计，然后套用延迟固定情况下的算法。其中，延迟估计在算法框架的每个阶段分步执行。目前，我们提出了两种复杂反馈下的赌博机凸优化算法。其中一个算法达到 $\Theta(T^{2/3})$ 损失上限，而另一个达到 $\Theta(\sqrt{T})$ 损失上限。值得一提的是，第二种算法具有在同等设定下，已知最佳的赌博机凸优化问题的损失上限。此外，它是一种最优算法，因为已经证明损失下限是 $\Omega(\sqrt{T})$。

最后，我们在本论文中进行了广泛的实验。初步目标是验证我们提出的算法的有效性和正确性。在整个数值实验中，算法的性能得到了验证，证实了我们算法分析的正确性。除了验证所提算法的损失上界之外，还将最优算法与一个基线算法进行了比较。我们的最有算法在很大程度上优于基线算法，这也证明了算法的最优性。

关键词：多臂赌博机，赌博机凸优化，复合损失，延迟反馈，未知延迟参数

MULTI-ARMED BANDITS IN VERSATILE SETTINGS

ABSTRACT

In this thesis, we study the multi-armed bandits (MAB) problem in versatile settings. Specifically, we consider the MAB problem itself and bandit convex optimization (BCO). Bandits convex optimization can be regarded as a general multi-armed bandits problem, where the action set is a continuous domain, instead of a discrete set. In particular, we pay our attention to MAB problem and BCO with complex feedback, such as delayed and anonymous feedback. Throughout the whole thesis, our work can be summarized in three parts. At first, we conduct an extensive literature survey on the field of MAB problem and BCO, including its variations with complex feedback. Then, we propose an optimal algorithm which reaches the best state-of-the-art regret upper bound of the BCO with a certain form of complex feedback. At last, we conduct extensive experiments to verify the effectiveness and correctness of our algorithm.

At the first place, we conduct an extensive literature survey on the field of MAB problem and BCO, which include the research progress on the basic MAB problem and BCO, and their corresponding variations with complex feedback. We have investigated the fundamental result for stochastic bandits and adversarial bandits. For MAB with complex feedback, a general idea is to fix the action for a certain length interval and observe the reward incurred by a single action in the consecutive interval, which can decouple the complex composite feedback. As for BCO, it can be viewed as a mixed problem of MAB problem and online convex optimization (OCO), which is much harder than any of the original problems. The general idea of BCO algorithm is to transform a gradient-based algorithm for OCO to a gradient-based algorithm for BCO by gradient estimation. In particular, we can perform gradient descent without calculating the gradient by gradient estimator.

Secondly, we study the problem of BCO with delayed and anonymous feedback while the delay parameter d is unknown. For the first of time, we propose a general algorithm framework which can be applied to the BCO problem in this hard setting. The underlying idea of this framework is straightforward, that is, first estimate the delay and then follow the fixed-delay algorithm. The delay estimation is performed during each phase of the framework. At present, we propose two algorithms for BCO with complex feedback. At the same time, one of the algorithm reach near $\Theta(T^{2/3})$ regret upper bound, while the other one reach near $\Theta(\sqrt{T})$ regret upper bound. To the best of the authors' knowledge, the second algorithm has the best state-of-the-art regret upper bound of the BCO with delayed and anonymous feedback while the delay parameter d is unknown. Moreover, it is an optimal algorithm as well, because it has been proved the regret lower bound is $\Omega(\sqrt{T})$.

Ultimately, we conduct extensive experiments of various mentioned algorithms in Chapter 4

in Chapter 6. The preliminary goal is to verify the effectiveness and correctness of our proposed algorithm, as well as some comparisons with existing works. Throughout the numerical experiment, the algorithm's performance has been verified, which confirms the correctness of our proof. In addition to the verification of the regret upper bound of the proposed algorithm, the proposed optimal algorithm has been compared with the baseline algorithm. Unexpectedly, our algorithm outperforms existing baseline algorithm largely, which proves the optimality of my algorithm as well.

KEY WORDS: Multi-armed bandits, Bandit convex optimization, Composite losses, Delayed feedback, Unknown Delay Parameter

Contents

List of Figures

List of Tables

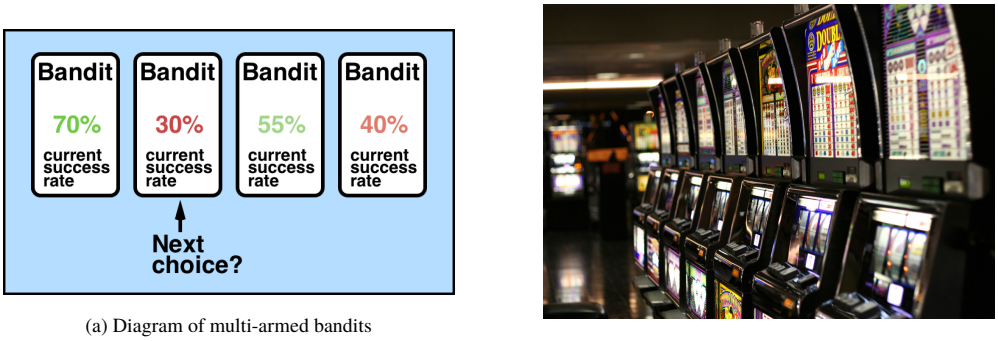
List of Algorithms

Notations

MAB	Multi-armed Bandit
BCO	Bandit Convex Optimization
OCO	Online Convex Optimization
UCB	Upper Confidence bounds
Exp3	Exponential-weight algorithm for Exploration and Exploitation
CMAB	Combinatorial Multi-armed Bandit
PMC	Probabilistic Maximum Coverage

Chapter 1 Introduction

The multi-armed bandit (MAB) problem is a sequential programming problem such that the player needs to select an action in each round, while the reward is only observed after the action is chosen. The basic motivation of the player is to maximize its gain among limited rounds [1] . The scenario comes up with the dilemma of exploration and exploitation. Specifically, the player can explore more actions to get more information on the reward of actions. Or the player can be fixed on the present best empirical action and get full use of the temporal largest reward while suffering from the danger of missing the best action. Fig. 1–1a shows the situation a play may face at the beginning of a specific round t.



(a) Diagram of multi-armed bandits

(b) Las Vegas slot machines

Figure 1–1 Multi-armed bandit problem and slot machines[2]

This problem comes from the game of a gambler, who sits at a row of slot machines and make decisions on machines to play, as shown in Fig. 1–1b, in each round of the game. The empirical motivation of MAB problem includes clinical trials [2] (investigating the effects of different experimental treatments while minimizing patient losses) and financial portfolio design [3]. Nowadays, the bandit problem or the idea of bandit (i.e. the payoff of an action is only observable after the action is taken) plays an important role in various tasks, including but not restricted to advertising placement [4], website optimization [5], and packet routing for minimizing delays in a network [6]. In particular, one of the most common applications of MAB problem is the content recommendation [7] system in e-commerce website or social media platform.

As for the mathematical form of the MAB problem, we can associated each arm with a distribution, i.e. distribution B_i for arm i, for $i \in \{1, ..., N\}$. Denote the mean value of distribution B_i as μ_i. There are a set of distributions $\{B_1, ..., B_N\}$. At round t, if player pull arm i, then he will get a reward $r_i(t)$ from a sample of distribution B_i. The player can only observe the reward $r_i(t)$ after he pulling the arm

[2]Fig. 1–1a is from https://towardsdatascience.com/solving-the-multi-armed-bandit-problem-b72de40db97c, and Fig. 1–1b is from Mr. Yamaguchi, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=57295504.

i at round t. In most of the cases, the player are only assumed to pull the arm for T times. In other words, there are only T rounds. The player needs to make decision in each round, with the objective is to maximize the sum of the collected rewards. The regret $R(T)$ after T rounds is defined as the difference between the maximum reward by an optimal policy and the total collected reward, which is given by

$$R(T) = T \max_{i \in \{1,...,N\}} \mu_i - \sum_{t=1}^{T} r_i(t)$$

In fact, the MAB problem can be regard as a one-state Markov decision process.

Apart from the traditional MAB problem, Bandit convex optimization (BCO) can be regarded as generalization of MAB problem. Specifically, in the context of BCO, the actions set becomes a continuous region instead of a discrete set for MAB. Besides, BCO is general and powerful framework for modeling learning problems with sequential data under partial feedback [8]. In the BCO model, at each round, the learner selects an action from a bounded convex set and observes the payoff of an action by a convex loss function. The feedback received is only the value of the loss function at $\mathbf{x}_t$, i.e. $f_t(\mathbf{x}_t)$, and no gradient or any other higher order information about the function is provided to the learner, not to mention the loss function itself at time t. BCO can model more complex scenario than traditional MAB and shows powerful expressiveness in many applications [9].

As for the mathematical form of the BCO, it is a iterative game between a player and an adversary, for T rounds. At each round $t \in \{1, ..., T\}$, the player will choose an action $\mathbf{x}_t \in \mathcal{K}$. Then, the adversary will choose a loss function $f_t \in \mathcal{F}$, independent of the user's action. Finally, the player will suffer the loss $f_t(\mathbf{x}_t)$. And the loss value $f_t(\mathbf{x}_t)$ is the only feedback that the player will receive. Here, $\mathcal{K}$ is the decision set, and $\mathcal{F}$ refers to the function set. Generally, for OCO (Online Convex Optimization, full information version of BCO) scenario, the decision set $\mathcal{K}$ is assumed to be convex, and that all functions in F are assumed to be convex as well. The similar assumption is adopted here for BCO framework. The regret $R(T)$ is defined as the difference of the sum of the received loss and the minimum sum of the loss by an optimal policy, which is given by

$$R(T) = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}_t)$$

The player's objective is the similar as that of MAB, i.e. minimizing his regret. The feedback sequence $f_1(\mathbf{x}_1), ..., f_T(\mathbf{x}_T)$ corresponding to its action sequence $\mathbf{x}_1, ..., \mathbf{x}_T$ helps the player to learn and improve its strategy. In the OCO framework, the loss function $f_t(\mathbf{x})$ will be send to the player as the feedback, thus the player will adopt a strategy by gradient. In other words, the player will calculate the gradient of the loss function within the action set. However, only limited information, i.e. the loss value can be observed by the player, for the BCO framework. The player will try to make gradient estimations by introducing randomness. The player' objective will be modified slightly, that is, to minimize the expected regret $\mathbb{E}\{R(T)\}$.

It is worth to mention that one of the most interesting applications of MAB or BCO is the content recommendation [7] system in e-commerce website or social media platform. However, the

original MAB or BCO framework is not powerful enough to model this scenario. Because, in these applications, the feedback of the user after receiving the recommendation is not immediately reported to the system. Instead, the feedback will be occurred sometime in the future and fades over time, after the recommendation was issued. The delay of feedback or reward makes the original MAB or BCO problem more intractable. Generally speaking, the delay depends on many factors, especially those related to users. This leads to the fact that the observable reward of the present time slot is a composite effect of some previously issued recommendation.

The scenarios of MAB problem with delayed feedback has been investigated in the literature. However, existing works make the assumption that the contributions of past recommendations to the combined reward is individually knowable, which is not always satisfied in the real system. Recently, in [10], the authors considered the bandits with delayed feedback and only the combined reward is available to the system, while the individual reward components remain unknown. Then in [11], the authors extended the work in [10]. In particular, they assumed that the payoff of action is spread among d continuous time slot. Besides, instead of a stochastic bandit, the adversarial bandit was considered here. The authors in [11] named this kind of feedback as composite anonymous feedback, which portrayed the effect of an action on the time scale.

However, even the composite anonymous feedback setting cannot model the complexity in the real system. The problem remains at the proper estimation of delay parameter d, which is the length of consecutive effect of a single action. Online advertising provide several use cases for this setting. The delay parameter d is always unknown in reality. For example, a click due to the browsing of pictures, later followed by conversion or a user that interacts with a recommended item multiple times over several days. In these cases, people are hard to tell how much time the user will respond to the system and for exactly how long it will last. Thus, it is urgent to investigate the MAB and BCO problem with composite anonymous feedback, where the delay parameter d is unknown.

In this project, we will be dedicated to solving the MAB and BCO problem in the harder settings than existing works to model more complex scenario. Specifically, we expect to solve bandit problem where the feedback or payoff of an action is more complex. For example, we may pay attention to the case where payoff is not observed immediately after the action is taken, but spread over multiple time slots. The problem will be tricky to handle once the length of the effective time slot of an action is unknown. In other words, d is the delay parameter which is unknown in our assumption. This implies that the instantaneous loss observed by the player at the end of each round is a sum of as many as d loss components of previously played actions. Hence, unlike the standard bandit setting with delayed feedback, where the player can observe the individual delayed losses, but only their sum. Besides, due to the agnosticism of the delay parameter d, the player even not know how many previous actions have an effect on the observed reward.

Apart from novel non-stochastic or adversarial MAB problem, the BCO with general feedback as the extension of our base algorithm and will be investigated as well. As we all know, BCO is a key framework for modeling learning problems with sequential data under partial feedback. The learner'

s objective is to minimize his regret, that is the difference between his cumulative loss over a finite number of rounds and that of the loss of the best-fixed action in hindsight. The limited feedback makes the BCO setup relevant to a number of applications. Moreover, it is exact the limited information that makes this problem hard. Generally, it falls into the area of online learning. The BCO problem with various feedback remains open and we are interested in solving this problem while preserving the optimality [12]. The possible contribution of this thesis can be given as follow:

1. Initially, we are dedicated to designing an algorithm or strategy for players under this setting of bandit problem and BCO. One the one hand, we will try to design a new algorithm which is suitable for players to adopt in this harder bandit setting. On the other hand, we will try to propose a general reduction, which can transform a standard bandit algorithm into one that can operate in this harder setting. Both kinds of research topic will be interesting. It will be challenging but meaningful to generalize our algorithm proposed in the context of the bandit problem to BCO and ensure its robustness. In other words, we will regard BCO as a generalized MBA problem with the same feedback setting.
2. Apart from designing the algorithm or general reduction method. It is of significant importance to show the performance of the algorithm. Specifically, it will be required to show the regret of the designed algorithm, including the upper bound and lower bound. It will be diverting to show how the regret of the transformed algorithm can be bounded in terms of the regret of the original algorithm if we focus on designing a general reduction. In a word, we will pay much attention to analyze the regret of the proposed algorithm.
3. Moreover, the optimality of the proposed algorithm will be our goal as well. It will be exciting to prove that our proposed algorithm is optimal or the reduction is optimal such that cannot be improved in general. For example, to prove the optimality of the reduction, one feasible idea is to prove a lower bound on the regret of any bandit algorithm in this setting that matches the upper bound obtained via our reduction.
4. Ultimately, numeric simulation is required to verify the effectiveness of the proposed algorithm. Extensive experiments should be conducted to prove the robustness of our algorithm in multiple applications. Besides, the comparison between our algorithm and other algorithms under the same setting will be convincing as well. It will be shown that in what a way our algorithm outperforms existing works.

Chapter 2 Related Work

In this chapter, we will list out some existing works related to MAB and BCO problem, including the research history, the state-of-the-art algorithms, variations and interesting applications.

2.1 Multi-armed Bandit

According to the nature of the reward process, bandit problem can be divided into three categories, including stochastic bandit, adversarial bandit, and Bayesian bandit. For each type of MAB problem, there is a effective algorithm to solve it.

Initially, for the stochastic bandit problem, the Upper Confidence bounds (UCB) algorithm was proposed in [13], which realized a $\log(T)$ regret upper bound. The authors considered a stochastic multi-armed bandit problem for infinite time horizon with the goal of proposing novel policies whose regret bound is small enough. They constructed index policies that relied on the playoff from each arm by their sample mean. These policies are computationally efficient and are also more general. They achieved a $O(\log(T))$ regret upper bound.

For the adversarial bandit problem, the Exp3 (which stands for “exponential-weight algorithm for exploration and exploitation") was proposed in [14]. In this paper, the author made no statistical assumptions about the distribution of the process generating the rewards for each arm. They gave a solution to the bandit problem in which an adversary is fighting against the player. For a game of T rounds, they showed that the reward of their policy approached that of the optimal policy at the rate $\sqrt{T}$. They also proved the optimality of this bound by a showing a feasible lower bound. Moreover, they also showed that their policy approached the reward of any policies at a similar rate. Finally, they proved their proposed algorithm approached the minimax reward of an unknown strategy at the rate $O(\sqrt{T})$.

Ultimately, for the Bayesian or Markovian bandit problem, the Gittins index method was proposed in [15], and then improved in [16] largely. By the way, the Bayesian or Markovian bandit means the bandit where played an action modifies its state in the Markovian space while it is not changed when not played. The player will get a reward, which depends on a state, after he pulling an arm. The number of states and the state transition probabilities of an action are unknown to the player. In this setting, they proved that with specific condition on the state transition probabilities of different arms, a sample mean based index policy reached logarithmic regret. Their work revealed that sample mean based index algorithms can be applied to the Markovian bandit problem without deterioration in the order.

In this thesis, we focus on adversarial bandit and adversarial bandit convex optimization with oblivious rewards.

2.2 Bandit Convex Optimization

The advantage of OCO framework is its power to generalize many problems from the field of online learning, and provide tools to fixing them. Adequate research throughout the past decade has broght many ingenious algorithms with worst case performance lower bound. That is why many researcher choose to adopt and study the OCO framework in their work. Find the tight regret upper bound of the BCO is an open problem and relatively challenging. The challenges comes from the limited informations bottleneck, that the player only receives a single point function value. A natural problem for the player is to balance between exploiting and exploring.

The general idea of BCO algorithm is to transform a gradient-based algorithm for OCO to a gradient-based algorithm for BCO by gradient estimation. Two common gradient estimators are the sphere sampling estimator and the ellipsoidal sampling estimator. In particular, we can perform gradient descent without calculating the gradient. Instead, we use the gradient estimator, which is based on the randomness. In [17], the author summarized a general reduction from limited information to full information as

1. One universal method for apply an OCO algorithm that relies only on the gradients of the loss functions, to a set of random vector variables with designed sophisticated rule .
2. Constructing the random variables that will help the reduction template to reach small enough regret bound.

For adversarial BCO, this problem was first considered by authors in [8]. They designed a clever algorithm to resolves the "exploration-exploitation" dilemma by proposing a gradient estimation algorithm. They reached an expected regret upper of $O(T^{3/4})$ in a game against an adversary with bounded and Lipschitz-continuous convex losses. Then in [18], the authors showed a lower bound of $\Omega(\sqrt{T})$ for the bandit with an adversary, having strongly-convex and smooth loss functions. Table 2–1 from [19] shows the state-of-the-art regret bound of OCO and BCO in versatile environments. In particular, the authors in [19] proposed an efficient algorithm that achieves a regret of $\Omega(\sqrt{T})$ for BCO with the setting assuming that the adversary's loss function is strongly-convex and smooth and the actions set is a constrained decision set. It is the work shown in the cell to the far right of the second line of the Table 2–1.

<table>
<tr><th>Setting</th><th>Convex</th><th>Linear</th><th>Smooth</th><th>Str.-Convex</th><th>Str.-Convex & Smooth</th></tr>
<tr><td>Full-Info.</td><td colspan="2">$\Theta(\sqrt{T})$</td><td colspan="2">$\Theta(\log(T))$</td></tr>
<tr><td rowspan="2">BCO</td><td>$O(T^{3/4})$</td><td>$O(\sqrt{T})$</td><td colspan="2">$O(T^{2/3})$</td><td>$O(\sqrt{T})$</td></tr>
<tr><td colspan="5">$\Omega(\sqrt{T})$</td></tr>
</table>

Table 2–1 Known regret bounds in the Full-Info./ BCO setting.

For stochastic BCO, the state-of-the-art progress was obtained by authors in [20]. Their work considered the problem stochastic BCO where loss functions are convex, Lipschitz on a convex and

compact set. The underlying ideal is to allow the algorithm to receive the noisy version of the function value at the chosen action point. Their algorithm reached $O(poly(n)\sqrt{T})$ regret, which means their algorithm is optimal in terms of the round number T. Here n is the dimension of the space.

Recently, there are some variations of traditional algorithm for BCO, such as multi-point query and multi-scale exploration. Specifically, the authors in [21] introduced the multi-point bandit setting, where the player can query a loss function for multiple times during one round of the game. They showed that the regret bound are very close to full information setting, i.e. the OCO setting where the loss function can be observed totally, if the player is allowed to query two point a time. Moreover, the regret bound can be even proved to be the same as that of full information setting, if the player is allowed to query m points a time. Then in [22], the author constructed a map to solve the adversarial BCO. The map is from a convex function to a multi-scale exploration distribution with respect to the function. The authors proved their algorithm reached $O(poly(n)\sqrt{T})$ regret bound.

2.3 MAB & BCO with Delayed Feedback

Online advertising and product recommendation are key applications for multi-armed bandit models. The reward is not always observable immediately after the action is taken. Instead, a delay is very common and it will be coupled with the feedback, which we refer to as a conversion. For example, when you are browsing an online shopping site, such as Taobao[1] or JD[2], you may need a few seconds, a few minutes, or even a few days to feel like clicking or buying a product, after the first time you see the picture of an item. It is of great practical significance to study the scenarios of MAB problem with delayed feedback.

The scenarios of MAB problem with delayed feedback has been investigated in the literature. In [23], the author analyzed the influence of delay on the regret bound for online learning strategies. They proposed a general reduction that transform an algorithm for non-delayed setting into algorithm that for the delay feedback. Then in [24], the author studied the MAB with delaye as well. They showed performance lower bounds as well as two simple but efficient algorithms based on the UCB [13] and KLUCB [25] frameworks. Besides, in [26], the authors was investigating on the networks of cooperative learning agents that communicate to solve a classical nonstochastic bandit problem. In particular, agents rely on a communication network to get informations about actions selected by other agents, and deliver messages with delay d. They introduced Exp3-Coop, a cooperative version of the Exp3 [14] algorithm and showed $O(\sqrt{T \ln K})$ regret bound, where K is the number of action. It is worth to mention that they provided the first characterization of the minimax regret for MAB with delayed feedback. However, these works like [23, 24, 26] make the assumption that the contributions of past recommendations to the combined reward is individually knowable, which is not always satisfied in the real systems.

Recently, in [10], the authors considered the bandits with delayed, aggregated anonymous feed-

[1] www.taobao.com
[2] www.jd.com

back, which is a variation of classical MAB problem. They assumed the payoff are postponed with a random delay. Besides, the information of which arm incurred to a specific payoff is lost. They proposed an algorithm that is the same as the worst regret of the non-anonymous setting when the delay parameter is bounded, and up to logarithmic factors for unbounded delay. Then in [11], the authors extended the work in [10]. In particular, they assumed that the payoff of action is spread among d continuous time slot, which makes this problem more intractable. Besides, instead of a stochastic bandit, the adversarial bandit was considered here. The authors in [11] named this kind of feedback as composite anonymous feedback, which portrayed the effect of an action on the time scale. They proposed a general reduction which can transform a base algorithm for MAB into ones that can be used in this hard setting. Moreover, a version for transform a base BCO algorithm is provided as well.

However, even the composite anonymous feedback setting is not powerful to model the real system due to the agnosticism of the delay parameter d. How to estimate the delay parameter d is a central problem to break the limitation of the model, where d is the length of consecutive effect of a single action. Online advertising provide several use cases for this setting. It is necessary to investigate the MAB and BCO problem with composite anonymous feedback, where the delay parameter d is unknown.

2.4 Application of Bandit Problem

The bandit has been applied in various applications. The empirical motivation of MAB problem includes clinical trials [2] and financial portfolio design [3]. Nowadays, the bandit has plays an important role on advertising placement [4], website optimization [5], and packet routing for minimizing delays in a network [6]. In particular, one of the most common applications of MAB problem is the content recommendation [7] system in e-commerce website or social media platform.

At the very beginning, in [2], the authors studied the problem of randomized clinical trials. They optimized the two-armed Bernoulli bandit problem to a variety of ethically motivated cost functions. They conducted numerical experiments that produced a heuristic approximation that applied even to very large horizons. Besides, they proposed a near-optimal strategy that is appropriate even when the horizon is unknown or unbounded. Since the tradeoff between exploration and exploitation to maximize rewards in bandit problem naturally establishes a connection to portfolio choice problems. Thus, in [3], the authors proposed an algorithm for conducting online portfolio choices by effectively exploiting correlations among multiple arms. Their algorithm was based on UCB policy, and an optimal portfolio strategy was proposed. The algorithm demonstrated advantage in risk-adjusted return and cumulative wealth.

In addition to the empirical motivation of MAB problem, many interesting applications have shown up in the recent years. In [5], the authors introduced the two characters: exploration and exploitation, to apply the bandit algorithm to website optimization. Thy further applied ϵ-Greedy algorithm, Softmax algorithm, and UCB algorithm on the domain of website optimization. Then,

in [4], the authors defined a universal framework for combinatorial multi-armed bandit (CMAB) problems, where novel arms with unknown distributions become super arms. They proposed CUCB algorithm that reached $O(\log(T))$ regret. Interestingly, they applied their CMAB algorithm to probabilistic maximum coverage (PMC) for online advertising and gained success. It is worth to mention that in [6], the authors studied problem of the opportunistic routing in wireless ad-hoc networks under an unknown probabilistic local broadcast model. They applied bandit algorithm to this problem and proposed both centralized and distributed online learning algorithms, which approached the optimal logarithmic regret bound, with the objective of governing the sequential choice of relay nodes.

Chapter 3 Problem Stetting and Background

3.1 Problem Setting

3.1.1 Bandit Convex Optimization

In the Bandit Convex Optimization (BCO) model, the player chooses $\mathbf{x} \in \mathcal{K}$ at round t, where $\mathcal{K}$ is a convex and compact domain with dimension n. After committing to this choice, a cost function $f_t \in \mathcal{F} : \mathcal{K} \mapsto [0, 1]$ is revealed, which is convex over Ω. Here $\mathcal{F}$ is the bounded family of cost functions available to the adversary. In particular, the player does not know the entire loss function f_t. Instead, the cost incurred to the player is exactly the value of the cost function at the point she committed to $f_t(\mathbf{x})$.

In this work, we consider BCO with anonymous and composite feedback, which is similar to the model in [11]. We assume each function f_t is the sum of d sub-components $f_t^{(0)}, ..., f_t^{(d-1)}$, with $f_t^{(s)} : \mathcal{K} \mapsto [0, 1]$ so that, for any $\mathbf{x} \in \mathcal{K}$,

$$f_t(\mathbf{x}) = \sum_{s=0}^{d-1} f_t^{(s)}(\mathbf{x}) \in [0, 1]$$

For each $f_t^{(s)}(\mathbf{x})$ sub-component of the loss function $f_t(\mathbf{x})$, it is assumed to be convex as well. Besides, we can regard the sub-component $f_t^{(s)}(\mathbf{text})$ as a the product of original loss function $f_t(\mathbf{x})$ and a constant ratio $r_t^{(s)}$, which is given by

$$f_t^{(s)} = r_t^{(s)} f_t(\mathbf{x})$$

for $s = 0, ..., d-1$ and $t = 1, ..., T$. Here, the constant ratio $r_t^{(s)}$ is assumed to be non-negative and subject to the law of conservation of summation, which is given by

$$\sum_{s=0}^{d-1} r_t^{(s)} = 1$$

for $t = 1, ..., T$.

In this thesis, we consider oblivious adversaries, which means all the loss functions $f_1(\mathbf{x}), ..., f_T(\mathbf{x})$ and constant ratios $r_1^{(1)}, ..., r_1^{(d)}, ..., r_T^{(1)}, ..., r_T^{(d)}$, are decided before the beginning of the game, which are independent of user's action. Let T denote the total number of game iterations. At each round $t = 1, 2, ..., T$, the payer select action $\mathbf{y}_t \in \mathcal{K}$ and receives loss $f_t(\mathbf{y}_t)$. Just as mentioned above, the loss spreads over d consecutive time slots, with constant ratio $r_t^{(s)}$ in round $t + s$. Thus, the action $\mathbf{y}_t$ will incur $f_t^{(0)}(\mathbf{y}_t)$ at time t, $f_t^{(1)}(\mathbf{y}_t)$ at time $t + 1$, ..., $f_t^{(d-1)}(\mathbf{y}_t)$ at time $t + d - 1$. Nevertheless, it should not be neglected that what the player really receive at round t is the sum of d-many sub-components incurred by actions chosen in the nearest d time slots. The received feedback (or received composite

loss) is given by $f_\tau^\circ(\mathbf{y}_{\tau-d+1}, ..., \mathbf{y}_\tau)$ as

$$f_\tau^\circ(\mathbf{y}_{\tau-d+1}, \mathbf{y}_{\tau-d+2}, ..., \mathbf{y}_\tau) = \sum_{s=0}^{d-1} f_{t-s}^{(s)}(\mathbf{y}_{t-s}) = \sum_{s=0}^{d-1} r_{t-s}^{(s)} f_{t-s}(\mathbf{x}_{d-s}) \tag{3–1}$$

where $r_t^{(s)} = 0$ when $t < s$.

Next, we need to bound the received composite loss. At the first glance, we may conclude $f_\tau^\circ(\mathbf{x}_{\tau-d+1}, \mathbf{x}_{\tau-d+2}, ..., \mathbf{x}_\tau)$ is within the region $[0, d]$, since $f_t(\mathbf{x})$ and its sub-component are within the region $[0, 1]$. In fact, the tight bound of composite loss can be given in the following theorem.

Theorem 3.1.
The cumulative sum of d recent composite losses up to round t by action $\mathbf{x}_t$ over d consecutive steps is within the region $[0, 2d - 1]$.

Proof.

$$\sum_{\tau=t-d+1}^{t} f_\tau^\circ(\mathbf{x}, ..., \mathbf{x}) = \sum_{\tau=t-d+1}^{t} \sum_{s=0}^{d-1} f_{t-s}^{(s)}(\mathbf{x}) \leq \sum_{\tau=t-2d+1}^{t} \sum_{s=0}^{d-1} f_\tau^{(s)}(\mathbf{x}) = \sum_{\tau=t-2d+1}^{t} f_\tau(\mathbf{x}) \leq 2d - 1$$

□

In fact, Theorem 3.1 is the BCO counterpart of Eq. (2) in [11].

The goal of the algorithm is to minimize its expected regret $R(T)$ against the best fixed action from the feasible domain $\mathcal{K}$. Thus, we have

$$R_T = \mathbb{E}\Big[\sum_{t=1}^{T} f_\tau^\circ(\mathbf{x}_{\tau-d+1}, ..., \mathbf{x}_\tau) \Big] - \min_{\mathbf{x}} \sum_{t=1}^{T} f_t^\circ(\mathbf{x}, \mathbf{x}, ..., \mathbf{x})$$

We define the expected regret with respect to the real received composite loss f_t° instead of the actual loss f_t, since the composite loss is the player really receives during the game.

One the on hand, setting is almost the same as that of [11], which generalizes the composite loss function of [27] as well. In particular, the linear composite loss function in [27] can be regarded as a instantiation of the composite loss (3–1) once we neglect different sub-components from the loss function components. It is worth to mention that in the linear case, the feedback in [27], individual loss be easily reconstructed in a recursive manner, which is impossible in our setting.

On the other hand, our model setting is very different from that in [11], that is our delay parameter d is assumed to be unknown. In other words, apart from the delayed and composite feedback, we consider a harder setting than the model in [11]. Specifically, we assume the delay parameter d is unknown, which means we do not have any prior knowledge about d. How to estimate the delay parameter is the key of the algorithm design to solve this harder problem.

3.1.2 Multi-armed Bandit

In addition to Bandit Convex Optimization, Multi-armed Bandit will be considered at the same time. Since MAB can be regarded as a simplification of BCO. We will introduce the model setting in

MAB shortly, which is given in the following. In fact, most of the parts are same as that of BCO with delayed and anonymous feedback while the delay parameter d is unknown, except the that the feasible set has changed to a discrete set from a continuous constrained set in BCO. By the way, MAB is not our major problem in this thesis.

The nonstochastic MAB with oblivious adversary is consisted of N arms. The feedback is delayed and anonymous while the delay parameter d is unknown. The loss at round t is given by $l_t(i) \in [0, 1]$ of action $i \in \{1, ..., N\}$ as

$$l_t(i) = \sum_{s=0}^{d-1} l_t^{(s)}(i)$$

The $l_t(i)$ is the sum of d sub-components, $l_t(i)^{(s)} \geq 0$ for $s = 0, ..., d-1$. Denote the action chosen by the player at round t as I_t. If $I_t = i$, action I_t will incur losses $l_t^{(0)}(i)$ at time t, ..., $l_t^{(d-1)}(i)$ at time $t + d - 1$. In this case, the player receives delayed and anonymous loss, which is the composite effect of recent d actions. Thus, we define composite loss function $\widetilde{l}_t$ of sequences of actions $i_1, ..., i_d$ as

$$\widetilde{l}_t(i_1, ..., i_d) = \sum_{s=0}^{d-1} l_{t-s}^{(s)}(i_{d-s})$$

where $l_{t-s}^{(s)}(i_{d-s}) = 0$ when $t < s$. In other words, what the player really observe at round t is the composite loss

$$\widetilde{l}_t(I_{t-d+1}, I_{t-d+2}, ..., I_t) = \sum_{s=0}^{d-1} l_{t-s}^{(s)}(I_{t-s})$$

The objective of the player is the same as that of BCO, which is to minimize the expected regret $R(T)$

$$R(T) = \mathbb{E}\Big[\sum_{t=1}^{T} \widetilde{l}_t(I_{t-d+1}, ..., I_t) \Big] - \min_{\mathbf{x} \in \{1,...,N\}} \sum_{t=1}^{T} \widetilde{l}_t(k, ..., k)$$

3.2 Background

In this section, we introduce the concept of strong convexity and smoothness. Besides, we will introduce the basic procedure to transform a algorithm works on online convex optimization into one works on bandit convex optimization.

3.2.1 Strong Convexity and Smoothness

As shown in the Chapter 4, we assume the loss functions to be smooth in Algorithm 4–6, and to be smooth and strong convex in 4–9, respectively. That is why we will introduce theses concepts here.

Dcefinition 3.1.

[Strong Convexity] *According to [28], we say that a function $f : \mathbb{R}^n \to \mathbb{R}$ is σ-strongly convex over the set $\mathcal{K}$ if for all $x, y \in \mathcal{K}$ it holds that*

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\sigma}{2} \parallel x - y \parallel^2$$

Dcefinition 3.2.

[Smoothness] *According to [28], we say that a convex function $f : \mathbb{R}^n \to \mathbb{R}$ is β-smooth over the set $\mathcal{K}$ if the following holds:*

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{\sigma}{2} \parallel x - y \parallel^2$$

3.2.2 Self Concordant Barriers

Let $\mathcal{K} \in \mathbb{R}^n$ be a convex set with a non empty interior int($\mathcal{K}$).

Dcefinition 3.3.

A function $\mathcal{R} : int(\mathcal{K}) \to \mathbb{R}$ is called ν-self=concordant if:

1. *$\mathcal{R}$ is three times continuously differentiable and convex, and approaches infinity along any sequence of points approaching the boundary of $\mathcal{K}$.*
2. *For every $h \in \mathbb{R}^n$ and $x \in int(\mathcal{K})$ the following holds:*

$$|\nabla^3 \mathcal{R}(x)[h, h, h]| \leq 2(\nabla^2 \mathcal{R}(x)[h, h])^{3/2}$$

and

$$|\nabla \mathcal{R}(x)[h]| \leq \nu^{1/2} (\nabla^2 \mathcal{R}(x)[h, h])^{1/2}$$

where

$$\nabla^3 \mathcal{R}(x)[h, h, h] \triangleq = \frac{\partial^3}{\partial t_1 \partial t_2 \partial t_3} \mathcal{R}(x + t_1 h + t_2 h + t_3 h)|_{t_1 = t_2 = t_3 = 0}$$

The Algorithm 4–4 and Algorithm 4–7 requires a ν-self-concordant barrier function over $\mathcal{K}$. There is a fact that any convex set in $\mathbb{R}^n$ admits a $\nu = O(n)$ barrier. The Hessian of a self-concordant barrier induces a local norm at every $x \in \text{int}(K)$, we denote this norm by $\| \cdot \|_x$ and its dual by $\| \cdot \|_x^*$ and define $\forall h \in \mathbb{R}^n$:

$$\begin{aligned} \parallel h \parallel_x &= \sqrt{h^\top \nabla^2 \mathcal{R}(x) h} \\ \parallel h \parallel_x^* &= \sqrt{h^\top (\nabla^2 \mathcal{R}(x)^{-1}) h} \end{aligned}$$

where we assume that $\nabla^2 \mathcal{R}(x)$ always has a full rank.

Let $\mathcal{R}$ to be a self-concordant barrier and $x \in \text{int}(\mathcal{K})$, then the Dikin Ellipsoide,

$$W_1(x) = \{y \in \mathbb{R}^n \mid\mid\mid y - x \parallel_x \leq 1\}$$

which is the $\| \cdot \|_x$-unit ball centered around x, is completely contained in $\mathcal{K}$.

3.2.3 Reduction from BCO to OCO

In [17], the author summarized a general reduction from limited information to full information as

1. One universal method for apply an OCO algorithm that relies only on the gradients of the loss functions, to a set of random vector variables with designed sophisticated rule .
2. Constructing the random variables that will help the reduction template to reach small enough regret bound.

3.2.3.1 Using Unbiased Estimators

It is of significant importance to find an observable random variable $\mathbf{g}_t$ that satisfies $\mathbf{E}[\mathbf{g}_t] \approx \nabla f_t(\mathbf{x}_t) = \nabla_t$ for designing algorithm solving bandit convex optimization. Only by this way, $\mathbf{g}_t$ can be seen as an estimator of the gradient, such that we can substitute $\mathbf{g}_t$ for ∇_t in an OCO algorithm. Besides, the family of regret minimization algorithms for which this reduction works is shown in the following definition from [17].

Dcefinition 3.4.

[First order OCO Algorithm] *Let $\mathcal{A}$ be an OCO (deterministic) algorithm receiving an arbitrary sequence of differential loss functions $f_1, ..., f_T$, and producing decisions $\mathbf{x}_{\leftarrow}\mathcal{A}(\emptyset), \mathbf{x}_t \leftarrow \mathcal{A}(f_1, ..., f_{t1})$. $\mathcal{A}$ is called a first order online algorithm if the following holds:*

- *The family of loss functions is closed under addition of linear functions: if $f \in \mathcal{F}_0$ and $\mathbf{u} \in \mathbb{R}^n$ then $f + \mathbf{u}^\top \mathbf{x} \in \mathcal{F}_0$.*
- *Let $\hat{f}_t$ be the linear function $\hat{f}_t = \nabla f_t(\mathbf{x}_t)^\top \mathbf{x}$, then for every iteration $t \in [T]$:*

$$\mathcal{A}(f_1, ..., f_{t-1}) = \mathcal{A}(\hat{f}_1, ..., \hat{f}_{t-1})$$

And a formal reduction from any first order online algorithm to a bandit convex optimization algorithm is also given in [17], which is shown in Algorithm 3–1

Algorithm 3–1 Reduction to bandit feedback.

Input: convex set $\mathcal{K} \subset \mathbb{R}^n$, first order online algorithm $\mathcal{A}$.

Initialize: Let $\mathbf{x}_1 = \mathcal{A}(\emptyset)$.

1: **for** $t = 1$ to T **do**
2: Generate distribution $\mathcal{D}_t$, sample $\mathbf{y}_t \sim \mathcal{D}_t$ with $\mathbb{E}[\mathbf{y}_t] = \mathbf{x}_t$.
3: Play $\mathbf{y}_t$.
4: Observe $f_t(\mathbf{y}_t)$, generate $\mathbf{g}_t$ with $\mathbb{E}[\mathbf{g}_t] = \nabla f_t(\mathbf{x}_t)$.
5: Let $\mathbf{x}_{t+1} = \mathcal{A}(\mathbf{g}_1, ..., \mathbf{g}_t)$.
6: **end for**

The reduction's regret bounds can be given by the Lemma 6.4 from [17], which is restated in the following:

Lemma 3.1.

Let $\mathbf{u}$ be a fixed point in $\mathcal{K}$. Let $f_1, ..., f_T : \mathcal{K} \to \mathbb{R}$ be a sequence of differentiable functions. Let $\mathcal{A}$ be a first order online algorithm that ensures a regret bound of the form $R_T(\mathcal{A}) \leq B_{\mathcal{A}}(\nabla f_1(\mathbf{x}1), ..., \nabla f_T(\mathbf{x}_T))$ in the full information setting. Define the points $\{\mathbf{x}_t\}$ as: $\mathbf{x}_{\leftarrow}\mathcal{A}(\emptyset), \mathbf{x}_t \leftarrow \mathcal{A}(\mathbf{g}_1, ..., \mathbf{g}_{t1})$ where each $\mathbf{g}_t$ is a vector valued random variable such that:

$$\mathbf{E}[\mathbf{g}_t \mid \mathbf{x}_1, f_1, ..., \mathbf{x}_t, f_t] = \nabla f_t(\mathbf{x}_t).$$

Then the following holds fro all $\mathbf{u} \in \mathcal{K}$*:*

$$\mathbb{E}[\sum_{t=1}^{T} f_t(\mathbf{x}_t)] - \sum_{t=1}^{T} f_t(\mathbf{u}) \leq \mathbf{E}[B_{\mathcal{A}}(\mathbf{g}_1, ..., \mathbf{g}_T)].$$

Proof. Define the functions $h_t : \mathcal{K} \to \mathbb{R}$ as follows:

$$h_t(\mathbf{x}) = f_t(\mathbf{x}) + \xi_t^\top \mathbf{x},$$

where $\xi_t = \mathbf{g}_t - \nabla f_t(\mathbf{x}_t)$. Notice that

$$\nabla h_t(\mathbf{x}_t) = \nabla f_t(\mathbf{x}_t) + \mathbf{g}_t - \nabla f_t(\mathbf{x}_t) = \mathbf{g}_t.$$

Therefore, deterministically applying a first order method $\mathcal{A}$ on the random functions h_t is equivalent to applying $\mathcal{A}$ on a stochastic first order approximation of the deterministic functions f_t. Thus by the full- information regret bound of $\mathcal{A}$ we have:

$$\sum_{t=1}^{T} h_t(\mathbf{x}_t) - \sum_{t=1}^{T} h_t(\mathbf{u}) \leq B_{\mathcal{A}}(\mathbf{g}_1, ..., \mathbf{g}_T). \tag{3–2}$$

Also note that:

$$\begin{aligned}
\mathbb{E}[h_t(\mathbf{x}_t)] &= \mathbb{E}[f_t(\mathbf{x}_t)] + \mathbb{E}[\xi_t \mathbf{x}_t] \\
&= \mathbb{E}[f_t(\mathbf{x}_t)] + \mathbb{E}[\mathbb{E}[\xi_t \mathbf{x}_t \mid \mathbf{x}_1, f_1, ..., \mathbf{x}_t, f_t]] \\
&= \mathbb{E}[f_t(\mathbf{x}_t)] + \mathbb{E}[\mathbb{E}[\xi_t \mid \mathbf{x}_1, f_1, ..., \mathbf{x}_t, f_t]^\top \mathbf{x}_t] \\
&= \mathbb{E}[f_t(\mathbf{x}_t)].
\end{aligned}$$

where we used $\mathbb{E}[\mathbb{E}[\xi_t \mid \mathbf{x}_1, f_1, ..., \mathbf{x}_t, f_t] = 0$. Similarly, since $\mathbf{u} \in \mathcal{K}$ is fixed we have that $\mathbb{E}[h_t(\mathbf{u})] = f_t(\mathbf{u})$. The lemma follows from taking the expectation of Equation (3–2). □

3.2.3.2 Point-wise Gradient Estimators

In the preceding part we have described how to transform a first order algorithm for OCO to one that for BCO, using specific random variables. We now describe how to design these vector random variables. There are two estimators, i.e. the sphere sampling estimator and the ellipsoidal sampling estimator.

- **Sphere Estimator** Let $\mathbf{x} \in \mathbf{R}^n$, and let $\mathbb{B}_\delta$ and $\mathbb{S}_\delta$ denote the n-dimensional ball and sphere with radius δ:

$$\begin{aligned}
\mathbb{B}_\delta &= \{\mathbf{x} \mid\mid \mathbf{x} \mid\mid \leq \delta\} \\
\mathbb{S}_\delta &= \{\mathbf{x} \mid\mid \mathbf{x} \mid\mid = \delta\}
\end{aligned}$$

  We define $\hat{f}(\mathbf{x}) = \hat{f}_\delta(\mathbf{x})$ to be a δ-smoothed version of $f(\mathbf{x})$:

$$\hat{f}_\delta(\mathbf{x}) = \mathbb{E}_{\mathbf{v} \in \mathbb{B}}[f(\mathbf{x} + \delta \mathbf{v})] \tag{3–3}$$

  where $\mathbf{v}$ is drawn uniform distribution over the unit ball. The following Lemma from [17] gives the property of the sphere sampling estimator:

Lemma 3.2.

Fix $\delta > 0$. Let $\hat{f}\delta(\mathbf{x})$ be as defined in Eq. (3–3), and let $\mathbf{u}$ *be a uniformly drawn unit vector* $\mathbf{u} \sim \mathbb{S}$. *Then*

$$\mathbb{E}_{\mathbf{v}\in\mathbb{B}}[f(\mathbf{x} + \delta\mathbf{v})] = \frac{\delta}{n}\nabla\hat{f}_\delta(\mathbf{x}).$$

- **Ellipsoidal Estimator** The sphere estimator above sometimes is difficult to apply if the center of the sphere is very close to the boundary of the feasible set. Thus, it is necessary to consider ellipsoidal estimators. The following corollary from [17] gives the property of the ellipsoidal sampling estimator:

  Corollary 3.1.

  Consider a continuous function $f : \mathbb{R}^n \to \mathbb{R}$, an invertible matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, and let $\mathbf{v} \sim \mathbb{B}^n$ and $\mathbf{u} \sim \mathbb{S}^n$. Define the smoothed version of f with respect to A:

  $$\hat{f}(\mathbf{x}) = \mathbb{E}[f(\mathbf{x} + \mathbf{A}\mathbf{v})].$$

  Then the following holds:

  $$\nabla\hat{f}(\mathbf{x}) = n\mathbb{E}[f(\mathbf{x} + \mathbf{A}\mathbf{u})\mathbf{A}^{-1}\mathbf{u}].$$

Chapter 4 Algorithm Framework for Delayed and Anonymous Feedback

In this chapter, we present our algorithm framework for bandit with delayed and anonymous feedback. There are two algorithm frameworks, one for MAB problem and the other for BCO. The procedure of two framework is almost the same. Specifically, we will be particularly interested on the framework for BCO, because MAB problem can be regarded as a simplification of BCO. BCO is more intractable and challenging. If we can solve a harder problem, there is no need to solve an easier one again. The framework for MAB will be presented in Section 4.3 for the integrity of the thesis. For both BCO and MAB, the instantiation of the algorithm framework will be given, in addition to the algorithm framework itself.

4.1 Algorithm Framework for the Bandit Convex Optimization

In this section, we present our algorithm framework for bandit convex optimization whit delayed and anonymous feedback, while the delay parameter d is unknown. Our algorithm framework can transform a base algorithm for BCO into one that can be operated under the setting of delayed and anonymous feedback while the delay parameter d is unknown. In this thesis, we assume the adversary is oblivious, which means the loss function is determined before the game starts. In other words, the adversary will not change no matter what action the player has taken. Our algorithm framework for BCO in this case is shown in Algorithm 4–1.

Algorithm 4–1 Algorithm framework for BCO with delayed and anonymous feedback (delay parameter d is unknown)

Input: Base BCO algorithm $\mathcal{A}$ with parameter h, T_1, T.

Initialize: $k = 1, t = 1, \mathcal{U} = \emptyset$.

Play any $\mathbf{y}_1 \in \Omega$.
while $t \leq T$ **do**
  $d_k \leftarrow h(T_k)$.
  $q_k \leftarrow \frac{1}{2d_k}$.
  Generate $2d_k - 1$ i.i.d. Bernoulli random variables $B_t, ..., B_{t+2d_k-2}$ with parameter q_k.
  while $t < T_k$ **do**
    if $t - 1 \in \mathcal{U}$ **then**
      play $\mathbf{y}_t$ by randomly perturbing state variable $\mathbf{x}_t$.
    else
      $\mathbf{y}_t \leftarrow \mathbf{y}_{t-1}$.
    end if
    Generate Bernoulli random variable B_{t+2d_k-1} with parameter q_k.
    if $B_t = 1, B_{t+1} = ... = B_{t-2d_k+1} = 0$ **then**
      Set $t \in \mathcal{U}$
      Feed the base BCO algorithm with average composite loss

$$\bar{f}_t = \frac{1}{2d_k} \sum_{\tau=t-d_k+1}^{t} f_\tau^\circ(\mathbf{y}_{\tau-d_k+1}, ..., \mathbf{y}_\tau)$$

      Use the update rule $\mathbf{x}_t \longrightarrow \mathbf{x}_{t+1}$ of base BCO to obtain the new state variable $\mathbf{x}_{t+1}$.
    end if
    $t \leftarrow t + 1$
  end while
  $k \leftarrow k + 1$
$T_k \leftarrow 2T_{k-1}$
end while

The algorithm has an outer loop. This loop decouples the algorithm framework into many rounds. In the following, we will illustrate on intuit of designing the algorithm by two aspects of among phases and within one phase, respectively.

4.1.1 Among Phases

Our algorithm framework extends the reduction in [11] to handle the setting where the delay parameter d is unknown. The algorithm framework can run without the prior information about the delay parameter d. Besides, it exploits a function $h : \mathbb{N}^+ \longrightarrow \mathbb{N}^+$ to make estimation on the precise

value of the delay parameter d. Shortly speaking, we estimate the delay parameter d by $d = h(T)$. The estimated delay parameter in the k phase is d_k with $d_k = h(T_k)$, while $T_k/2$ is the length of this phase. The phase length in increased exponentially, i.e. $T_{k+1} = 2T_k$. Thus, the k-th phase is exactly round $[T_{k-1}, T_k]$. Within each phase, we just substitute the reduction from [11] to the algorithm in each phase. The basic procedure described above is shown in Algorithm 4–2.

Algorithm 4–2 Algorithm framework among phases

Input: Base BCO algorithm $\mathcal{A}$ with parameter h, T_1, T.

Initialize: $k = 1, t = 1, \mathcal{U} = \emptyset$.

1: **while** $t \leq T$ **do**
2: $d_k \leftarrow h(T_k)$.
3: $q_k \leftarrow \frac{1}{2d_k}$.
4: Play any $\mathbf{y}_1 \in \Omega$.
5: Reduction from [11] for phase k.
6: $k \leftarrow k + 1$
7: $T_k \leftarrow 2T_{k-1}$
8: **end while**

4.1.2 Within One Phase

Within each phase, our algorithm framework adopts a similar procedure as that of the work in [11]. Specifically, [11] proposed a reduction based on a randomized approach. The basic ideal is to split the game into many rounds. Within each round, only partial information is exploited. The stochasticity ensure that the expected payoff in each time slot is the same, regardless of whether it is considered in the algorithm or not. This is the main trick of the reduction in [11].

The algorithm within one phase is the same as that of [11], where the authors named it as "Composite Loss Wrapper" algorithm, as shown in Algorithm 4–3. Here the base BCO algorithm works on normal losses within $[0, 1]$, producing state variable $\mathbf{x}_t$ on the feasible set $\mathcal{K}$. Algorithm 4–3 produces a sequence of i.i.d. Bernoulli random variables, $B_0, ..., B_T$, with parameter q. There are three kind of rounds in Algorithm 4–3, i.e. update round, silent round, random round.

- Update round: In this round, the state variable is updated by the rule of the base BCO algorithm by $\mathbf{x}_t \rightarrow \mathbf{x}_{t+1}$. The chosen action is the same as the last time slot, i.e. $\mathbf{y}_t = \mathbf{y}_{t-1}$.
- Silent round: In this round, the state variable $\mathbf{x}_t$ remain unchanged, and the chosen action is the same as last time slot, which is given by $\mathbf{y}_t = \mathbf{y}_{t-1}$.
- Random round: In this round, the chosen action $\mathbf{y}_t$ is generated by randomly perturbing state variable $\mathbf{x}_t$.

Algorithm 4–3 The Composite Loss Wrapper for BCO.

Input: Base BCO algorithm $\mathcal{A}$ with parameter $\eta \in (0, 1]$.

Initialize: $t = 1, \mathcal{U} = \emptyset$.

Play any $\mathbf{y}_1 \in \Omega$.
Generate $2d - 1$ i.i.d. Bernoulli random variables $B_t, ..., B_{t+2d-2}$ with parameter q.
while $t \leq T$ **do**
  if $t - 1 \in \mathcal{U}$ **then**
    play $\mathbf{y}_t$ by randomly perturbing state variable $\mathbf{x}_t$.
  else
    $\mathbf{y}_t \leftarrow \mathbf{y}_{t-1}$.
  end if
  Generate Bernoulli random variable B_{t+2d-1} with parameter q.
  if $B_t = 1, B_{t+1} = ... = B_{t-2d+1} = 0$ **then**
    Set $t \in \mathcal{U}$
    Feed the base BCO algorithm with average composite loss

$$\bar{f}_t = \frac{1}{2d} \sum_{\tau=t-d+1}^{t} f_\tau^\circ(\mathbf{y}_{\tau-d+1}, ..., \mathbf{y}_\tau)$$

    Use the update rule $\mathbf{x}_t \longrightarrow \mathbf{x}_{t+1}$ of base BCO to obtain the new state variable $\mathbf{x}_{t+1}$.
  end if
  $t \leftarrow t + 1$
end while

4.2 Instantiation of Algorithm Framework (BCO)

In this section, we substitute two base algorithm for bandit convex optimization into the algorithm framework in Algorithm 4–1. The first base BCO algorithm is proposed in [29]. This algorithm is designed to solve BCO with adversary having smooth convex loss functions, and reaches expected regret of $O(T^{2/3})$, ignoring constant and logarithmic factors. This algorithm is not the optimal algorithm. Then in [19], the authors proposed an algorithm for BCO with adversary having smooth and strong convex loss. It is worth to mention that this algorithm reaches $O(\sqrt{T})$ regret upper bound, which is optimal. Because the regret lower bound for BCO has been proved to be $\Omega(\sqrt{T})$ in [18].

4.2.1 A Basic BCO Algorithm

At first, we give the base bandit algorithm from [29] in Algorithm 4–4.

Algorithm 4–4 Bandit OCO Algorithm for Smooth Functions

Input: $\eta > 0$, $\delta \in [0, 1]$, ν-self-concordant barrier $\mathcal{R}$.

Initialize: Choose $\mathbf{x}_1 \in \mathcal{K}$ randomly.

1: **for** t=1,2,...,T **do**
2: Define $\mathbf{A}_t = (\nabla^2 \mathcal{R}(\mathbf{x}_t))^{-1/2}$.
3: Draw $\mathbf{u}_t \sim \mathbb{S}^n$ uniformly at random.
4: $\mathbf{y}_t = \mathbf{x}_t + \delta \mathbf{A}_t \mathbf{u}_t$
5: Play $\mathbf{y}_t$ and receive $f_t(\mathbf{y})_t \in \mathbb{R}$.
6: $\mathbf{g}_t = \frac{n}{\delta} f_t(\mathbf{y}_t) \mathbf{A}_t^{-1} \mathbf{u}_t$.
7: Update $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in \mathcal{K}} \eta \sum_{\tau=1}^{t} g_\tau^\top \mathbf{x} + \mathcal{R}(\mathbf{x})$.
8: **end for**

Like common bandit algorithms, Algorithm 4–4 is based on the estimation gradient and then passing the estimated gradient to a full information algorithm. The underlying full information algorithm is an algorithm from [30], which is given in Algorithm 4–5.

Algorithm 4–5 Bandit Online Linear Optimization

Input: $\eta > 0$, ν-self-concordant barrier $\mathcal{R}$.

Initialize: Choose $\mathbf{x}_1 = \arg\min_{\mathbf{x} \in \mathcal{K}} \mathcal{R}(\mathbf{x})$.

1: **for** t=1,2,...,T **do**
2: Let $\{\mathbf{e}_1, ..., \mathbf{e}_n\}$ and $\{\lambda_1, ..., \lambda_n\}$ be the set of eigenvectors and eigenvalues of $\nabla^2 \mathcal{R}(\mathbf{x}_t)$.
3: Choose i_t uniformly at random from $\{1, ..., n\}$ and $\epsilon_t = \pm 1$ with probability 1/2.
4: Predict $\mathbf{y}_t = \mathbf{x}_t + \epsilon \lambda_{i_t}^{-1/2} \mathbf{e}_{i_t}$.
5: Observe the gain $\mathbf{f}_t^\top \mathbf{y}_t \in \mathbb{R}$.
6: Define $\widetilde{\mathbf{f}}_t = n(\mathbf{f}_t^\top \mathbf{y}_t) \epsilon_t \lambda_{i_t}^{1/2} \cdot e_{i)t}$.
7: Update

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in \mathcal{K}} [\eta \sum_{\tau=1}^{t} \widetilde{\mathbf{f}}_t^\top \mathbf{x} + \mathcal{R}(\mathbf{x})]$$

.
8: **end for**

Theorem 4.1.

Let the set $\mathcal{K}$ have diameter $\mathcal{D}$. Suppose we run Algorithm 4–4 against an arbitrary sequence of functions f_t all drawn from smooth set and bounded by C. Then, for appropriate choices of the parameters η, δ, the expected regret is bounded as:

$$R(T) \leq 3(H\nu \log T)^{1/3} (Cd\mathcal{D})^{2/3} T^{2/3} + (\frac{2C}{\mathcal{D}} + \mathcal{D}H)\sqrt{T} = O(T^{2/3} (\log T)^{1/3})$$

Then, we are ready to give the algorithm for BCO with delayed and anonymous feedback where the delay parameter d is unknown with base algorithm by Algorithm 4–4. The full algorithm is given in Algorithm 4–6. The analysis of this algorithm is in next chapter.

Algorithm 4–6 Algorithm for BCO with delayed and anonymous feedback (basic version)

Input: Base BCO algorithm $\mathcal{A}$ with parameter h, T_1, T.

Initialize: $k = 1, t = 1, \mathcal{U} = \emptyset$.

Play any $\mathbf{y}_1 \in \Omega$.
while $t \leq T$ **do**
  $d_k \leftarrow h(T_k)$.
  $q_k \leftarrow \frac{1}{2d_k}$.
  Generate $2d_k - 1$ i.i.d. Bernoulli random variables $B_t, ..., B_{t+2d_k-2}$ with parameter q_k.
  while $t < T_k$ **do**
    if $t - 1 \in \mathcal{U}$ **then**
      play $\mathbf{y}_t$ by randomly perturbing state variable $\mathbf{x}_t$.
    else
      $\mathbf{y}_t \leftarrow \mathbf{y}_{t-1}$.
    end if
    Generate Bernoulli random variable B_{t+2d_k-1} with parameter q_k.
    if $B_t = 1, B_{t+1} = ... = B_{t-2d_k+1} = 0$ **then**
      Set $t \in \mathcal{U}$
      Feed the algorithm by [29] with average composite loss

$$\bar{f}_t = \frac{1}{2d_k} \sum_{\tau=t-d_k+1}^{t} f_\tau^\circ(\mathbf{y}_{\tau-d_k+1}, ..., \mathbf{y}_\tau)$$

  and set the parameter as $\eta_k = O\left(\left(\frac{2d_k \log(T_k/(2d_k))}{nT_k}\right)^{2/3}\right)$.
      Use the update rule $\mathbf{x}_t \rightarrow \mathbf{x}_{t+1}$ of base BCO to obtain the new state variable $\mathbf{x}_{t+1}$.
    end if
    $t \leftarrow t + 1$
  end while
  $k \leftarrow k + 1$
$T_k \leftarrow 2T_{k-1}$
end while

4.2.2 An Optimal BCO Algorithm

If we replace the base BCO algorithm in Algorithm 4–6 by the algorithm proposed in [19], the regret of the new algorithm, i.e. the Algorithm 4–9 in the following, will reach a better regret upper bound. The algorithm proposed in [19] is given Algorithm 4–8.

Algorithm 4–7 BCO Algorithm for Str.-convex & Smooth losses

Input: $\eta > 0$, $\sigma > 0$, ν-self-concordant barrier $\mathcal{R}$.

Initialize: Choose $\mathbf{x}_1 = \arg\min_{\mathbf{x}\in\mathcal{K}} \mathcal{R}(\mathbf{x})$.

1: **for** t=1,2,...,T **do**
2: Define $\mathbf{B}_t = (\nabla^2\mathcal{R}(\mathbf{x}_t) + \eta\sigma t\mathcal{I})^{-1/2}$.
3: Draw $\mathbf{u}_t \sim \mathbb{S}^n$.
4: Play $\mathbf{y}_t = \mathbf{x}_t + \mathbf{B}_t\mathbf{u}_t$.
5: Observe $f_t(\mathbf{x}_t + \mathbf{B}_t\mathbf{u}_t)$ and define $g_t = nf_t(\mathbf{x}_t + \mathbf{B}_t\mathbf{u}_t)\mathbf{B}_t^{-1}\mathbf{u}_t$.
6: Update $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{K}} \eta\sum_{\tau=1}^{t}\{g_\tau^\top\mathbf{x} + \frac{\sigma}{2} \parallel \mathbf{x} - \mathbf{x}_\tau \parallel\} + \mathcal{R}(\mathbf{x})$.
7: **end for**

Just as that stated in [19], Algorithm 4–7 shrinks the exploration magnitude with time due to the strong-convexity of the loss functions. In fact, it follows a full-information first-order algorithm for online convex optimization, which is given in Algorithm 4–8, denoted as FTARL-σ.

Algorithm 4–8 FTARL-σ

Input: $\eta > 0$, ν-self-concordant barrier $\mathcal{R}$.

Initialize: Choose $\mathbf{x}_1 = \arg\min_{\mathbf{x}\in\mathcal{K}} \mathcal{R}(\mathbf{x})$.

1: **for** t=1,2,...,T **do**
2: Receive $\nabla h_t(\mathbf{x}_t)$.
3: Update $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{K}} \eta\sum_{\tau=1}^{t}\{\nabla h_t(\mathbf{x}_t)^\top\mathbf{x} + \frac{\sigma}{2} \parallel \mathbf{x} - \mathbf{x}_\tau \parallel\} + \mathcal{R}(\mathbf{x})$.
4: **end for**

Algorithm 4–8 is a variant of the FTRL algorithm, which is given in [31]. The following theorem is given in [19].

Theorem 4.2.
Let $\mathcal{K}$ be a convex set with diameter $\mathcal{D}$ and R be a ν-self-concordant barrier over $\mathcal{K}$. Then in the BCO setting where the adversary is limited to choosing β-smooth and σ-strongly- convex functions and $|f_t(\mathbf{x})| \le L$, $\forall\mathbf{x} \in \mathcal{K}$, then the expected regret of Algorithm 4–4 with $\eta = \sqrt{\frac{(\nu+2\beta/\sigma)\log T}{2n^2L^2T}}$ is upper bounded as

$$R(T) \le 4nL\sqrt{(\nu + \frac{2\beta}{\sigma})T\log T} + 2L + \frac{\beta\mathcal{D}^2}{2} = O(\sqrt{\frac{\beta\nu}{\sigma}T\log T})$$

whenever $T/\log T \ge 2(\nu + 2\beta/\sigma)$.

Then, we are ready to give the algorithm for BCO with delayed and anonymous feedback where the delay parameter d is unknown with base algorithm by Algorithm 4–7. The full algorithm is given in Algorithm 4–9. The analysis of this algorithm is in next chapter.

Algorithm 4–9 Algorithm for BCO with delayed and anonymous feedback (optimal version)

Input: Base BCO algorithm $\mathcal{A}$ with parameter h, T_1, T.

Initialize: $k = 1, t = 1, \mathcal{U} = \emptyset$.

Play any $\mathbf{y}_1 \in \Omega$.
while $t \leq T$ **do**
  $d_k \leftarrow h(T_k)$.
  $q_k \leftarrow \frac{1}{2d_k}$.
  Generate $2d_k - 1$ i.i.d. Bernoulli random variables $B_t, ..., B_{t+2d_k-2}$ with parameter q_k.
  while $t < T_k$ **do**
    if $t - 1 \in \mathcal{U}$ **then**
      play $\mathbf{y}_t$ by randomly perturbing state variable $\mathbf{x}_t$.
    else
      $\mathbf{y}_t \leftarrow \mathbf{y}_{t-1}$.
    end if
    Generate Bernoulli random variable B_{t+2d_k-1} with parameter q_k.
    if $B_t = 1, B_{t+1} = ... = B_{t-2d_k+1} = 0$ **then**
      Set $t \in \mathcal{U}$
      Feed the algorithm by [19] with average composite loss

$$\bar{f}_t = \frac{1}{2d_k} \sum_{\tau=t-d_k+1}^{t} f_\tau^\circ(\mathbf{y}_{\tau-d_k+1}, ..., \mathbf{y}_\tau)$$

  and set the parameter as $\eta_k = O\Big((\frac{2d_k \log(T_k/2d_k)}{nT_k})^{2/3}\Big)$.
      Use the update rule $\mathbf{x}_t \rightarrow \mathbf{x}_{t+1}$ of base BCO to obtain the new state variable $\mathbf{x}_{t+1}$.
    end if
    $t \leftarrow t + 1$
  end while
  $k \leftarrow k + 1$
$T_k \leftarrow 2T_{k-1}$
end while

Algorithm 4–7 works under the assumption of β-smooth and σ-strongly-convex functions. It updates according to a full-information first-order algorithm denoted FTARL-σ. This algorithm is a variant of the FTRL methodology as defined in [29]. The algorithm takes in input a learning rate η, a convexity parameter $\sigma > 0$, and a ν-self-concordant (barrier) function $\mathcal{R}$. The algorithm maintains at each round t the state variable $\mathbf{x}_t \in \mathcal{K}$, of the form

$$\mathbf{x}_t = \arg\min_{\mathbf{x}\in\mathcal{K}} \eta \sum_{s=1}^{t} (g_s^\top \mathbf{x} + \frac{\sigma}{2} \parallel \mathbf{x} - \mathbf{x}_s \parallel^2) + R(\mathbf{x}) \tag{4–1}$$

Then, it computes a perturbed version $\mathbf{y}_t$ of $\mathbf{x}_t$ as $\mathbf{y}_t = \mathbf{x}_t + \mathbf{B}_t \mathbf{u}_t$, where $\mathbf{B}_t$ is the Hessian matrix

$(\nabla^2\mathcal{R}(\mathbf{x}_t + \eta\sigma t\mathcal{I}))^{-1/2}$, and $\mathbf{s}_t$ is drawn uniformly at random from the surface of the Euclidean n-dimensional unit ball $\mathbb{B}^n$. The update $\mathbf{x}_t \rightarrow \mathbf{x}_{t+1}$ amounts to computing the next vector g_t in Eq. (4–1) as $g_t = nf_t(\mathbf{y}_t)\mathbf{B}_t^{-1}\mathbf{u}_t$, an unbiased estimate of the gradient at $\mathbf{x}_t$ of a smoothed version of f_t. From [19] one can bound

$$R(T, n, \eta) \leq 2\eta T n^2 + \frac{\log T}{\eta}(\nu + \frac{2\beta}{\sigma}) + 2 + \frac{\beta\mathcal{D}^2}{2}$$

4.3 Algorithm Framework for Multi-armed Bandit Problem

In this section, we present our algorithm framework for Multi-armed bandit problem whit delayed and anonymous feedback, while the delay parameter d is unknown.[1] Our algorithm framework can transform a base algorithm for MAB into one that can be operated under the setting of delayed and anonymous feedback while the delay parameter d is unknown. In this thesis, we assume the adversary is oblivious, which means the reward vector is determined at the beginning of the game. In other words, the adversary will not change no matter what action the player has taken. Our algorithm framework for MAB in this case is shown in Algorithm 4–10.

[1]This part of the work is first proposed by [32], Thus, it is listed here for purpose of the integrity of the thesis, but not the contribution of the authors' of this thesis.

Algorithm 4–10 Algorithm framework for MAB with delayed and anonymous feedback (delay parameter d is unknown)

Input: Base BCO algorithm $\mathcal{A}$ with parameter h, T_1, T.

Initialize: $k = 1, t = 1, \mathcal{U} = \emptyset$.

Draw a_0 from the uniform distribution $\mathbf{p}_1$ over $\{1, ..., K\}$.
while $t \leq T$ **do**
  $d_k \leftarrow h(T_k)$.
  $q_k \leftarrow \frac{1}{2d_k}$.
  Generate $2d_k - 1$ i.i.d. Bernoulli random variables $B_t, ..., B_{t+2d_k-2}$ with parameter q_k.
  while $t \leq T$ **do**
    if $t - 1 \in \mathcal{U}$ **then**
      Draw $a_t \sim \mathbf{p}_t$ and play it without updating $\mathbf{p}_t$ ($\mathbf{p}_{t+1} = \mathbf{p}_t$).
    else
      $a_t \leftarrow a_{t-1}$.
    end if
    Generate Bernoulli random variable B_{t+2d-1} with parameter q.
    if $B_t = 1, B_{t+1} = ... = B_{t-2d_k+1} = 0$ **then**
      Set $t \in \mathcal{U}$
      Feed the base MAB algorithm with average composite loss

$$\bar{l}_t = \frac{1}{2d} \sum_{\tau=t-d+1}^{t} l_\tau^\circ(a_{\tau-d+1}, ..., a_\tau)$$

      Use the update rule $\mathbf{p}_t \rightarrow \mathbf{p}_{t+1}$ of base MAB to obtain the new distribution $\mathbf{p}_{t+1}$.
    end if
    $t \leftarrow t + 1$
  end while
  $k \leftarrow k + 1$
$T_k \leftarrow 2T_{k-1}$
end while

The algorithm has an outer loop. This loop decouples the algorithm framework into many rounds. In each round, the algorithm follows the procedure of Algorithm 1 in [11], which shown in Algorithm 4–11.

Algorithm 4–11 The Composite Loss Wrapper for MAB.

Input: Base BCO algorithm $\mathcal{A}$ with parameter $\eta \in (0, 1]$.

Initialize: $t = 1, \mathcal{U} = \emptyset$.

Draw a_0 from the uniform distribution $\mathbf{p}_1$ over $\{1, ..., K\}$.
Generate $2d - 1$ i.i.d. Bernoulli random variables $B_t, ..., B_{t+2d-2}$ with parameter q.
while $t \leq T$ **do**
  if $t - 1 \in \mathcal{U}$ **then**
    Draw $a_t \sim \mathbf{p}_t$ and play it without updating $\mathbf{p}_t$ ($\mathbf{p}_{t+1} = \mathbf{p}_t$).
  else
    $a_t \leftarrow a_{t-1}$.
  end if
  Generate Bernoulli random variable B_{t+2d-1} with parameter q.
  if $B_t = 1, B_{t+1} = ... = B_{t-2d+1} = 0$ **then**
    Set $t \in \mathcal{U}$
    Feed the base MAB algorithm with average composite loss

$$\bar{l}_t = \frac{1}{2d} \sum_{\tau=t-d+1}^{t} l_\tau^\circ(a_{\tau-d+1}, ..., a_\tau)$$

    Use the update rule $\mathbf{p}_t \rightarrow \mathbf{p}_{t+1}$ of base MAB to obtain the new distribution $\mathbf{p}_{t+1}$.
  end if
  $t \leftarrow t + 1$
end while

Remark 4.1.

The algorithm framework for MAB with delayed and anonymous feedback while the delay parameter d is unknown can be regarded as simplification of the algorithm framework for BCO in the same feedback setting. Thus, we need not to repeat the analysis as well as algorithm instantiation again. However, these part will be given in the following section, for the sake of integrity of the paper. The algorithm framework with respect to MAB is first proposed in [32].

4.3.1 Instantiation of Algorithm Framework (MAB)

We substitute Exp3 [14] in Algorithm 4–10 to get an instantiation of algorithm framework for MAB with delayed and anonymous feedback while the delay parameter d is unknown. The instantiation is shown in Algorithm 4–13. At first the Exp3 algorithm is shown in Algorithm 4–12.

Algorithm 4–12 Algorithm Exp3

Input: Real $\gamma \in (0, 1]$.

Initialize: $\omega_i(1) = 1$ for $i = 1, ..., N$,

1: **for** t=1,2,...,T **do**

2: Set

$$p_i(t) = (1 - \gamma)\frac{\omega_i(t)}{\sum_{j=1}^{N} w_j(t)} + \frac{\gamma}{N}, \quad i = 1, ..., N$$

3: Draw i_t randomly accordingly to the probabilities $p_1(t), ..., p_N(t)$.

4: Receive reward $x_{i_t}(t) \in [0, 1]$.

5: **for** $j = 1, ..., N$ **do**

6: Set

$$\hat{x}_j(t) = \begin{cases} x_j(t)/p_j(t), & \text{if } j = i_t. \\ 0, & \text{otherwise.} \end{cases}$$

$$\omega_j(t + 1) = \omega_j(t) \exp(\gamma \hat{x}_j(t)/K)$$

7: **end for**

8: **end for**

The regret bound of Exp3 algorithm can be given in Theorem 3.1 from [14], which is shown in the following:

Theorem 4.3.

For any $N > 0$ and for any $\gamma \in (0, 1]$

$$G_{\max} - \mathbb{E}[G_{Exp3}] \leq (e - 1)\gamma G_{\max} + \frac{NlnN}{T}$$

holds for any assignment of rewards and for any $T > 0$.

Moreover, the Corollary 3.2 in [14] bound the regret of Exp3 algorithm explicitly in the following:

Corollary 4.1.

For any $T > 0$, assume that $g \leq G_{\max}$ and that algorithm Exp3 is run with input parameter

$$\gamma = \min\left\{1, \sqrt{\frac{N \ln N}{(e - 1)g}}\right\}.$$

Then

$$G_{\max} - \mathbb{E}[G_{Exp3}] \leq 2\sqrt{e - 1}\sqrt{gN \ln N} \leq 2.63\sqrt{gN \ln N}.$$

Then the Algorithm 4–10 with base Algorithm 4–12 is shown in the following:

Algorithm 4–13 Algorithm for MAB with delayed and anonymous feedback (Exp3 version)

Input: Base BCO algorithm $\mathcal{A}$ with parameter h, T_1, T.

Initialize: $k = 1, t = 1, \mathcal{U} = \emptyset$.

Draw a_0 from the uniform distribution $\mathbf{p}_1$ over $\{1, ..., K\}$.
while $t \leq T$ **do**
  $d_k \leftarrow h(T_k)$.
  $q_k \leftarrow \frac{1}{2d_k}$.
  $\gamma_k = \sqrt{\frac{2d_k N \log N}{T_k + d_k}}$.
  Generate $2d_k - 1$ i.i.d. Bernoulli random variables $B_t, ..., B_{t+2d_k-2}$ with parameter q_k.
  while $t \leq T$ **do**
    if $t - 1 \in \mathcal{U}$ **then**
      Draw $a_t \sim \mathbf{p}_t$ and play it without updating $\mathbf{p}_t$ ($\mathbf{p}_{t+1} = \mathbf{p}_t$).
    else
      $a_t \leftarrow a_{t-1}$.
    end if
    Generate Bernoulli random variable B_{t+2d-1} with parameter q.
    if $B_t = 1, B_{t+1} = ... = B_{t-2d_k+1} = 0$ **then**
      Set $t \in \mathcal{U}$
      Update $\mathbf{p}_t$ using Exp3 policy by pulling arm a_t and obtain reward

$$\frac{1}{2d_k} \sum_{\tau=t-d+1}^{t} l_\tau^\circ(a_{\tau-d+1}, ..., a_\tau)$$

    .
    end if
    $t \leftarrow t + 1$
  end while
  $k \leftarrow k + 1$
$T_k \leftarrow 2T_{k-1}$
end while

Chapter 5 Regret Analysis

In this chapter, we present our analysis on the algorithm framework for bandit with delayed and anonymous feedback. There are two algorithm frameworks, one for MAB problem and the other for BCO. The procedure of analysis on MAB and BCO is almost the same. Specifically, we will be particularly interested on the analysis for BCO, because MAB problem can be regarded as a simplification of BCO. BCO is more intractable and challenging. If we can give analysis on a harder problem, there is no need to analyze an easier one again. The analysis for BCO and MAB will be presented in Section 5.1 and Section 5.2, respectively, for the integrity of the thesis. For bandit convex optimization, both the analysis for basic version algorithm and optimal version algorithm will be given.

5.1 Bandit Convex Optimization

5.1.1 Basic Algorithm

The proof of Theorem 5.1 follows a similar procedure as the proof of Theorem 3 in [32].

Theorem 5.1.
If $h : N_+ \to N_+$ is an increasing function such that $h(T) = o(T)$ holds for any T, and

1. *for some constant $L \geq 0$, the loss functions $f_1, ..., f_T$ are L-Lipschitz on Ω w.r.t. $\|\cdot\|$,*
2. *for some constant $\beta \geq 0$, the loss functions $f_1, ..., f_T$ are β-smooth w.r.t. $\|\cdot\|$,*

then Algorithm 4–6 satisfies

$$R(T) = O\Big((h(2T)\log(T/h(2T)))^{1/3}(nT)^{2/3} + \sqrt{h(2T)T} + h^{-1}(d)\Big)$$

Proof. Let k_0 be the first round that $d_{k_0} \geq d$. Then, T_{k_0} satisfies $h(T_{k_0-1}) < d$. Since h is an increasing function, h^{-1} is also increasing. Thus, $T_{k_0-1} < h^{-1}(d)$, which implies $T_{k_0} < 2h^{-1}(d)$.

We now ignore the phases before k_0, as these time steps will have an additional regret of at most $2h^{-1}(d)$. Consider the phases with $d_k \geq d$. Notice that from Corollary 15 in [11], each phase k has regret $O\Big((d_k \log(t_k/d_k))^{1/3}(nt_k)^{2/3} + \sqrt{d_k t_k}\Big)$, where $t_k = T_k/2$ is the number of time steps in this phase. Specifically, Corollary 15 in [11] is given in the following:

Corollary 5.1.
If Algorithm 4–3 is run with the Algorithm 4–4 by [29] as Base BCO algorithm, with $\eta = O\Big(\big(\frac{d\log(T/d)}{nT}\big)^{2/3}\Big)$ and $\delta = O(\eta^{1/4}n^{1/2})$, then its regret for BCO with d-delayed composite anonymous feedback satisfies

$$R(T) = O\Big(\big(d\log(T/d)^{1/3}\big)(nT)^{2/3} + \sqrt{dT}\Big)$$

where the O notation in the tuning of η, δ and in the bound on $R(T)$ hides the constants β, $\mathcal{D}$ and ν.

Then, the total regret in these phases is upper bounded by:

$$\sum_{k=k_0}^{K} O\Big((d_k \log(t_k/d_k))^{1/3}(nt_k)^{2/3} + \sqrt{d_k t_k}\Big) \leq O\Big((d_K \log(t_K/d_K))^{1/3}(nt_K)^{2/3} + \sqrt{d_K t_K}\Big)$$

where K is the last phase number.

From the description of Algorithm 4–6, we have that $t_K \leq T$ and $d_K \leq h(2T)$. Thus, the total regret of Algorithm 4–6 satisfies $R(T) \leq O\Big((h(2T)\log(T/h(2T)))^{1/3}(nT)^{2/3} + \sqrt{h(2T)T} + h^{-1}(d)\Big)$. □

Remark 5.1.
Surprisingly, since the last term of regret of Theorem 5.1 does not depend on T, our regret bound can be arbitrary close to the regret lower bound given in [11], by restricting the increasing rate of h, without using any prior information on d. However, a small increasing rate of the round size will causes a large regret before we have $d_k \geq d$. In fact, the regret bound can be compressed to $\Theta(T^{2/3+\epsilon}(\log T^{1-\epsilon})^{1/3})$, which is quit close as the case of convex and smooth losses in [29], who attained an upper bound of $\Theta(T^{2/3}(\log T)^{2/3})$. This tells us the regret loss in anonymous and composite feedback is $O(h^{-1}(d))$.

5.1.2 Optimal Algorithm

It should be noted that the Theorem 5.2 is from Theorem 2.2 in [17].

Theorem 5.2 (Karush-Kuhn-Tucker)**.**
Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a convex set, $\mathbf{x}^* \in \arg\min_{x\in\mathcal{K}} f(\mathbf{x})$. Then for any $\mathbf{y} \in \mathcal{K}$ we have

$$\nabla f(\mathbf{x}^*)^\top(\mathbf{y} - \mathbf{x}^*) \geq 0.$$

Proof. The generalization of the fact that a minimum of a convex differentiable function on $\mathbb{R}^n$ is a point in which its derivative is equal to zero, is given by the multi-dimensional analogue that its gradient is zero:

$$\nabla f(\mathbf{x}) = 0 \Leftrightarrow \mathbf{x} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}).$$

We will require a slightly more general, but equally intuitive, fact for constrained optimization: at a minimum point of a constrained convex function, the inner product between the negative gradient and direction towards the interior of $\mathcal{K}$ is non-positive. This is depicted in Figure 5–1, which shows that $\nabla f(\mathbf{x}^*)$ defines a supporting hyperplane to $\mathcal{K}$. The intuition is that if the inner product were positive, one could im- prove the objective by moving in the direction of the projected negative gradient. This fact is stated formally in the above theorem.

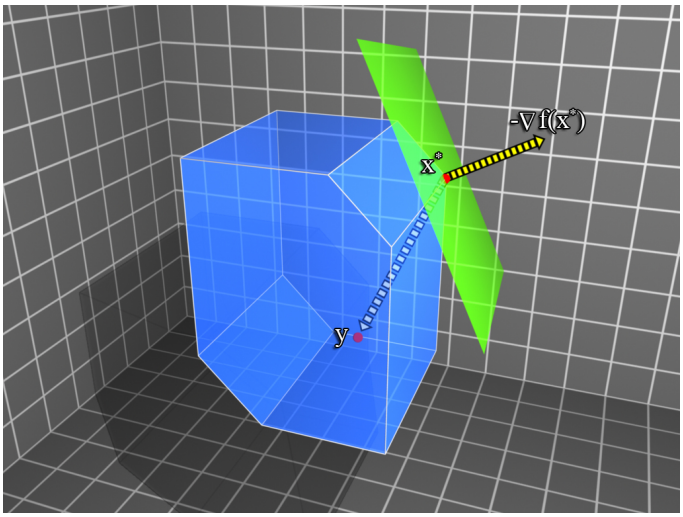


Figure 5–1 Optimality conditions: negative sub-gradient pointing outwards.[2]

□

Lemma 5.1.

$$\| \mathbf{x}_t - \mathbf{x}_{t+1} \| = O(\eta n)$$

Proof. The proof is similar as that of the proof of Lemma 14 in [11]. Consider the Bregman divergence associated with the (strongly convex) barrier function

$$B_R(\mathbf{x}||\mathbf{y}) = R(\mathbf{x}) - R(\mathbf{y}) - \nabla R(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}).$$

Recall that $R(\mathbf{x})$ is a convex function and $\mathcal{K}$ is a convex set. Denote:

$$\Phi_t(\mathbf{x}) \triangleq \left\{ \eta \sum_{s=1}^{t} (\mathbf{g}_s^\top \mathbf{x} + \frac{\sigma}{2} \| \mathbf{x} - \mathbf{x}_s \|^2) + R(\mathbf{x}) \right\}$$

For twice differentiable functions, by the mean-value theorem, the Taylor expansion shows that the Bregman divergence is the same as the second derivative at an intermediate point, i.e.,

$$B_R(\mathbf{x}||\mathbf{y}) = \frac{1}{2} \| \mathbf{x} - \mathbf{y} \|_{\mathbf{z}}$$

for $\mathbf{z} \in [\mathbf{x}, \mathbf{y}]$, or $\mathbf{z} = \alpha \mathbf{x} + (1 - \alpha)\mathbf{y}$ for $\alpha \in [0, 1]$.

By the Taylor expansion (with its obvious remainder term through the mean-value theorem) at $\mathbf{x}_{t+1}$, and by the definition of the Bregman divergence,

$$\begin{aligned} \Phi_t(\mathbf{x}_t) &= \Phi_t(\mathbf{x}_{t+1}) + (\mathbf{x}_t - \mathbf{x}_{t+1})^\top \nabla \Phi_t(\mathbf{x}_{t+1}) + B_{\Phi_t}(\mathbf{x}_t||\mathbf{x}_{t+1}) \\ &\geq \Phi_t(\mathbf{x}_{t+1}) + B_{\Phi_t}(\mathbf{x}_t||\mathbf{x}_{t+1}) \end{aligned} \tag{5–1}$$

[2]Fig. 5–1 is from Fig. 2.2 in [17]

The inequality holds because $\mathbf{x}_{t+1}$ is a minimum point of Φ_t over $\mathcal{K}$, as in Theorem 5.2. Denote

$$\Psi_t(\mathbf{x}) \triangleq \left\{\eta \sum_{s=1}^{t} \frac{\sigma}{2} \parallel \mathbf{x} - \mathbf{x}_s \parallel^2 + R(\mathbf{x})\right\}$$

Then,

$$\begin{aligned}
& B_{\Psi_t}(\mathbf{x}_t||\mathbf{x}_{t+1}) - B_R(\mathbf{x}_t||\mathbf{x}_{t+1}) \\
= \; & \eta \sum_{s=1}^{t} \frac{\sigma}{2} \parallel \mathbf{x}_t - \mathbf{x}_s \parallel^2 - \eta \sum_{s=1}^{t} \frac{\sigma}{2} \parallel \mathbf{x}_{t+1} - \mathbf{x}_s \parallel^2 - \eta \sum_{s=1}^{t} \sigma(\mathbf{x}_{t+1} - \mathbf{x}_s)^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \\
= \; & \frac{\eta\sigma}{2} \sum_{s=1}^{t} \Big((\mathbf{x}_t - \mathbf{x}_s)^\top (\mathbf{x}_t - \mathbf{x}_s) - (\mathbf{x}_{t+1} - \mathbf{x}_s)^\top (\mathbf{x}_{t+1} - \mathbf{x}_s) - 2(\mathbf{x}_{t+1} - \mathbf{x}_s)^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \Big) \\
= \; & \frac{\eta\sigma}{2} \sum_{s=1}^{t} \Big((\mathbf{x}_t - \mathbf{x}_s)^\top (\mathbf{x}_t - \mathbf{x}_s) - (\mathbf{x}_{t+1} - \mathbf{x}_s)^\top (\mathbf{x}_t - \mathbf{x}_s) - (\mathbf{x}_{t+1} - \mathbf{x}_s)^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \Big) \\
= \; & \frac{\eta\sigma}{2} \sum_{s=1}^{t} \Big((\mathbf{x}_t - \mathbf{x}_{t+1})^\top (\mathbf{x}_t - \mathbf{x}_s) - (\mathbf{x}_{t+1} - \mathbf{x}_s)^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \Big) \\
= \; & \frac{\eta\sigma}{2} \sum_{s=1}^{t} (\mathbf{x}_t - \mathbf{x}_{t+1})^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \\
= \; & \frac{\eta\sigma t}{2} \parallel \mathbf{x}_t - \mathbf{x}_{t+1} \parallel^2
\end{aligned}$$

Thus, we have

$$B_{\Phi_t}(\mathbf{x}_t||\mathbf{x}_{t+1}) = B_{\Psi_t}(\mathbf{x}_t||\mathbf{x}_{t+1}) = B_R(\mathbf{x}_t||\mathbf{x}_{t+1}) + \frac{\eta\sigma t}{2} \parallel \mathbf{x}_t - \mathbf{x}_{t+1} \parallel^2 \tag{5–2}$$

The first equality holds because the term $\mathbf{g}_s^\top \mathbf{x}$ is linear and thus does not have effect on the Bregman divergence. Combine Eq. (5–1) with Eq. (5–2), we have

$$\begin{aligned}
B_R(\mathbf{x}_t||\mathbf{x}_{t+1}) \; & \leq \; \Phi_t(\mathbf{x}_t) - \Phi_t(\mathbf{x}_{t+1}) - \frac{\eta\sigma t}{2} \parallel \mathbf{x}_t - \mathbf{x}_{t+1} \parallel^2 \\
& = \; (\Phi_{t-1}(\mathbf{x}_t) - \Phi_{t-1}(\mathbf{x}_{t+1})) + \eta \mathbf{g}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) - \frac{\eta\sigma(t+1)}{2} \parallel \mathbf{x}_t - \mathbf{x}_{t+1} \parallel^2 \\
& \leq \; (\Phi_{t-1}(\mathbf{x}_t) - \Phi_{t-1}(\mathbf{x}_{t+1})) + \eta \mathbf{g}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \\
& \leq \; \eta \mathbf{g}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1})
\end{aligned} \tag{5–3}$$

The last inequality holds since $\mathbf{x}_t$ is the minimizer.

Denote the norm produced by the Bregman divergence with respect to $\mathcal{R}$ on point $\mathbf{x}_t$, $\mathbf{x}_{t+1}$ as $\|\cdot\|_t = \|\cdot\|_{\mathbf{x}_t, \mathbf{x}_{t+1}}$. The case is similar for the dual local norm $\|\cdot\|_t^* = \|\cdot\|_{\mathbf{x}_t, \mathbf{x}_{t+1}}^*$. By this notation, we have $B_{\mathcal{R}}(\mathbf{x}_t||\mathbf{x}_{t+1}) = \frac{1}{2} \parallel \mathbf{x}_t - \mathbf{x}_{t+1} \parallel_t^2$. The generalized Cauchy-Schwarz theorem asserts

$$\langle \mathbf{x}, \mathbf{y} \rangle \leq \parallel \mathbf{x} \parallel \parallel \mathbf{y} \parallel^*$$

and in particular for matrix norms,

$$\langle \mathbf{x}, \mathbf{y} \rangle \leq \parallel \mathbf{x} \parallel_{\mathbf{A}} \parallel \mathbf{y} \parallel_{\mathbf{A}}^*$$

where $\mathbf{A}$ is a positive definite matrix and $\| \mathbf{x} \|_{\mathbf{A}} = \sqrt{\mathbf{x}^\top \mathbf{A} \mathbf{x}}$. The dual norm of a matrix norm is $\| \mathbf{x} \|_{\mathbf{A}}^* = \| \mathbf{x} \|_{\mathbf{A}^{-1}}$.

Then, with the generalized Cauchy-Schwarz inequality, we have

$$\mathbf{g}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \leq \| \mathbf{g}_t \|_t^* \cdot \| \mathbf{x}_t - \mathbf{x}_{t+1} \|_t \tag{5–4}$$

To find the range of the last equation, we use Lemma 6.9 in [17], and the definition of $\mathbf{x}_t = \arg\min_{\mathbf{x} \in \mathcal{K}} \Phi_t(\mathbf{x})$ where

$$\Phi_t(\mathbf{x}) \triangleq \left\{ \eta \sum_{s=1}^{t} (\mathbf{g}_s^\top \mathbf{x} + \frac{\sigma}{2} \| \mathbf{x} - \mathbf{x}_s \|^2) + R(\mathbf{x}) \right\}$$

is a self-concordant barrier. Here Lemma 6.9 in [17] is restated in Lemma 5.2.

Lemma 5.2.
Let $\mathbf{x} \in int(\mathcal{K})$ be such that $\| \nabla \mathcal{R}(\mathbf{x}) \|_{\mathbf{x}}^* \leq \frac{1}{4}$, and let $\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathcal{K}} \mathcal{R}(\mathbf{x})$. Then

$$\| \mathbf{x} - \mathbf{x}^* \|_x \leq 2 \| \nabla \mathcal{R}(\mathbf{x}) \|_{\mathbf{x}}^*$$

Thus,

$$\| \mathbf{x}_t - \mathbf{x}_{t+1} \|_t \leq 2 \| \nabla \Phi_t(\mathbf{x}_t) \|_t^* = 2 \| \nabla \Phi_{t-1}(\mathbf{x}_t) + \eta \mathbf{g}_t \|_t^* = 2\eta \| \mathbf{g}_t \|_t^* \tag{5–5}$$

since $\Phi_{t-1}(\mathbf{x}_t) = 0$ by definition of $\mathbf{x}_t$. Recall that to use Lemma 6.9 in [17], we need $\| \nabla \Phi_t(\mathbf{x}_t) \|_t^* = \eta \| \mathbf{g}_t \|_t^* \leq \frac{1}{4}$, which is true by choice of η and since

$$\| \mathbf{g}_t \|_t^{*2} \leq n^2 \mathbf{u}^\top \mathbf{A}_t^{-\top} \nabla^{-2} \mathcal{R}(\mathbf{x}_t) \mathbf{A}_t^{-1} \mathbf{u} \leq n^2$$

By the strong convexity of Φ_t w.r.t. $\| \cdot \|$ we have

$$B_{\Phi_t}(\mathbf{x}_t || \mathbf{x}_{t+1}) \geq \frac{\alpha}{2} \| \mathbf{x}_t - \mathbf{x}_{t+1} \|^2$$

for some constant $\alpha > 0$. Moreover, one can show that $\mathbb{E}[\| \mathbf{g}_t \|_t^{*2} | \mathbf{x}_t] \leq n^2$ from Lemma 11 in [19], which is restated in Lemman 5.3.

Lemma 5.3.
Let $\mathcal{R}$ be a self-concordant barrier over a convex set $\mathcal{K}$, and $\eta > 0$. Consider an online player receiving σ-strongly-convex loss functions $h_1, ..., h_T$ and choosing points according to FTARL-σ (Algorithm 4–8), and $\eta \| \nabla h_t(\mathbf{x}_t) \|_t^* \leq 1/2$, $\forall t \in [T]$. Then the player's regret is upper bounded as follows:

$$\sum_{t=1}^{T} h_t(\mathbf{x}_t) - \sum_{t=1}^{T} h_t(\omega) \leq 2\eta \sum_{t=1}^{T} (\| \nabla h_t(\mathbf{x}_t) \|_t^*)^2 + \eta^{-1} \mathcal{R}(\omega), \quad \forall \mathbf{z} \in \mathcal{K}$$

where $(\| \mathbf{a} \|_t^*)^2 = \mathbf{a}^\top (\nabla^2 \mathcal{R}(\mathbf{x}_t) + \eta \sigma t \mathcal{I})^{-1} \mathbf{a}$

Put Eq. (5–4) and (5–5) together in Eq. (5–3) gives

$$\| \mathbf{x}_t - \mathbf{x}_{t+1} \| = O(\eta n)$$

where the O notation hides here the inverse dependence on α. □

The notion of stability of the Base BCO has now to refer also to the sequence of loss functions the algorithm is operating with. We have to consider only the positive part of the backward difference. The following definition of ξ-stable is first given in [11], which modified in this paper and given here as $\xi(t)$-stable.

Dcefinition 5.1.
Let $\mathcal{A}(\eta)$ be a Base BCO with learning rate η, and $\{\mathbf{y}_t\}_{t=1}^T$ be the sequence of plays produced by $\mathcal{A}(\eta)$ during a run over T rounds on the sequence of convex losses $\{f_t\}_{t=1}^T$. We say that $\mathcal{A}(\eta)$ is $\xi(t)$-stable w.r.t. $\{f_t\}_{t=1}^T$ if for any round t we have that

$$\Big[\mathbb{E}\big[f_{t+1}(\mathbf{y}_{t+1}) - f_{t+1}(\mathbf{y}_t)\big]\Big]_+ \leq \xi(t)$$

where $\xi(t)$ is a function of t.

Lemma 5.4.
Let $f_1, ..., f_T \colon \Omega \subseteq R^n \to [0,1]$ be a sequence of β-smooth convex losses w.r.t. $\|\cdot\|$, and $\mathcal{D}$ be the diameter of $\mathbf{x}$. Then the Base BCO algorithm by [19] is ξ-stable, with $\xi(t) = O\Big(\Big(\frac{1}{\mathcal{D}} + \mathcal{D}\beta\Big)\eta n + \frac{\beta}{\eta\sigma(t+1)}\Big)$.

Proof. Since $f : \Omega \to [0,1]$ is β-smooth w.r.t. $\|\cdot\|$, then

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y}-\mathbf{x}) + \frac{\beta}{2}\| \mathbf{y}-\mathbf{x} \|^2 .$$

Let $\mathbb{E}_t[\cdot]$ denote expectation conditioned on all random events up tp time $t-1$. Then, by the convexity of f_{t+1}, we have

$$\mathbb{E}[f_{t+1}(\mathbf{y}_t)] = \mathbb{E}\big[\mathbb{E}_t[f_{t+1}(\mathbf{y}_t)]\big] \geq \mathbb{E}\big[f_{t+1}(\mathbb{E}_t[\mathbf{y}_t])\big] = \mathbb{E}\big[f_{t+1}(\mathbb{E}_t[\mathbf{x}_t])\big] = \mathbb{E}[f_{t+1}(\mathbf{x}_t)].$$

The next equation holds by the remark that follows Lemma 7 and by the lemma itself in [19]:

$$\mathbb{E}[f_t(\mathbf{y}_t) - f_t(\mathbf{x}_t)] = \mathbb{E}\big[\mathbb{E}_{\mathbf{u}_t}[f_t(\mathbf{x}_t + \mathbf{B}_t\mathbf{u}_t)] - f_t(\mathbf{x}_t)|\mathbf{x}_t\big] \leq \frac{\beta}{2}\mathbb{E}[\| \mathbf{B}_t^2 \|] \leq \frac{\beta}{2\eta\sigma t},$$

where Lemma 7 and the following remark in [19] are given in the following:

Lemma 5.5.
Consider a function $f : \mathbb{R}^n \to \mathbb{R}$, and a positive definite matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$. Let $\hat{f}$ be the smoothed version of f with respect to $\mathbf{A}$ as defined in Equation (3–3). Then the following holds:

- *If f is σ-strongly convex then so is f.*
- *If f is convex and β-smooth, and $\lambda_{\max}$ be the largest eigenvalue of $\mathbf{A}$.*

Then:

$$0 \leq \hat{f}(\mathbf{x}) - f(\mathbf{x}) \leq \frac{\beta}{2}\| \mathbf{A}^2 \|_2 = \frac{\beta}{2}\lambda_{\max}^2$$

Remark 5.2.
Lemma 7 also holds if we define the smoothed version of f as $f(\mathbf{x}) = \mathbb{E}_{\mathbf{u}\sim\mathbb{S}^n}[f(\mathbf{x}+\mathbf{A}\mathbf{u})]$, i.e. an average of the original function values over the unit sphere rather than the unit ball as defined in Equation (3–3). Proof is similar to the one of Lemma 7.

Thus,

$$\mathbb{E}[f_{t+1}(\mathbf{y}_{t+1}) - f_{t+1}(\mathbf{x}_{t+1})] \leq \frac{\beta}{2\eta\sigma(t+1)}$$

Putting together, we have so far obtained

$$\begin{aligned}\Big[\mathbb{E}[f_{t+1}(\mathbf{y}_{t+1}) - f_{t+1}(\mathbf{y}_t)]\Big]_+ &\leq \Big[\mathbb{E}[f_{t+1}(\mathbf{x}_{t+1}) - f_{t+1}(\mathbf{x}_t)] + \frac{\beta}{2\eta\sigma(t+1)}\Big]_+ \\ &\leq \Big[\mathbb{E}[f_{t+1}(\mathbf{x}_{t+1}) - f_{t+1}(\mathbf{x}_t)]\Big]_+ + \frac{\beta}{2\eta\sigma(t+1)} \end{aligned} \tag{5–6}$$

where we have further used the fact that $[a]_+$ is nondecreasing in $a \in \mathbb{R}$, and that $[a+b]_+ \leq [a]_+ + [b]_+$ for all $a, b \in \mathbb{R}$.

Since f_t is [0,1]-bounded and β-smooth on a set of diameter $\mathcal{D}$, it must be that f_t is also Lipschitz with constant $L \leq \frac{2}{\mathcal{D}} + \mathcal{D}\beta$, so that combining with (5–6) and Lemma 5.1 yields

$$\Big[\mathbb{E}[f_{t+1}(\mathbf{y}_{t+1}) - f_{t+1}(\mathbf{y}_t)]\Big]_+ \leq (\frac{2}{\mathcal{D}} + \mathcal{D}\beta)\mathbb{E}[\| \mathbf{x}_{t+1} - \mathbf{x}_t \|] + \frac{\beta}{2\eta\sigma} = O\Big(\Big(\frac{1}{\mathcal{D}} + \mathcal{D}\beta\Big)\eta n + \frac{\beta}{\eta\sigma(t+1)}\Big)$$

□

Theorem 5.3.
If Algorithm 2 in [11] is run with the above mentioned Algorithm 4–7 with $\eta = O\Big(\sqrt{\frac{d\log(T/d)}{n^2T}}\Big)$*, then its regret for the BCO setting where the adversary is limited to choosing* β*-smooth and* σ*-strongly-convex functions with d-delayed composite anonymous feedback satisfies*

$$R(T) = \begin{cases} O\Big(\sqrt{d\log(T/d)T} + d\Big), & \text{if } d > n. \\ O\Big(n\sqrt{\frac{T\log(T/d)}{d}} + d\Big), & \text{otherwise.} \end{cases} \tag{5–7}$$

where the O *notation in the tuning of* η *and in the bound on* $R(T)$ *hides the constants* β*,* $\mathcal{D}$*,* ν *and* σ*.*

Proof. From the proof of Theorem 13 in [11], we have

$$\begin{aligned} & \mathbb{E}\Big[f_t^\circ(\mathbf{y}_{t-d+1}, ..., \mathbf{y}_t) - f_t^\circ(\mathbf{y}_{t-d+1}, ..., \mathbf{y}_{t-d+1})\Big] \\ = \ & \mathbb{E}\Big[\sum_{s=0}^{d-1}\big(f_{t-s}^{(s)}(\mathbf{y}_{t-s}) - f_{t-s}^{(s)}(\mathbf{y}_{t-d+1})\big)\Big] \\ = \ & \sum_{s=0}^{d-1}\big[\mathbb{E}[f_{t-s}^{(s)}(\mathbf{y}_{t-s}) - f_{t-s}^{(s)}(\mathbf{y}_{t-d+1})]\big]_+ \\ \leq \ & \xi(t) \end{aligned}$$

since there is at most one update of the underlying state variable $\mathbf{x}_t$ (which in turn determines the distribution of the corresponding $\mathbf{y}_t$) during the rounds from $t-d+1$ to t, and Base BCO is assumed to be $\xi(t)$-stable in the sense of Definition 5.1. Piecing together as in the proof of Theorem 2 in [11],

we have

$$\begin{aligned}
R(T) &\le \mathbb{E}\Big[\sum_{t=1}^{T} f_t^\circ(\mathbf{y}_{t-d+1},...,\mathbf{y}_t)\Big] - \sum_{t=1}^{T} f_t^\circ(\mathbf{x},...,\mathbf{x}) \\
&= \mathbb{E}\Big[\sum_{t=1}^{T} f_t^\circ(\mathbf{y}_{t-d+1},...,\mathbf{y}_t) - \sum_{t=1}^{T} f_t^\circ(\mathbf{y}_{t-d+1},...,\mathbf{y}_{t-d+1})\Big] \\
&+ \mathbb{E}\Big[\sum_{t=1}^{T} f_t^\circ(\mathbf{y}_{t-d+1},...,\mathbf{y}_{t-d+1})\Big] - \sum_{t=1}^{T} f_t^\circ(\mathbf{x},...,\mathbf{x}) \\
&\le \sum_{t=1}^{T} \xi(t) + 8(2d-1)R_{\mathcal{A}}(T/2d, K, \eta) + O(d) \\
&= \sum_{t=1}^{T} O\Big(\Big(\frac{1}{\mathcal{D}} + \mathcal{D}\beta\Big)\eta n + \frac{\beta}{\eta\sigma(t+1)}\Big) + 8(2d-1)\Big(\frac{\eta n^2 T}{d} + \frac{\log(\frac{T}{2d})}{\eta}(\nu + \frac{2\beta}{\sigma}) + 2 + \frac{\beta\mathcal{D}^2}{2}\Big) + O(d) \\
&= O\Big(\Big(\frac{1}{\mathcal{D}} + \mathcal{D}\beta\Big)\eta n T + \sum_{t=1}^{T} \frac{\beta}{\eta\sigma(t+1)}\Big) + 8(2d-1)\Big(\frac{\eta n^2 T}{d} + \frac{\log(\frac{T}{2d})}{\eta}(\nu + \frac{2\beta}{\sigma}) + 2 + \frac{\beta\mathcal{D}^2}{2}\Big) + O(d) \\
&= O\Big(\Big(\frac{1}{\mathcal{D}} + \mathcal{D}\beta\Big)\eta n T + \frac{\beta \log T}{\eta\sigma}\Big) + 8(2d-1)\Big(\frac{\eta n^2 T}{d} + \frac{\log(\frac{T}{2d})}{\eta}(\nu + \frac{2\beta}{\sigma}) + 2 + \frac{\beta\mathcal{D}^2}{2}\Big) + O(d) \\
&= O\Big(\eta n T + \eta n^2 T + \frac{\log T}{\eta} + \frac{d\log(\frac{T}{2d})}{\eta} + d\Big),
\end{aligned}$$

where the last equality follows the fact that $\beta, \mathcal{D}, \sigma, \nu$ are regarded as constants.

Take $\eta = O\Big(\sqrt{\frac{d\log(T/d)}{n^2 T}}\Big)$, we get $R(T) = O\Big(\sqrt{d\log(T/d)T} + n\sqrt{\frac{T\log(T/2d)}{d}} + d\Big)$ and Eq. (5–7) can be get easily. □

Theorem 5.4.

If Algorithm 4–6 is run with the above mentioned Algorithm 4–7 as Base BCO algorithm, then its regret for BCO with d-delayed composite anonymous feedback while d is unknown satisfies

$$R(T) = O\Big(\sqrt{h(2T)\log(T/h(2T))T} + h^{-1}(d)\Big)$$

If $h : N_+ \to N_+$ is an increasing function such that $h(T) = o(T)$ holds for any T, and

1. *$\mathcal{R}$ be a ν-self-concordant barrier over Ω,*
2. *$f_1, ..., f_T$ are β-smooth and σ-strongly-convex functions and $|f_t(x)| \le 1$, $\forall x \in \mathcal{K}$,*
3. *$d > n$.*

Proof. Let k_0 be the first round that $d_{k_0} \ge d$. Then, T_{k_0} satisfies $h(T_{k_0-1}) < d$. Since h is an increasing function, h^{-1} is also increasing. Thus, $T_{k_0-1} < h^{-1}(d)$, which implies $T_{k_0} < 2h^{-1}(d)$.

We now ignore the phases before k_0, as these time steps will have an additional regret of at most $2h^{-1}(d)$. Consider the phases with $d_k \ge d$. Notice that from Theorem 5.3, each phase k has regret $O\Big(\sqrt{d_k \log(t_k/d_k)t_k} + d_k\Big)$ when $d > n$, where $t_k = T_k/2$ is the number of time steps in this phase.

Then, the total regret in these phases is upper bounded by:

$$\sum_{k=k_0}^{K} O\left(\sqrt{d_k \log(t_k/d_k)t_k} + d_k\right) \leq O\left(\sqrt{d_K \log(t_K/d_K)t_K} + d_K\right)$$

where K is the last phase number.

From the description of Algorithm 4–6, we have that $t_K \leq T$ and $d_K \leq h(2T)$. Thus, the total regret of Algorithm 4–6 satisfies $R(T) \leq O\left(\sqrt{h(2T)\log(T/h(2T))T} + h^{-1}(d)\right)$. □

Remark 5.3.

Surprisingly, since the last term of regret of Theorem 5.4 does not depend on T, our regret bound can be arbitrary close to the regret lower bound given in Theorem 5.3, by restricting the increasing rate of h, without using any prior information on d. However, a small increasing rate of the round size will causes a large regret before we have $d_k \geq d$. In fact, the regret bound can be compressed to $\Theta(\sqrt{T^\epsilon \log T^{1-\epsilon}})$, which is quit close as the case of convex and smooth losses in [19], which attained an upper bound of $\Theta(\sqrt{T \log T})$. This tells us the regret loss in anonymous and composite feedback is $O(h^{-1}(d))$.

Remark 5.4.

We reach near $\Theta(T^{2/3})$ regret upper bound in Theorem 5.1 with a $\Theta(T^{2/3})$ base algorithm in [29]. And we reduce the regret upper bound further to $\Theta(\sqrt{T})$ by adopting $\Theta(\sqrt{T})$-base algorithm in [19]. This indicates that the anonymity of the delay parameter d can only cause limited damage to the regret of the algorithm. Moreover, the base BCO algorithm proposed in [19] is investigated under the BCO setting where the adversary is limited to inflicting strongly-convex and smooth losses and the player may choose points from a constrained decision set. In this setting with complex feedback, i.e. delayed and anonymous feedback with unknown d, we devise an efficient algorithm that achieves a regret of $\Theta(\sqrt{T})$. This rate is the best possible up to logarithmic factors as implied by a work of [18], cleverly obtaining a lower bound of $\Omega(\sqrt{T})$ for the basic BCO setting. Thus, we conclude that we have reach the optimal regret upper bound for this harder setting. Besides, the regret deterioration is neglectable incurred by the anonymity of delay parameter d.

5.2 Multi-armed Bandit

The algorithm for multi-armed bandit with delayed and anonymous feedback while the delay parameter d is unknown, is almost the same as that for BCO in the same hard setting, except the base algorithm has been changed from an algorithm for BCO to one for MAB. We substitute Exp3 [14] in Algorithm 4–10 and obtain Algorithm 4–13. The regret bound of this algorithm is given in the following theorem.

Theorem 5.5.

If $h : N_+ \to N_+$ is an increasing function such that $h(T) = o(T)$ holds for any T, and then Algorithm

4–13 satisfies

$$R(T) = O\left(\sqrt{Th(2T)N\log N} + h^{-1}(d)\right)$$

Proof. Let k_0 be the first round that $d_{k_0} \geq d$. Then, T_{k_0} satisfies $h(T_{k_0-1}) < d$. Since h is an increasing function, h^{-1} is also increasing. Thus, $T_{k_0-1} < h^{-1}(d)$, which implies $T_{k_0} < 2h^{-1}(d)$.

We now ignore the phases before k_0, as these time steps will have an additional regret of at most $2h^{-1}(d)$. Consider the phases with $d_k \geq d$. Notice that from Corollary 4 in [11], each phase k has regret $O(\sqrt{d_k t_k N\log N})$, where $t_k = T_k/2$ is the number of time steps in this phase.

Specifically, Corollary 15 in [11] is given in the following:

Corollary 5.2.
If Algorithm 4–11 is run with Exp3(η) with $= 4\sqrt{\frac{d\ln N}{(4N+1)T}}$ *as Base MAB, then its regret for N-armed bandits with d-delayed composite anonymous feedback satisfies*

$$R(T) \leq 8\sqrt{d(4N+1)T\ln N} + O(d) = O(\sqrt{dNT\ln N})$$

Then, the total regret in these phases is upper bounded by:

$$\sum_{k=k_0}^{K} O\left(\sqrt{d_k t_k N\log N}\right) \leq O\left(\sqrt{d_K t_K N\log N}\right)$$

where K is the last phase number.

From the description of Algorithm 4–6, we have that $t_K \leq T$ and $d_K \leq h(2T)$. Thus, the total regret of Algorithm 4–6 satisfies $R(T) \leq O(\sqrt{Th(2T)N\log N} + h^{-1}(d))$. □

Remark 5.5.
Algorithm 4–13 is first proposed by a senior in my lab. Compare to the Algorithm 4–11 defined in [11], Algorithm 4–13 modify the approach to decide whether $t \in \mathcal{U}$*, i.e. the method to determine whether to end the current round at round t (lines 14-17 in Algorithm 4–13). This modification is designed to ensure that in each phase, we have*

$$\mathbb{P}(t \in \mathcal{U}) = p, \quad \forall t \in \{1, ..., T\},$$

where p is a constant. This will make sure us to use any inequalities in the proof and analysis of the algorithm regret bound with respect to constant p. Besides, it will not changed the regret bound of the original Algorithm 4–11. Moreover, the expected round size remains a constant, because we maintain a constant $\mathbb{P}(t \in \mathcal{U}) = p$ *in Algorithm 4–13. This leads to a consequence that we cannot increase the round size continuously in this algorithm, instead we have to changed the round size at an exponential rate. That is why we try to split the whole time horizon into different phase. In addition to the idea of phases division, we regard the sub-game in phase k within each phase has fixed time horizon* $\frac{T_k}{2}$ *and delay parameter* d_k*.*

Remark 5.6.
Surprisingly, since the last term of regret of Theorem 5.5 does not depend on T, our regret bound

can be arbitrary close to the regret lower bound given in [11], by restricting the increasing rate of h, without using any prior information on d. However, a small increasing rate of the round size will causes a large regret before we have $d_k \geq d$. In fact, the regret bound can be compressed to $\Theta(T^{1/2+\epsilon})$, which is quit close as the case of d-known in [11], who attained an upper bound of $O(\sqrt{T})$. This tells us the regret loss in anonymous and composite feedback is $O(h^{-1}(d))$.

Chapter 6 Numerical Result

In this chapter, we present our experiment results. We conduct extensive experiment to study the regret bound of the algorithms proposed in Chapter 4. Specifically, we present the experiment on algorithms for bandit convex optimization. Experiments on BCO without delay, BCO with delayed and anonymous feedback while delay parameter d is known, and BCO in same setting while the delay parameter d is unknown are presented for both basic algorithm based on Algorithm 4–4 from [29], and optimal algorithm based on Algorithm 4–7 from [19], respectively. By the way, experiments on multi-armed bandit will be omitted, for the simplicity of the thesis. Ultimately, comparison between different algorithm will be presented. In this chapter, several loss functions are considered, which are defined in the following equation:

$$Loss_1: \quad f_t(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} x_i^2 = \frac{1}{n}\mathbf{x}^\top\mathbf{x} \tag{6–1}$$

$$Loss_2: \quad f_t(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} x_i^4 \tag{6–2}$$

$$Loss_3: \quad f_t(\mathbf{x}) = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}(x_i-1)^2, & \text{if} \quad t \mid 2 = 0 \\ \frac{1}{n}\sum_{i=1}^{n}(x_i+1)^2, & \text{if} \quad t \mid 2 = 1 \end{cases} \tag{6–3}$$

6.1 BCO without Delay

In this section, we consider the performances of basic algorithm (Algorithm 4–4) and optimal algorithm (Algorithm 4–7) on bandit convex optimization with simple feedback (no delay), respectively.

6.1.1 Basic Algorithm

In this subsection, we regard Algorithm 4–4 as basic algorithm and study its performance with two different loss functions $Loss_1$ and $Loss_2$, which are defined in Eq. (6–1) and Eq. (6–2), respectively. We plot the cumulative regret and instantaneous regret (the value of loss function $f_t(\mathbf{x}_t)$ at the query point $\mathbf{x}_t$) at the same time, which are shown in Fig. 6–2 and Fig. 6–1, respectively.

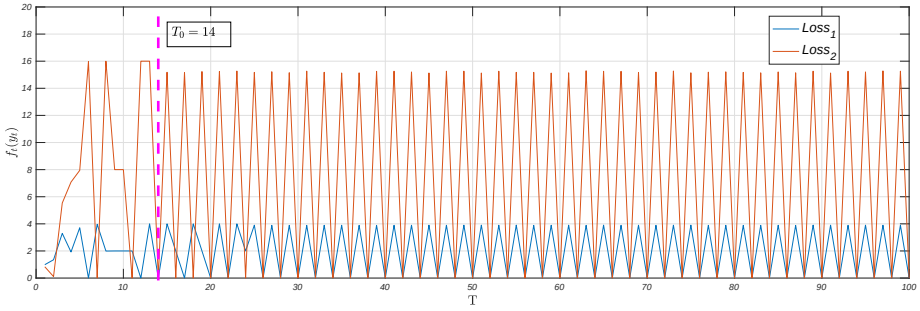


Figure 6–1 Instantaneous regret for Algorithm 4–4 on $Loss_1$ and $Loss_2$.

From Fig. 6–1, we find that the instantaneous regret is oscillating for both $Loss_1$ and $Loss_2$. The oscillations of the two functions have the same frequency but different amplitudes. In particular, the instantaneous regret gets into oscillation after $T_0 = 14$. In particular, $Loss_2$ has larger amplitude than $Loss_1$, which is consistent with their definitions in Eq. (6–1) and Eq. (6–2).

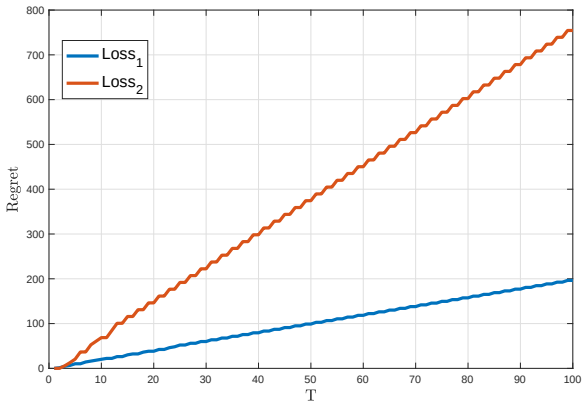


Figure 6–2 Cumulative regret for Algorithm 4–4 on $Loss_1$ and $Loss_2$.

Since the amplitude of instantaneous regret remains unchanged, the cumulative regret grows linearly for both $Loss_1$ and $Loss_2$, which is shown in Fig. 6–2.

6.1.2 Optimal Algorithm

In this subsection, we regard Algorithm 4–7 as optimal algorithm and study its performance with loss function $Loss_1$ with two different value of σ. We plot the cumulative regret and instantaneous regret at the same time, which are shown in Fig. 6–4 and Fig. 6–3, respectively.

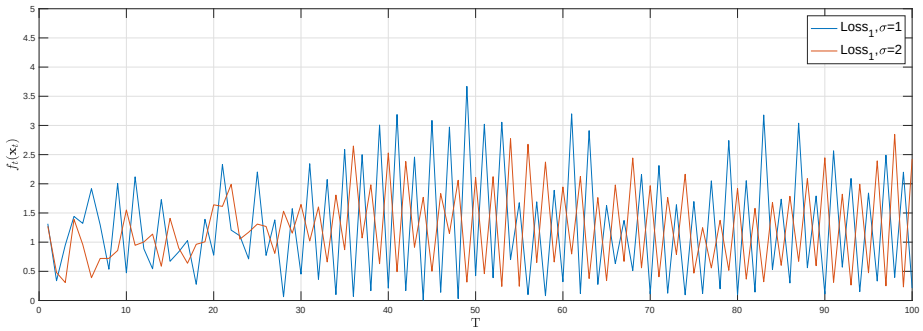


Figure 6–3 Instantaneous regret for Algorithm 4–7 on $Loss_1$ and $Loss_2$.

From Fig. 6–3, we find that the instantaneous regret is oscillating for both $\sigma = 1$ and $\sigma = 2$. The oscillations of the two functions have the same frequency but different amplitudes. In particular, the case when $\sigma = 1$ has larger amplitude than that of $\sigma = 2$.

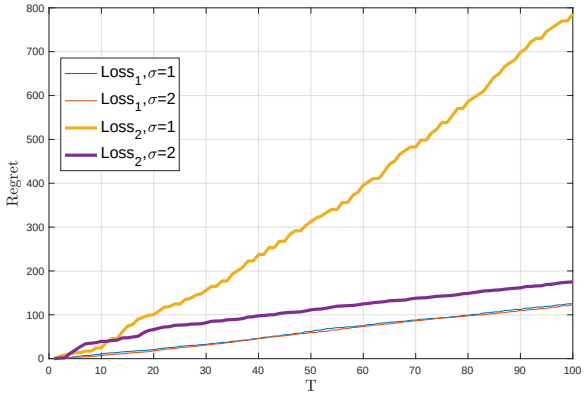


Figure 6–4 Cumulative regret for Algorithm 4–7 on $Loss_1$ and $Loss_2$.

Though the amplitude of instantaneous regret is changing constantly, the cumulative regret still grows more linearly for $Loss_1$ with $\sigma = 1$ and $\sigma = 2$, which is shown in Fig. 6–4, except that the cumulative regret grows faster than linear growth for $Loss_2$ with $\sigma = 1$.

Then, we analyze the Algorithm 4–7's performance with loss function $Loss_3$ with four different value of σ. We plot the cumulative regret and instantaneous regret at the same time, which are shown in Fig. 6–6 and Fig. 6–5, respectively.

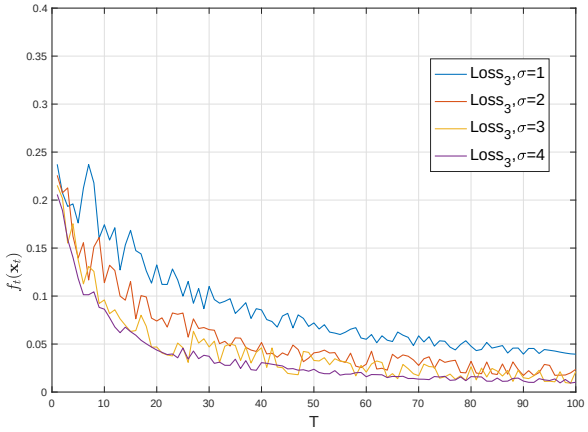


Figure 6–5 Instantaneous regret for Algorithm 4–7 on $Loss_3$.

From Fig. 6–5, we find that the instantaneous regret is decaying for both $\sigma = 1, 2, 3, 4$. In particular, the larger the σ is, the larger decaying rate.

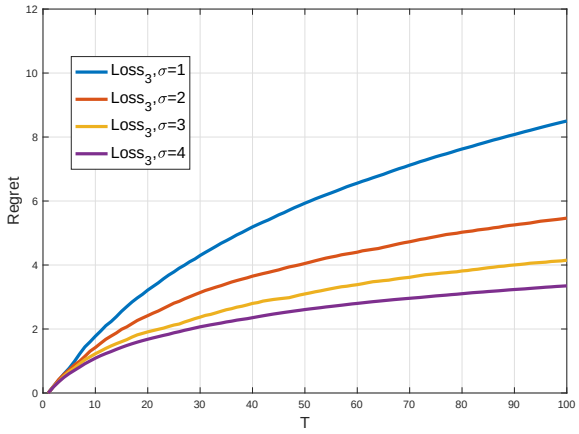


Figure 6–6 Cumulative regret for Algorithm 4–7 on $Loss_3$.

Since the instantaneous regret is decaying, the cumulative regret grows sub-linearly for $Loss_3$ with $\sigma = 1, 2, 3, 4$, which is shown in Fig. 6–6.

6.2 BCO with Known Delay

In this section, we show the performances of Algorithm 4–3 with base Algorithm 4–4 and Algorithm 4–7, respectively. Similar as the above section, we will show the instantaneous and cumulative regret for the algorithm in different parameter settings, with respect to time horizon T. In this section, we only consider a constant loss function $Loss_3$.

6.2.1 Basic Algorithm

We first present the instantaneous regret for Algorithm 4–3 with base Algorithm 4–4 on $Loss_3$ in Fig. 6–7. We find that the instantaneous regret is oscillating around the line $f_t(\mathbf{x}_t) = 0.06$. Interestingly, the instantaneous regret split into two groups, i.e. one for $\delta = 0.5$ and the other one for $\delta = 0.1$. In particular, the group of $\delta = 0.5$ has much larger instantaneous regret than that of group with $\delta = 0.1$.

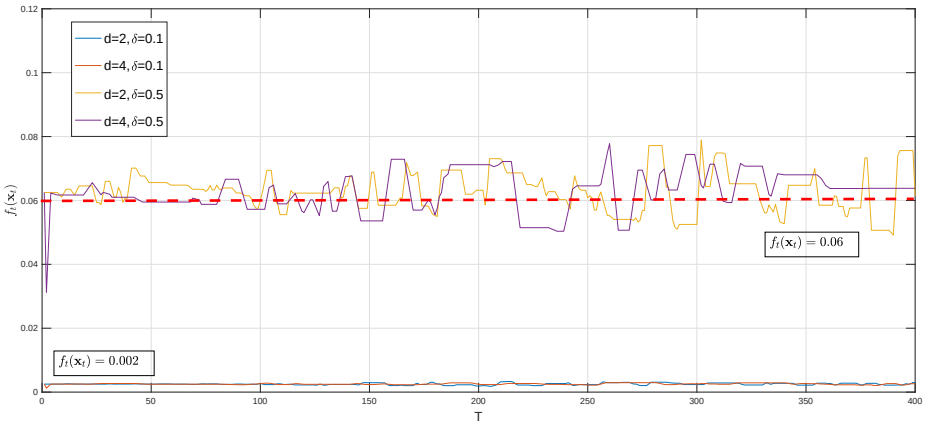


Figure 6–7 Instantaneous regret for Algorithm 4–3 with base Algorithm 4–4 on $Loss_3$.

Then, we present the cumulative regret for Algorithm 4–3 with base Algorithm 4–4 on $Loss_3$ in Fig. 6–8. We find all four regret grows linearly with respect to T, since the instantaneous regret oscillates around a constant. And there are still two groups of cumulative regret, just as the case of instantaneous regret, which is consistent with Fig. 6–7.

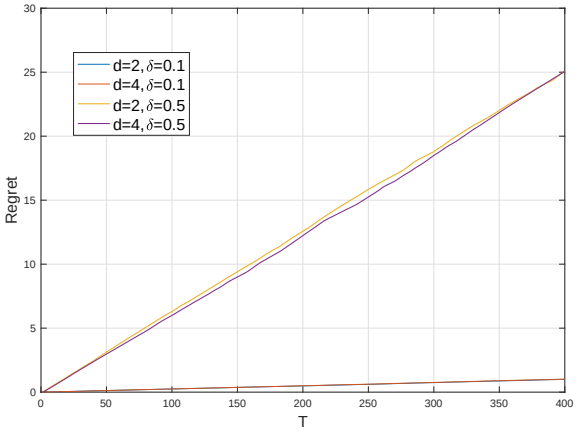


Figure 6–8 Cumulative regret for Algorithm 4–3 with base Algorithm 4–4 on $Loss_3$.

6.2.2 Optimal Algorithm

We first present the instantaneous regret for Algorithm 4–3 with base Algorithm 4–4 on $Loss_3$ in Fig. 6–9. We find that the instantaneous regret is decaying with respect to T. Besides, the larger the σ, the larger the decaying ratio of the instantaneous regret.

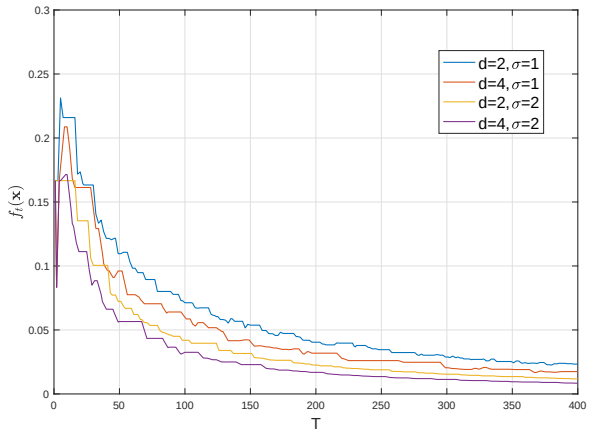


Figure 6–9 Instantaneous regret for Algorithm 4–3 with base Algorithm 4–7 on $Loss_3$.

Then, we present the cumulative regret for Algorithm 4–3 with base Algorithm 4–4 on $Loss_3$ in Fig. 6–10. We find all four regret grows sub-linearly with respect to T. The underlying reason is that the instantaneous regret is decaying. And the larger the σ, the smaller the cumulative regret, which is consistent with Fig. 6–9.

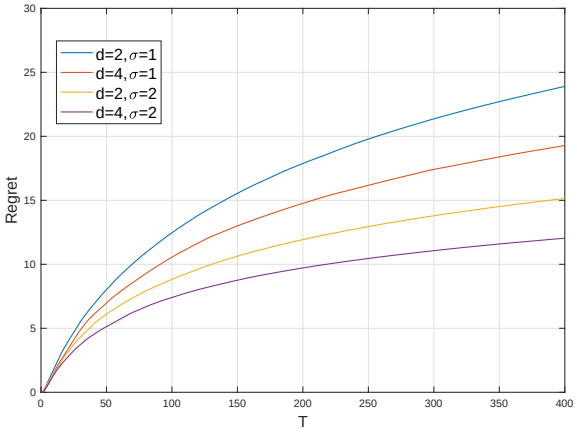


Figure 6–10 Cumulative regret for Algorithm 4–3 with base Algorithm 4–7 on $Loss_3$.

6.3 BCO with Unknown Delay

In this section, we study the algorithms' performance for bandit convex optimization with delayed and anonymous feedback, while the delay parameter d is unknown. Two algorithms, i.e. Algorithm 4–6 and Algorithm 4–9 are considered in this section, which are obtained from substituting Algorithm 4–4 and Algorithm 4–7 in Algorithm 4–1, respectively. Here, we only consider one loss function, i.e. $Loss_3$. And the delay parameter is set to $d = 8$. Apart from the regret behavior of each algorithm individually, the comparison of two algorithms' regret behavior will be given as well.

6.3.1 Basic Algorithm

At first, we present instantaneous regret for Algorithm 4–6 on $Loss_3$ in Fig. 6–11. We find that the instantaneous regret is oscillating around the line $f_t(\mathbf{x}_t) = 0.06$ for all four curves. Roughly speaking, all four curves seem to have similar amplitudes.

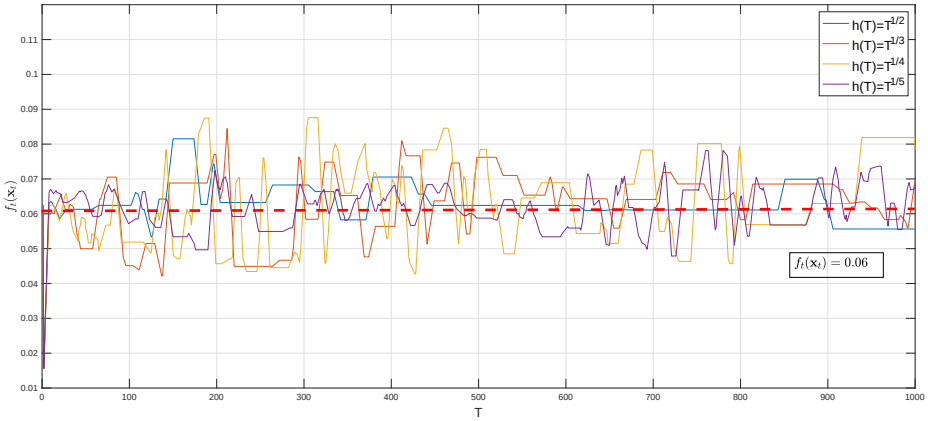


Figure 6–11 Instantaneous regret for Algorithm 4–6 on $Loss_3$.

Then, we present cumulative regret for Algorithm 4–6 on $Loss_3$ in Fig. 6–12. We find all four regret grows linearly with respect to T, since the instantaneous regret oscillates around a constant. And four curves almost have same cumulative regret (except that the case with $h(T) = T^{1/2}$ has notable largest regret), just as the case of instantaneous regret, which is consistent with Fig. 6–11.

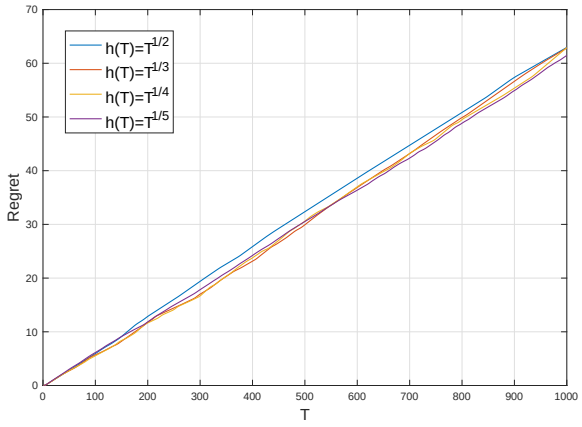


Figure 6–12 Cumulative regret for Algorithm 4–6 on $Loss_3$.

6.3.2 Optimal Algorithm

We first present the instantaneous regret for Algorithm 4–6 on $Loss_3$ in Fig. 6–9. We find that the instantaneous regret is decaying with respect to T. Besides, the larger the power of $h(T)$ with respect to T, the larger the decaying ratio of the instantaneous regret.

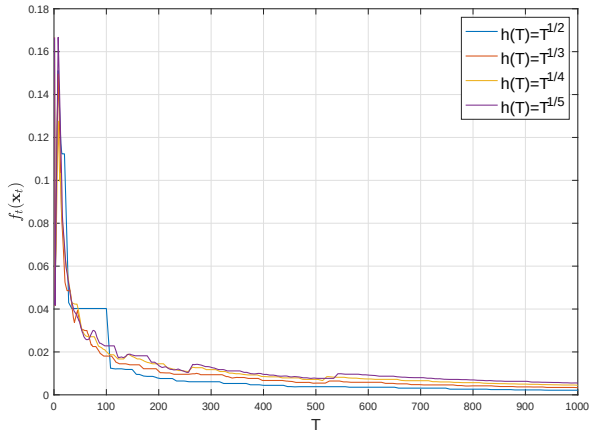


Figure 6–13 Instantaneous regret for Algorithm 4–9 on $Loss_3$.

Then, we present the cumulative regret for Algorithm 4–6 on $Loss_3$ in Fig. 6–14. We find all four regret grows sub-linearly with respect to T. The underlying reason is that the instantaneous regret is decaying. And the larger the power of $h(T)$ with respect to T, the smaller the cumulative regret, which is consistent with Fig. 6–13.

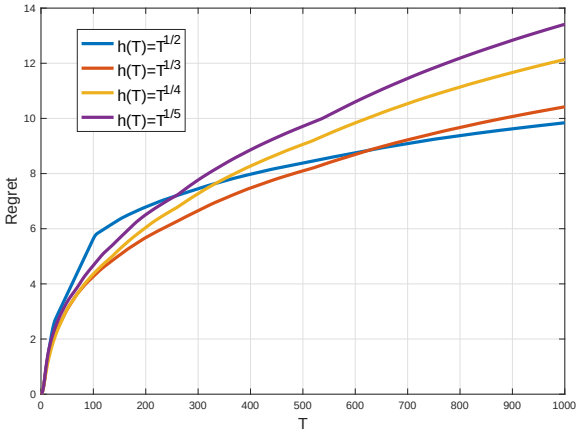


Figure 6–14 Cumulative regret for Algorithm 4–9 on $Loss_3$.

Further, we show the regret behavior under different estimated d_k for Algorithm 4–9 with the same delay d with $T = 10^3$ and $h(T) = T^{1/2}$ in Fig. 6–15. In particular, $d = 2$ and $d = 4$ cases are presented, respectively. The dashed blue and red curve are the immediate result of the experiment for $d = 2$ and $d = 4$ case, respectively. We find that the curve is oscillating. The underlying reason comes from the instabilities of module for finding minimum point a function, which does not guarantee to find the optimal solution. Thus, we smooth the curve by taking running average of the curve with span length for 10 rounds. The smoothed curves are plotted in solid lines (red for $d = 2$ and green for $d = 4$). Two curve are both increasing sub-linearly. Interestingly, we find that the cumulative regret for $d = 2$ is larger than that of $d = 4$.

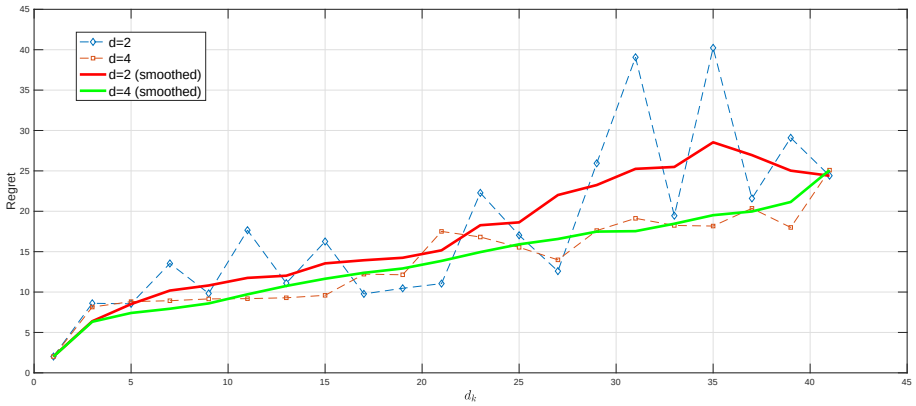


Figure 6–15 Behavior under different estimated d_k for Algorithm 4–9 on $Loss_3$ with the same delay d.

6.3.3 Comparison

We present the algorithms' regret behavior of Algorithm 4–6 and Algorithm 4–9 on the same setting here, where $d = 8$, $T = 1000$, $h(T) = T^{1/2}$ and $f_t(\mathbf{x}) = Loss_3$.

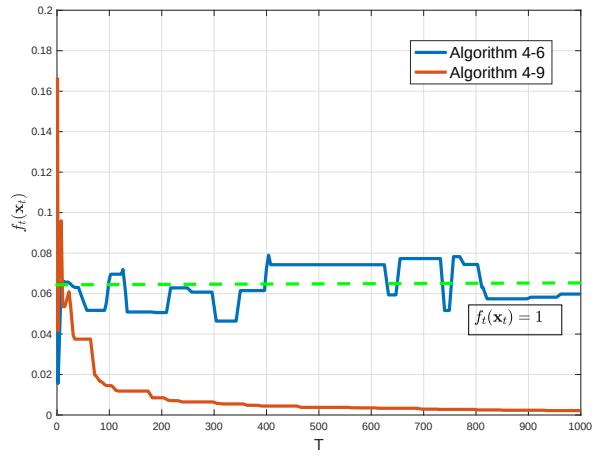


Figure 6–16 Comparison of instantaneous regret of Algorithm 4–6 and Algorithm 4–9 on $Loss_3$.

From Fig. 6–16, we find that the instantaneous regret for Algorithm 4–6 is oscillating around line $f_t(\mathbf{x}_t) = 0.06$, while the that for the other algorithm is decaying.

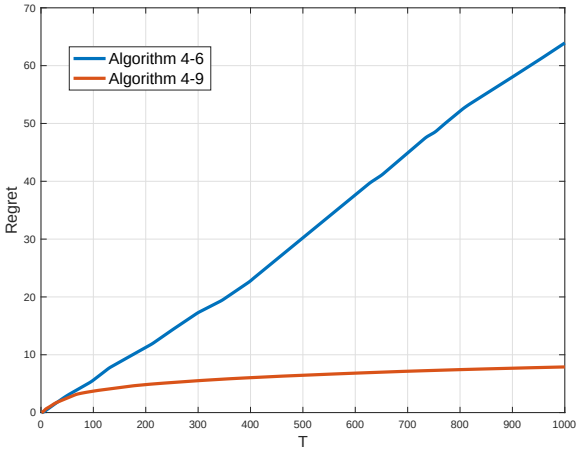


Figure 6–17 Comparison of cumulative regret of Algorithm 4–6 and Algorithm 4–9 on $Loss_3$.

As we expected, the cumulative regret for Algorithm 4–6 grows linearly, while that for Algorithm 4–9 grows sub-linearly. Besides, the cumulative regret for Algorithm 4–9 is smaller than that of Algorithm 4–6, which proves the optimality of Algorithm 4–9 from the side.

Chapter 7 Conclusion

In this thesis, we study the multi-armed bandit problem in versatile settings. We pay extreme attention to bandit convex optimization since bandits convex optimization can be regarded as the generalization of the multi-armed bandit problem. In particular, we study the BCO with complex feedback, i.e. the delayed and anonymous feedback, while the delay parameter d is unknown.

We propose a general algorithm framework which can be applied to the BCO in the mentioned hard setting, for the first of time. The underlying idea of this framework is straightforward, that is first estimate the delay and then follow the fixed-delay algorithm (i.e. Algorithm 4–3 first proposed in [11]). The delay estimation is performed during each phase of the framework. At present, we propose two algorithms (i.e. Algorithm 4–6 and Algorithm 4–9) for BCO with complex feedback. The immediate result is that the first algorithm reach near $\Theta(T^{2/3})$ regret upper bound, while the second one reach near $\Theta(\sqrt{T})$ regret upper bound, except a logarithmic order. To the best of the authors' knowledge, the second algorithm has the best state-of-the-art regret upper bound of the BCO with delayed and anonymous feedback while the delay parameter is unknown. Moreover, it is an optimal algorithm as well, because it has been proved the regret lower bound is $\Omega(\sqrt{T})$.

In addition to the proposed algorithm and regret analysis, we conduct extensive experiments to verify the regret behavior of the proposed algorithms. Throughout the numerical experiment, the algorithm's performance has been verified, which confirms the correctness of our proof. In addition to the verification of the regret upper bound of the proposed algorithm, the comparison of regret behaviors between Algorithm 4–6 and Algorithm 4–9 proves the optimality of the algorithm as well.

In the future, we would like to study the multi-armed bandit problem and bandit optimization in other forms of complex feedback, with a similar goal to minimize the expected regret. Also, we will try to figure out some interesting scenario where our model setting can be fitted such that our algorithm can take advantage of it. We hope our work can shed more lights on the future study of MAB in versatile settings.

Bibliography

[1] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[2] William H Press. Bandit solutions provide unified ethical models for randomized clinical trials and comparative effectiveness research. *Proceedings of the National Academy of Sciences*, 106(52):22387–22392, 2009.

[3] Weiwei Shen, Jun Wang, Yu-Gang Jiang, and Hongyuan Zha. Portfolio choices with orthogonal bandit learning. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[4] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159, 2013.

[5] John White. *Bandit algorithms for website optimization*. " O'Reilly Media, Inc.", 2012.

[6] Pouya Tehrani, Qing Zhao, and Tara Javidi. Opportunistic routing under unknown stochastic models. In *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 145–148. IEEE, 2013.

[7] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[8] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

[9] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[10] Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvari, and Steffen Grunewalder. Bandits with delayed, aggregated anonymous feedback. *arXiv preprint arXiv:1709.06853*, 2017.

[11] Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Nonstochastic bandits with composite anonymous feedback. In *Conference On Learning Theory*, pages 750–773, 2018.

[12] Scott Yang and Mehryar Mohri. Optimistic bandit convex optimization. In *Advances in Neural Information Processing Systems*, pages 2297–2305, 2016.

[13] Rajeev Agrawal. Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078, 1995.

[14] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[15] John C Gittins, Kevin D Glazebrook, Richard Weber, and Richard Weber. *Multi-armed bandit allocation indices*, volume 25. Wiley Online Library, 1989.

[16] Cem Tekin and Mingyan Liu. Online algorithms for the multi-armed bandit problem with markovian rewards. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1675–1682. IEEE, 2010.

[17] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[18] Ohad Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In *Conference on Learning Theory*, pages 3–24, 2013.

[19] Elad Hazan and Kfir Levy. Bandit convex optimization: Towards tight bounds. In *Advances in Neural Information Processing Systems*, pages 784–792, 2014.

[20] Alekh Agarwal, Dean P Foster, Daniel J Hsu, Sham M Kakade, and Alexander Rakhlin. Stochastic convex optimization with bandit feedback. In *Advances in Neural Information Processing Systems*, pages 1035–1043, 2011.

[21] Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pages 28–40. Citeseer, 2010.

[22] Sébastien Bubeck and Ronen Eldan. Multi-scale exploration of convex functions and bandit convex optimization. In *Conference on Learning Theory*, pages 583–589, 2016.

[23] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International Conference on Machine Learning*, pages 1453–1461, 2013.

[24] Claire Vernade, Olivier Cappé, and Vianney Perchet. Stochastic bandit models for delayed conversions. *arXiv preprint arXiv:1706.09186*, 2017.

[25] Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual Conference On Learning Theory*, pages 359–376, 2011.

[26] Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Delay and cooperation in non-stochastic bandits. *The Journal of Machine Learning Research*, 20(1):613–650, 2019.

[27] Ofer Dekel, Jian Ding, Tomer Koren, and Yuval Peres. Online learning with composite loss functions. In *Conference on Learning Theory*, pages 1214–1231, 2014.

[28] Miroslav Bačák and Jonathan M Borwein. On difference convexity of locally lipschitz functions. *Optimization*, 60(8-9):961–978, 2011.

[29] Ankan Saha and Ambuj Tewari. Improved regret guarantees for online smooth convex optimization with bandit feedback. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 636–642, 2011.

[30] Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. 2009.

[31] Elad Hazan. A survey: The convex optimization approach to regret minimization. 2009.

[32] Longbo Huang Siwei Wang. Adaptive algorithms for multi-armed bandit with delayed and anonymous feedback. In *Submitted to Internation Conference on Machine Learning*, 2019.

Acknowledgements

At the very beginning, I would like to express my great gratitude to my advisor Prof. Longbo Huang, and Prof. Xiaofeng Gao, for their professional guidance and creative suggestions during the whole process of the graduation project. It is worth to mention that Prof. Huang gives me the direction of my research work, i.e. the bandit convex optimization with d-unknown feedback. Besides, when I was in the toughest stage of the project, it was Prof. Huang who provided me with the direction of next step. After adopting his opinions, I solved the most difficult problem in the paper, which is the proof of the optimality of the algorithm. At the same time, Prof. Gao also made an indelible contribution to the completion of my thesis. From the beginning of the thesis and to the finalization, Mr. Gao provided the guidance document for the whole process, and also reminded me to submit the materials on time. Experiences from students in her laboratory provided a great reference for me to successfully complete the thesis.

Secondly, I would like to express sincere thanks to my senior Siwei Wang, who is a Ph.D student of Prof. Huang. It is him that work with me on the problem of multi-armed bandit and bandit convex optimization during the whole process of the graduation project. We have had many discussions. We explored a lot of specific algorithms and shared a lot of new ideas. And, whenever I encounter specific academic problems, I will always be the first to ask him for advice. He took the trouble to answer questions for me and took time to comment on my work. It is no exaggeration to say that he can also be called my teacher, a great one.

Then, I would like to thank the many authors for their indelible groundbreaking work in the field of the multi-armed bandit and bandit convex optimization. Because their work provides a lot of research foundation for my graduation thesis. Standing on the shoulders of giants, the scientific work in my graduation thesis has become more meaningful. Especially, there authors : Cesa-Bianchi in [11] for algorithm wrapper for MAB with composite feedback, Saha in [29] for basic algorithm for BCO, Hazan in [19] for optimal algorithm for BCO and great introduction of [17].

Besides, I would like to thank Professor Xia Bin, senior Li Cheng, and all the seniors in the laboratory of Professor Xia. I am very grateful to Prof. Xia for giving me the opportunity to join his lab. In the undergraduate stage, I exercised valuable scientific research skills and gained valuable scientific research experience, in the laboratory of intelligent network transmission. Especially under the guidance of Li Cheng, the master, I know how to do research, how to model and analyze a problem, how to write a thesis. I also harvested fruitful research results in this lab, including two articles and three patents. I once again express my sincere gratitude to Prof. Xia and Li Cheng.

I would express my sincere thanks to the School of Electronic information and Electrical Engineering, and Shanghai Jiao Tong University, for providing me with academic opportunities, abundant resources, and necessary facilities.

I would express my sincere thanks to the Institute for Interdisciplinary Information Sciences, and Tsinghua University, for providing me with the opportunity to finish my graduation project under the guidance of Prof. Huang, who is my advisor for my Ph.D study.

I also appreciate all my instructors in Shanghai Jiao Tong University, especially the faculties in the IEEE Honored Class, the Department of Computer Science Engineering and the Department of Electronic Engineering. I could never finish my study and this thesis without their well-designed courses, assignments, and projects. In particular, I would like to express my gratitude to our department head, Professor Xinbing, Wang.

Last but not least, I would like to express my sincere gratitude to my intimate lover, Zhou. Thanks for her encouragement, support and care during my studies, and I will be with her forever.

Publications

[1] Li, Cheng, Pihe Hu, Yao Yao, Bin Xia, and Zhiyong Chen. "Optimal Multi-User Scheduling for the Unbalanced Full-Duplex Buffer-Aided Relay Systems." IEEE Transactions on Wireless Communications (2019).

[2] Hu, Pihe, Cheng Li, Dingjie Xu, and Bin Xia. "Optimal Multi-User Scheduling of Buffer-Aided Relay Systems." In 2018 IEEE International Conference on Communications (ICC), pp. 1-6. IEEE, 2018.

Projects

[1] 自然科学基金重点项目“同时同频全双工通信理论与关键技术”（项目编号：61531009）

[2] GF 项目“同时同频全双工 ****** 技术”

Patents

[1] 第一发明人，“全双工缓存中继系统多用户调度方法及系统”，专利申请号 CN201810005151.4，法律状态：实审

[2] 第二发明人，“基于功率自适应的全双工中继系统多用户调度方法”，专利申请号 CN201810241964.3，法律状态：实审

[3] 第二发明人，“基于统计概率选择的混合双工中继实现方法”，专利申请号 CN201810092204.0，法律状态：实审