# Beyond the Lower Bound: Bridging Regret Minimization and Best Arm Identification in Lexicographic Bandits

**Bo Xue**[1,2], **Yuanyu Wan**[3], **Zhichao Lu**[1], **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] School of Software Technology, Zhejiang University, Ningbo, China
boxue4-c@my.cityu.edu.hk, wanyy@zju.edu.cn, {zhichao.lu, qingfu.zhang}@cityu.edu.hk

## Abstract

In multi-objective decision-making with hierarchical preferences, lexicographic bandits provide a natural framework for optimizing multiple objectives in a prioritized order. In this setting, a learner repeatedly selects arms and observes reward vectors, aiming to maximize the reward for the highest-priority objective, then the next, and so on. While previous studies have primarily focused on regret minimization, this work bridges the gap between *regret minimization* and *best arm identification* under lexicographic preferences. We propose two elimination-based algorithms to address this joint objective. The first algorithm eliminates suboptimal arms sequentially, layer by layer, in accordance with the objective priorities, and achieves sample complexity and regret bounds comparable to those of the best single-objective algorithms. The second algorithm simultaneously leverages reward information from all objectives in each round, effectively exploiting cross-objective dependencies. Remarkably, it outperforms the known lower bound for the single-objective bandit problem, highlighting the benefit of cross-objective information sharing in the multi-objective setting. Empirical results further validate their superior performance over baselines.

## Introduction

The multi-armed bandit (MAB) problem is a foundational framework for sequential decision-making under uncertainty (Robbins 1952; Lai and Robbins 1985; Auer 2002), with widespread applications in domains such as online recommendation systems (Schwartz, Bradlow, and Fader 2017), clinical trials (Villar, Bowden, and Wason 2015), and adaptive routing (Awerbuch and Kleinberg 2008). In the classical MAB setting (Bubeck and Cesa-Bianchi 2012), a learner repeatedly selects one arm from a finite set of $K$ arms, each associated with an unknown reward distribution. Upon each selection, the learner observes a stochastic reward sampled from the distribution of the chosen arm. Depending on the learning objective, bandit algorithms are generally categorized into two primary paradigms: (1) regret minimization (RM), which aims to minimize the cumulative regret incurred by not always selecting the optimal arm (Auer, Cesa-Bianchi, and Fischer 2002; Abbasi-yadkori, Pál, and Szepesvári 2011; Lykouris, Mirrokni, and Paes Leme 2018); and (2) best arm identification (BAI), which aims to identify the optimal arm using as few samples as possible (Audibert and Bubeck 2010; Karnin, Koren, and Somekh 2013; Jamieson et al. 2014; Kaufmann, Cappé, and Garivier 2016; Jin et al. 2024).

While traditional bandit algorithms focus on optimizing a scalar reward (Auer, Cesa-Bianchi, and Fischer 2002), many real-world applications involve multiple, often conflicting objectives (Xie et al. 2021; Shu et al. 2024), which motivate the study of the multi-objective bandit problem (Drugan and Nowe 2013). Several formulations have been proposed in this context, including scalarized regret minimization (Q. Yahyaa, M. Drugan, and Manderick 2015), Pareto regret minimization (Lu et al. 2019; Xu and Klabjan 2023), and Pareto set identification (Auer et al. 2016). These methods offer different strategies for managing trade-offs among objectives, but generally assume that all objectives are equally important or can be aggregated into a single scalar value. However, in many practical scenarios, objectives have inherently different priorities. For instance, in medical diagnosis (Alkaabneh and Diabat 2023), patient safety typically outweighs considerations such as cost or treatment speed; in recommendation systems (Li et al. 2023), fairness may be prioritized over user engagement.

An effective framework for modeling such hierarchical decision-making is lexicographic bandits (Tekin and Turgay 2018; Hüyük and Tekin 2021), where the agent seeks to optimize multiple objectives according to the lexicographic (i.e., priority-based) order. Unlike approaches that aggregate objectives into a single scalar using linear weights, the lexicographic bandit framework preserves the dominance structure: higher-priority objectives must be optimized before lower-priority ones are considered. This formulation provides a more faithful representation of structured decision-making in sensitive applications such as hyperparameter optimization (Zhang et al. 2023) and multi-criteria resource allocation (Kurokawa, Procaccia, and Shah 2018).

Research on lexicographic bandits has attracted increasing attention in recent years, with most studies focusing on the RM task (Tekin and Turgay 2018; Hüyük and Tekin 2021; Xue et al. 2024). However, to the best of our knowledge, another significant task in the bandit literature, BAI, has not yet been explored in the context of lexicographic bandits. In many real-world scenarios, it is important to min-

*Qingfu Zhang is the corresponding author.

imize regret during the learning phase while also accurately identifying the optimal arm at the end (Zhong, Cheung, and Tan 2023). For instance, in clinical trials, ethical considerations require providing effective treatments during the study (low regret), while the ultimate goal is to determine the most effective treatment (accurate BAI). These dual requirements motivate a central research question:

*Can we design algorithms for lexicographic bandits that effectively unify RM and BAI?*

In this work, we answer this question affirmatively and demonstrate that a unified treatment of RM and BAI in lexicographic bandits is not only possible, but also yields *surprising benefits*. In particular, the rich multi-objective feedback naturally accelerates the elimination of suboptimal arms during the BAI process, thereby reducing the need to explore inferior actions and mitigating cumulative regret. This positive feedback loop between accurate identification and efficient learning highlights an unexpected advantage of jointly addressing BAI and RM in lexicographic bandits.

This paper presents the first algorithmic framework for lexicographic bandits that simultaneously tackles both RM and BAI tasks. Our main contributions are as follows:

- We propose a simple yet effective elimination-based algorithm, **LexElim-Out**, which sequentially filters suboptimal arms, starting from the highest-priority objective and proceeding to the lowest. This top-down elimination strategy ensures that lower-priority objectives are only considered after higher-priority objectives have been sufficiently optimized. Theoretically, LexElim-Out matches the best-known problem-dependent BAI guarantees for the primary objective, without compromising performance when optimizing additional objectives.
- We further develop an enhanced algorithm, **LexElim-In**, which eliminates arms using joint reward information from all objectives in each round. By simultaneously incorporating information across objectives during each decision step, LexElim-In accelerates the identification and elimination of suboptimal arms. We show that it surpasses the known lower bounds for single-objective bandits in both regret and sample complexity, highlighting the advantage of exploiting the multi-objective structure.
- LexElim-In also enjoys anytime performance guarantees. Specifically, we establish a minimax regret bound of $\widetilde{O}(\Lambda^i(\lambda) \cdot \sqrt{Kt})$ for each objective $i \in [m]$ at any round $t \geq 1$, ensuring that the regret grows at most at a square-root rate over time. This bound is comparable to the best-known results in single-objective bandits, while operating in a more challenging multi-objective setting.
- Through extensive experiments on synthetic data, we demonstrate that both LexElim-Out and LexElim-In outperform existing baselines in cumulative regret and BAI sample complexity. Notably, LexElim-In exhibits superior performance on some instances, validating the benefit of joint exploitation of multi-objective reward signals.

## Preliminaries

This paper studies the lexicographic bandit problem, where a learner selects arms to simultaneously optimize multiple objectives that are ranked according to their importance.

Let $K \in \mathbb{N}_+$ denote the number of objectives, and $m \in \mathbb{N}_+$ be the number of objectives. For any $N \in \mathbb{N}_+$, let $[N] = \{1, 2, \ldots, N\}$ denote the index set. At each round $t \in [T]$, the learner chooses an arm $a_t \in [K]$ and receives a stochastic reward vector $\boldsymbol{r}_t(a_t) = [r_t^1(a_t), r_t^2(a_t), \ldots, r_t^m(a_t)] \in \mathbb{R}^m$. The component $r_t^i(a_t)$ corresponds to the reward for the $i$-th objective and is independently drawn from a 1-sub-Gaussian distribution with an *unknown* mean $\mu^i(a_t) \in [0, 1]$. That is, for all $\beta \in \mathbb{R}$ and $i \in [m]$,

$$\mathrm{E}[e^{\beta r_t^i(a_t)}] \leq \exp\left(\beta^2/2\right), \quad \mu^i(a_t) = \mathrm{E}[r_t^i(a_t)]. \tag{1}$$

The key challenge in lexicographic bandits is managing the hierarchical structure of objectives: the learner must optimize the most important objective first, followed by the second-most important, and so on. To formalize this, we adopt the standard notion of lexicographic dominance from prior work (Hüyük and Tekin 2021; Xue et al. 2024).

**Definition 1 (Lexicographic Order)** *Let $a_1$, $a_2 \in [K]$ be two arms. We say that $a_1$ lexicographically dominates $a_2$ if there exists an index $i \in [m]$ such that $\mu^j(a_1) = \mu^j(a_2)$ for all $j < i$, and $\mu^i(a_1) > \mu^i(a_2)$.*

An illustrate example is that the arm with expected rewards $[5, 5, 2]$ lexicographically dominates the arm with expected rewards $[5, 4, 8]$, even though the latter has a higher value on the third objective. Lexicographic order induces a total order over arms, enabling the comparison of any two arms and thereby defining the notion of the lex-optimal arm.

**Definition 2 (Lex-optimal Arm)** *An arm $a_*$ is lex-optimal if no other arm in $[K]$ lexicographically dominates it.*

We study two classical goals in the bandit literature, and adapt them to the lexicographic multi-objective setting. The first is **Regret Minimization (RM)**, which aims to minimize the cumulative regret for each objective over $T$ rounds,

$$R^i(T) = T \cdot \mu^i(a_*) - \sum_{t=1}^{T} \mu^i(a_t), i \in [m].$$

The second is **Best Arm Identification (BAI)** with fixed confidence. Given a confidence level $\delta \in (0, 1)$, the goal is to identify the optimal arm (or optimal arm set) with probability at least $1 - \delta$, using as few samples as possible.

Unlike the single-objective setting where the optimal arm is uniquely defined, in the multi-objective case, different objectives may induce different optimal arms. To capture this, we consider the following two types of optimal arm sets for each objective $i \in [m]$:

- $\mathcal{O}_*(i) = \{a \in [K] \mid \mu^j(a) = \mu^j(a_*) \text{ for all } j \in [i]\}$: the set of arms that match $a_*$ on the top $i$ objectives;
- $\widetilde{\mathcal{O}}_*(i) = \{a \in [K] \mid \mu^i(a) \geq \mu^i(a_*)\}$: the set of arms that are optimal with respect to the $i$-th objective alone.

| Algorithm | Sample Complexity | Regret Bound | # Objectives |
|---|---|---|---|
| Auer, Cesa-Bianchi, and Fischer (2002) | – | $\widetilde{O}\left(\sum_{\Delta(a)>0} \frac{1}{\Delta(a)}\right)$ | 1 |
| Degenne and Perchet (2016) | – | $O\left(\sqrt{KT}\right)$ | 1 |
| Lattimore (2018) **(Lower Bound)** | – | $\Omega\left(\sum_{\Delta(a)>0} \frac{1}{\Delta(a)}\right)$ | 1 |
| Karnin, Koren, and Somekh (2013) | $\widetilde{O}\left(\sum_{\Delta(a)>0} \frac{1}{(\Delta(a))^2}\right)$ | – | 1 |
| Jamieson et al. (2014) **(Lower Bound)** | $\Omega\left(\sum_{\Delta(a)>0} \frac{1}{(\Delta(a))^2}\right)$ | – | 1 |
| Degenne et al. (2019) | $\widetilde{O}\left(\sum_{\Delta(a)>0} \frac{1}{(\Delta(a))^2}\right)$ | $\widetilde{O}\left(\sum_{\Delta(a)>0} \frac{1}{\Delta(a)}\right)$ | 1 |
| LexElim-Out **(Ours)** | $\widetilde{O}\left(\sum_{j=1}^{i} \sum_{a\in\mathcal{S}(j)} \frac{1}{(\Delta^j(a))^2}\right)$ | $\widetilde{O}\left(\sum_{j=1}^{i} \sum_{a\in\mathcal{S}(j)} \frac{\Delta^i(a)}{(\Delta^j(a))^2}\right)$ | $i \in [m]$ |
| LexElim-In **(Ours)** | $\widetilde{O}\left(\sum_{\Delta^i(a)>0} \frac{1}{(\tilde{\Delta}(a))^2}\right)$ | $\widetilde{O}\left(\sum_{\Delta^i(a)>0} \frac{\Delta^i(a)}{(\tilde{\Delta}(a))^2}\right)$<br>$\widetilde{O}\left(\Lambda^i(\lambda)\cdot\sqrt{KT}\right)$ | $i \in [m]$ |

[1.] $\Delta^i(a) = \mu^i(a_*) - \mu^i(a)$ for all $a \in [K]$ and $i \in [m]$, where $a_*$ is the lex-optimal arm defined in Definition 2.
[2.] For single-objective works, we simplify the notation by letting $\Delta(a) := \Delta^1(a)$.
[3.] $\mathcal{S}(i) = \{a \in \mathcal{O}_*(i-1) \mid \Delta^i(a) > 0\}, \mathcal{O}_*(i-1) = \{a \in [K] \mid \mu^j(a_*) = \mu^j(a),\ \forall j \in [i-1]\}$ and $\mathcal{O}_*(0) = [K]$.
[4.] $\tilde{\Delta}(a) = \max_{i\in[m]} \left\{\frac{\Delta^i(a)}{\Lambda^i(\lambda)} \cdot \mathbb{I}[\Delta^i(a) > 0]\right\}$, where $\Lambda^i(\lambda) = 1 + \lambda + \cdots + \lambda^{i-1}$ and $\lambda \geq 0$ is defined in Eq. (2).

Table 1: Overview of Our Results and Comparisons with RM and BAI Methods: Since $\tilde{\Delta}(a) \geq \Delta^1(a)$ for all $a \in [K]$, LexElim-In outperforms the lower bounds of the single-objective problem (Jamieson et al. 2014; Lattimore 2018).

Let $T^i(\delta)$ and $\widetilde{T}^i(\delta)$ denote the number of samples used to identify $\mathcal{O}_*(i)$ and $\widetilde{\mathcal{O}}_*(i)$, respectively. Thus, the sample complexity of identifying $a_*$ is $T^m(\delta)$ or $\max_{i\in[m]} \widetilde{T}^i(\delta)$.

Finally, we introduce a parameter $\lambda$ to capture the trade-offs among conflicting objectives. In the lexicographic bandit problem, we assume that for any $i \geq 2$ and $a \in [K]$,

$$\mu^i(a) - \mu^i(a_*) \leq \lambda \cdot \max_{j\in[i-1]} \{\mu^j(a_*) - \mu^j(a)\}. \quad (2)$$

## Related Work

We review bandit work on four directions: regret minimization (RM), best arm identification (BAI), joint optimization of RM and BAI, and multi-objective bandits (MOB).

**RM.** The seminal work of Robbins (1952) initiated the study of the MAB problem. A foundational algorithm for minimizing regret in stochastic MABs is the Upper Confidence Bound (UCB) algorithm (Auer, Cesa-Bianchi, and Fischer 2002), which achieves a problem-dependent regret bound of $\widetilde{O}\left(\sum_{\Delta(a)>0} 1/\Delta(a)\right)$. To improve worst-case performance, Audibert and Bubeck (2009) proposed the MOSS algorithm, which attains the minimax-optimal regret bound of $O(\sqrt{KT})$. This was further improved by Degenne and Perchet (2016), who developed an anytime variant of MOSS that removes the need for prior knowledge of the time horizon $T$, thereby improving its practicality. Additionally, Lattimore (2018) established a fundamental lower bound of $\Omega\left(\sum_{\Delta(a)>0} 1/\Delta(a)\right)$, highlighting the intrinsic complexity of the problem. These foundational results have been extended to structured bandit settings, such as linear bandits (Dani, Hayes, and Kakade 2008), graphical bandits (Alon et al. 2015) and combinatorial bandits (Chen et al. 2016).

**BAI.** Existing work on BAI can be categorized into two primary settings: (a) *Fixed-confidence setting:* The algorithm aims to identify the best arm with probability at least $1 - \delta$, using as few samples as possible. Early approaches include the Successive Elimination algorithm (Even-Dar, Mannor, and Mansour 2006), which sequentially discards suboptimal arms based on empirical comparisons. Later works (Karnin, Koren, and Somekh 2013; Garivier and Kaufmann 2016) introduced more refined strategies that achieve near-optimal sample complexity by adaptively allocating samples to competitive arms. Jamieson et al. (2014) established a lower bound showing that the sample complexity of any algorithm is at least $\Omega(\sum_{\Delta(a)>0} 1/(\Delta(a))^2)$.

(b) *Fixed-budget setting:* Given a fixed budget $T \in \mathbb{N}$, the objective is to minimize the probability of incorrect identification at time $T$. Audibert and Bubeck (2010) first studied this setting and designed an algorithm based on successive rejects, proved its optimality up to logarithmic factors. A subsequent work of Karnin, Koren, and Somekh (2013) further improved the theoretic guarantees, leaving only doubly-logarithmic gap. Carpentier and Locatelli (2016) constructed lower bounds to confirm the near-optimality of these results.

**RM and BAI.** While RM and BAI have traditionally been treated as separate goals, recent studies have sought to address them jointly. Degenne et al. (2019) explored both goals with a fixed confidence and introduced an algorithm $\text{UCB}_\alpha$, where the parameter $\alpha > 1$ controls the trade-off between regret and sample complexity. Subsequently, Zhong,

Cheung, and Tan (2023) quantified the trade-off between RM and BAI in the fixed-budget setting. In parallel, Zhang and Ying (2023) developed algorithms that achieve asymptotic regret optimality in Gaussian bandit models. Most recently, Yang, Tan, and Jin (2024) established an information-theoretic lower bound for BAI with minimal regret and proposed an algorithm that attains asymptotic optimality.

**MOB.** Multi-objective bandits aim to balance competing objectives, often without a unique optimal solution. Prior research has explored various notions of optimality and preference structures to address this challenge. Early studies focus extended the Pareto optimality concept to online learning (Auer et al. 2016; Kone, Kaufmann, and Richert 2024; Crepon, Garivier, and M Koolen 2024), where the learner aims to approximate the Pareto front. Another line of work employs scalarization techniques (Drugan and Nowe 2013; Q. Yahyaa, M. Drugan, and Manderick 2015; Wanigasekara et al. 2019) to guide learning, based on utility functions or user-specified preferences. Lexicographic bandits, a specific form of preference-based MOB, have been studied under the RM framework (Hüyük and Tekin 2021; Tekin 2019; Xue et al. 2024). Our work contributes the first unified framework that simultaneously addresses RM and BAI under lexicographic preference, and we theoretically demonstrates how joint rewards signals lead to improved performance.

## Algorithms

In this section, we propose two algorithms tailored for lexicographic bandits: LexElim-Out and LexElim-In. Both algorithms are based on the principle of arm elimination, but differ in how they utilize multi-objective information.

### Warm-up: LexElim-Out

We begin by introducing LexElim-Out, a warm-up algorithm for the lexicographic MAB problem. This algorithm follows an outer-layer elimination strategy, where arms are pruned layer-by-layer according to the lexicographic priority of objectives. Details are provided in Algorithm 1.

LexElim-Out requires prior knowledge of $|\mathcal{O}_*(i)|$, i.e., the number of arms that are optimal up to objective $i \in [m]$. This aligns with common practices in the single-objective BAI literature (Bubeck, Munos, and Stoltz 2009; Audibert and Bubeck 2010; Zhang and Ying 2023), where the optimal arm is typically assumed to be unique. Therefore, our setting does not require any additional information beyond what is standard in the single-objective BAI methods.

Given a confidence parameter $\delta \in (0,1)$, the number of arms $K$, the number of objectives $m$, and the cardinalities $|\mathcal{O}_*(i)|$ for all $i \in [m]$, LexElim-Out proceeds as follows. For each arm $a \in [K]$ and objective $i \in [m]$, it initializes the empirical mean reward $\hat{\mu}^i(a)$ and pull count $n(a)$ to zero, and the confidence width $c(a)$ to $+\infty$. The active arm set is initialized as $\mathcal{A}_1 = [K]$, and the round index as $t = 1$.

Then, LexElim-Out performs iterations over the objectives in order of priority, from the most to the least important. For each objective $i \in [m]$, it repeatedly performs elimination rounds until the size of the active arm set is reduced to the known optimal set size, i.e., $|\mathcal{A}_t| = |\mathcal{O}_*(i)|$. In each round, the algorithm selects the arm with the highest uncertainty,

$$a_t = \operatorname*{argmax}_{a \in \mathcal{A}_t} c(a).$$

It then identifies the empirical best arm with respect to the current objective, i.e., $\hat{a}_t^i = \arg\max_{a \in \mathcal{A}_t} \hat{\mu}^i(a)$. The active arm set is updated by retaining only those arms whose empirical means are within $2c(a_t)$ of the best empirical arm $\hat{a}_t^i$,

$$\mathcal{A}_{t+1} = \{a \in \mathcal{A}_t \mid \hat{\mu}^i(\hat{a}_t^i) - \hat{\mu}^i(a) \le 2c(a_t)\}.$$

This ensures that arms that are suboptimal on the $i$-th objective are eliminated.

After the elimination step, LexElim-Out plays the most uncertain arm $a_t$ and observes its reward vector $\boldsymbol{r}_t(a_t) = [r_t^1(a_t), r_t^2(a_t), \ldots, r_t^m(a_t)]$. The empirical mean for each objective $i \in [m]$ is updated using an incremental average,

$$\hat{\mu}^i(a_t) = \frac{n(a_t) \cdot \hat{\mu}^i(a_t) + r_t^i(a_t)}{n(a_t) + 1}. \tag{3}$$

Next, the pull count $n(a_t)$ is incremented, and the confidence width $c(a_t)$ is updated by a concentration inequality,

$$\begin{aligned} n(a_t) &= n(a_t) + 1, \\ c(a_t) &= \sqrt{\frac{4}{n(a_t)} \log\left(\frac{6Km \cdot n(a_t)}{\delta}\right)}. \end{aligned} \tag{4}$$

The round index $t$ is then incremented to $t+1$.

Once all objectives have been processed, LexElim-Out terminates and outputs the sole remaining arm in the final active set. The regret bounds and sample complexity of the algorithm are established in Theorems 1 and 2, respectively.

**Theorem 1** *Suppose that Eq.* (1) *holds, define* $\mathcal{S}(i) = \{a \in \mathcal{O}_*(i-1) \mid \Delta^i(a) > 0\}$ *with* $\mathcal{O}_*(0) = [K]$*, and set* $\gamma^i(\delta) = 64 \log\left(\frac{392Km}{(\Delta^i(a))^2 \cdot \delta}\right)$*. With probability at least* $1-\delta$*, for any objective* $i \in [m]$*, the regret of LexElim-Out satisfies*

$$R^i(t) \le \sum_{j=1}^{i} \sum_{a \in \mathcal{S}(j)} \frac{\gamma^j(\delta) \cdot \Delta^i(a)}{(\Delta^j(a))^2}.$$

---

Algorithm 1: Outer-layer Active Arm Elimination in Lexicographic Bandits (LexElim-Out)

---

**Input:** $\delta \in (0,1), K, m, \{|\mathcal{O}_*(i)|, \forall i \in [m]\}$

Initialize empirical mean $\hat{\mu}^i(a) = 0$, counter $n(a) = 0$, and confidence width $c(a) = +\infty$ for $i \in [m]$, $a \in [K]$
Initialize active set $\mathcal{A}_1 = [K]$ and round counter $t = 1$
**for** $i = 1, 2, \ldots, m$ **do**
  **while** $|\mathcal{A}_t| > |\mathcal{O}_*(i)|$ **do**
    Choose the arm $a_t = \operatorname{argmax}_{a \in \mathcal{A}_t} c(a)$
    $\hat{a}_t^i = \operatorname{argmax}_{a \in \mathcal{A}_t} \hat{\mu}^i(a)$
    $\mathcal{A}_{t+1} = \{a \in \mathcal{A}_t | \hat{\mu}^i(\hat{a}_t^i) - \hat{\mu}^i(a) \le 2c(a_t)\}$
    Play $a_t$ and observe reward vectors $\boldsymbol{r}_t(a_t)$
    Update $\hat{\mu}^i(a_t)$ for all $i \in [m]$ by Eq. (3)
    Update $n(a_t)$ and $c(a_t)$ by Eq. (4)
    Set $t = t + 1$
  **end while**
**end for**
**Output** the arm in $\mathcal{A}_t$

---

**Remark 1** Theorem 1 states that LexElim-Out achieves a regret bound of $\widetilde{O}\left(\sum_{j=1}^{i}\sum_{a\in\mathcal{S}(j)}\frac{\Delta^i(a)}{(\Delta^j(a))^2}\right)$ for any objective $i \in [m]$, with the following key implications.

- For the primary objective ($i = 1$), its regret bound is $\widetilde{O}\left(\sum_{\Delta^1(a)>0}\frac{1}{\Delta^1(a)}\right)$, matching the known lower bound for single-objective bandits (Lattimore 2018). This ensures no performance degradation for the highest-priority objective when optimizing additional objectives.
- For the secondary objective ($i = 2$), its regret bound includes two terms:

$$\underbrace{\widetilde{O}\left(\sum_{\Delta^1(a)>0}\frac{\Delta^2(a)}{(\Delta^1(a))^2}\right)}_{\text{cross-objective cost}}+\underbrace{\widetilde{O}\left(\sum_{a\in\mathcal{S}(2)}\frac{1}{\Delta^2(a)}\right)}_{\text{single-objective term}}.$$

  The second term aligns with the regret bound in the single-objective setting. The first term captures the cost incurred on the second objective due to the need to prioritize the first objective. This cost becomes negligible if $(\Delta^1(a))^2 \gg \Delta^2(a)$, i.e., when arm $a$ is clearly suboptimal on the first objective and thus quickly eliminated.

The same decomposition can be applied to $i > 2$, where the regret bound includes cumulative cross-objective costs from all higher-priority objectives $j < i$, and a local term that matches the single-objective bound for objective $i$.

**Theorem 2** *Suppose the same conditions and notations as in Theorem 1. With probability at least $1-\delta$, for any objective $i \in [m]$, the number of samples required by LexElim-Out to identify $\mathcal{O}_*(i)$ satisfies*

$$T^i(\delta) \le \sum_{j=1}^{i}\sum_{a\in\mathcal{S}(j)}\frac{\gamma^j(\delta)}{(\Delta^j(a))^2}.$$

**Remark 2** From Theorem 2, LexElim-Out identifies the optimal arm set for the first $i$ objectives using at most $\widetilde{O}\left(\sum_{j=1}^{i}\sum_{a\in\mathcal{S}(j)}\frac{1}{(\Delta^j(a))^2}\right)$ samples. In particular, for the highest-priority objective ($i = 1$), the sample complexity simplifies to $\widetilde{O}\left(\sum_{\Delta^1(a)>0}\frac{1}{(\Delta^1(a))^2}\right)$, which matches the known lower bound for single-objective bandits (Jamieson et al. 2014). This implies that LexElim-Out identifies the optimal arm for the primary objective as efficiently as state-of-the-art single-objective algorithms (Karnin, Koren, and Somekh 2013). For general $i \in [m]$, the bound reflects that identifying the lex-optimal arm requires solving a sequence of BAI problems, where suboptimal arms for higher-priority objectives are progressively eliminated before being evaluated on lower-priority ones.

## Improved Algorithm: LexElim-In

LexElim-Out handles objectives layer by layer, it ignores lower-priority objectives when optimizing higher-priority ones. As a result, the arm selection for lower-priority objectives in early rounds is purely random, lacking any targeted exploration. To address this limitation, we propose an improved algorithm, LexElim-In, which adopts an inner-layer elimination strategy that leverages information from all objectives throughout the decision-making process. The complete procedure is presented in Algorithm 2.

Algorithm 2: Inner-layer Active Arm Elimination in Lexicographic Bandits (LexElim-In)

**Input:** $\delta \in (0,1), K, m, \lambda \ge 0$

Initialize empirical mean $\hat{\mu}^i(a) = 0$, counter $n(a) = 0$, and confidence width $c(a) = +\infty$ for $i \in [m]$, $a \in [K]$
Initialize active set $\mathcal{A}_1 = [K]$ and round counter $t = 1$
**while** $|\mathcal{A}_t| > 1$ **do**
  Choose the arm $a_t = \text{argmax}_{a\in\mathcal{A}_t} c(a)$
  Initialize the arm set $\mathcal{A}_t^0 = \mathcal{A}_t$
  **for** $i = 1, 2, \ldots, m$ **do**
    $\hat{a}_t^i = \text{argmax}_{a\in\mathcal{A}_t^{i-1}} \hat{\mu}^i(a)$
    $\mathcal{A}_t^i = \{a \in \mathcal{A}_t^{i-1} | \hat{\mu}^i(\hat{a}_t^i) - \hat{\mu}^i(a) \le (2+4\lambda+\cdots+4\lambda^{i-1}) \cdot c(a_t)\}$
  **end for**
  Play $a_t$ and observe reward vectors $\boldsymbol{r}_t(a_t)$
  Update $\hat{\mu}^i(a_t)$ for all $i \in [m]$ by Eq. (3)
  Update $n(a_t)$ and $c(a_t)$ by Eq. (4)
  Update $\mathcal{A}_{t+1} = \mathcal{A}_t^m$ and $t = t+1$
**end while**
**Output** the arm in $\mathcal{A}_t$

Given a confidence level $\delta \in (0,1)$, the number of arms $K$, the number of objectives $m$, and a trade-off parameter $\lambda \ge 0$, LexElim-In begins with an initialization phase similar to that of LexElim-Out. Specifically, for each arm $a \in [K]$ and each objective $i \in [m]$, the empirical mean reward $\hat{\mu}^i(a)$ and pull count $n(a)$ are set to zero, and the confidence width $c(a)$ is initialized to $+\infty$. The initial active set of arms is defined as $\mathcal{A}_1 = [K]$, and the round index is initialized as $t = 1$.

At each round, LexElim-In selects the arm $a_t \in \mathcal{A}_t$ with the largest confidence width $c(a)$, corresponding to the highest uncertainty, and plays this arm. It then updates the active arm set through a layered filtering process that incorporates empirical means across all objectives in a nested fashion.

Specifically, let $\mathcal{A}_t^0 = \mathcal{A}_t$ and for each objective $i = 1, 2, \ldots, m$, LexElim-In identifies the empirical best arm $\hat{a}_t^i = \arg\max_{a\in\mathcal{A}_t^{i-1}} \hat{\mu}^i(a)$, and eliminates arms in $\mathcal{A}_t^{i-1}$ whose empirical mean falls below that of $\hat{a}_t^i$ by more than a scaled confidence threshold. Formally, the updated set is

$$\mathcal{A}_t^i = \left\{a \in \mathcal{A}_t^{i-1} \mid \hat{\mu}^i(\hat{a}_t^i) - \hat{\mu}^i(a) \le \left(2+4\lambda+\cdots+4\lambda^{i-1}\right)\cdot c(a_t)\right\}. \tag{5}$$

The scaling factor $2+4\lambda+\cdots+4\lambda^{i-1}$ grows geometrically with $i$, allowing lower-priority objectives to tolerate larger reward gaps while still contributing to elimination decisions.

After completing the elimination process across all $m$ objectives, LexElim-In updates the active set to $\mathcal{A}_{t+1} = \mathcal{A}_t^m$. It then pulls arm $a_t$ to observe the full reward vector $\boldsymbol{r}_t(a_t) = [r_t^1(a_t), \ldots, r_t^m(a_t)]$. For each objective $i \in [m]$, the empirical mean $\hat{\mu}^i(a_t)$ is updated using an incremental average defined in Eq. (3). The pull count $n(a_t)$ and the confidence width $c(a_t)$ are then updated by Eq. (4). The round index is

incremented, and the procedure repeats until the active set contains only a single arm.

The key innovation of LexElim-In is its *cross-objective elimination* strategy, which utilizes information from all objectives at each round. By jointly incorporating elimination evidence across objectives, LexElim-In more efficiently eliminates suboptimal arms, especially when lower-priority objectives provide stronger signals. This approach leads to faster identification of the lexicographic optimum compared to LexElim-Out, albeit at the cost of requiring the prior knowledge $\lambda$. Formal regret and sample complexity guarantees are presented in Theorems 3 and 4, respectively.

**Theorem 3** *Suppose that Eq.* (1) *and Eq.* (2) *hold. Define*

$$\Lambda^i(\lambda) = \sum_{j=0}^{i-1} \lambda^j, \text{ and } \gamma^i(\delta) = 64 \log\left(\frac{392Km}{(\Delta^i(a))^2 \cdot \delta}\right).$$

*With probability at least $1-\delta$, for any objective $i \in [m]$, the regret of LexElim-In satisfies*

$$R^i(t) \leq \sum_{\Delta^i(a)>0} \min_{j\in[m]} \left\{ \frac{(\Lambda^j(\lambda))^2 \cdot \Delta^i(a) \cdot \gamma^j(\delta)}{(\Delta^j(a))^2 \cdot \mathbb{I}(\Delta^j(a) > 0)} \right\}.$$

**Remark 3** For the primary objective ($i = 1$), the regret incurred due to $\Delta^1(a) > 0$ is bounded by

$$\min_{j\in[m]} \left\{ \frac{\Delta^1(a) \cdot (\Lambda^j(\lambda))^2}{(\Delta^j(a))^2 \cdot \mathbb{I}(\Delta^j(a) > 0)} \right\} \leq \frac{1}{\Delta^1(a)},$$

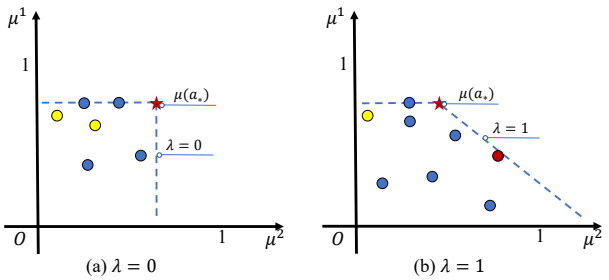
where the right-hand side matches the known lower bound (Lattimore 2018). The existence of $\min_{j\in[m]}$ allows the bound to go beyond the lower bound: if for some $j \geq 2$, the suboptimality gap $\Delta^j(a)$ is much larger than $\Delta^1(a) \cdot \Lambda^j(\lambda)$, the corresponding regret term can become significantly smaller than $1/\Delta^1(a)$. Thus, LexElim-In can adaptively exploit auxiliary objectives to accelerate learning.

Moreover, while the gap-dependent bound in Theorem 3 highlights how LexElim-In can exploit the relative gap structures among objectives to reduce regret, it remains essential to understand the algorithm's behavior in the worst case.

**Corollary 1** *Suppose the same conditions and notations as in Theorem 3. With probability at least $1-\delta$, for any objective $i \in [m]$, the regret of LexElim-In satisfies*

$$R^i(t) \leq \widetilde{O}\left(\Lambda^i(\lambda) \cdot \sqrt{Kt}\right).$$

Corollary 1 shows that for any objective $i \in [m]$, the worst-case regret of LexElim-In grows at most as $\widetilde{O}(\Lambda^i(\lambda)\sqrt{Kt})$. This matches the minimax bound $\widetilde{O}(\sqrt{Kt})$ of single-objective bandits (Degenne and Perchet 2016), up to the factor $\Lambda^i(\lambda)$. Hence, LexElim-In achieves minimax-optimal regret rates in terms of $K$ and $t$. Importantly, since $\Lambda^1(\lambda) = 1$, the regret for the highest-priority objective remains unaffected by the inclusion of lower-priority objectives, ensuring no performance degradation when optimizing multiple objectives simultaneously.



(a) $\lambda = 0$ (b) $\lambda = 1$

Figure 1: Cross-objective Acceleration

**Theorem 4** *Suppose the same conditions and notations as in Theorem 3. With probability at least $1-\delta$, for any objective $i \in [m]$, the number of samples required by LexElim-In to identify $\widetilde{\mathcal{O}}_*(i)$ satisfies*

$$\widetilde{T}^i(\delta) \leq \sum_{\Delta^i(a)>0} \min_{j\in[m]} \left\{ \frac{(\Lambda^j(\lambda))^2 \cdot \gamma^j(\delta)}{(\Delta^j(a))^2 \cdot \mathbb{I}(\Delta^j(a) > 0)} \right\}.$$

**Remark 4** Theorem 4 characterizes the sample complexity of LexElim-In for identifying the optimal arm set $\widetilde{\mathcal{O}}_*(i)$ for the $i$-th objective, revealing an objective-adaptive complexity. For each suboptimal arm $a$, the cost of distinguishing it is governed by the most distinguishable objective $j \in [m]$. In particular, if some objective $j$ exhibits a large suboptimality gap $\Delta^j(a)$ for a given arm $a$, that arm can often be eliminated early, without requiring extensive exploration of other objectives. In such case, LexElim-In adaptively leverages the reward structure across objectives to accelerate the identification process. Notably, in the single-objective setting, the lower bound on sample complexity is known to be $\Omega(\sum_{\Delta(a)>0} \frac{1}{(\Delta(a))^2})$ (Jamieson et al. 2014). Our bound recovers this result when $i = 1$, since $\Lambda^1(\lambda) = 1$, and the $\min_{j\in[m]}$ term ensures our result surpasses this lower bound.

**Cross-objective Acceleration.** Figure 1 illustrates how the second objective can accelerate BAI under varying degrees of trade-offs. The red star denotes the lex-optimal arm, while the circles represent suboptimal arms. In Figure 1(a), there is no conflict between other arms and the lex-optimal arm, resulting in $\lambda = 0$. The *two yellow arms* exhibit much larger reward gaps in the second objective than in the first, enabling LexElim-In to efficiently eliminate them by leveraging second objective information. Figure 1(b) shows a conflict between the lex-optimal arm and the red suboptimal arm, leading to $\lambda = 1$. In this case, only *the yellow arm that is far from the optimal arm* can be quickly eliminated, as the confidence term for the second objective is scaled by $2 + 4\lambda = 6$, as specified in Eq. (5).

## Experiments

In this section, we evaluate the empirical performance of our proposed algorithms, LexElim-Out and LexElim-In, on both RM and BAI tasks in lexicographic multi-objective bandits. Experiments are conducted on a Windows 10 laptop with Intel(R) Core(TM) i7-1170 CPU and 32GB memory.

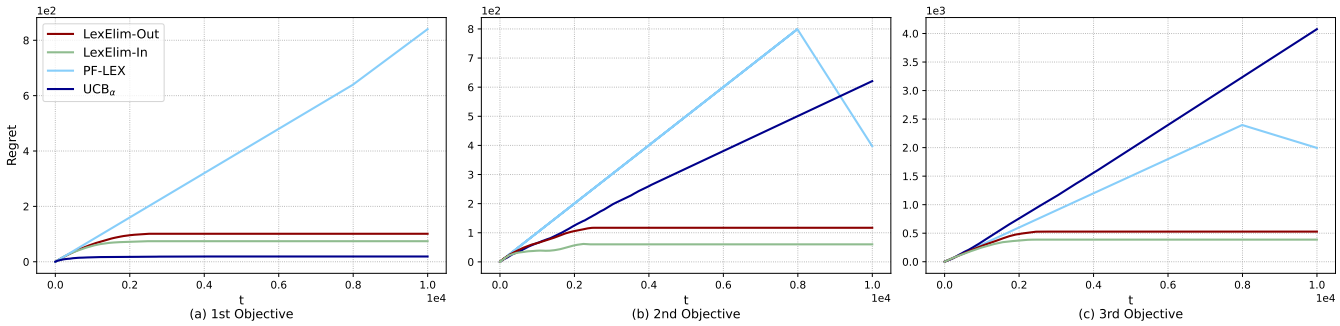



Figure 2: Regret Comparison of Our Algorithms versus PF-LEX and UCB$_\alpha$: $K = 10$

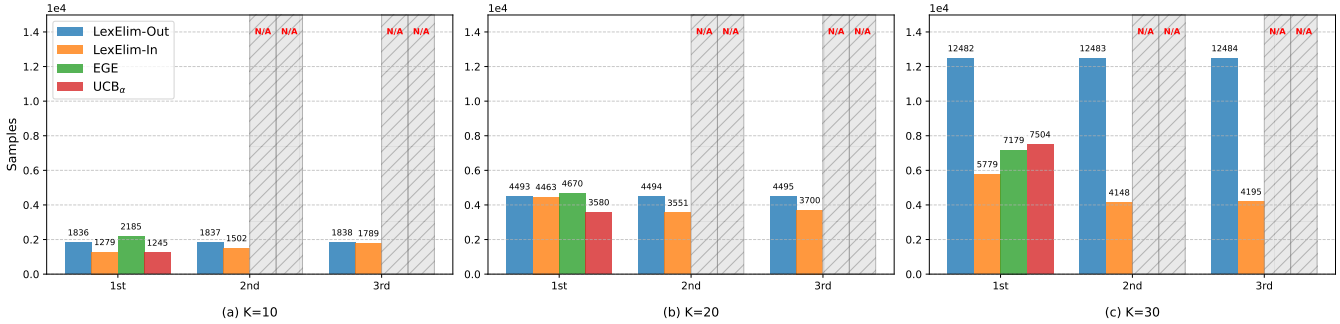



Figure 3: Sample Complexity Comparison of Our Algorithms versus EGE and UCB$_\alpha$

**Baselines.** There are three baselines. The first is EGE, which addresses BAI in single-objective MAB (Karnin, Koren, and Somekh 2013). The second is UCB$_\alpha$, designed to handle both BAI and RM in the single-objective MAB setting (Degenne et al. 2019). The third is PF-LEX, an algorithm tailored to lexicographic MAB, which focuses on the RM task (Hüyük and Tekin 2021).

**Experimental Setup.** We consider settings with $m = 3$. The expected rewards across the three objectives are defined as: $\mu^1(a) = 1 - \min_{p\in\{0.3,0.6,0.9\}} |a/K - p|$, $\mu^2(a) = 1 - 2 \times \min_{p\in\{0.5,0.8\}} |a/K - p|$, $\mu^3(a) = 1 - 2 \times |a/K - 0.5|$, $a \in [K]$. This construction ensures that multiple arms are optimal for the higher-priority objectives: $\{0.3K, 0.6K, 0.9K\}$ are optimal for the first objective, while $\{0.6K, 0.9K\}$ are optimal for both the first and second objectives. To identify the unique lex-optimal arm $a_* = 0.6K$, all three objectives must be considered. Stochastic rewards $r_t^i(a)$ are drawn from Gaussian distributions with mean $\mu^i(a)$ and variance 0.1. Each algorithm is run for 10 independent trials, and we report the average regret and sample complexity.

**RM Results.** For those RM algorithms (UCB$_\alpha$, PF-LEX, LexElim-Out, and LexElim-In), we fix $K = 10$ and run each algorithm for $T = 10{,}000$. Figure 2 presents the cumulative regret over time, where Panels (a), (b), and (c) correspond to objectives 1, 2, and 3, respectively. LexElim-Out and LexElim-In exhibit uniformly sublinear regret growth across all objectives, demonstrating their ability to optimize multiple objectives simultaneously. In contrast, UCB$_\alpha$ tailored for single-objective optimization, only achieves low regret for the first objective, while incurring linear regret on the second and third. Although PF-LEX is designed for multi-objective settings, it lacks theoretical guarantees under general regret metrics and suffers from a slower convergence rate, as reflected in its $\widetilde{O}(T^{2/3})$ regret bound.

**BAI Results.** For BAI algorithms (EGE, UCB$_\alpha$, LexElim-Out, and LexElim-In), we set the confidence level $\delta = 0.01$ and evaluate their performance under varying numbers of arms $K \in \{10, 20, 30\}$. The results are shown in Figure 3, where Panels (a) – (c) correspond to increasing $K$. All algorithms require more samples as $K$ increases, reflecting the greater difficulty of distinguishing between arms when reward gaps shrink. LexElim-In consistently outperforms the baselines, and its advantage becomes more significant with larger $K$. This is because LexElim-In exploits information from lower-priority objectives, which have larger reward gaps and provide stronger signals for elimination. In our setting, the reward gaps for the second and third objectives are twice as large as that of the first, allowing LexElim-In to identify the optimal arm more efficiently.

## Conclusion and Future work

This paper develops the first unified framework for simultaneously addressing both RM and BAI tasks in lexicographic multi-objective bandits. We propose two principled algorithms, LexElim-Out and LexElim-In, which adhere to the lexicographic preference structure while optimizing multiple objectives. LexElim-Out adopts a conservative elimination strategy that sequentially filters arms based on priority, ensuring no compromise on higher-priority objectives. LexElim-In exploits the joint reward signals across all objectives to perform more efficient arm elimination. We provide a comprehensive theoretical analysis for

both algorithms: LexElim-Out matches the known instance-dependent lower bounds for the primary objective, while LexElim-In achieves better instance-dependent bounds than classical single-objective methods.

An interesting direction for future work is to establish tighter lower bounds for lexicographic RM and BAI that explicitly capture the interactions among objectives. Additionally, eliminating the need for prior knowledge of the parameter $\lambda$ would further enhance the applicability of LexElim-In.

## 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.

Alkaabneh, F.; and Diabat, A. 2023. A multi-objective home healthcare delivery model and its solution using a branch-and-price algorithm and a two-stage meta-heuristic algorithm. *Transportation Research Part C: Emerging Technologies*, 147: 103838.

Alon, N.; Cesa-Bianchi, N.; Dekel, O.; and Koren, T. 2015. Online Learning with Feedback Graphs: Beyond Bandits. In *Proceedings of the 28th Conference on Learning Theory*, 23–35.

Audibert, J.-Y.; and Bubeck, S. 2009. Minimax policies for adversarial and stochastic bandits. In *Proceedings of the 22th annual conference on learning theory*, 217–226.

Audibert, J.-Y.; and Bubeck, S. 2010. Best Arm Identification in Multi-Armed Bandits. In *Proceedings of the 23rd Annual Conference on Learning Theory*, 41–53.

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

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47(2–3): 235–256.

Auer, P.; Chiang, C.-K.; Ortner, R.; and Drugan, M. 2016. Pareto Front Identification from Stochastic Bandit Feedback. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 939–947.

Awerbuch, B.; and Kleinberg, R. 2008. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1): 97–114.

Bubeck, S.; and Cesa-Bianchi, N. 2012. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5(1): 1–122.

Bubeck, S.; Munos, R.; and Stoltz, G. 2009. Pure exploration in multi-armed bandits problems. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory*, 23–37.

Carpentier, A.; and Locatelli, A. 2016. Tight (Lower) Bounds for the Fixed Budget Best Arm Identification Bandit Problem. In *Proceedings of the 29th Annual Conference on Learning Theory*, 590–604.

Chen, W.; Hu, W.; Li, F.; Li, J.; Liu, Y.; and Lu, P. 2016. Combinatorial Multi-Armed Bandit with General Reward Functions. In *Advances in Neural Information Processing Systems 29*, 1659–1667.

Crepon, E.; Garivier, A.; and M Koolen, W. 2024. Sequential learning of the Pareto front for multi-objective bandits. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, 3583–3591.

Dani, V.; Hayes, T. P.; and Kakade, S. M. 2008. Stochastic Linear Optimization under Bandit Feedback. In *Proceedings of the 21st Annual Conference on Learning*, 355–366.

Degenne, R.; Nedelec, T.; Calauzenes, C.; and Perchet, V. 2019. Bridging the gap between regret minimization and best arm identification, with application to A/B tests. In *Proceedings of the 32nd International Conference on Artificial Intelligence and Statistics*, 1988–1996.

Degenne, R.; and Perchet, V. 2016. Anytime optimal algorithms in stochastic multi-armed bandits. In *Proceedings of The 33rd International Conference on Machine Learning*, 1587–1595.

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.

Even-Dar, E.; Mannor, S.; and Mansour, Y. 2006. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *Journal of Machine Learning Research*, 7(39): 1079–1105.

Garivier, A.; and Kaufmann, E. 2016. Optimal Best Arm Identification with Fixed Confidence. In *Proceedings of the 29th Annual Conference on Learning Theory*, 998–1027.

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

Jamieson, K.; Malloy, M.; Nowak, R.; and Bubeck, S. 2014. lil' UCB : An Optimal Exploration Algorithm for Multi-Armed Bandits. In *Proceedings of The 27th Conference on Learning Theory*, 423–439.

Jin, T.; Yang, Y.; Tang, J.; Xiao, X.; and Xu, P. 2024. Optimal Batched Best Arm Identification. In *Advances in Neural Information Processing Systems 37*, 134947–134980.

Karnin, Z.; Koren, T.; and Somekh, O. 2013. Almost Optimal Exploration in Multi-Armed Bandits. In *Proceedings of the 30th International Conference on Machine Learning*, 1238–1246.

Kaufmann, E.; Cappé, O.; and Garivier, A. 2016. On the Complexity of Best-Arm Identification in Multi-Armed Bandit Models. *Journal of Machine Learning Research*, 17(1): 1–42.

Kone, C.; Kaufmann, E.; and Richert, L. 2024. Bandit Pareto Set Identification: the Fixed Budget Setting. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, 2548–2556.

Kurokawa, D.; Procaccia, A. D.; and Shah, N. 2018. Leximin Allocations in the Real World. *ACM Transactions on Economics and Computation*, 6(3–4): 1–24.

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

Lattimore, T. 2018. Refining the Confidence Level for Optimistic Bandit Strategies. *Journal of Machine Learning Research*, 19(20): 1–32.

Li, Y.; Chen, H.; Xu, S.; Ge, Y.; Tan, J.; Liu, S.; and Zhang, Y. 2023. Fairness in Recommendation: Foundations, Methods, and Applications. *ACM Transactions on Intelligent Systems and Technology*, 14(5): 1–48.

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

Lykouris, T.; Mirrokni, V.; and Paes Leme, R. 2018. Stochastic Bandits Robust to Adversarial Corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 114–122.

Q. Yahyaa, S.; M. Drugan, M.; and Manderick, B. 2015. Thompson Sampling in the Adaptive Linear Scalarized Multi Objective Multi Armed Bandit. In *International Conference on Agents and Artificial Intelligence*, 55–65.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5): 527–535.

Schwartz, E.; Bradlow, E.; and Fader, P. 2017. Customer Acquisition via Display Advertising Using Multi-Armed Bandit Experiments. *Marketing Science*, 36(2): 500–522.

Shu, T.; Shang, K.; Gong, C.; Nan, Y.; and Ishibuchi, H. 2024. Learning pareto set for multi-objective continuous robot control. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*, 4920 – 4928.

Tekin, C. 2019. The biobjective multiarmed bandit: learning approximate lexicographic optimal allocations. *Turkish Journal of Electrical Engineering and Computer Sciences*, 27(2): 1065–1080.

Tekin, C.; and Turgay, E. 2018. Multi-objective Contextual Multi-armed Bandit With a Dominant Objective. *IEEE Transactions on Signal Processing*, 66(14): 3799–3813.

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.

Wanigasekara, N.; Liang, Y.; Goh, S. T.; Liu, Y.; Williams, J. J.; and Rosenblum, D. S. 2019. Learning Multi-Objective Rewards and User Utility Function in Contextual Bandits for Personalized Ranking. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 3835–3841.

Xie, Y.; Shi, C.; Zhou, H.; Yang, Y.; Zhang, W.; Yu, Y.; and Li, L. 2021. MARS: Markov Molecular Sampling for Multi-objective Drug Discovery. In *International Conference on Learning Representations*.

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.; Cheng, J.; Liu, F.; Wang, Y.; and Zhang, Q. 2024. Multiobjective Lipschitz Bandits under Lexicographic Ordering. *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, 16238–16246.

Yang, J.; Tan, V. Y. F.; and Jin, T. 2024. Best Arm Identification with Minimal Regret. arXiv:2409.18909.

Zhang, Q.; and Ying, L. 2023. Fast and Regret Optimal Best Arm Identification: Fundamental Limits and Low-Complexity Algorithms. In *Advances in Neural Information Processing Systems 36*, 16729–16769.

Zhang, S.; Jia, F.; Wang, C.; and Wu, Q. 2023. Targeted Hyperparameter Optimization with Lexicographic Preferences Over Multiple Objectives. In *The 11th International Conference on Learning Representations*.

Zhong, Z.; Cheung, W. C.; and Tan, V. 2023. Achieving the Pareto Frontier of Regret Minimization and Best Arm Identification in Multi-Armed Bandits. *Transactions on Machine Learning Research*.

## Proof of Theorem 1

For clarity, throughout the proof in appendix, we use the notations $\hat{\mu}_t^i(a)$, $n_t(a)$, and $c_t(a)$ to denote the values of $\hat{\mu}^i(a)$, $n(a)$, and $c(a)$ at the beginning of round $t$, respectively.

We first present a high-probability confidence interval for the expected rewards of all objectives.

**Lemma 1** *With probability at least $1-\delta$, for any $t \geq 1$,*

$$\left|\hat{\mu}_t^i(a) - \mu^i(a)\right| \leq c_t(a) = \sqrt{\frac{4}{n(a_t)} \log\left(\frac{6Km \cdot n(a_t)}{\delta}\right)}, i \in [m], a \in [K].$$

This lemma provides a standard concentration inequality, bounding the deviation between the empirical and true rewards, and serves as a foundation for the subsequent analysis.

Let $\mathcal{E}$ denote the following high-probability event:

$$\mathcal{E} = \left\{\forall t \in [T],\ \forall a \in [K],\ \forall i \in [m] :\ \left|\hat{\mu}_t^i(a) - \mu^i(a)\right| \leq c_t(a)\right\}. \tag{6}$$

By the argument in Lemma 1, the event $\mathcal{E}$ holds with probability at least $1-\delta$.

Next, we present three technical lemmas that characterize how many times an arm can be pulled before it is eliminated. These lemmas serve as analytical tools to facilitate the regret analysis.

The first lemma provides a useful inequality for comparing logarithmic expressions, which will be instrumental in simplifying bounds on the number of arm pulls.

**Lemma 2** *Let $a > 0$, $b > 0$ and $ab > e$. If $x > a\log(ab)$, then $x > a\log(bx)$.*

The second lemma analyzes the behavior of the confidence radius function and shows that it decreases as the number of pulls increases.

**Lemma 3** *Let $f(n) = 4\sqrt{\frac{4}{n}\log\left(\frac{6Km \cdot n}{\delta}\right)}$ for $n > 0$. Then, $f(n)$ is strictly decreasing for all*

$$n > \frac{e\delta}{6Km}.$$

*In particular, since $\frac{e\delta}{6Km} \ll 1$ in typical applications, the function $f(n)$ is strictly decreasing for all $n \geq 1$.*

The detailed proofs of these three technical lemmas are deferred to the end of this appendix. The following lemma shows that the number of times any two active arms have been pulled remains nearly balanced throughout the execution of Algorithm 1.

**Lemma 4** *In Algorithm 1, for any arm $a_1, a_2 \in \mathcal{A}_t$, their pull counts satisfy:*

$$n_t(a_1) - 1 \leq n_t(a_2) \leq n_t(a_1) + 1.$$

**Proof.** We prove this lemma by induction. At the initialization step, all arms have $n_1(a) = 0$, and the condition holds trivially.

Suppose at round $t$, for all $a_1, a_2 \in \mathcal{A}_t$, the pull counts satisfy $|n_t(a_1) - n_t(a_2)| \leq 1$. Now consider how the pull counts change at round $t$. LexElim-Out selects the arm $a_t$ as:

$$a_t = \arg\max_{a \in \mathcal{A}_t} c(a),$$

where $c(a) = \sqrt{\frac{4}{n(a)}\log\left(\frac{6Km \cdot n(a)}{\delta}\right)}$ is a strictly decreasing function of $n(a)$ (cf. Lemma 3).

Thus, at each round, LexElim-Out chooses the arm with the *fewest number of pulls*. Let $n_{\min} := \min_{a \in \mathcal{A}_t} n_t(a)$. At round $t$, LexElim-Out chooses some arm $a_t$ with $n_t(a_t) = n_{\min}$ and increments its count:

$$n_{t+1}(a_t) = n_t(a_t) + 1, \quad \text{while for all } a \neq a_t,\ n_{t+1}(a) = n_t(a).$$

After this round, the former minimum becomes $n_{\min} + 1$. All arms now have pull counts either $n_{\min}$ or $n_{\min} + 1$. Therefore, all arms' pull counts differ by at most 1. This completes the induction. □

Equipped with the previous lemmas, we can now bound the number of times a suboptimal arm (with respect to the $i$-th objective) can be pulled before elimination.

**Lemma 5** *Suppose the event $\mathcal{E}$ in (6) holds. For each objective $i \in [m]$, define:*

- $\mathcal{O}_*(i-1) := \left\{a \in [K] \;\middle|\; \mu^j(a) = \mu^j(a_*) \text{ for all } j \in [i-1]\right\}$;
- $\mathcal{S}(i) := \left\{a \in \mathcal{O}_*(i-1) \;\middle|\; \Delta^i(a) > 0\right\}$.

*In Algorithm 1, for any arm $a \in \mathcal{S}(i)$, the number of times it is played is at most*

$$n_t(a) \leq \frac{64}{(\Delta^i(a))^2}\log\left(\frac{392Km}{(\Delta^i(a))^2 \cdot \delta}\right).$$

**Proof.** Let $a \in \mathcal{S}(i)$ and suppose it is eliminated at round $t$. By Algorithm 1, the elimination condition is:

$$\hat{\mu}_t^i(\hat{a}_t^i) - \hat{\mu}_t^i(a) > 2c_t(a_t) \quad \Longleftrightarrow \quad \hat{\mu}_t^i(\hat{a}_t^i) - c_t(a_t) > \hat{\mu}_t^i(a) + c_t(a_t),$$

where $a_t = \arg\max_{a \in \mathcal{A}_t} c_t(a)$ denotes the arm with the largest confidence width.

Since $\hat{a}_t^i = \operatorname{argmax}_{a \in \mathcal{A}_t} \hat{\mu}_t^i(\hat{a}_t^i)$ and $a_* \in \mathcal{A}_t$, a sufficient condition to eliminate $a$ is:

$$\hat{\mu}_t^i(a_*) - c_t(a_t) > \hat{\mu}_t^i(a) + c_t(a_t).$$

Using the confidence event in (6), we have that for all $t \geq 1$,

$$|\hat{\mu}_t^i(a) - \mu^i(a)| \leq c_t(a) \leq c_t(a_t), \quad \text{for all } i \in [m],\ a \in [K].$$

Thus, it follows another sufficient condition to eliminate $a$:

$$\mu^i(a_*) - 2c_t(a_t) > \mu^i(a) + 2c_t(a_t) \Leftrightarrow \Delta^i(a) = \mu^i(a_*) - \mu^i(a) > 4c_t(a_t).$$

Now recall that the confidence width is defined as

$$c_t(a_t) = \sqrt{\frac{4}{n_t(a_t)} \log\left(\frac{6Km \cdot n_t(a_t)}{\delta}\right)}.$$

Substituting into the inequality above, we obtain:

$$\Delta^i(a) > 4\sqrt{\frac{4}{n_t(a_t)} \log\left(\frac{6Km \cdot n_t(a_t)}{\delta}\right)}.$$

By Lemma 4, the number of pulls among arms in $\mathcal{A}_t$ differs by at most one, i.e., $n_t(a_t) \geq n_t(a) - 1$. Meanwhile, Lemma 4 tells that $c_t(\cdot)$ is decreasing with respect to $n_t(a_t)$. Thus, a sufficient condition for eliminating $a$ becomes:

$$\Delta^i(a) > 4\sqrt{\frac{4}{n_t(a) - 1} \log\left(\frac{6Km \cdot (n_t(a) - 1)}{\delta}\right)}.$$

Squaring both sides gives:

$$(\Delta^i(a))^2 > \frac{64}{n_t(a) - 1} \log\left(\frac{6Km \cdot (n_t(a) - 1)}{\delta}\right).$$

Rewriting this inequality yields:

$$n_t(a) - 1 > \frac{64}{(\Delta^i(a))^2} \log\left(\frac{6Km \cdot (n_t(a) - 1)}{\delta}\right). \tag{7}$$

To obtain an explicit upper bound on $n_t(a)$, we apply Lemma 2 with:

$$a = \frac{64}{(\Delta^i(a))^2}, \quad b = \frac{6Km}{\delta}, \quad x = n_t(a).$$

According to Lemma 2, if

$$n_t(a) > \frac{64}{(\Delta^i(a))^2} \log\left(\frac{384Km}{(\Delta^i(a))^2 \cdot \delta}\right) + 1,$$

then inequality (7) holds, which implies that arm $a$ will be eliminated at that point.

Therefore, the number of times arm $a$ is pulled is at most

$$n_t(a) \leq \frac{64}{(\Delta^i(a))^2} \log\left(\frac{392Km}{(\Delta^i(a))^2 \cdot \delta}\right),$$

which completes the proof of Lemma 5. □

We now complete the proof of Theorem 1. Recall that the regret for each objective arises only from suboptimal arms that are not eliminated early enough. For the first objective, only the arms in $\mathcal{S}(1)$ incur regret, and by Lemma 5, each such arm is played at most

$$\frac{64}{(\Delta^1(a))^2} \log\left(\frac{392Km}{(\Delta^1(a))^2 \cdot \delta}\right) = \frac{\gamma^1(\delta)}{(\Delta^1(a))^2}$$

times. Therefore, the total regret for the first objective is bounded by:

$$R^1(t) \leq \sum_{a \in \mathcal{S}(1)} \frac{\gamma^1(\delta)}{\Delta^1(a)}.$$

For the second objective, regret may arise from both $\mathcal{S}(1)$ and $\mathcal{S}(2)$. Any arm $a \in \mathcal{S}(1)$ may continue to be pulled before being eliminated, thereby contributing regret proportional to $\Delta^2(a)$. Its regret contribution is bounded by:

$$\frac{\gamma^1(\delta) \cdot \Delta^2(a)}{(\Delta^1(a))^2}.$$

Meanwhile, for arms $a \in \mathcal{S}(2)$, each is played at most

$$\frac{64}{(\Delta^2(a))^2} \log\left(\frac{392Km}{(\Delta^2(a))^2 \cdot \delta}\right) = \frac{\gamma^2(\delta)}{(\Delta^2(a))^2}$$

times, incurring regret at most $\frac{\gamma^2(\delta)}{\Delta^2(a)}$ each. Hence, the total regret for the second objective satisfies:

$$R^2(t) \leq \sum_{a \in \mathcal{S}(1)} \frac{\gamma^1(\delta) \cdot \Delta^2(a)}{(\Delta^1(a))^2} + \sum_{a \in \mathcal{S}(2)} \frac{\gamma^2(\delta)}{\Delta^2(a)}.$$

By the same reasoning, for the $i$-th objective ($i \in [m]$), regret may be contributed by all arms in $\mathcal{S}(1), \ldots, \mathcal{S}(i)$. Specifically, an arm $a \in \mathcal{S}(j)$ contributes regret to the $i$-th objective as long as it is not eliminated before stage $j$, and is pulled while optimizing objectives 1 through $j$. Each such arm contributes at most

$$\frac{64 \cdot \Delta^i(a)}{(\Delta^j(a))^2} \log\left(\frac{392Km}{(\Delta^j(a))^2 \cdot \delta}\right) = \frac{\gamma^j(\delta) \cdot \Delta^i(a)}{(\Delta^j(a))^2}$$

to the $i$-th objective's regret. Summing over all $j \leq i$ gives the bound:

$$R^i(t) \leq \sum_{j=1}^{i} \sum_{a \in \mathcal{S}(j)} \frac{\gamma^j(\delta) \cdot \Delta^i(a)}{(\Delta^j(a))^2}.$$

This completes the proof of Theorem 1. □

## Proof of Theorem 2

With Lemma 5 in hand, the proof of Theorem 2 follows directly. To eliminate any suboptimal arm $a \in \mathcal{S}(i)$, the algorithm requires at most

$$\frac{64}{(\Delta^i(a))^2} \log\left(\frac{392Km}{(\Delta^i(a))^2 \cdot \delta}\right) = \frac{\gamma^i(\delta)}{(\Delta^i(a))^2} \tag{8}$$

pulls.

To identify the set $\mathcal{O}_*(i)$, the set of arms that are optimal up to objective $i$, the algorithm must eliminate all arms in $\mathcal{S}(j)$ for every $j \leq i$. Therefore, the total number of samples required by LexElim-Out to identify $\mathcal{O}_*(i)$ is bounded by:

$$T^i(\delta) \leq \sum_{j=1}^{i} \sum_{a \in \mathcal{S}(j)} \frac{\gamma^j(\delta)}{(\Delta^j(a))^2}.$$

This concludes the proof of Theorem 2. □

## Proof of Theorem 3

To begin with, we prove that the lex-optimal arm $a_*$ is not eliminated during the Steps 6 to 9 in Algorithm 2.

**Lemma 6** *Suppose $\mathcal{E}$ in Eq.* (6) *holds. In Steps 6 to 9 of Algorithm 2, if $a_* \in \mathcal{A}_t^0$, then*

$$a_* \in \mathcal{A}_t^m \quad \textit{and} \quad \Delta^i(a) \leq 4(1 + \lambda + \cdots + \lambda^{i-1}) \cdot c_t(a_t), \quad \forall i \in [m],\ \forall a \in \mathcal{A}_t^m.$$

**Proof:** We prove the lemma via induction on the objective index $i \in [m]$.

**Base case ($i = 1$):** Since $\hat{a}_t^1 = \operatorname{argmax}_{a \in \mathcal{A}_t^0} \hat{\mu}_t^1(a)$ and $a_* \in \mathcal{A}_t^0$, for all $a \in \mathcal{A}_t^1$, we have

$$\Delta^1(a) = \mu^1(a_*) - \mu^1(a) \leq \mu^1(a_*) - \hat{\mu}_t^1(a_*) + \hat{\mu}_t^1(\hat{a}_t^1) - \mu^1(a). \tag{9}$$

Under event $\mathcal{E}$, it holds that

$$\mu^1(a_*) - \hat{\mu}_t^1(a_*) \leq c_t(a_*), \qquad \hat{\mu}_t^1(a) - \mu^1(a) \leq c_t(a), \quad \forall a \in \mathcal{A}_t^1.$$

Plugging these into Eq. (9), we obtain

$$\Delta^1(a) \le c_t(a_*) + \hat{\mu}_t^1(\hat{a}_t^1) - \hat{\mu}_t^1(a) + c_t(a), \quad \forall a \in \mathcal{A}_t^1.$$

By the elimination rule, for all $a \in \mathcal{A}_t^1$,

$$\hat{\mu}_t^1(\hat{a}_t^1) - \hat{\mu}_t^1(a) \le 2c_t(a_t).$$

Moreover, since $a_t = \operatorname{argmax}_{a \in \mathcal{A}_t^0} c_t(a)$, it holds that $c_t(a) \le c_t(a_t)$ and $c_t(a_*) \le c_t(a_t)$. Hence,

$$\Delta^1(a) \le c_t(a_t) + 2c_t(a_t) + c_t(a_t) = 4c_t(a_t), \quad \forall a \in \mathcal{A}_t^1.$$

Finally, since

$$\hat{\mu}_t^1(\hat{a}_t^1) - \hat{\mu}_t^1(a_*) \le \mu^1(\hat{a}_t^1) + c_t(\hat{a}_t^1) - \mu^1(a_*) + c_t(a_*) \le 2c_t(a_t),$$

we conclude that $a_* \in \mathcal{A}_t^1$.

**Inductive step:** Suppose that for all $j \le i-1$, it holds that $a_* \in \mathcal{A}_t^j$ and

$$\Delta^j(a) \le 4(1 + \lambda + \cdots + \lambda^{j-1}) \cdot c_t(a_t), \quad \forall a \in \mathcal{A}_t^j.$$

We now prove the statement for $j = i$. Since $\hat{a}_t^i = \operatorname{argmax}_{a \in \mathcal{A}_t^{i-1}} \hat{\mu}_t^i(a)$ and $a_* \in \mathcal{A}_t^{i-1}$, then for all $a \in \mathcal{A}_t^i \subseteq \mathcal{A}_t^{i-1}$,

$$\Delta^i(a) = \mu^i(a_*) - \mu^i(a) \le \mu^i(a_*) - \hat{\mu}_t^i(a_*) + \hat{\mu}_t^i(\hat{a}_t^i) - \mu^i(a). \tag{10}$$

By the event $\mathcal{E}$, we have

$$\mu^i(a_*) - \hat{\mu}_t^i(a_*) \le c_t(a_*), \qquad \hat{\mu}_t^i(a) - \mu^i(a) \le c_t(a), \quad \forall a \in \mathcal{A}_t^i. \tag{11}$$

Substituting into Eq. (10), we get

$$\Delta^i(a) \le c_t(a_*) + \hat{\mu}_t^i(\hat{a}_t^i) - \hat{\mu}_t^i(a) + c_t(a).$$

From the elimination rule in Algorithm 2, it follows that

$$\hat{\mu}_t^i(\hat{a}_t^i) - \hat{\mu}_t^i(a) \le (2 + 4\lambda + \cdots + 4\lambda^{i-1}) \cdot c_t(a_t), \quad \forall a \in \mathcal{A}_t^i.$$

Also, since $a_t = \operatorname{argmax}_{a \in \mathcal{A}_t^0} c_t(a)$, we have $c_t(a), c_t(a_*) \le c_t(a_t)$, thus

$$\Delta^i(a) \le 2c_t(a_t) + (2 + 4\lambda + \cdots + 4\lambda^{i-1}) \cdot c_t(a_t) = 4(1 + \lambda + \cdots + \lambda^{i-1}) \cdot c_t(a_t).$$

Next, we show $a_* \in \mathcal{A}_t^i$. By the same reasoning as above,

$$\hat{\mu}_t^i(\hat{a}_t^i) - \hat{\mu}_t^i(a_*) \le \mu^i(\hat{a}_t^i) + c_t(\hat{a}_t^i) - \mu^i(a_*) + c_t(a_*).$$

From the lexicographic trade-off in Eq. (2) and the inductive assumption,

$$\mu^i(\hat{a}_t^i) - \mu^i(a_*) \le \lambda \cdot \max_{j \in [i-1]} \{\mu^j(a_*) - \mu^j(\hat{a}_t^i)\} \le \lambda \cdot 4(1 + \lambda + \cdots + \lambda^{i-2}) \cdot c_t(a_t).$$

Using $c_t(\hat{a}_t^i), c_t(a_*) \le c_t(a_t)$, it follows that

$$\hat{\mu}_t^i(\hat{a}_t^i) - \hat{\mu}_t^i(a_*) \le 4\lambda(1 + \lambda + \cdots + \lambda^{i-2}) \cdot c_t(a_t) + 2c_t(a_t) = (2 + 4\lambda + \cdots + 4\lambda^{i-1}) \cdot c_t(a_t).$$

Thus, $a_* \in \mathcal{A}_t^i$. By induction, this holds for all $i \in [m]$. Therefore, we conclude

$$a_* \in \mathcal{A}_t^m, \quad \Delta^i(a) \le 4(1 + \lambda + \cdots + \lambda^{i-1}) \cdot c_t(a_t), \quad \forall i \in [m],\ \forall a \in \mathcal{A}_t^m.$$

This completes the proof. □

Then, we provide an upper bound on the number of times a suboptimal arm can be pulled in the LexElim-In algorithm.

**Lemma 7** *Suppose the event $\mathcal{E}$ in* (6) *holds. In Algorithm 2, for any arm $a \in [K]$, the number of times it is played is at most*

$$n_t(a) \le \min_{i \in [m]} \left\{ \frac{64(\Lambda^i(\lambda))^2}{(\Delta^i(a))^2 \cdot \mathbb{I}(\Delta^i(a) > 0)} \log\left( \frac{392Km}{(\Delta^i(a))^2 \cdot \delta} \right) \right\},$$

*where $\Lambda^i(\lambda) = 1 + \lambda + \cdots + \lambda^{i-1}$.*

**Proof.** Fix any arm $a \in [K]$. Let $i$ be an index such that $a$ is suboptimal with respect to the $i$-th objective, i.e., $\Delta^i(a) > 0$, and is eliminated based on the reward estimates of objective $i$ in some round $t$.

In Algorithm 2, an arm $a$ is removed from $\mathcal{A}_t^{i-1}$ according to objective $i$ if

$$\hat{\mu}_t^i(\hat{a}_t^i) - \hat{\mu}_t^i(a) > (2 + 4\lambda + \cdots + 4\lambda^{i-1}) \cdot c_t(a_t), \tag{12}$$

where $a_t = \operatorname{argmax}_{a \in \mathcal{A}_t} c_t(a)$ denotes the arm with the largest confidence width.

Since $\hat{a}_t^i = \operatorname{argmax}_{a \in \mathcal{A}_t^{i-1}} \hat{\mu}_t^i(a)$ and by Lemma 6 we know $a_* \in \mathcal{A}_t^{i-1}$, a sufficient condition for (12) is

$$\hat{\mu}_t^i(a_*) - \hat{\mu}_t^i(a) > (2 + 4\lambda + \cdots + 4\lambda^{i-1}) \cdot c_t(a_t). \tag{13}$$

Under the confidence event $\mathcal{E}$, for all $t$ and $a \in [K]$, we have

$$|\hat{\mu}_t^i(a) - \mu^i(a)| \le c_t(a) \le c_t(a_t). \tag{14}$$

Using (14), inequality (13) holds if

$$\mu^i(a_*) - \mu^i(a) > 4(1 + \lambda + \cdots + \lambda^{i-1}) \cdot c_t(a_t). \tag{15}$$

Define $\Lambda^i(\lambda) = 1 + \lambda + \cdots + \lambda^{i-1}$. Then, (15) becomes

$$\Delta^i(a) > 4\Lambda^i(\lambda) \cdot c_t(a_t). \tag{16}$$

Recall the form of the confidence radius:

$$c_t(a_t) = \sqrt{\frac{4}{n_t(a_t)} \log\left(\frac{6Km \cdot n_t(a_t)}{\delta}\right)}. \tag{17}$$

Combining (16) and (17), we obtain:

$$\Delta^i(a) > 4\Lambda^i(\lambda) \cdot \sqrt{\frac{4}{n_t(a_t)} \log\left(\frac{6Km \cdot n_t(a_t)}{\delta}\right)}. \tag{18}$$

By Lemma 3 and Lemma 4, $c_t(\cdot)$ is decreasing in $n_t$ and $n_t(a_t) \ge n_t(a) - 1$, (18) still holds if we replace $n_t(a_t)$ with $n_t(a) - 1$:

$$\Delta^i(a) > 2\Lambda^i(\lambda) \cdot \sqrt{\frac{4}{n_t(a) - 1} \log\left(\frac{6Km \cdot (n_t(a) - 1)}{\delta}\right)}.$$

Squaring both sides yields:

$$(\Delta^i(a))^2 > \frac{16(\Lambda^i(\lambda))^2}{n_t(a) - 1} \log\left(\frac{6Km \cdot (n_t(a) - 1)}{\delta}\right).$$

Rewriting this inequality gives:

$$n_t(a) - 1 > \frac{16(\Lambda^i(\lambda))^2}{(\Delta^i(a))^2} \log\left(\frac{6Km \cdot (n_t(a) - 1)}{\delta}\right). \tag{19}$$

To get an explicit bound, apply Lemma 2 with

$$a = \frac{16(\Lambda^i(\lambda))^2}{(\Delta^i(a))^2}, \quad b = \frac{6Km}{\delta}, \quad x = n_t(a).$$

According to Lemma 2, inequality (19) holds if

$$n_t(a) > \frac{16(\Lambda^i(\lambda))^2}{(\Delta^i(a))^2} \log\left(\frac{384Km}{(\Delta^i(a))^2 \cdot \delta}\right) + 1.$$

Therefore, the number of times arm $a$ is played is at most

$$n_t(a) \le \frac{16(\Lambda^i(\lambda))^2}{(\Delta^i(a))^2} \log\left(\frac{392Km}{(\Delta^i(a))^2 \cdot \delta}\right).$$

Since this holds for every $i \in [m]$ with $\Delta^i(a) > 0$, we obtain

$$n_t(a) \le \min_{i \in [m]} \left\{ \frac{64(\Lambda^i(\lambda))^2}{(\Delta^i(a))^2 \cdot \mathbb{I}(\Delta^i(a) > 0)} \log\left(\frac{392Km}{(\Delta^i(a))^2 \cdot \delta}\right) \right\}.$$

This concludes the proof of Lemma 7. □

We now complete the proof of Theorem 3. Recall that for each objective $i \in [m]$, he regret arises solely from the suboptimal arms with $\Delta^i(a) > 0$. The contribution of each such arm $a$ to the regret is given by $\Delta^i(a) \cdot n_t(a)$. Therefore, the cumulative regret for the $i$-th objective can be bounded as follows:

$$R^i(t) = \sum_{\Delta^i(a) > 0} \Delta^i(a) \cdot n_t(a) \le \sum_{\Delta^i(a) > 0} \min_{j \in [m]} \left\{ \frac{64(\Lambda^j(\lambda))^2 \Delta^i(a)}{(\Delta^j(a))^2 \cdot \mathbb{I}(\Delta^j(a) > 0)} \log\left(\frac{392Km}{(\Delta^j(a))^2 \cdot \delta}\right) \right\}.$$

Finally, noting that $\gamma^j(\delta) = 64 \log\left(\frac{392Km}{(\Delta^j(a))^2 \cdot \delta}\right)$, the proof is finished. □

## Proof of Corollary 1

From Lemma 6, we know that for any arm $a \in \mathcal{A}_t^m$, the suboptimality gap satisfies

$$\Delta^i(a) \leq 4(1 + \lambda + \cdots + \lambda^{i-1}) \cdot c_t(a_t), \quad \forall i \in [m].$$

Define the scaling factor $\Lambda^i(\lambda) = 1 + \lambda + \cdots + \lambda^{i-1}$. Since $a_t \in \mathcal{A}_{t-1}^m$, it follows that

$$\Delta^i(a_t) \leq 4\Lambda^i(\lambda) \cdot c_{t-1}(a_{t-1}).$$

By the definition of regret, we have

$$R^i(t) = \sum_{\tau=1}^{t} \Delta^i(a_\tau) \leq \sum_{\tau=1}^{t} 4\Lambda^i(\lambda) \cdot c_{\tau-1}(a_{\tau-1}). \tag{20}$$

Recall that the confidence radius is defined as

$$c_t(a_t) = \sqrt{\frac{4}{n_t(a_t)} \log\left(\frac{6Km \cdot n_t(a_t)}{\delta}\right)}.$$

Substituting the definition of the confidence radius $c_{\tau-1}(a_{\tau-1})$ into Eq. (20), we obtain:

$$R^i(t) \leq \sum_{\tau=1}^{t} 4\Lambda^i(\lambda) \sqrt{\frac{4}{n_{\tau-1}(a_{\tau-1})} \log\left(\frac{6Kmt}{\delta}\right)}.$$

We regroup the terms by arm $a \in [K]$ and the number of times each arm has been pulled up to round $t$:

$$R^i(t) \leq \sum_{a \in [K]} \sum_{n=1}^{n_{t-1}(a)} 4\Lambda^i(\lambda) \sqrt{\frac{4}{n} \log\left(\frac{6Kmt}{\delta}\right)}. \tag{21}$$

Using the standard inequality

$$\sum_{n=1}^{N} \frac{1}{\sqrt{n}} \leq 2\sqrt{N},$$

we upper-bound the inner sum of Eq. (21) as

$$R^i(t) \leq \sum_{a \in [K]} 8\Lambda^i(\lambda) \cdot \sqrt{4 \cdot n_{t-1}(a) \cdot \log\left(\frac{6Kmt}{\delta}\right)}.$$

Simplifying constants, we arrive at,

$$R^i(t) \leq \sum_{a \in [K]} 16\Lambda^i(\lambda) \sqrt{n_{t-1}(a) \cdot \log\left(\frac{6Kmt}{\delta}\right)}.$$

Finally, applying Jensen's inequality (or concavity of the square root), we bound the total sum

$$\sum_{a \in [K]} \sqrt{n_{t-1}(a)} \leq \sqrt{K \cdot \sum_{a} n_{t-1}(a)} \leq \sqrt{Kt}.$$

Therefore, the regret is bounded as

$$R^i(t) \leq 16\Lambda^i(\lambda) \cdot \sqrt{Kt \cdot \log\left(\frac{6Kmt}{\delta}\right)} = \widetilde{O}(\Lambda^i(\lambda) \cdot \sqrt{Kt}).$$

The proof of Corollary 1 is finished. □

## Proof of Theorem 4

With Lemma 7 in hand, the proof of Theorem 4 follows directly. To eliminate any suboptimal arm $a$ that $\Delta^i(a) > 0$, the algorithm requires at most

$$\min_{j\in[m]} \left\{ \frac{64(\Lambda^j(\lambda))^2}{(\Delta^j(a))^2 \cdot \mathbb{I}(\Delta^j(a) > 0)} \log\left(\frac{392Km}{(\Delta^j(a))^2 \cdot \delta}\right) \right\}$$

pulls.

To identify the set $\widetilde{\mathcal{O}}_*(i) = \{a \in [K] \mid \Delta^i(a) \le 0\}$, the set of arms that are optimal up to objective $i$, the algorithm must eliminate all arms $\Delta^i(a) > 0$. Therefore, the total number of samples required by LexElim-In to identify $\widetilde{\mathcal{O}}_*(i)$ is bounded by:

$$\widetilde{T}^i(\delta) \le \sum_{\Delta^i(a)>0} \min_{j\in[m]} \left\{ \frac{(\Lambda^j(\lambda))^2 \cdot \gamma^j(\delta)}{(\Delta^j(a))^2 \cdot \mathbb{I}(\Delta^j(a) > 0)} \right\}, \gamma^j(\delta) = 64 \log\left(\frac{392Km}{(\Delta^j(a))^2 \cdot \delta}\right).$$

This concludes the proof. □

## Proof of Technical Lemmas

**Lemma 1** *With probability at least* $1-\delta$*, for any* $t \ge 1$*,*

$$\left|\hat{\mu}_t^i(a) - \mu^i(a)\right| \le c_t(a), i \in [m], a \in [K].$$

**Proof.** If $n_t(a) = 0$, then by definition $c_t(a) = +\infty$, the inequality holds trivially. We therefore consider the case $n_t(a) \ge 1$.

Fix any objective $i \in [m]$, according to Lemma 6 of Abbasi-yadkori, Pál, and Szepesvári (2011), we have that with probability at least $1-\delta$, for any $t \ge 1$ and any arm $a \in [K]$, the empirical mean satisfies:

$$\left| \frac{1}{n_t(a)} \sum_{\tau=1}^{t-1} r_\tau^i(a_\tau)\mathbb{I}(a_\tau = a) - \mu^i(a) \right| \le \sqrt{\left(1 + 2\log\left(\frac{K\sqrt{1+n_t(a)}}{\delta}\right)\right)\frac{1+n_t(a)}{n_t^2(a)}}.$$

Noting that $\hat{\mu}_t^i(a) = \frac{1}{n_t(a)} \sum_{\tau=1}^{t-1} r_\tau^i(a_\tau) \cdot \mathbb{I}(a_\tau = a)$, the above bound directly applies to $\left|\hat{\mu}_t^i(a) - \mu^i(a)\right|$.

Applying a union bound over all $m$ objectives, and replacing $\delta$ with $\delta/m$, we get that with probability at least $1-\delta$, for all $i \in [m]$, $a \in [K]$, and $t \ge 1$,

$$\left|\hat{\mu}_t^i(a) - \mu^i(a)\right| \le \sqrt{\left(1 + 2\log\left(\frac{Km\sqrt{1+n_t(a)}}{\delta}\right)\right)\frac{1+n_t(a)}{n_t^2(a)}}. \tag{22}$$

Using the inequality $\log(Km \cdot \sqrt{e} \cdot \sqrt{1+n_t(a)}/\delta) \le \log(6Km \cdot n_t(a)/\delta)$ for $n_t(a) \ge 1$, we can further relax the bound in Eq. (22) to:

$$\left|\hat{\mu}_t^i(a) - \mu^i(a)\right| \le \sqrt{\frac{4}{n_t(a)} \log\left(\frac{6Km \cdot n_t(a)}{\delta}\right)} =: c_t(a).$$

This completes the proof. □

**Lemma 2** *Let* $a > 0$*,* $b > 0$ *and* $ab > e$*. If* $x > a\log(ab)$*, then* $x > a\log(bx)$*.*

**Proof.** Define the function $f(x) = x - a\log x$. We aim to find a value $x_0$ such that $f(x_0) > a\log b$, which implies

$$x_0 - a\log x_0 > a\log b \quad \Leftrightarrow \quad x_0 > a\log(bx_0).$$

First, observe that $f(x)$ is differentiable and its derivative is given by

$$f'(x) = 1 - \frac{a}{x}.$$

Thus, $f(x)$ is strictly increasing for all $x > a$.

Now, let us consider $x_0 = a\log(ab)$. Note that $\log(ab) = \log a + \log b$, and so $x_0 = a(\log a + \log b)$. We compute

$$a\log(bx_0) = a\log\left(ba\log(ab)\right) = a(\log a + \log b + \log\log(ab)).$$

Since $\log\log(ab) < \log(ab)$ for all $ab > e$, it follows that

$$x_0 = a\log(ab) > a\log(bx_0).$$

Hence, $x_0$ satisfies the inequality, and due to the monotonicity of $f(x)$ for $x > a$, any $x > x_0$ also satisfies

$$x > a\log(bx).$$

The proof is finished. □

**Lemma 3** *Let $f(n) = 4\sqrt{\frac{4}{n}\log\left(\frac{6Km\cdot n}{\delta}\right)}$ for $n > 0$. Then, $f(n)$ is strictly decreasing for all*

$$n > \frac{e\delta}{6Km}.$$

*In particular, since $\frac{e\delta}{6Km} \ll 1$ in typical applications, the function $f(n)$ is strictly decreasing for all $n \geq 1$.*

**Proof.** Let $C = \frac{6Km}{\delta}$, so that the function becomes:

$$f(n) = 8\sqrt{\frac{\log(Cn)}{n}}.$$

Define the inner function $h(n) = \frac{\log(Cn)}{n}$, so that $f(n) = 8\sqrt{h(n)}$. It suffices to show that $h(n)$ is strictly decreasing. Taking the derivative:

$$h'(n) = \frac{1 - \log(Cn)}{n^2}.$$

Hence, $h'(n) < 0$ if and only if $\log(Cn) > 1$, which is equivalent to $Cn > e$. Therefore, $f(n)$ is strictly decreasing for all $n > \frac{e}{C} = \frac{e\delta}{6Km}$, as claimed. □