Multiobjective Lipschitz Bandits under Lexicographic Ordering

Bo Xue[1,2], **Ji Cheng**[1,2], **Fei Liu**[1,2], **Yimu Wang**[3], **Qingfu Zhang**[1,2,*]

[1]Department of Computer Science, City University of Hong Kong, Hong Kong, China
[2]The City University of Hong Kong Shenzhen Research Institute, Shenzhen, China
[3]Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada
{boxue4, jcheng27, fliu36}-c@my.cityu.edu.hk, yimu.wang@uwaterloo.ca, qingfu.zhang@cityu.edu.hk

Abstract

This paper studies the multiobjective bandit problem under lexicographic ordering, wherein the learner aims to simultaneously maximize m objectives hierarchically. The only existing algorithm for this problem considers the multi-armed bandit model, and its regret bound is $\widetilde{O}((KT)^{2/3})$ under a metric called priority-based regret. However, this bound is suboptimal, as the lower bound for single objective multi-armed bandits is $\Omega(K \log T)$. Moreover, this bound becomes vacuous when the arm number K is infinite. To address these limitations, we investigate the multiobjective Lipschitz bandit model, which allows for an infinite arm set. Utilizing a newly designed multi-stage decision-making strategy, we develop an improved algorithm that achieves a general regret bound of $\widetilde{O}(T^{(d_z^i+1)/(d_z^i+2)})$ for the i-th objective, where d_z^i is the zooming dimension for the i-th objective, with $i \in \{1, 2, \ldots, m\}$. This bound matches the lower bound of the single objective Lipschitz bandit problem in terms of T, indicating that our algorithm is almost optimal. Numerical experiments confirm the effectiveness of our algorithm.

Introduction

Online learning with bandit feedback is a powerful paradigm for modeling sequential decision-making cases (Robbins 1952), such as clinical trials (Villar, Bowden, and Wason 2015), news recommendation (Li et al. 2010), and website optimization (White 2012). The fundamental model of this paradigm is multi-armed bandits (MAB), where a learner repeatedly selects one arm from K available arms and receives a stochastic payoff drawn from an unknown distribution associated with the chosen arm (Bubeck et al. 2015; Luo et al. 2018; Zhou, Xu, and Blanchet 2019; Xue et al. 2020; Zhu and Mineiro 2022; Qin et al. 2023; Gou, Yi, and Zhang 2023). The goal of learner is to minimize regret, defined as the cumulative difference between the expected payoff of the selected arm and that of the inherently best arm. To achieve this goal, the learner must strike a balance between exploration and exploitation, attempting potentially better arms while concurrently employing the best arm identified so far.

Although MAB is powerful, many real-world applications involve multiple and potentially conflicting objectives, such as the Click-Through Rate (CTR) and the Post-Click Conversion Rate (CVR) in advertising recommendation systems (Ma et al. 2018). This has led to the study of multiobjective multi-armed bandits (MOMAB), in which payoffs are vectors containing multiple elements, and the learner aims to simultaneously minimize the regret for all objectives (Drugan and Nowe 2013). A commonly used criterion for evaluating the performance of MOMAB is Pareto regret, which regards all objectives as equivalent (Van Moffaert et al. 2014; Q. Yahyaa, M. Drugan, and Manderick 2014; Turgay, Oner, and Tekin 2018; Lu et al. 2019a; Xu and Klabjan 2023). However, some scenarios may require varying levels of importance among objectives, such as radiation treatment for cancer patients, where the primary objective is target coverage and the secondary objective is the therapy's proximity to organs at risk (Jee, McShan, and Fraass 2007). Similarly, water resource planning legally mandates the prioritization of objectives such as flood protection, supply shortage for irrigation, and electricity generation (Weber et al. 2002).

To deal with these real-world applications, a natural idea is to utilize lexicographic ordering, as it ranks the objectives according to their importance (Ehrgott 2005; Wray and Zilberstein 2015; Wray, Zilberstein, and Mouaddib 2015; Hüyük and Tekin 2021; Hosseini et al. 2021; Skalse et al. 2022). Let $\mathcal{X}$ represent an arm space, and the expected payoffs for $a, b \in \mathcal{X}$ are $[\mu^1(a), \mu^2(a), \ldots, \mu^m(a)] \in \mathbb{R}^m$ and $[\mu^1(b), \mu^2(b), \ldots, \mu^m(b)] \in \mathbb{R}^m$. The arm a is said to **lexicographically dominate** arm b if and only if $\mu^1(a) > \mu^1(b)$, or there exists an $i^* \in \{2, 3, \ldots, m\}$, such that $\mu^i(a) = \mu^i(b)$ for $1 \leq i \leq i^* - 1$ and $\mu^{i^*}(a) > \mu^{i^*}(b)$. The **lexicographically optimal** arm is the one that is not lexicographically dominated by any other arms (Hüyük and Tekin 2021).

The only existing algorithm for multiobjective bandits under lexicographic ordering is specifically designed for the MOMAB model (Hüyük and Tekin 2021), whose arm set is finite, i.e., $\mathcal{X} = [K]^1$. Let x^* denote the lexicographically optimal arm among $\mathcal{X}$ and x_t be the arm chosen at t-th epoch. Hüyük and Tekin (2021) defined a priority-based regret to evaluate the performance of their algorithm, given by

$$\widetilde{R}^i(T) = \sum_{t=1}^{T} \left(\mu^i(x^*) - \mu^i(x_t)\right) \mathbb{I}(A^i(x_t)), i \in [m]. \quad (1)$$

Here, $\mathbb{I}(\cdot)$ is the indicator function, and $A^i(x_t)$ denotes the

[*] Qingfu Zhang is the corresponding author.

event that the previous $i-1$ expected payoffs of the chosen arm are optimal, i.e., $A^i(x_t) = \{\mu^j(x^*) - \mu^j(x_t) = 0, j \in [i-1]\}$. Hüyük and Tekin (2021) proposed an algorithm with a bound of $\widetilde{O}((KT)^{2/3})$ under this priority-based regret.

There are three limitations of the existing algorithm (Hüyük and Tekin 2021). First, the performance declines as K increases and becomes ineffective when the arm set is infinite. Second, the regret bound is suboptimal with respect to T when the number of objectives reduces to one, as the lower bound for single objective MAB is $\Omega(K \log T)$ (Lai and Robbins 1985). Third, regret (1) is not practicable due to the impact of $\mathbb{I}(\cdot)$. Specifically, for the i-th objective, if there exists an objective $j \in [i-1]$, the expected payoff of the chosen arm x_t is not optimal, i.e., $\mu^j(x^*) \neq \mu^j(x_t)$, the gap $\mu^i(x^*) - \mu^i(x_t)$ does not accumulate to regret (1).

To remove these limitations, we investigate the multiobjective Lipschitz bandit (MOLB) model under lexicographic ordering, where the arm set $\mathcal{X}$ can be infinite. We adopt the general regret to evaluate our algorithms, as expressed by

$$R^i(T) = T\mu^i(x^*) - \sum_{t=1}^{T} \mu^i(x_t), i \in [m]. \tag{2}$$

To the best of our knowledge, this work is the first to explore the MOLB model under lexicographic ordering, and the main contributions can be summarized as follows:

- We develop an algorithm that achieves the regret bound $\widetilde{O}(T^{(d_z^i+1)/(d_z^i+2)})$ for the i-th objective, where d_z^i is the zooming dimension to be introduced in the next section. This result matches the lower bound of single objective Lipschitz bandit problem (Kleinberg, Slivkins, and Upfal 2008), indicating that our algorithm is almost optimal.
- We propose a novel multi-stage decision-making strategy that delicately balances exploration and exploitation, which is crucial for improving the existing suboptimal result (Hüyük and Tekin 2021).
- We extend the metric of lexicographically ordered multiobjective bandits from the priority-based regret (1) to the general regret (2), which more accurately evaluates the performance of the learner.

Preliminaries

In this section, we first present the learning setting of MOLP, and then introduce two necessary concepts, covering dimension and zooming dimension.

Learning Setting

MOLP is a T-round decision-making system indexed by $t \in [T]$. At each epoch t, the learner selects an arm x_t from the metric space $\mathcal{X}$ and receives a stochastic payoff vector $[y_t^1, y_t^2, \dots, y_t^m] \in \mathbb{R}^m$, where y_t^i is the payoff of the i-th objective and m is the number of objectives. The payoffs are conditionally 1-sub-Gaussian, such that

$$\mathrm{E}\left[e^{\alpha(y_t^i - \mu^i(x_t))} | \mathcal{F}_{t-1}\right] \leq e^{\alpha^2/2}, \forall \alpha \in \mathbb{R} \tag{3}$$

where $\mu^i(x_t)$ denotes the i-th expected payoff of arm x_t , $i \in [m]$, and $\mathcal{F}_{t-1} = \{x_1, x_2, \dots, x_t\} \cup \{y_1^1, y_2^1, \dots, y_{t-1}^1\} \cup \dots \cup \{y_1^m, y_2^m, \dots, y_{t-1}^m\}$ is a σ-filtration (Auer 2002; Bubeck, Stoltz, and Yu 2011; Abbasi-yadkori, Pál, and Szepesvári 2011; Shao et al. 2018). Another common assumption on the Lipschitz bandit model is that the expected functions satisfy the Lipschitz property (Turgay, Oner, and Tekin 2018; Wanigasekara and Yu 2019; Podimata and Slivkins 2021), such that

$$|\mu^i(x) - \mu^i(x')| \leq \mathcal{D}(x, x'), \forall x, x' \in \mathcal{X}, i \in [m] \tag{4}$$

where $\mathcal{D}(\cdot,\cdot)$ is the distance function on metric space $\mathcal{X}$. Without loss of generality, we assume the diameter of $\mathcal{X}$ is smaller than 1, i.e., $\mathcal{D}(x, x') \leq 1, \forall x, x' \in \mathcal{X}$.

Furthermore, we propose a parameter λ to depict the difficulty of identifying the lexicographically optimal arm. Precisely, we assume there exists some $\lambda \geq 0$,

$$\mu^i(x) - \mu^i(x^*) \leq \lambda \cdot \max_{j \in [i-1]} \{\mu^j(x^*) - \mu^j(x)\} \tag{5}$$

for any $i \in \{2, 3, \dots, m\}$ and $x \in \mathcal{X}$. An exceptionally large λ implies the existence of an arm $x \in \mathcal{X}$ with a significantly higher payoff than the optimal arm x^* for the i-th objective while maintaining similar payoffs for the preceding $i-1$ objectives, which makes the identification of the lexicographically optimal arm challenging.

Covering Dimension and Zooming Dimension

Let $B(\bar{x}, r)$ denote the ball with center $\bar{x} \in \mathcal{X}$ and radius $r \geq 0$, such that $B(\bar{x}, r) = \{x \in \mathcal{X} | \mathcal{D}(\bar{x}, x) \leq r\}$. The ***r*-covering number** of $\mathcal{X}$ is the minimal number of balls with radius r to cover $\mathcal{X}$, i.e.,

$$N_c(r) = \min\{n \in \mathbb{N} \mid \mathcal{X} \subseteq \cup_{k \in [n]} B(\bar{x}_k, r)\}. \tag{6}$$

Based on the covering number, the **covering dimension** of $\mathcal{X}$ is defined as

$$d_c = \min\{d \geq 0 \mid \exists\, C > 0, N_c(r) \leq Cr^{-d}, \forall\, r > 0\}. \tag{7}$$

We present two specific cases to help with the understanding of covering dimension. One is the unit ball in d-dimensional Euclidean space, whose covering dimension is d and $C = 1$. Another case is any set containing finite elements, i.e., $|\mathcal{X}| = K$, whose covering dimension is 0 and $C = K$.

The covering dimension does not account for the structure of expected payoff functions, thus failing to reflect the complexity of a Lipschitz bandit problem accurately. To illustrate this issue, we provide a simple example. Suppose the arm space is $\mathcal{X} \subset \mathbb{R}^d$ with a Euclidean metric, and the expected functions are $\mu^i(x) = x_1, i \in [m]$ for all $x \in \mathcal{X}$. Here, x_1 denotes the first element of vector x. In this case, the complexity of identifying the optimal arm remains the same, regardless of the covering dimension of $\mathcal{X}$.

To deal with this issue, another concept termed zooming dimension was proposed (Kleinberg, Slivkins, and Upfal 2008). In this paper, we extend this concept to multiobjective setting. First, we define the ***r*-optimal region** for the i-th objective as

$$\mathcal{X}^i(r) = \{x \in \mathcal{X} | \lambda_i r/2 < \mu^i(x^*) - \mu^i(x) \leq \lambda_i r\} \tag{8}$$

[1] For any $n \in \mathbb{N}_+$, $[n]$ denotes the set $\{1, 2, \dots, n\}$.

where $\lambda_i = 1 + \lambda + \ldots + \lambda^{i-1}$ is a fixed constant and $r \geq 0$. Then, similar to the r-covering number, the ***r*-zooming number** can be defined as the minimal number of balls with radius $r/96$ to cover $\mathcal{X}^i(r)$, denoted by $N_z^i(r)$, i.e.,

$$N_z^i(r) = \min\{n \in N \mid \mathcal{X}^i(r) \subseteq \cup_{k\in[n]} B(\bar{x}_k, r/96)\}. \quad (9)$$

Now, we are ready to define the **zooming dimension** for the i-th objective, which is

$$d_z^i = \min\{d \geq 0 \mid \exists\, Z_i > 0, N_z^i(r) \leq Z_i r^{-d}, \forall r > 0\}. \quad (10)$$

Compared with the zooming dimension of single objective Lipschitz bandits (Kleinberg, Slivkins, and Upfal 2008), the only difference is the adoption of the constant λ_i in (8), which is due to technical reasons (see Theoretical Analysis section) and does not constitute an essential different, as r can approach zero arbitrarily closely.

Related Work

In this section, we give a brief review of the research for Lipschitz bandits and multiobjective bandits.

Lipschitz Bandits

Plenties of work on Lipschitz bandits have been conducted in recent years, and most of them employ two basic techniques: static discretization (Agrawal 1995; Kleinberg 2004; Auer, Ortner, and Szepesvári 2007) and adaptive discretization (Kleinberg, Slivkins, and Upfal 2008; Bubeck et al. 2008, 2011; Lu et al. 2019b; Wang et al. 2020; Feng, Huang, and Wang 2022). Static discretization involves dividing the arm space into a uniform mesh and directly applying MAB algorithms to the mesh regions, such as UCB (Auer 2002). The seminal work of Agrawal (1995) investigated a specific case called continuum-armed bandits, wherein the arm set is a compact interval (i.e., $\mathcal{X} \in [0, 1]$). Building upon this research, Kleinberg (2004) proposed a near-optimal algorithm with a bound of $O(T^{2/3})$ and established a matching lower bound. Subsequently, Auer, Ortner, and Szepesvári (2007) improved this result by achieving a regret bound of $O(\sqrt{T})$ under mild assumptions.

Adaptive discretization dynamically discretizes the arm space according to observed payoffs and allocates more trials to promising regions. This technique was first proposed by Kleinberg, Slivkins, and Upfal (2008), who extended the arm set into a general metric space and introduced the zooming algorithm, achieving a regret bound of $\widetilde{O}(T^{(d_z+1)/(d_z+2)})$. Here, d_z represents the zooming dimension of the expected payoff function. Furthermore, Kleinberg, Slivkins, and Upfal (2008) provided a matching lower bound of $\Omega(T^{(d_z+1)/(d_z+2)})$. A subsequent work of Bubeck et al. (2011) relaxed the Lipschitz assumption to locally Lipschitz and proposed a tree-based algorithm that attains a regret bound of $\widetilde{O}(T^{(d_z+1)/(d_z+2)})$. Wang et al. (2020) connected tree-based methods with Gaussian processes and developed a new analytical framework.

Multiobjective Banidts

Drugan and Nowe (2013) initially formalized the MOMAB model and introduced two algorithms enjoying the bounds $O(K \log T)$ under the metrics of scalarized regret and Pareto regret, respectively. Scalarized regret refers to the weighted sum of all objectives' regret, while Pareto regret measures the cumulative Pareto distance between the obtained payoff vectors and the Pareto optimal payoff vector. Turgay, Oner, and Tekin (2018) studied the multiobjective contextual bandit model and proposed a zooming-based algorithm that achieves a Pareto regret bound of $\widetilde{O}(T^{(d_p+1)/(d_p+2)})$, where d_p represents the Pareto zooming dimension. Subsequently, Lu et al. (2019a) investigated a parameterized bandit model called multiobjective generalized linear bandits. To our knowledge, the study by Hüyük and Tekin (2021) is the only one that focuses on bandits with lexicographic ordering. They proposed the algorithm PF-LEX, which enjoys a regret bound of $\widetilde{O}((KT)^{2/3})$. However, this bound is suboptimal as existing single objective MAB algorithms attain a regret bound of $O(K \log T)$ (Lai and Robbins 1985).

To illustrate the intuition for improving PF-LEX, we briefly introduce the decision-making strategy of PF-LEX. The fundamental framework to settle the bandit problem is the upper confidence bound (UCB), which first constructs confidence intervals for all arms, and then selects the arm with the highest upper confidence bound (Lattimore and Szepesvári 2020). When adapting UCB to MOMAB, the main modification is considering all objectives in the arm selection. Let $c_t(a)$ denote the confidence term of arm $a \in [m]$ at round t. PF-LEX considers two cases for arm selection. If some arm $a_t \in [K]$ satisfies $c_t(a_t) > \epsilon$ for a given criterion $\epsilon > 0$, PF-LEX chooses this arm a_t. Otherwise, if $c_t(a) < \epsilon$ for all arms $a \in [K]$, PF-LEX filters promising arms based on the confidence intervals sequentially, ranging from the first to the m-th objective, and ultimately selects an arm in the m-th filtered set. PF-LEX consumes numerous trials in the first case, which is a pure exploration case and leads to suboptimal regret. Therefore, we consider avoiding the pure exploration case by dividing the decision-making process into multiple stages.

Algorithms

In this section, we first introduce a simple algorithm based on static discretization, which is easy to understand but needs an oracle. Then, we use adaptive discretization to remove the oracle, creating an almost optimal algorithm.

Warm-up: SDLO

As a warm-up, we propose a simple algorithm called Static Discretization under Lexicographic Ordering (SDLO), which first discretizes the arm set $\mathcal{X}$ statically, and then utilizes a multi-stage decision-making strategy to select arms.

According to the Lipschitz property of expected payoff functions, knowing the expected payoff of $\bar{x} \in \mathcal{X}$ enables us to estimate the expected payoff of any arm $x \in B(\bar{x}, r)$, i.e., $|\mu^i(x) - \mu^i(\bar{x})| \leq r, i \in [m]$. Consequently, a natural strategy for addressing the Lipschitz bandit problem is to discretize the arm space $\mathcal{X}$ into a collection of small balls and identify the best one. Given the radius r, using fewer balls to cover the arm space simplifies the task of identifying the optimal ball. Thus, covering $\mathcal{X}$ with $N_c(r)$ balls is

Algorithm 1: Static Discretization under Lexicographic Ordering (SDLO)

```
Input: confidence parameter $\delta \in (0,1)$, query radius $r \geq 0$
Query the oracle with $r$ to obtain the static arm set $\mathcal{A} = \{\bar{x}_1, \ldots, \bar{x}_{N_c(r)}\}$ satisfying $\mathcal{X} \subseteq \cup_{k \in [N_c(r)]} B(\bar{x}_k, r)$
Initialize $\hat{\mu}^i(x) = 0, i \in [m]$ for $x \in \mathcal{A}$
Initialize $r(x) = +\infty$ and $n(x) = 0$ for $x \in \mathcal{A}$
for $t = 1, 2, \ldots, T$ do
    Invoke the Algorithm 2 to select the arm $x_t$ = MSDM-SD $\left(\{\hat{\mu}^i(x), i \in [m]\}_{x \in \mathcal{A}}, \{r(x)\}_{x \in \mathcal{A}}, r\right)$
    Play arm $x_t$ and receive the payoff $[y_t^1, y_t^2, \ldots, y_t^m]$
    Update $\hat{\mu}^i(x_t), i \in [m]$ and $n(x_t)$ according to (14)
    Compute $r(x_t)$ according to (15)
end for
```

Algorithm 2: Multi-stage Decision-Making under Static Discretization (MSDM-SD)

```
Input: estimated payoffs $\{\hat{\mu}^i(x), i \in [m]\}_{x \in \mathcal{A}}$, confidence interval width $\{r(x)\}_{x \in \mathcal{A}}$, query radius $r \geq 0$
Initialize $s = 1$ and $\mathcal{A}_1 = \mathcal{A}$
repeat
    if $r(x_t) > 2^{-s}$ for some $x_t \in \mathcal{A}_s$ then
        Choose this arm $x_t$
    else
        Initialize the arm set $\mathcal{A}_s^0 = \mathcal{A}_s$
        for $i = 1, 2, \ldots, m$ do
            $\hat{x}_t^i = \text{argmax}_{x \in \mathcal{A}_s^{i-1}} \hat{\mu}^i(x) + r(x)$
            $\mathcal{A}_s^i = \{x \in \mathcal{A}_s^{i-1} | \hat{\mu}^i(x) + r(x) \geq \hat{\mu}^i(\hat{x}_t^i) + r(\hat{x}_t^i) - (1 + 2\lambda + \ldots + 2\lambda^{i-1}) \cdot (r + 2 \cdot 2^{-s})\}$
        end for
        Update $\mathcal{A}_{s+1} = \mathcal{A}_s^m$ and $s = s + 1$
    end if
until an arm $x_t$ is chosen
Return the chosen arm $x_t$
```

the best choice, as $N_c(r)$ is the minimum number of balls with radius r needed to cover $\mathcal{X}$. However, constructing this minimal coverage is challenging due to the potentially intricate structure of $\mathcal{X}$. Hence, we assume there exists an oracle that takes radius r as input and outputs the minimal arm set $\mathcal{A} = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{N_c(r)}\}$ satisfying

$$\mathcal{X} \subseteq \bigcup_{k \in [N_c(r)]} B(\bar{x}_k, r). \tag{11}$$

Note that for any $x \in \mathcal{X}$, there always exists an arm $\bar{x}_k \in \mathcal{A}$ satisfying $\mathcal{D}(x, \bar{x}_k) \leq r$, which reduces our MOLP problem to a MOMAB problem with $N_c(r)$ arms.

Similar to existing MAB algorithms (Auer 2002; Yu et al. 2018), SDLO initializes the mean payoffs $\hat{\mu}^i(x)$, $i \in [m]$ and the counter $n(x)$ to zero for all $x \in \mathcal{A}$, where $n(x)$ counts the times arm x is played. Meanwhile, the confidence term $r(x)$ is initialized to infinity. These terms will be updated with new trial data as learning goes, whose details are given in equations (14) and (15). Equipped with the mean payoffs and confidence terms, SDLO is ready to make a decision. At each epoch t, SDLO utilizes a novel decision-making method to select an arm x_t from $\mathcal{A}$, whose details are outlined in Algorithm 2, referred to as Multi-stage Decision-Making under Static Discretization (MSDM-SD).

Starting with the initialized arm set $\mathcal{A}_1 = \mathcal{A}$ and stage index $s = 1$, MSDM-SD enters a loop that continues until an arm is chosen. In each stage s, MSDM-SD first checks if there exists an arm $x_t \in \mathcal{A}_s$ whose confidence term $r(x_t)$ is greater than 2^{-s}. If such an arm exists, MSDM-SD chooses this arm x_t. If no arm in $\mathcal{A}_s$ meets this criterion, MSDM-SD proceeds to an inner loop containing m iterations, whhich sequentially filters promising arms from the first objective to the m-th objective. The initialized arm set for the inner loop is $\mathcal{A}_s^0 = \mathcal{A}_s$. For the i-th objective, MSDM-SD first selects the arm $\hat{x}_t^i$ who is highest in mean payoff plus confidence term from the previously filtered arm set $\mathcal{A}_s^{i-1}$, i.e.,

$$\hat{x}_t^i = \underset{x \in \mathcal{A}_s^{i-1}}{\text{argmax}}\, \hat{\mu}^i(x) + r(x). \tag{12}$$

Then MSDM-SD updates the arm set $\mathcal{A}_s^{i-1}$ to $\mathcal{A}_s^i$ by keeping the promising arms, such that

$$\mathcal{A}_s^i = \big\{x \in \mathcal{A}_s^{i-1} | \hat{\mu}^i(x) + r(x) \geq \hat{\mu}^i(\hat{x}_t^i) + r(\hat{x}_t^i) \\ -(1 + 2\lambda + \ldots + 2\lambda^{i-1}) \cdot (r + 2 \cdot 2^{-s})\big\}. \tag{13}$$

After filtering on the last objective, MSDM-SD sets the arm set $\mathcal{A}_{s+1} = \mathcal{A}_s^m$ and proceeds to the next stage $s = s + 1$. According to equation (15), $r(x) > 1/\sqrt{T}$ for all $x \in \mathcal{A}$, MSDM-SD will return an arm x_t before $s = \log_2(T)$.

Once SDLO plays the arm x_t returned by MSDM-SD and receives payoff vector $[y_t^1, y_t^2, \ldots, y_t^m]$, it updates the mean payoffs $\hat{\mu}^i(x), i \in [m]$ and counter $n(x_t)$ as follows:

$$\hat{\mu}^i(x_t) = \frac{n(x_t)\hat{\mu}^i(x_t) + y_t^i}{n(x_t) + 1}, n(x_t) = n(x_t) + 1. \tag{14}$$

Meanwhile, SDLO updates the confidence term of the chosen arm x_t as

$$r(x_t) = \sqrt{2\alpha(x_t)/n(x_t)}. \tag{15}$$

Here, $\alpha(x_t) = 1 + 2\ln(mN_c(r)\sqrt{1 + n(x_t)}/\delta)$ and δ is an input confidence parameter. The following theorem provides a theoretical guarantee for the SDLO algorithm.

Theorem 1 *Suppose that* (3), (4) *and* (5) *hold. If SDLO is run with* $\delta \in (0,1)$ *and* $r \geq 0$, *then with probability at least* $1 - \delta$, *the regret of SDLO can be bounded as*

$$R^i(T) \leq 2\lambda_i \left(rT + 8\sqrt{\alpha_T N_c(r) T}\right), i \in [m]$$

where $\lambda_i = 1 + \lambda + \ldots + \lambda^{i-1}$ *and* $\alpha_T = 1 + 2\ln(mN_c(r)\sqrt{1+T}/\delta)$.

Remark: Theorem 1 states that SDLO achieves a regret bound of $\widetilde{O}(rT + \sqrt{N_c(r)T})$. If the arm set is finite, i.e., $|\mathcal{X}| = K$, the query radius r can be 0 and $N_c(r) = K$. Thus, Theorem 1 provides a regret bound $\widetilde{O}(\sqrt{KT})$ for MOMAB, which not only improves the existing results of Hüyük and

Algorithm 3: Adaptive Discretization under Lexicographic Ordering (ADLO)

```
Input: confidence parameter $\delta \in (0, 1)$
Initialize $\widetilde{\mathcal{A}} = \emptyset$
for $t = 1, 2, \ldots, T$ do
    if $\mathcal{X} \not\subset \cup_{x \in \widetilde{\mathcal{A}}} B(x, r(x))$ then
        Pick an arm $x$ randomly from the uncovered region $\mathcal{X} - \cup_{x \in \widetilde{\mathcal{A}}} B(x, r(x))$
        $\widetilde{\mathcal{A}} = \widetilde{\mathcal{A}} \cup \{x\}$
        Initialize $\hat{\mu}^i(x) = 0, i \in [m]$ and $n(x_t) = 0$
        Play $x_t = x$ and receive the reward $[y_t^1, \ldots, y_t^m]$
    else
        Invoke the Algorithm 4 to select the arm $x_t =$ MSDM-AD $\left(\{\hat{\mu}^i(x), i \in [m]\}_{x \in \widetilde{\mathcal{A}}}, \{r(x)\}_{x \in \widetilde{\mathcal{A}}}\right)$
        Play $x_t$ and receive the reward $[y_t^1, \ldots, y_t^m]$
    end if
    Update $\hat{\mu}^i(x_t), i \in [m]$ and $n(x_t)$ according to (14)
    Compute $r(x_t)$ according to (18)
end for
```

Tekin (2021) by order of $\widetilde{O}((KT)^{1/6})$, but also extends the priority-based regret (1) to general regret (2).

Recalling the definition of covering dimension d_c in (7), $N_c(r) \leq Cr^{-d_c}$ for some constant $C > 0$. Taking this inequality into Theorem 1 and minimizing the regret with respect to r results in a tight bound, as presented below.

Corollary 1 *Suppose that* (3), (4) *and* (5) *hold. If SDLO is run with* $\delta \in (0, 1)$ *and* $r = T^{-\frac{1}{2+d_c}}$, *then with probability at least* $1 - \delta$, *the regret of SDLO can be bounded as*

$$R^i(T) \leq \widetilde{O}\left(\left(1 + \lambda + \ldots + \lambda^{i-1}\right) T^{\frac{1+d_c}{2+d_c}}\right), i \in [m].$$

Improved Algorithm: ADLO

Although SDLO is easy to understand, it presents two limitations. Firstly, it requires a complicated oracle to discretize the arm space $\mathcal{X}$. Secondly, SDLO fails to match the lower bound of the single objective Lipschitz bandit problem (Kleinberg, Slivkins, and Upfal 2008), which means SDLO can be further improved. Therefore, we adopt the adaptive discretization method proposed by Kleinberg, Slivkins, and Upfal (2008) to improve it. Our second algorithm based on adaptive discretization is called Adaptive Discretization under Lexicographic Ordering (ADLO), and the detailed procedure can be found in Algorithm 3.

ADLO maintains an adaptive arm set $\widetilde{\mathcal{A}}$ to construct a collection of balls that cover the arm space $\mathcal{X}$, and the radius of these balls is the confidence term $r(x)$, which is dynamically adjusted as the learning goes. To begin with, ADLO initializes the adaptive arms set $\widetilde{\mathcal{A}}$ with the empty set $\emptyset$. In each round t, if the arm space $\mathcal{X}$ is not covered by the set of balls constructed by $\widetilde{\mathcal{A}}$, i.e., $\mathcal{X} \not\subset \cup_{x \in \widetilde{\mathcal{A}}} B(x, r(x))$, ADLO selects an arm x randomly from the uncovered region, adds it to the arm set $\widetilde{\mathcal{A}}$, and plays this arm. The mean payoffs $\hat{\mu}^i(x), i \in [m]$ and the counter $n(x)$ of the new arm x are initialized to zero. If the arm space is covered, ADLO employs a multi-stage decision-making method called Multi-Stage Decision-Making under Adaptive Discretization (MSDM-AD) to select the most promising arm from $\widetilde{\mathcal{A}}$.

Algorithm 4: Multi-stage Decision-Making under Adaptive Discretization (MSDM-AD)

```
Input: estimated payoffs $\{\hat{\mu}^i(x), i \in [m]\}_{x \in \widetilde{\mathcal{A}}}$, confidence interval width $\{r(x)\}_{x \in \widetilde{\mathcal{A}}}$
Initialize $s = 1$ and $\widetilde{\mathcal{A}}_1 = \widetilde{\mathcal{A}}$
repeat
    if $r(x_t) > 2^{-s}$ for some $x_t \in \widetilde{\mathcal{A}}_s$ then
        Choose this arm $x_t$
    else
        Initialize the arm set $\widetilde{\mathcal{A}}_s^0 = \widetilde{\mathcal{A}}_s$
        for $i = 1, 2, \ldots, m$ do
            $\hat{x}_t^i = \mathrm{argmax}_{x \in \widetilde{\mathcal{A}}_s^{i-1}} \hat{\mu}^i(x) + 2r(x)$
            $\widetilde{\mathcal{A}}_s^i = \{x \in \widetilde{\mathcal{A}}_s^{i-1} | \hat{\mu}^i(x) + 2r(x) \geq \hat{\mu}^i(\hat{x}_t^i) + 2r(\hat{x}_t^i) - (3 + 6\lambda + \ldots + 6\lambda^{i-1}) \cdot 2^{-s}\}$
        end for
        Update $\widetilde{\mathcal{A}}_{s+1} = \widetilde{\mathcal{A}}_s^m$ and $s = s + 1$
    end if
until an arm $x_t$ is chosen
Return the chosen arm $x_t$
```

As shown in Algorithm 4, MSDM-AD takes a similar framework to MSDM-SD, which utilizes an outer loop to restrict the confidence term $r(x)$ of the arms to be chosen, and an inner loop to filter promising arms from the first objective to the m-th objective. Unlike MSDM-SD, MSDM-AD does not take the query radius r as input, and the candidate arm set $\widetilde{\mathcal{A}}$ changes as ADLO goes, resulting in a different filtering mechanism within the inner loop of MSDM-AD. Precisely, for the i-th objective, MSDM-AD first selects an arm $\hat{x}_t^i$ that maximizes the mean payoff plus twice the confidence term, i.e.,

$$\hat{x}_t^i = \underset{x \in \widetilde{\mathcal{A}}_s^{i-1}}{\mathrm{argmax}}\, \hat{\mu}^i(x) + 2r(x) \tag{16}$$

where $\widetilde{\mathcal{A}}_s^{i-1}$ is the set filtered on the previous $i - 1$ objectives and $\widetilde{\mathcal{A}}_s^0 = \widetilde{\mathcal{A}}_s$. Then, MSDM-AD eliminates arms from $\widetilde{\mathcal{A}}_s^{i-1}$ who are less promising on the i-th objective as follows,

$$\begin{aligned} \widetilde{\mathcal{A}}_s^i = \Big\{ & x \in \widetilde{\mathcal{A}}_s^{i-1} | \hat{\mu}^i(x) + 2r(x) \geq \hat{\mu}^i(\hat{x}_t^i) \\ & + 2r(\hat{x}_t^i) - (3 + 6\lambda + \ldots + 6\lambda^{i-1}) \cdot 2^{-s} \Big\}. \end{aligned} \tag{17}$$

After the inner loop ends, MSDM-AD obtains a set $\widetilde{\mathcal{A}}_s^m$ containing arms that are promising for all m objectives. MSDM-AD then passes $\widetilde{\mathcal{A}}_s^m$ to the next stage $s + 1$ as $\widetilde{\mathcal{A}}_{s+1} = \widetilde{\mathcal{A}}_s^m$ for a more refined filtration. Similar to MSDM-SD, MSDM-AD chooses an arm x_t before the stage s increases to $\log(T)$.

Upon playing the arm x_t and receiving the corresponding payoff vector, ADLO proceeds to update the mean payoffs $\hat{\mu}^i(x), i \in [m]$ according to the equation (14). Note that the update of the confidence term in SDLO relies on the arm number $N_c(r)$, and ADLO picks at most T arms, thus ADLO updates the confidence term as follows,

$$r(x_t) = \sqrt{2\tilde{\alpha}(x_t)/n(x_t)} \tag{18}$$

where $\tilde{\alpha}(x_t) = 1 + 2\ln(mT\sqrt{1+n(x_t)}/\delta)$.

There are two main differences between SDLO and ADLO. The first one is the construction of the candidate arm set. SDLO constructs the arm set statically at the beginning of the algorithm by querying an oracle, while ADLO grows arm set to cover previously uncovered regions. The second difference is the filtering mechanism in the decision-making stage. MSDM-SD filters arms by the operation (13), which relies on the parameter r. In contrast, MSDM-AD employs a mechanism for the adaptive arm set, as demonstrated by equation (17), which removes the dependence on r. The following theorem provides a theoretical guarantee for ADLO.

Theorem 2 *Suppose that* (3), (4) *and* (5) *hold. If ADLO is run with* $\delta \in (0,1)$*, then with probability at least* $1-\delta$*, the regret of ADLO can be bounded as*

$$R^i(T) \leq 96\lambda_i \left(\tilde{\alpha}_T Z_i\right)^{\frac{1}{d_z^i+2}} T^{\frac{d_z^i+1}{d_z^i+2}}, i \in [m]$$

where $\lambda_i = 1 + \lambda + \ldots + \lambda^{i-1}$ *and* $\tilde{\alpha}_T = 1 + 2\ln(mT\sqrt{1+T}/\delta)$.

Remark: Theorem 2 states that ADLO attains a regret bound $\widetilde{O}(\lambda_i T^{(1+d_z^i)/(2+d_z^i)})$, which matches the lower bound of the Lipschitz bandit problem with respect to T (Kleinberg, Slivkins, and Upfal 2008). When applied to the single objective problem, ADLO removes the dependence on λ and achieves the regret bound $\widetilde{O}(T^{(1+d_z^1)/(2+d_z^1)})$, which is the same as the optimal single objective Lipschitz bandit algorithm (Kleinberg, Slivkins, and Upfal 2008).

Theoretical Analysis

In this section, we provide a proof sketch for Theorem 2. The omitted details can be found in the supplementary material due to the page limit. For clarity, we use the notation $\hat{\mu}_t^i(x)$, $n_t(x)$, and $r_t(x)$ to represent the values of $\hat{\mu}^i(x)$, $n(x)$, and $r(x)$ at the end of the t-th epoch, respectively. Furthermore, $\widetilde{\mathcal{A}}_t$ denotes the adaptive arm set $\widetilde{\mathcal{A}}$ at the end of the t-th epoch, and $\widetilde{\mathcal{A}}_{t,s}$ represents the arm set $\widetilde{\mathcal{A}}_s$ in MSDM-AD.

Proof of Theorem 2

First, we present Lemma 1 to show that mean payoff $\hat{\mu}_t^i(x)$ and confidence term $r_t(x)$ construct a reliable confidence interval for the expected payoff $\mu^i(x)$.

Lemma 1 *With probability at least* $1-\delta$*, for any* $x \in \widetilde{\mathcal{A}}_t$*,*

$$\left|\hat{\mu}_t^i(x) - \mu^i(x)\right| \leq r_t(x), i \in [m], t \in [T].$$

Next, we demonstrate an essential property of the multi-stage decision-making strategy by the following lemma.

Lemma 2 *With probability at least* $1-\delta$*, for any* $x \in \widetilde{\mathcal{A}}_{t,s}$*,*

$$\mu^i(x^*) - \mu^i(x) \leq 6\lambda_i \cdot 2^{-s+1}, i \in [m], t \in [T]$$

where x^* *is the optimal arm and* $\lambda_i = 1 + \lambda + \ldots + \lambda^{i-1}$.

Remark: Lemma 2 establishes a bound on the instantaneous regret for any arm $x \in \widetilde{\mathcal{A}}_{t,s}$, indicating an exponential decrease as the stage advances. This property is crucial for bounding the cumulative regret, which we will further illustrate in the proof of Lemma 4.

Let $\Delta^i(x) = \mu^i(x^*) - \mu^i(x)$, $i \in [m]$. To proceed with the analysis, we partition the adaptive arm set $\widetilde{\mathcal{A}}_+^i = \{x \in \widetilde{\mathcal{A}}_T \mid \Delta^i(x) > 0\}$ into a set of disjoint subsets. Specifically, we define

$$\widetilde{\mathcal{A}}_j^i = \{x \in \widetilde{\mathcal{A}}_+^i | \lambda_i 2^{-j-1} < \Delta^i(x) \leq \lambda_i 2^{-j}\}, \tag{19}$$

thus $\widetilde{\mathcal{A}}_+^i = \cup_{j\in\mathbb{N}} \widetilde{\mathcal{A}}_j^i$. Recall the definitions of r-optimal region in (8) and zooming dimension d_z^i in (10), we can easily bound the number of arms in $\widetilde{\mathcal{A}}_j^i$ by the following lemma.

Lemma 3 *With probability at least* $1-\delta$*, for any* $j \in \mathbb{N}$*,*

$$|\widetilde{\mathcal{A}}_j^i| \leq Z_i 2^{j \cdot d_z^i}, i \in [m].$$

Then, we give Lemma 4 to analyze the cumulative regret of any arm in $\widetilde{\mathcal{A}}_j^i$. A detailed proof of Lemma 4 is provided since it illustrates the capacity to divide the decision-making process into multiple stages.

Lemma 4 *With probability at least* $1-\delta$*, for all* $j \in \mathbb{N}$*, the regret for any* $x \in \widetilde{\mathcal{A}}_j^i$ *can be bounded as*

$$n_T(x)\Delta^i(x) \leq 1152\lambda_i\tilde{\alpha}_T \cdot 2^j, i \in [m]$$

where $\tilde{\alpha}_T = 1 + 2\ln(mT\sqrt{1+T}/\delta)$.

Proof. For any $x \in \widetilde{\mathcal{A}}_j^i$, if $n_T(x) = 1$, Lemma 4 holds trivially. Now, we assume $n_T(x) \geq 2$. Recalling Step 3 of MSDM-AD, if the last time x is chosen occurs at the $s_T(x)$-th stage among the total T rounds, we get

$$n_T(x) - 1 \leq 2^{s_T(x)}\sqrt{2\tilde{\alpha}_T(n_T(x)-1)}$$

since x is played $n_T(x)-1$ times before this round. Then, due to the fact that $1 \leq (2-\sqrt{2})2^{s_T(x)}\sqrt{\tilde{\alpha}_T n_T(x)}$, we have

$$n_T(x)\Delta^i(x) \leq 2^{s_T(x)+1}\sqrt{\tilde{\alpha}_T n_T(x)}\Delta^i(x). \tag{20}$$

Taking Lemma 2 into the right-hand side of (20) yields

$$n_T(x)\Delta^i(x) \leq 24\lambda_i\sqrt{\tilde{\alpha}_T n_T(x)}. \tag{21}$$

This step reduces the linear term $n_T(x)\Delta^i(x)$ to a sublinear term $\widetilde{O}(\sqrt{n_T(x)})$, which serves as the crucial function for dividing the decision-making process into multiple stages. Squaring both sides of (21) gives

$$n_T(x)\Delta^i(x) \leq 576\lambda_i^2\tilde{\alpha}_T/\Delta^i(x).$$

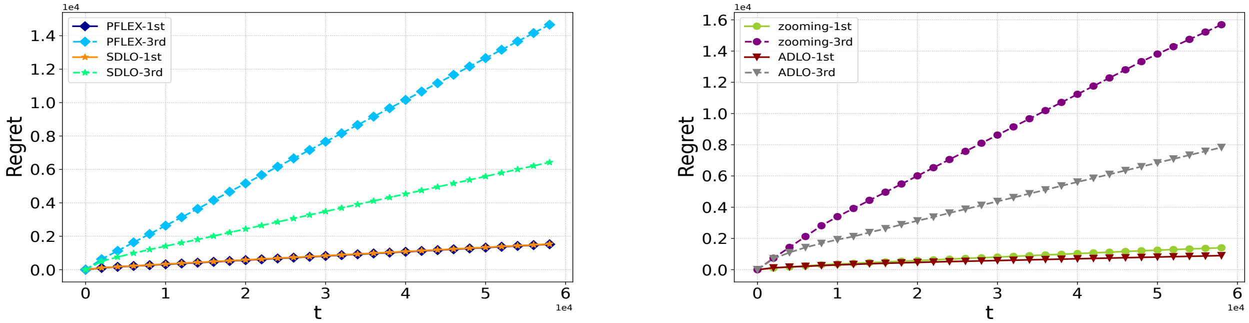
The definition of $\widetilde{\mathcal{A}}_j^i$ implies that $1/\Delta^i(x) < 2^{j+1}/\lambda_i$ for any $x \in \widetilde{\mathcal{A}}_j^i$. Taking it into the right-hand side of the above equation finishes the proof of Lemma 4. □

Now, we are ready to prove Theorem 2. First, we relax $R^i(T)$ by some $r_0 > 0$ as follows,

$$R^i(T) \leq \lambda_i r_0 T + \sum_{x\in\widetilde{\mathcal{A}}_+^i} n_T(x)\Delta^i(x)\mathbb{I}(\Delta^i(x) > \lambda_i r_0).$$

Next, due to $\widetilde{\mathcal{A}}_+^i = \cup_{j\in\mathbb{N}}\widetilde{\mathcal{A}}_j^i$ and the definition of $\widetilde{\mathcal{A}}_j^i$ in (19), we rewrite above equation as

$$R^i(T) \leq \lambda_i r_0 T + \sum_{j\in\mathbb{N}}\sum_{x\in\widetilde{\mathcal{A}}_j^i} n_T(x)\Delta^i(x)\mathbb{I}(2^{-j} \geq r_0).$$



(a) Static Methods

(b) Adaptive Methods

Figure 1: Comparison of our algorithms versus PF-LEX and zooming algorithm.

Then, Lemma 3 and Lemma 4 tell that

$$R^i(T) \leq \lambda_i r_0 T + 1152\lambda_i \tilde{\alpha}_T Z_i \sum_{j\in\mathbb{N}} 2^{j(d_z^i+1)} \mathbb{I}(2^{-j} \geq r_0).$$

Utilizing the sum formula of the geometric sequence, we get

$$\begin{aligned} R^i(T) \leq\ & \lambda_i r_0 T + 1152\lambda_i \tilde{\alpha}_T Z_i \sum_{j=0}^{\lfloor -\log_2(r_0)\rfloor} 2^{j(d_z^i+1)} \\ \leq\ & \lambda_i r_0 T + 1152\lambda_i \tilde{\alpha}_T Z_i (2/r_0)^{d_z^i+1}. \end{aligned}$$

Finally, we minimize the right side of the above equation by taking

$$r_0 = (1152\tilde{\alpha}_T Z_i 2^{d_z^i+1}/T)^{\frac{1}{d_z^i+2}}.$$

This concludes the proof. □

Experiments

In this section, we conduct numerical experiments to verify the effectiveness of our algorithms. We adopt PF-LEX (Hüyük and Tekin 2021) and zooming algorithm (Kleinberg, Slivkins, and Upfal 2008) as baselines. PF-LEX is a static method designed for MOMAB under lexicographic ordering, and the zooming algorithm is an adaptive method that is optimal for single objective Lipschitz bandits.

Following the existing experimental setup (Magureanu, Combes, and Proutiere 2014), we set the arm space $\mathcal{X} = [0, 1]$ with a Euclidean metric on it. The number of objectives is set as $m = 3$, and the expected payoff functions are given as $\mu^1(x) = 1 - \min_{p\in\{0.1,0.4,0.8\}} |x-p|$, $\mu^2(x) = 1 - 2\min_{p\in\{0.3,0.7\}} |x-p|$ and $\mu^3(x) = 1-2|x-0.3|$. Note that the optimal arms for the first objective are $\{0.1, 0.4, 0.8\}$, and the optimal arms for both the first and second objectives are $\{0.4, 0.8\}$. Thus, all three objectives must be considered to determine the lexicographically optimal arm 0.4. We set the payoff $y_t^i = \mu^i(x_t) + \eta_t$, where η_t is drawn from a Gaussian distribution with mean 0 and variance 1. The time horizon T is 6×10^4, and thus the nearly optimal query parameter r for SDLO is 0.025, as stated in Corollary 1. The static arm set for SDLO and PF-LEX is constructed as $\mathcal{A} = \{0.025 + 0.05 \times (k-1) | k \in [20]\}$. The confidence term (15) is scaled by a factor searched in $[1e^{-2}, 1]$, which is a common practice in bandit learning (Chapelle and Li 2011; Li et al. 2012; Zhang et al. 2016; Jun et al. 2017).

We present the cumulative regret for the first and third objectives. To reduce the randomness, each algorithm is repeated 10 times, and the average regret is reported. Figure 1(a) presents the performance of the static methods, PF-LEX and SDLO. SDLO and PF-LEX exhibit comparable performance in the first objective, while SDLO significantly outperforms PF-LEX in the third objective. The primary reason for this difference is that the theoretical guarantee of PF-LEX is constructed under the priority-based regret (1) and is not reliable for general regret (2). Figure 1(b) showcases the performance of two adaptive methods, the zooming algorithm and ADLO. ADLO demonstrates a similar regret to the zooming algorithm in the first objective but surpasses it in the third objective. This result confirms the effectiveness of ADLO in addressing the MOLP problem.

Conclusion and Future Work

We investigated the MOLB model under lexicographic ordering and proposed two algorithms: SDLO and ADLO. The SDLO algorithm is straightforward but requires an oracle, yielding a regret bound of $\widetilde{O}((1+\lambda^{i-1})T^{(1+d_c)/(2+d_c)})$ for the i-th objective, where $i \in [m]$. In contrast, the ADLO algorithm removes the dependence on oracle and achieves an almost optimal bound of $\widetilde{O}((1+\lambda^{i-1})T^{(1+d_z^i)/(2+d_z^i)})$ for the i-th objective, which matches the lower bound of the Lipschitz bandit problem with respect to T (Kleinberg, Slivkins, and Upfal 2008). Both SDLO and ADLO improve the regret bounds by order of $O((KT)^{1/6})$ compared to the recent work of Hüyük and Tekin (2021), as $d_c = 0$ and $d_z = 0$ for K-armed bandit problem. Moreover, we extended the metric of lexicographically ordered multiobjective bandits from the priority-based regret to the general regret, which more accurately evaluates the performance of algorithms.

However, both SDLO and ADLO require the prior knowledge λ. Thus, a challenging open problem is to eliminate the dependence on λ and achieve a regret bound of $\widetilde{O}(T^{(1+d_z^i)/(2+d_z^i)})$ for all objectives.

Acknowledgments

The work described in this paper was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China [GRF Project No. CityU 11215622] and by Natural Science Foundation of China [Project No: 62276223].

References

Abbasi-yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved Algorithms for Linear Stochastic Bandits. In *Advances in Neural Information Processing Systems 24*, 2312–2320.

Agrawal, R. 1995. The Continuum-Armed Bandit Problem. *SIAM Journal on Control and Optimization*, 33(6): 1926–1951.

Auer, P. 2002. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research*, 3(11): 397–422.

Auer, P.; Ortner, R.; and Szepesvári, C. 2007. Improved Rates for the Stochastic Continuum-Armed Bandit Problem. In *Proceedings of the 20th Annual Conference on Learning Theory*, 454–468.

Bubeck, S.; Dekel, O.; Koren, T.; and Peres, Y. 2015. Bandit Convex Optimization: $\sqrt{T}$ Regret in One Dimension. In *Proceedings of The 28th Conference on Learning Theory*, 266–278.

Bubeck, S.; Munos, R.; Stoltz, G.; and Szepesvári, C. 2011. X-Armed Bandits. *Journal of Machine Learning Research*, 12(46): 1655–1695.

Bubeck, S.; Stoltz, G.; Szepesvári, C.; and Munos, R. 2008. Online Optimization in X-Armed Bandits. In *Advances in Neural Information Processing Systems 21*, 201–208.

Bubeck, S.; Stoltz, G.; and Yu, J. Y. 2011. Lipschitz Bandits Without the Lipschitz Constant. In *In Proceedings of the 22nd International Conference on Algorithmic Learning Theory*, 144–158.

Chapelle, O.; and Li, L. 2011. An Empirical Evaluation of Thompson Sampling. In *Advances in Neural Information Processing Systems 24*, 2249–2257.

Drugan, M. M.; and Nowe, A. 2013. Designing multi-objective multi-armed bandits algorithms: A study. In *The 2013 International Joint Conference on Neural Networks*, 1–8.

Ehrgott, M. 2005. *Multicriteria Optimization*. Berlin, Heidelberg: Springer-Verlag.

Feng, Y.; Huang, Z.; and Wang, T. 2022. Lipschitz Bandits with Batched Feedback. In *Advances in Neural Information Processing Systems 35*, 19836–19848.

Gou, Y.; Yi, J.; and Zhang, L. 2023. Stochastic Graphical Bandits with Heavy-Tailed Rewards. In *Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence*, 734–744.

Hosseini, H.; Sikdar, S.; Vaish, R.; and Xia, L. 2021. Fair and Efficient Allocations under Lexicographic Preferences. *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 5472–5480.

Hüyük, A.; and Tekin, C. 2021. Multi-objective multi-armed bandit with lexicographically ordered and satisficing objectives. *Machine Learning*, 110(6): 1233–1266.

Jee, K.-W.; McShan, D. L.; and Fraass, B. A. 2007. Lexicographic ordering: intuitive multicriteria optimization for IMRT. *Physics in Medicine & Biology*, 52: 1845–1861.

Jun, K.-S.; Bhargava, A.; Nowak, R.; and Willett, R. 2017. Scalable Generalized Linear Bandits: Online Computation and Hashing. In *Advances in Neural Information Processing Systems 30*, 99–109.

Kleinberg, R. 2004. Nearly Tight Bounds for the Continuum-armed Bandit Problem. *Advances in Neural Information Processing Systems 17*, 697–704.

Kleinberg, R.; Slivkins, A.; and Upfal, E. 2008. Multi-armed Bandits in Metric Spaces. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, 681–690.

Lai, T. L.; and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1): 4–22.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit Algorithms*. Cambridge University Press.

Li, L.; Chu, W.; Langford, J.; Moon, T.; and Wang, X. 2012. An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. In *Proceedings of the Workshop on On-line Trading of Exploration and Exploitation 2*, 19–36.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670.

Lu, S.; Wang, G.; Hu, Y.; and Zhang, L. 2019a. Multi-Objective Generalized Linear Bandits. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 3080–3086.

Lu, S.; Wang, G.; Hu, Y.; and Zhang, L. 2019b. Optimal Algorithms for Lipschitz Bandits with Heavy-tailed Rewards. In *Proceedings of the 36th International Conference on Machine Learning*, 4154–4163.

Luo, H.; Wei, C.-Y.; Agarwal, A.; and Langford, J. 2018. Efficient Contextual Bandits in Non-stationary Worlds. In *Proceedings of the 31st Conference On Learning Theory*, 1739–1776.

Ma, X.; Zhao, L.; Huang, G.; Wang, Z.; Hu, Z.; Zhu, X.; and Gai, K. 2018. Entire Space Multi-Task Model: An Effective Approach for Estimating Post-Click Conversion Rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 1137–1140.

Magureanu, S.; Combes, R.; and Proutiere, A. 2014. Lipschitz Bandits: Regret Lower Bound and Optimal Algorithms. In *Proceedings of The 27th Conference on Learning Theory*, 975–999.

Podimata, C.; and Slivkins, A. 2021. Adaptive Discretization for Adversarial Lipschitz Bandits. In *Proceedings of the 34st Conference On Learning Theory*, 3788–3805.

Q. Yahyaa, S.; M. Drugan, M.; and Manderick, B. 2014. Knowledge Gradient for Multi-Objective Multi-Armed Bandit Algorithms. In *Proceedings of the 6th International Conference on Agents and Artificial Intelligence 1*, 74–83.

Qin, Y.; Li, Y.; Pasqualetti, F.; Fazel, M.; and Oymak, S. 2023. Stochastic Contextual Bandits with Long Horizon Rewards. *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, 9525–9533.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5): 527–535.

Shao, H.; Yu, X.; King, I.; and Lyu, M. R. 2018. Almost Optimal Algorithms for Linear Stochastic Bandits with Heavy-tailed Payoffs. In *Advances in Neural Information Processing Systems 31*, 8430–8439.

Skalse, J.; Hammond, L.; Griffin, C.; and Abate, A. 2022. Lexicographic Multi-Objective Reinforcement Learning. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, 3430–3436.

Turgay, E.; Oner, D.; and Tekin, C. 2018. Multi-objective contextual bandit problem with similarity information. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, 1673–1681.

Van Moffaert, K.; Van Vaerenbergh, K.; Vrancx, P.; and Nowe, A. 2014. Multi-objective $\mathcal{X}$-Armed bandits. In *2014 International Joint Conference on Neural Networks*, 2331–2338.

Villar, S. S.; Bowden, J.; and Wason, J. 2015. Multi-armed Bandit Models for the Optimal Design of Clinical Trials: Benefits and Challenges. *Statistical Science*, 30(2): 199 – 215.

Wang, T.; Ye, W.; Geng, D.; and Rudin, C. 2020. Towards Practical Lipschitz Bandits. In *Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference*, 129–138.

Wanigasekara, N.; and Yu, C. L. 2019. Nonparametric Contextual Bandits in Metric Spaces with Unknown Metric. In *Advances in Neural Information Processing Systems 32*, 14657–14667.

Weber, E.; Rizzoli, A. E.; Soncini-Sessa, R.; and Castelletti, A. 2002. Lexicographic Optimisation for Water Resources Planning: the Case of Lake Verbano, Italy. 235–240.

White, J. M. 2012. *Bandit Algorithms for Website Optimization*. O'Reilly Media, Inc.

Wray, K.; Zilberstein, S.; and Mouaddib, A.-I. 2015. Multi-Objective MDPs with Conditional Lexicographic Reward Preferences. *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 3418–3424.

Wray, K. H.; and Zilberstein, S. 2015. Multi-Objective POMDPs with Lexicographic Reward Preferences. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 1719–1725.

Xu, M.; and Klabjan, D. 2023. Pareto Regret Analyses in Multi-objective Multi-armed Bandit. In *Proceedings of the 40th International Conference on International Conference on Machine Learning*, 38499–38517.

Xue, B.; Wang, G.; Wang, Y.; and Zhang, L. 2020. Nearly Optimal Regret for Stochastic Linear Bandits with Heavy-Tailed Payoffs. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 2936–2942.

Yu, X.; Shao, H.; Lyu, M. R.; and King, I. 2018. Pure exploration of multi-armed bandits with heavy-tailed payoffs. *In Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, 937–946.

Zhang, L.; Yang, T.; Jin, R.; Xiao, Y.; and Zhou, Z. 2016. Online Stochastic Linear Optimization under One-bit Feedback. In *Proceedings of the 33rd International Conference on Machine Learning*, 392–401.

Zhou, Z.; Xu, R.; and Blanchet, J. 2019. Learning in Generalized Linear Contextual Bandits with Stochastic Delays. In *Advances in Neural Information Processing Systems 32*, 5197–5208.

Zhu, Y.; and Mineiro, P. 2022. Contextual Bandits with Smooth Regret: Efficient Learning in Continuous Action Spaces. In *Proceedings of the 39th International Conference on Machine Learning*, 27574–27590.