Cascading Bandits: Minimizing Regret with Respect to the Optimal Ordering

Anonymous authors
Paper under double-blind review

Abstract

We study a variant of cascading bandits, motivated by applications such as cognitive radios and carousel-based recommendation systems, where early successes reduce sensing costs or enhance user experience. In this setting, a *learner* sequentially interacts with a set of K arms over T rounds, where each arm yields a binary reward governed by an unknown success probability. In each round, the learner selects an ordering of the arms and pulls them sequentially until the first success is observed or all the arms are exhausted. The objective is to minimize the cumulative number of arm pulls required to find a success. We define regret as the expected number of additional pulls compared to an optimal ordering of the arms. To minimize regret, we propose two algorithms, *O-UCB* and *O-Greedy*, which extend the classical UCB and Greedy algorithms to our setting. For *O-UCB*, we derive two regret bounds with respect to the time horizon T: a logarithmic bound of $O(\log T)$ that depends on suboptimality gaps, and a constant bound of $O(1)$ that additionally depends on a problem-dependent parameter p_{all}, which quantifies the extent of inherent exploration. Furthermore, we show that *O-Greedy*, despite lacking explicit exploration, also achieves $O(1)$ regret under the same conditions. Extensive experiments on both synthetic and real-world datasets corroborate our theoretical results and delineate the regimes where each algorithm is most effective.

1 Introduction

The cascading-bandit model serves as a fundamental framework for online learning in structured decision-making, particularly in ranked retrieval systems such as web search and recommender systems. It extends the classical multi-armed bandit (MAB) paradigm by incorporating sequential interactions over a ranked list of items (Kveton et al., 2015a). In each round, the learner selects an ordered list of arms that are examined sequentially from the top until the first successful outcome (e.g., a click) occurs or the list is exhausted. Crucially, the learner observes the outcomes only up to this first success, leaving the remaining arms unobserved. This *partial and position-dependent feedback* introduces a distinctive exploration–exploitation trade-off that fundamentally shapes both algorithm design and theoretical analysis (Kiyohara et al., 2022).

Most existing works on cascading bandits assume large item sets where only a subset can be presented to the user in each round. The primary focus has been on selecting the optimal subset of items to maximize the likelihood of success, such as clicks in web search or ad engagement (Kveton et al., 2015b; Li et al., 2016). In contrast, our work addresses a distinct yet practically relevant setting where the number of items is small enough that all can be examined. In this scenario, the challenge shifts from item selection to optimizing the order in which items are presented to minimize the number of pulls required for success—thereby enhancing user experience or reducing sensing costs.

Such ordering problems arise naturally in domains where the item sets are small and all items can be sequentially probed at a cost. Representative examples of such ordering problems include:

- **Cognitive Radio Networks:** In dynamic spectrum access, secondary users sequentially sense a limited number of channels to find an idle one for transmission. Each sensing action incurs energy or

latency cost, and the process continues until an idle channel is found or all channels are exhausted. Prior work (e.g., (Raj et al., 2018)) emphasizes this behavior: "The sensing operation is repeated until we find a channel that is idle or we exhaust all available channels." Optimizing the sensing order can significantly reduce the total cost—an aspect overlooked by classical cascading models focused solely on selection.

- **Carousel-Based Recommendation Systems:** Platforms like Netflix or Spotify present content in a small number of curated carousels (e.g., *New Releases*, *Action Movies*), where the items within each carousel are preselected—either by algorithms or human editors. Thus, the system's decision lies in determining the order of the carousels rather than selecting their content. Users typically browse the carousels sequentially and stop at the first one they find relevant. As observed by Rahdari et al. (2022), "the user examines all carousels, from top to bottom, until they stop at a relevant carousel." In such cases, optimizing the display order reduces user effort (searching time) and improves engagement—directly aligning with our model.

These examples demonstrate that settings where all items can be presented and where the goal is to minimize the number of pulls until success arise naturally in various domains. Despite this, most prior work on cascading bandits has focused on subset selection, with limited attention to the role of ordering. Our work addresses this gap by formalizing the problem of learning an optimal ordering under partial feedback, and by proposing algorithms that minimize the expected number of pulls required to observe a success.

1.1 Key Contributions

1. We introduce a novel variant of the cascading bandit problem in which all items can be examined in each round, and the objective is to minimize the cumulative number of pulls until a success occurs. This shifts the focus from item selection to learning the optimal ordering, with applications in cognitive radio and curated recommendation systems.

2. We propose two algorithms, *O-UCB* (Order-based Upper Confidence Bound) and *O-Greedy* (Order-based Greedy). For *O-UCB*, we establish two regret bounds with respect to the time horizon T: a logarithmic bound of $O(\log T)$ that depends on the suboptimality gaps, and a constant bound of $O(1)$ that additionally depends on an inherent exploration parameter p_{all}, which quantifies the probability that all arms in the cascade are explored. Despite the absence of explicit exploration, the *O-Greedy* algorithm leverages the inherent exploration in the problem setup and achieves constant regret, which depends on p_{all} and the suboptimality gaps.

3. Through extensive simulations on both synthetic and real-world data sets, we evaluate the performance of these algorithms. Our results show that *O-Greedy* outperforms *O-UCB* in environments with high inherent exploration, whereas *O-UCB* is observed to be more robust across different levels of inherent exploration. We further benchmark their performance against standard baselines, including Thompson Sampling and ε-greedy variants adapted to our setting.

1.2 Related Work

The classical stochastic MAB problem, where a learner pulls a single arm per round to maximize cumulative reward, has been extensively studied; see Lattimore & Szepesvári (2020); Slivkins (2019) for comprehensive overviews. Several extensions generalize this classical setting to structured or multiple-action environments, such as multiple-play bandits (Anantharam et al., 1987), combinatorial bandits (Cesa-Bianchi & Lugosi, 2012; Chen et al., 2013), and batched exploration (Gao et al., 2019). These models typically assume full feedback for all selected arms and primarily focus on cumulative reward maximization.

The cascading bandit framework, introduced by Kveton et al. (2015a), captures partial feedback by modeling sequential user examination with stopping after the first success. Most subsequent works assume large item sets and focus on selecting a subset to maximize success probability (Kveton et al., 2015b; Zhong et al., 2021; Vial et al., 2022; Kiyohara et al., 2022). In contrast, we consider the setting where the item set is

small enough to allow examining all arms, and the goal is to minimize the expected number of pulls required before observing a success.

Our setting also differs from ranked bandits (Radlinski et al., 2008), where item orderings are learned to maximize binary click feedback, often allowing multiple successes per round and position-dependent click probabilities. In our model, the success probabilities are position-independent, and the feedback terminates at the first success. Multi-slot sponsored search models share structural similarities but typically aim to maximize aggregate reward across positions, rather than stopping early.

Cost-aware bandit settings have also been studied, where only a subset of arms is pulled based on reward-cost trade-offs (e.g., Gan et al., 2020), even at the possibility of missing a success. In contrast, our setting assumes that all arms can be examined, and focuses on learning an optimal ordering to minimize the number of pulls before success.

To our knowledge, this is the first work to formalize the learning problem of ordering a small set of arms in a pull-until-success cascading bandit setting. Our regret analysis highlights the role of inherent exploration in the environment and establishes both logarithmic and constant bounds under different structural conditions.

The paper is organized as follows: Section 2 presents the problem setup; Section 3 introduces the *O-UCB* and *O-Greedy* algorithms along with their performance guarantees; Section 4 presents numerical simulations that validate our theoretical findings; Section 5 provides regret analysis with proof sketches; and Section 6 concludes. Full proofs and additional numerical results are provided in the appendix.

2 Problem Setup

We consider a *cascading bandit problem* that involves a *learner* interacting with a set of K arms over a time horizon of T rounds, where T is typically assumed to be much larger than K. In each round, every arm independently generates a stochastic reward drawn from an unknown probability distribution. More specifically, we assume that the reward generated by arm i follows a Bernoulli distribution with an unknown success probability p_i. That is, the (independent) reward associated with arm i in any given round is a binary random variable that takes the value 1 with probability p_i and 0 otherwise. The vector of success probabilities for all arms is denoted by $\nu = [p_1, p_2, \ldots, p_K]$, which characterizes the underlying problem instance.

An arm is considered *successful* in a given round if it generates a reward of 1. The learner's objective in each round is to identify at least one successful arm, provided that such a successful arm exists in that round. In order to identify a successful arm, the learner determines an ordering of the arms in each round based on observations from the previous rounds and sequentially pulls the arms in the chosen order until one of the following conditions is met:

(i) A successful arm is pulled,

(ii) All K arms have been pulled.

The objective is to design an algorithm that sequentially determines the ordering of the arms in each round, leveraging observations from previous rounds, to minimize the total number of arm pulls incurred over T rounds.

It is important to note that the learner only observes the outcomes of the arms pulled up to the first success. Since the remaining arms in the chosen order are not pulled once a success is observed, their outcomes remain unobserved. This partial feedback introduces a fundamental exploration-exploitation trade-off, which plays a crucial role in shaping the algorithm's design and analysis.

In order to minimize the cumulative number of arm pulls, an algorithm chooses an ordering of the arms in each round t based on observations from the previous rounds. Consider an algorithm $\pi = [\boldsymbol{\sigma}(t)]_{t=1}^{T}$, where $\boldsymbol{\sigma}(t) := [\sigma_1(t), \sigma_2(t), \ldots, \sigma_K(t)]$ represents the ordering of the arms chosen by the algorithm in round t, and $\sigma_i(t)$ denotes the ith arm to be pulled in round t. Let $N(\boldsymbol{\sigma})$ denote the number of arms pulled in a round for

a chosen order $\boldsymbol{\sigma}$, and let $\mu(\boldsymbol{\sigma}) = \mathbb{E}_\nu[N(\boldsymbol{\sigma})]$, where the expectation is taken with respect to the randomness induced by the problem instance ν. Therefore, the algorithm π's expected cumulative number of arm pulls is given by $\sum_{t=1}^{T} \mathbb{E}_{\nu,\pi}[N(\boldsymbol{\sigma}(t))]$, where the expectation is taken with respect to the randomness induced by both the problem instance ν and the algorithm π.

To evaluate an algorithm's performance, we compare its expected cumulative number of arm pulls with that of an optimal algorithm π^*, which pulls the arms according to a fixed order $\boldsymbol{\sigma}^*$, arranged in non-increasing order of their success probabilities p_i. Thus, $\boldsymbol{\sigma}(t) = \boldsymbol{\sigma}^*$ for all t, and the expected cumulative number of arm pulls incurred by π^* is $\sum_{t=1}^{T} \mathbb{E}_\nu[N(\boldsymbol{\sigma}^*)]$. Therefore, the expected cumulative regret of an algorithm π for a problem instance ν is

$$R_{\nu,\pi}(T) = \sum_{t=1}^{T} \mathbb{E}_{\nu,\pi}[N(\boldsymbol{\sigma}(t))] - \sum_{t=1}^{T} \mathbb{E}_\nu[N(\boldsymbol{\sigma}^*)].$$

The cumulative regret can be equivalently expressed in the following useful form:

$$R_{\nu,\pi}(T) = \mathbb{E}_{\nu,\pi}\left[\sum_{t=1}^{T} \Delta(\boldsymbol{\sigma}(t))\right],$$

where $\Delta(\boldsymbol{\sigma}) := \mu(\boldsymbol{\sigma}) - \mu(\boldsymbol{\sigma}^*)$ denotes the suboptimality gap of the ordering $\boldsymbol{\sigma}$, defined as the difference in the expected number of arm pulls between $\boldsymbol{\sigma}$ and the optimal ordering $\boldsymbol{\sigma}^*$.

Given the inherent exploration-exploitation trade-off introduced by partial feedback, our aim is to design an algorithm π that minimizes the cumulative regret $R_{\nu,\pi}(T)$ with respect to the optimal policy. To this end, we propose two algorithms: the *O-UCB* algorithm and the *O-Greedy* algorithm, and analyze their regret performance.

3 Proposed Algorithms

In this section, we introduce our proposed algorithms along with their theoretical performance guarantees. The first algorithm, termed the *O-UCB* algorithm, is an adaptation of the classic UCB algorithm (Auer et al., 2002), originally designed for regret minimization in MAB problems. It balances both exploration and exploitation. The second algorithm, referred to as the *O-Greedy* algorithm, is a greedy strategy that prioritizes exploitation based on empirical success probabilities.

3.1 *O-UCB* algorithm

During the initial K rounds, the *O-UCB* algorithm selects orderings that ensure each arm is pulled in at least one round. For all subsequent rounds $t > K$, the algorithm computes the UCB score for each arm based on its empirical mean of observed rewards and the corresponding confidence interval. The arms are then ordered in decreasing order of their UCB scores. In each round, the algorithm sequentially pulls the arms according to this ordering, terminating the sequence either upon observing the first success or after all K arms have been pulled without success. The detailed procedure of the algorithm is presented in Algorithm 0. We use the following notations in our algorithms: Let $\bar{p}_t(a_i)$ denote the empirical success probability of arm a_i based on observations up to round t, and let $n_t(a_i)$ denote the number of times arm a_i has been pulled up to round t.

We now proceed to analyze the performance of the *O-UCB* algorithm. Without loss of generality, we assume that the arms are indexed such that $p_1 \geq p_2 \geq \cdots \geq p_K$, where p_i denotes the success probability of arm i. Let $\Delta_{i,j} := p_i - p_j$ denote the gap between the success probabilities of arms i and j.

We derive two regret bounds for the *O-UCB* algorithm: (i) a logarithmic bound in T that depends on the suboptimality gaps; and (ii) a constant bound, independent of T, that depends on both the suboptimality gaps and an inherent exploration parameter p_{all}, which will be defined later in (1). We first present the regret bound that grows logarithmically with T.

Algorithm 1 *O-UCB* algorithm

Input: Total rounds T, Number of arms K
Initialization: For each arm a_i, set $\bar{p}_0(a_i) = 0$ and $n_0(a_i) = 0$
for $t = 1$ to K **do**
    Choose $\boldsymbol{\sigma}(t) = [\sigma_1(t), \ldots, \sigma_K(t)]$, where

$$\sigma_i(t) = ((t + i - 1) \mod K) + 1$$

end for
for $t = K + 1$ to T **do**
    Step 1: Compute the UCB scores for all the arms based on the observations up to round $t - 1$:

$$\mathrm{UCB}_{t-1}(a_i) = \bar{p}_{t-1}(a_i) + \sqrt{\frac{3 \ln t}{2\, n_{t-1}(a_i)}}$$

    Step 2: Compute $\boldsymbol{\sigma}(t)$ by sorting the arms in the decreasing order of their UCB scores.
    Step 3: Play the arms sequentially in the order given by $\boldsymbol{\sigma}(t)$ until a success is observed or all the K arms are played out.
    Step 4: Update $\bar{p}_t(a_i)$ and $n_t(a_i)$ based on observed outcomes.
end for

$O(\ln T)$ **Regret Bound:**

Theorem 1. *The expected cumulative regret of the O-UCB algorithm, denoted by $R_{\nu,u}(T)$, satisfies*

$$R_{\nu,u}(T) \leq \sum_{j=2}^{K} \sum_{i=1}^{j-1} (K - i)\Delta_{i,j}(n_{i,j} - n_{i-1,j}) + \frac{K^2\pi^2}{3} + K^2,$$

where

$$n_{i,j} := \begin{cases} \frac{6 \ln T}{\Delta_{i,j}^2}, & 1 \leq i < j \leq K, \\ 0, & i = 0. \end{cases}$$

This result establishes an $O(\ln T)$ upper bound on the regret of the *O-UCB* algorithm, highlighting its logarithmic growth with time. Unlike the standard UCB regret bound for multi-armed bandits, which depends solely on the gaps between suboptimal arms and the optimal arm, the *O-UCB* regret depends on all pairwise gaps $\Delta_{i,j}$. This distinction reflects *O-UCB*'s goal of identifying the optimal ordering of arms rather than just the best arm.

We now present a simplified regret bound in the following corollary, along with its structural interpretation, to highlight key insights.

Corollary 1. *The expected cumulative regret of the O-UCB algorithm, denoted by $R_{\nu,u}(T)$, satisfies*

$$R_{\nu,u}(T) \leq \sum_{i=1}^{K-1} (K - i) \sum_{j=i+1}^{K} \frac{6 \ln T}{\Delta_{i,j}} + \frac{K^2\pi^2}{3} + K^2.$$

Structural Interpretation of the Regret Bound: The corollary reveals that the regret bound can be decomposed into several MAB instances, each corresponding to a specific position i in the arm ordering. At each position, only suboptimal arms—those with lower success probabilities compared to the optimal arm in that position—contribute to the regret. The regret bound at each position mirrors the standard UCB form, $\sum_{j=i+1}^{K} \frac{6 \ln T}{\Delta_{i,j}}$, where the summation is taken over the suboptimal arms for that position. Additionally, the regret contribution associated with position i is weighted by $(K - i)$, which corresponds to the number of positions remaining after position i. This weighting highlights that mistakes at earlier positions have a greater impact on overall regret, as they affect both the immediate outcome and all subsequent decisions.

Remark 1. *The $O(K^3)$ dependence in this corollary can be tightened to $O(K^2)$ via an algebraic simplification (Kveton et al., 2014, Lemma 3) applied to Theorem 1, which rewrites the summation of difference terms into a telescoping form; see Appendix A.7 for a proof.*

$O(1)$ **Regret Bound:**

While the regret bound in Theorem 1 scales logarithmically with the time horizon T, we next derive an $O(1)$ regret bound that is independent of T. Note that the constant factor in this bound depends not only on the sub-optimality gaps between the arms but also on an additional problem-dependent parameter

$$p_{\text{all}} = \min_{\boldsymbol{\sigma} \in \mathcal{S}_K} \prod_{i=1}^{K-1} (1 - p_{\sigma(i)}) = \prod_{i=1}^{K-1} (1 - p_i), \tag{1}$$

where $\mathcal{S}_K$ denotes the set of all possible permutations of the K arms. The parameter p_{all} represents the minimum probability of observing all arms irrespective of the ordering, and attains its minimum value under the optimal ordering. The quantity p_{all} quantifies the inherent exploration in the problem. The following theorem formalizes the $O(1)$ regret bound for the *O-UCB* algorithm.

Theorem 2. *The expected cumulative regret of the O-UCB algorithm, denoted by $R_{\nu,u}(T)$, satisfies*

$$R_{\nu,u}(T) \leq \sum_{j=1}^{K} \sum_{i=1}^{j-1} (K-i) \left[\frac{12}{\Delta_{i,j} p_{all}} \ln \left(\frac{12}{\Delta_{i,j}^2 p_{all}} \right) + \frac{1}{\exp(p_{all}/8) - 1} \right] + \frac{K^2 \pi^2}{3} + K^2.$$

From Theorems 1 and 2, we obtain the following corollary.

Corollary 2. *The expected cumulative regret of the O-UCB algorithm, denoted by $R_{\nu,u}(T)$, satisfies*

$$R_{\nu,u}(T) = O\Bigg(\min \Bigg\{ \frac{K^2 \ln T}{\Delta}, \frac{K^3 \ln \left(\frac{1}{\Delta^2 p_{all}} \right)}{\Delta p_{all}} \Bigg\} \Bigg),$$

where $\Delta = \min_{j \in \{1,2,\ldots,K-1\}} \Delta_{j,j+1}$.

3.2 *O-Greedy* Algorithm

We now describe the *O-Greedy* algorithm and provide its regret guarantees.

Following the same initial exploration strategy as the *O-UCB* algorithm, the *O-Greedy* algorithm ensures that each arm is sampled at least once in the first K rounds. From round $K+1$ onward, the arm ordering is obtained by sorting the arms in decreasing order of their empirical mean rewards. A regret bound of the *O-Greedy* algorithm is formally characterized in the following theorem.

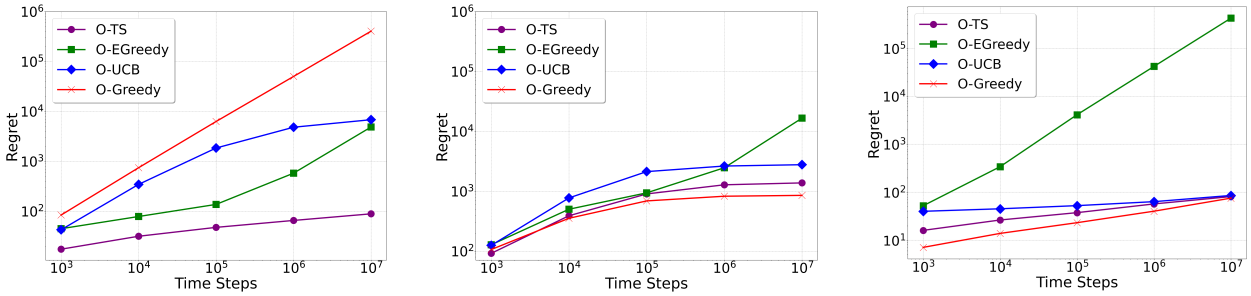
Theorem 3. *The expected cumulative regret of the O-Greedy algorithm, denoted by $R_{\nu,g}(T)$, satisfies*

$$R_{\nu,g}(T) \leq \frac{16K}{p_{all}} \sum_{j=1}^{K-1} \frac{1}{\Delta_{j,j+1}^2} + \frac{8K^2}{p_{all}} + K^2 = O\Bigg(\frac{K^2}{\Delta^2 p_{all}} \Bigg).$$

Remark 2. *Unlike standard multi-armed bandits—where the greedy algorithm incurs linear regret due to insufficient exploration—the cascading feedback structure here ensures* ***constant*** *regret. Nonetheless, because the bound scales as $\frac{1}{p_{all}}$, environments with low inherent exploration (i.e., small p_{all}) can lead to significantly higher regret, reflecting the poorer performance of the O-Greedy algorithm in such settings. In contrast, when p_{all} is sufficiently large, the inherent exploration of the O-Greedy algorithm enables it to outperform O-UCB, as observed in the numerical results later.*

4 Numerical Results

In this section, we numerically evaluate the performance of the proposed *O-UCB* and *O-Greedy* algorithms through extensive simulations on both synthetic and real-world datasets. We also include *O-TS* and *O-EGreedy*, which are adapted versions of Thompson Sampling (Agrawal & Goyal, 2012) and ε-Greedy for our



(a) Low p_{all} env (b) High p_{all} env (c) Real-world env (Yandex Dataset)

Figure 1: Regret as a function of time horizon T across synthetic and real-world environments

ordered setting. *O-TS* uses posterior sampling to select the ordering, while *O-EGreedy* selects the greedy ordering with probability $1 - \varepsilon$, and a uniformly random ordering with probability ε. Our objective is to analyze how the cumulative regret evolves with the number of rounds T, the sub-optimality gap Δ, the inherent exploration parameter p_{all}, and the number of arms K. For completeness, the pseudocode for the adapted *O-TS* and *O-EGreedy* baselines is provided in Appendix B.1.

4.1 Evaluation on Synthetic Data

We conduct simulations across a variety of environments to understand the performance of each algorithm under different conditions. Unless stated otherwise, all results are averaged over 100 independent runs with $T = 10^7$, $K = 10$, and $\varepsilon = 0.01$.

Regret as a function of the time horizon T:

We evaluate the regret as a function of the time horizon T in two environments that differ in the level of inherent exploration: one with low p_{all} and another with high p_{all}.

In the first environment, shown in Figure 1a, the true success probabilities are evenly spaced in the range $[0.9, 0.99]$, implying a minimum gap $\Delta = 0.01$, and $p_{\text{all}} = 3.6 \times 10^{-13}$, representing a regime with negligible inherent exploration. Results show that *O-Greedy* performs significantly worse than the other algorithms due to the lack of sufficient inherent exploration. Among the rest, *O-TS* achieves the lowest regret, while *O-UCB* and *O-EGreedy* perform moderately well.

In the second environment, shown in Figure 1b, the true success probabilities are evenly spaced in the range $[0.4, 0.49]$, implying a minimum gap $\Delta = 0.01$ and $p_{\text{all}} = 4.55 \times 10^{-3}$, indicating sufficient inherent exploration. The regret curves of *O-Greedy*, *O-UCB*, and *O-TS* saturate and remain nearly flat from 10^5 to 10^7 rounds. Among these, *O-Greedy* achieves the lowest regret by exploiting the inherent exploration present in the environment. In contrast, *O-EGreedy* continues to accumulate regret due to persistent exploration, which is unnecessary and even detrimental when the environment already provides sufficient exploration. Appendix B.2 provides the corresponding regret plots with shaded standard-deviation bands, which show that the observed trends are consistent across independent trials.

Regret as a function of the inherent exploration p_{all}:

To evaluate how regret varies with the level of inherent exploration, we consider six environments with corresponding p_{all} values ranging from 10^{-10} to 10^{-2}. Specifically, the true success probabilities in each environment are evenly spaced in the ranges $[0.83, 0.99]$, $[0.73, 0.89]$, $[0.63, 0.79]$, $[0.53, 0.69]$, $[0.43, 0.59]$, and $[0.33, 0.49]$, with a suboptimality gap of $\Delta = 0.02$.

As seen in Figure 2, the performance of *O-Greedy* is highly sensitive to the level of inherent exploration. When p_{all} is high—indicating sufficient inherent exploration—*O-Greedy* performs exceptionally well, often outperforming all other algorithms. However, as p_{all} decreases, its performance deteriorates rapidly. In

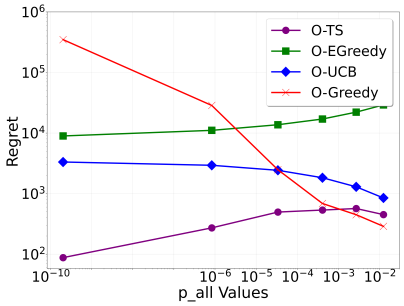


Figure 2: Regret vs. p_{all} parameter.

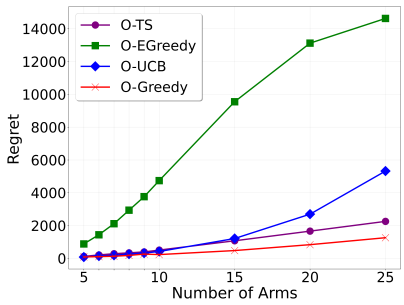


Figure 3: Regret vs Number of arms K

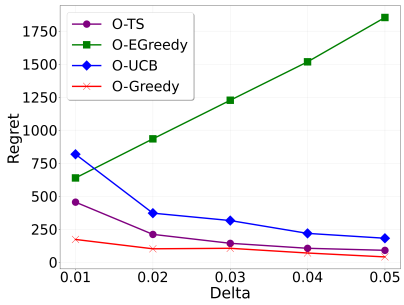


Figure 4: Regret vs Suboptimality gap Δ

contrast, *O-EGreedy* exhibits the opposite behavior. Its exploratory mechanism enables better performance in low-p_{all} regimes, but leads to increasing regret as p_{all} becomes larger, where explicit exploration is no longer necessary. This inability to adapt to the level of inherent exploration of the environment limits the effectiveness of both *O-Greedy* and *O-EGreedy* across varying environments. On the other hand, *O-UCB* and *O-TS* maintain consistently low regret across all p_{all} values, demonstrating their robustness to varying levels of inherent exploration.

Regret as a function of the number of arms K:

To evaluate the effect of the number of arms K, we consider a sequence of environments with true success probabilities starting at 0.65 and increasing in steps of 0.01. The number of arms varies from 5 to 25 across the environments, and the regret is measured over a time horizon of $T = 10^6$. As shown in Figure 3, regret increases with K for all algorithms, with the growth being particularly pronounced for *O-EGreedy*.

Regret as a function of the sub-optimality gap Δ:

To study the effect of suboptimality gaps, we consider five environments with Δ values ranging from 0.01 to 0.05 and analyze the regret over a time horizon of $T = 10^6$. Specifically, the true success probabilities of the environments are evenly spaced in the following ranges: $[0.5, 0.59]$, $[0.5, 0.68]$, $[0.5, 0.77]$, $[0.5, 0.86]$, and $[0.5, 0.95]$. As shown in Figure 4, the regret of *O-Greedy*, *O-UCB*, and *O-TS* decreases with increasing suboptimality gap Δ, as larger gaps make it easier to distinguish between arms. In contrast, *O-EGreedy* exhibits increasing regret with Δ, as its continued exploration results in higher expected regret per exploratory action when suboptimal arms are significantly worse than the optimal one.

4.2 Evaluation on Real-World Dataset

We use the Yandex dataset (Yandex, 2014), a web search click log that records user interactions with ranked lists. We extract sessions containing consistent sets of 10 URLs. Most items have low click probabilities, with over 80% having empirical click-through rates below 0.2, resembling an environment with high inherent exploration (p_{all}). We randomly select one such instance and evaluate regret over $T = 10^7$ rounds; the corresponding URLs and empirical click-through rates are listed in Appendix B.3. As seen in Figure 1c, *O-Greedy* excels in this high-p_{all} regime. *O-UCB* and *O-TS* also perform well, while *O-EGreedy* incurs high regret due to persistent exploration. Appendix B.3 further reports results on six additional instances, all of which show the same qualitative behavior.

5 Proofs of Regret Bounds

In this section, we provide proof sketches for the regret analysis of the *O-UCB* and *O-Greedy* algorithms. The detailed proofs are provided in Appendix.

5.1 *O-UCB* Regret Analysis

To prove the regret bounds, we follow these two steps:

1. Reducing the combinatorial complexity of the regret expression by expressing it in terms of position-specific arm pulls, and
2. Bounding the position-specific pull counts of suboptimal arms.

Reducing combinatorial complexity:

Let $N_T(\boldsymbol{\sigma})$ denote the total number of times an algorithm selects the ordering $\boldsymbol{\sigma}$ in T rounds. A naive expression for the regret sums over all suboptimal permutations:

$$R_{\nu,\pi}(T) = \sum_{\boldsymbol{\sigma}\neq\boldsymbol{\sigma}^*} \Delta(\boldsymbol{\sigma})\mathbb{E}[N_T(\boldsymbol{\sigma})].$$

However, this formulation leads to an intractable sum of $K!$ terms. To bypass this explosion in complexity, we upper bound the suboptimality gap $\Delta(\boldsymbol{\sigma})$, and subsequently the regret $R_{\nu,\pi}(T)$ in terms of *primitive components*: the pairwise suboptimality gaps $\Delta_{i,j}$ between arms, and the number of times an arm is pulled at a specific position. This yields a decomposition with only K^2 terms, exploiting the structure that regret arises from placing worse arms ahead of better ones. This is formalized through Lemma 1 and Lemma 2 below.

Lemma 1. *The sub-optimality gap of an order $\boldsymbol{\sigma}$ satisfies*

$$\Delta(\boldsymbol{\sigma}) = \mu(\boldsymbol{\sigma}) - \mu(\boldsymbol{\sigma}^*) \leq \sum_{i=1}^{K-1}(K-i)\left(\prod_{l=1}^{i-1}(1-p_{\sigma_l})\right)\max\{\Delta_{i,\sigma_i}, 0\},$$

where $\Delta_{i,\sigma_i} = p_i - p_{\sigma_i}$ is the difference between the success probabilities of arms i and σ_i.

Remark 3. *The upper bound in Lemma 1 highlights that suboptimal arms placed earlier in the order contribute more significantly to the regret. Specifically, whenever a suboptimal arm is placed in position i, its regret contribution is weighted by two factors: (i) the probability that it is sampled, given by $\prod_{l=1}^{i-1}(1-p_{\sigma_l})$, which corresponds to the event that all earlier arms in $\boldsymbol{\sigma}$ fail; and (ii) the factor $K-i$, which reflects how many additional arms may be unnecessarily probed due to having a worse arm at position i.*

Let $n_T^{(i)}(a_j)$ denote the total number of times arm a_j is pulled in the ith position up to round T.

Lemma 2. *The regret of an algorithm π satisfies*

$$R_{\nu,\pi}(T) \leq \sum_{j=1}^{K}\sum_{i<j}(K-i)\,\Delta_{i,j}\,\mathbb{E}_{\nu,\pi}[n_T^{(i)}(a_j)].$$

Note that in the upper bound given in Lemma 2, an arm j contributes to the regret only when it is played in a position $i < j$, that is, ahead of better arms in the ordering.

We now decompose the regret according to whether the UCB confidence bounds hold at each round. The regret incurred when the bounds fail is controlled using standard concentration arguments, while the regret accumulated when the bounds hold is analyzed through the position-specific pull-count decomposition underlying Lemma 2. To this end, let

$$\mathcal{E}_t := \left\{\exists\, j : |\bar{p}_t(a_j) - p_j| > \sqrt{\frac{3\log t}{2\,n_t(a_j)}}\right\}$$

denote the bad event that at least one confidence bound fails at round t. The cumulative regret contribution from the bad events can be bounded as

$$\mathbb{E}\left[\sum_{t=1}^{T}\Delta(\boldsymbol{\sigma}(t))\mathbf{1}\{\mathcal{E}_t\}\right] \leq \frac{K^2\pi^2}{3}.$$

We next analyze the position-specific pull counts associated with the good-event rounds.

Analysis of position-specific pull counts:

Let $\mathcal{A}_t^{ij}$ denote the event that arm j is sampled in position i at round t, i.e., $\sigma_i(t) = j$ and all earlier arms—namely $\sigma_1(t), \ldots, \sigma_{i-1}(t)$—fail to produce a success. Define the number of such pulls under the good events as

$$Z_T^{(i)}(a_j) := \sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}_t^{ij}\}\mathbf{1}\{\mathcal{E}_t^c\}.$$

By restricting the argument used to prove Lemma 2 to the good-event rounds, the regret contribution from these rounds can be shown to satisfy

$$\mathbb{E}\left[\sum_{t=1}^{T} \Delta(\boldsymbol{\sigma}(t))\mathbf{1}\{\mathcal{E}_t^c\}\right] \leq \sum_{j=2}^{K}\sum_{i=1}^{j-1}(K-i)\Delta_{i,j}\mathbb{E}\left[Z_T^{(i)}(a_j)\right].$$

We present two complementary approaches to bound the position-specific pull counts $\mathbb{E}\left[Z_T^{(i)}(a_j)\right]$, yielding a logarithmic bound in T and a constant bound independent of T.

Logarithmic Regret Bound (Theorem 1): A standard argument for UCB-type algorithms shows that if an arm has been sampled sufficiently, its confidence interval shrinks enough to prevent it from being placed ahead of a better arm. Specifically, at round t, arm j will not be placed in position m if

$$n_t(a_j) \geq \frac{6\ln t}{\Delta_{m,j}^2}.$$

Moreover, in the cascading setting, this implies that arm j cannot appear in any position earlier than m, since those positions are occupied by arms with even higher UCB scores. This allows us to jointly bound the number of times a suboptimal arm is pulled in any of the first m positions—an effect unique to the cascade model, which is formalized below.

Lemma 3. *Let $m < j$. Then, under the UCB algorithm, we have*

$$\mathbb{E}\left[\sum_{i=1}^{m} Z_T^{(i)}(a_j)\right] \leq \frac{6\ln T}{\Delta_{m,j}^2}.$$

Further, as highlighted in Lemmas 1 and 2, misplacements at earlier positions contribute more to the regret due to the factor $K - i$, leading to a telescoping structure in the total regret contribution of an arm across positions. The following lemma formalizes this and leads to the logarithmic bound stated in Theorem 1.

Lemma 4. *In the UCB algorithm, for any arm j, we have*

$$\sum_{i=1}^{j-1}(K-i)\Delta_{i,j}\,\mathbb{E}\left[Z_T^{(i)}(a_j)\right] \leq \sum_{i=1}^{j-1}(K-i)\Delta_{i,j}(\eta_{ij} - \eta_{i-1,j}),$$

where $\eta_{ij} := \frac{6\ln T}{\Delta_{i,j}^2}$ and $\eta_{0j} := 0$.

Constant Regret Bound (Theorem 2): In standard multi-armed bandit analyses, arm selection conditions such as $n_t(a_j) \geq \frac{6\ln t}{\Delta_{i,j}^2}$ are typically addressed by simply noting that $t \leq T$. Thus, once $n_t(a_j)$ exceeds $\frac{6\ln T}{\Delta_{i,j}^2}$, the condition is never triggered again, and the arm is excluded from future selections. Such standard arguments yield an $O(\ln T)$ regret bound.

In contrast, our constant regret bound leverages a structural property unique to the cascading setting: due to inherent exploration, every arm has a non-zero probability of being sampled in each round. Specifically, each arm a_j is pulled with probability at least p_{all}, so $\mathbb{E}[n_t(a_j)] \geq t\,p_{\text{all}}$. Since the expected number of pulls grows linearly while the confidence threshold grows only logarithmically, the condition $n_t(a_j) \geq \frac{6\ln t}{\Delta_{i,j}^2}$ is satisfied

after a certain number of rounds—determined by p_{all} and $\Delta_{i,j}$, but independent of T. Specifically, we show that for

$$t > \frac{12}{\Delta_{i,j}^2 p_{\text{all}}} \ln\left(\frac{12}{\Delta_{i,j}^2 p_{\text{all}}}\right),$$

the condition holds with high probability, leading to the constant regret bound stated in Theorem 2.

5.2 *O-Greedy* Regret Analysis (Theorem 3)

Let $\mathcal{E}_G(t)$ denote the event that the *O-Greedy* algorithm pulls each arm at least $tp_{\text{all}}/2$ times by round t. The regret can be decomposed into contributions from the events $\mathcal{E}_G(t)$ and their complements $\mathcal{E}_G^c(t)$. Using concentration inequalities, we bound the total regret due to the complement events by $\frac{8K^2}{p_{\text{all}}}$. In each round, the maximum possible regret is at most K, and under the *O-Greedy* algorithm, if $\boldsymbol{\sigma}(t) \neq \boldsymbol{\sigma}^*$, then there exists an index $j \in \{1, \ldots, K-1\}$ such that $\bar{p}_t(a_j) \leq \bar{p}_t(a_{j+1})$. Based on these observations, the total regret contributed by the events $\mathcal{E}_G(t)$ can be bounded by $\frac{16K}{p_{\text{all}}} \sum_{j=1}^{K-1} \frac{1}{\Delta_{j,j+1}^2}$. Therefore, the cumulative regret of the *O-Greedy* algorithm is

$$R_{\nu,g}(T) \leq \frac{16K}{p_{\text{all}}} \sum_{j=1}^{K-1} \frac{1}{\Delta_{j,j+1}^2} + \frac{8K^2}{p_{\text{all}}} + K^2.$$

Note that the additional K^2 term upper bounds the regret accumulated during the initial K rounds.

6 Concluding Remarks

We studied a cascading bandit setting where, in each round, the algorithm selects an ordering of all arms, and regret captures the extra pulls needed to reach success compared to the optimal order. This framework is motivated by practical applications such as dynamic spectrum access in cognitive radio networks and carousel-based recommendation systems, where minimizing the time to first success directly impacts user experience or system efficiency.

To address this, we proposed two algorithms, *O-UCB* and *O-Greedy*, that learn the optimal ordering from partial feedback in an online manner. Our analysis shows that *O-UCB* enjoys both a logarithmic regret bound in the time horizon T, and a constant regret bound that depends on the suboptimality gaps and an inherent exploration parameter p_{all}. We also prove that *O-Greedy* achieves constant regret under the same dependencies, despite lacking explicit exploration.

Simulations corroborate our theoretical results: *O-Greedy* performs exceptionally well in environments with high inherent exploration, while *O-UCB* maintains reliable performance across a wide range of conditions. We also benchmark against baseline algorithms such as *O-TS* and *O-EGreedy* to provide a comprehensive comparison.

Promising directions for future research include extending our results to contextual settings, where the learner leverages side information or features to adaptively order the arms based on the context. Furthermore, exploring multi-agent cascading bandit settings, where multiple learners interact with shared or overlapping environments, opens up new challenges.

References

Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pp. 39.1–39.26. JMLR Workshop and Conference Proceedings, 2012.

Venkatachalam Anantharam, Pravin Varaiya, and Jean Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays—Part I: I.I.D. rewards. *IEEE Transactions on Automatic Control*, 32(11):968–976, 1987.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002. doi: 10.1023/A:1013689704352.

Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

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

Chao Gan, Ruida Zhou, Jing Yang, and Cong Shen. Cost-aware cascading bandits. *IEEE Transactions on Signal Processing*, 68:3692–3706, 2020.

Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. Batched multi-armed bandits problem. *Advances in Neural Information Processing Systems*, 32, 2019.

Haruka Kiyohara, Yuta Saito, Tatsuya Matsuhiro, Yusuke Narita, Nobuyuki Shimizu, and Yasuo Yamamoto. Doubly robust off-policy evaluation for ranking policies under the cascade behavior model. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 487–497, 2022.

Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. Matroid bandits: Fast combinatorial optimization with learning. *arXiv preprint arXiv:1403.5045*, 2014.

Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning*, pp. 767–776. PMLR, 2015a.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Combinatorial cascading bandits. *Advances in Neural Information Processing Systems*, 28, 2015b.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Shuang Li, Shuxin Tang, Yuan Zhou, and Wei Chen. Contextual combinatorial cascading bandits. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 1245–1253, 2016.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 784–791, 2008.

Behnam Rahdari, Branislav Kveton, and Peter Brusilovsky. The magic of carousels: Single vs. multi-list recommender systems. In *Proceedings of the 33rd ACM Conference on Hypertext and Social Media*, pp. 166–174, 2022.

Vishnu Raj, Irene Dias, Thulasi Tholeti, and Sheetal Kalyani. Spectrum access in cognitive radio using a two-stage reinforcement learning approach. *IEEE Journal of Selected Topics in Signal Processing*, 12(1): 20–34, 2018.

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

Daniel Vial, Sujay Sanghavi, Sanjay Shakkottai, and Rayadurgam Srikant. Minimax regret for cascading bandits. *Advances in Neural Information Processing Systems*, 35:29126–29138, 2022.

Yandex. Yandex personalized web search challenge. `https://www.kaggle.com/c/yandex-personalized-web-search-challenge`, 2014. Accessed: 2025-08-01.

Zixin Zhong, Wang Chi Chueng, and Vincent YF Tan. Thompson sampling algorithms for cascading bandits. *Journal of Machine Learning Research*, 22(218):1–66, 2021.

A Detailed Proofs

In this appendix, we provide detailed theoretical proofs of all lemmas and theorems presented in the main paper.

A.1 Proof of Lemma 1

To prove the lemma, we first state and prove an auxiliary claim.

Claim 1. *Let $\{c_i\}_{i=1}^n$, $\{d_i\}_{i=1}^n$, and $\{\epsilon_i\}_{i=1}^n$ be sequences such that $c_i, d_i \in [0,1]$, $\epsilon_i \geq 0$, and $c_i - d_i \leq \epsilon_i$ for all $i \in [n]$. Then,*

$$\prod_{i=1}^{n} c_i - \prod_{i=1}^{n} d_i \leq \sum_{i=1}^{n} \left(\prod_{j=1}^{i-1} c_j \right) \epsilon_i,$$

where we define the empty product $\prod_{j=1}^{0} c_j = 1$.

Proof. We prove the claim by induction on n.

Base case ($n = 1$):

$$c_1 - d_1 \leq \epsilon_1.$$

So the inequality holds for $n = 1$.

Inductive step: Assume the claim holds for $n = k$, i.e.,

$$\prod_{i=1}^{k} c_i - \prod_{i=1}^{k} d_i \leq \sum_{i=1}^{k} \left(\prod_{j=1}^{i-1} c_j \right) \epsilon_i.$$

We now prove that the claim also holds for $n = k + 1$. Let

$$x := \prod_{i=1}^{k} c_i, \quad y := \prod_{i=1}^{k} d_i.$$

Then,

$$\begin{aligned}
\prod_{i=1}^{k+1} c_i - \prod_{i=1}^{k+1} d_i &= x c_{k+1} - y d_{k+1} \\
&= x(c_{k+1} - d_{k+1}) + (x - y) d_{k+1} \\
&\leq x \epsilon_{k+1} + \left(\sum_{i=1}^{k} \left(\prod_{j=1}^{i-1} c_j \right) \epsilon_i \right) d_{k+1},
\end{aligned}$$

where the first term uses $c_{k+1} - d_{k+1} \leq \epsilon_{k+1}$, and the second term uses the induction hypothesis.

Since $d_{k+1} \leq 1$, we have

$$\left(\sum_{i=1}^{k} \left(\prod_{j=1}^{i-1} c_j \right) \epsilon_i \right) d_{k+1} \leq \sum_{i=1}^{k} \left(\prod_{j=1}^{i-1} c_j \right) \epsilon_i.$$

Therefore,

$$\prod_{i=1}^{k+1} c_i - \prod_{i=1}^{k+1} d_i \leq \left(\prod_{j=1}^{k} c_j \right) \epsilon_{k+1} + \sum_{i=1}^{k} \left(\prod_{j=1}^{i-1} c_j \right) \epsilon_i = \sum_{i=1}^{k+1} \left(\prod_{j=1}^{i-1} c_j \right) \epsilon_i.$$

Thus, the inequality holds for $n = k + 1$. Thus by induction, the result holds for all $n \in \mathbb{N}$. □

Having proved the claim, we now proceed with the main proof of the lemma. Let $q_i := 1 - p_i$ denote the failure probability of arm i. Observe that the expected cost of an order $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_K]$ is

$$\mu(\boldsymbol{\sigma}) = 1 + \sum_{i=1}^{K-1} \prod_{j=1}^{i} q_{\sigma_j},$$

which follows from the standard identity, $\mathbb{E}[X] = \sum_{i=0}^{\infty} \mathbb{P}(X > i)$, for non-negative integer-valued random variables. Then the sub-optimality gap is given by

$$\begin{aligned}
\Delta(\boldsymbol{\sigma}) &= \mu(\boldsymbol{\sigma}) - \mu(\boldsymbol{\sigma}^*) \\
&= \sum_{i=1}^{K-1} \prod_{j=1}^{i} q_{\sigma_j} - \sum_{i=1}^{K-1} \prod_{j=1}^{i} q_j \\
&= \sum_{i=1}^{K-1} \left(\prod_{j=1}^{i} q_{\sigma_j} - \prod_{j=1}^{i} q_j \right).
\end{aligned}$$

Let $\epsilon_j := \max\{0, \Delta_{j,\sigma_j}\}$. Then we have $\Delta_{j,\sigma_j} = p_j - p_{\sigma_j} = q_{\sigma_j} - q_j \leq \epsilon_j$, for $j \in [K]$. Using Claim 1, we can bound the sub-optimality gap as follows. We have

$$\prod_{j=1}^{i} q_{\sigma_j} - \prod_{j=1}^{i} q_j \leq \sum_{j=1}^{i} \left(\prod_{l=1}^{j-1} q_{\sigma_l} \right) \epsilon_j.$$

Therefore, summing over $i = 1$ to $K - 1$, we get:

$$\begin{aligned}
\Delta(\boldsymbol{\sigma}) &\leq \sum_{i=1}^{K-1} \sum_{j=1}^{i} \left(\prod_{l=1}^{j-1} q_{\sigma_l} \right) \epsilon_j \\
&= \sum_{j=1}^{K-1} \sum_{i=j}^{K-1} \left(\prod_{l=1}^{j-1} q_{\sigma_l} \right) \epsilon_j \\
&= \sum_{j=1}^{K-1} (K - j) \times \left(\prod_{l=1}^{j-1} q_{\sigma_l} \right) \epsilon_j.
\end{aligned}$$

Hence

$$\Delta(\boldsymbol{\sigma}) \leq \sum_{j=1}^{K-1} (K - j) \left(\prod_{l=1}^{j-1} q_{\sigma_l} \right) \max\{0, \Delta_{j,\sigma_j}\}.$$

A.2 Proof of Lemma 2

$$R_{\nu,\pi}(T) = \mathbb{E}\left[\sum_{t=1}^{T} \Delta(\sigma(t)) \right].$$

Let $q_i := 1 - p_i$ denote the failure probability of arm i. Applying Lemma 1 to each round and taking expectations:

$$\begin{aligned}
R_{\nu,\pi}(T) &\leq \mathbb{E}\left[\sum_{t=1}^{T} \sum_{i=1}^{K-1} (K - i) \left(\prod_{\ell=1}^{i-1} q_{\sigma_\ell(t)} \right) \max\{\Delta_{i,\sigma_i(t)}, 0\} \right] \\
&= \sum_{i=1}^{K-1} (K - i) \cdot \mathbb{E}\left[\sum_{t=1}^{T} \left(\prod_{\ell=1}^{i-1} q_{\sigma_\ell(t)} \right) \max\{\Delta_{i,\sigma_i(t)}, 0\} \right].
\end{aligned}$$

The expectation term can be rewritten using the indicator variable $\mathbf{1}\{\sigma_i(t) = j\}$ as:

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(\prod_{\ell=1}^{i-1} q_{\sigma_\ell(t)}\right)\sum_{j=1}^{K}\max\{\Delta_{i,j}, 0\}\mathbf{1}\{\sigma_i(t) = j\}\right].$$

Let $\mathcal{A}_t^{ij}$ denote the indicator that arm j is *pulled* at position i in round t; this occurs if arm j is placed in position i, and all arms placed before it (in positions 1 to $i-1$) fail. The expected value of this indicator is:

$$\mathbb{E}\left[\mathcal{A}_t^{ij}\right] = \mathbb{E}\left[\left(\prod_{\ell=1}^{i-1} q_{\sigma_\ell(t)}\right)\cdot \mathbf{1}\{\sigma_i(t) = j\}\right].$$

Using the above observation, and the fact that $\max\{\Delta_{i,j}, 0\} = 0 \quad \text{for } j \leq i$, we get:

$$\begin{aligned} R_{\nu,\pi}(T) &\leq \sum_{i=1}^{K-1}(K-i)\sum_{j=i+1}^{K}\Delta_{i,j}\cdot\mathbb{E}\left[n_T^{(i)}(a_j)\right] \\ &= \sum_{j=1}^{K}\sum_{i<j}(K-i)\cdot\Delta_{i,j}\cdot\mathbb{E}\left[n_T^{(i)}(a_j)\right]. \end{aligned}$$

A.3 Proof of Lemma 3

Fix any arm j and position $m < j$. Under the good event $\mathcal{E}_t^c$, we know for all j:

$$\begin{aligned} |\bar{p}_t(a_j) - p_j| &\leq \sqrt{\frac{3}{2}\cdot\frac{\ln t}{n_t(a_j)}} \\ \Rightarrow \quad p_j \leq UCB_t(a_j) &\leq p_j + 2\sqrt{\frac{3}{2}\cdot\frac{\ln t}{n_t(a_j)}} \end{aligned}$$

We now define the threshold:

$$\eta_{m,j} := \frac{6\ln T}{\Delta_{m,j}^2}.$$

Let us consider a round t such that $n_t(a_j) \geq \eta_{m,j}$. Then, under the good event $\mathcal{E}_t^c$, we have:

$$\mathrm{UCB}_t(a_j) \leq p_j + 2\sqrt{\frac{3}{2}\cdot\frac{\ln T}{\eta_{m,j}}} = p_j + \Delta_{m,j} = p_m.$$

Now for any arm $i \leq m$, since $p_i \geq p_m$, we also have:

$$\mathrm{UCB}_t(a_i) \geq p_i \geq p_m \geq \mathrm{UCB}_t(a_j).$$

Thus, $\mathrm{UCB}_t(a_j) < \mathrm{UCB}_t(a_i)$ for all $i \leq m$. Hence, arm j cannot appear in any of the first m positions in round t under the good event, once $n_t(a_j) \geq \eta_{m,j}$.

We now use this fact to argue that the sum $\sum_{i=1}^{m} Z_T^{(i)}(a_j)$ is bounded by $\eta_{m,j}$. From the definition of $Z_T^{(i)}(a_j)$, we have:

$$\begin{aligned} \sum_{i=1}^{m} Z_T^{(i)}(a_j) &= \sum_{i=1}^{m}\sum_{t=1}^{T}\mathcal{A}_t^{ij}\cdot\mathbf{1}\{\mathcal{E}_t^c\} \\ &= \sum_{i=1}^{m}\sum_{t=1}^{T}\mathcal{A}_t^{ij}\mathbf{1}\{\mathcal{E}_t^c\}\mathbf{1}\{n_t(a_j) < \eta_{m,j}\} \quad + \sum_{i=1}^{m}\sum_{t=1}^{T}\mathcal{A}_t^{ij}\mathbf{1}\{\mathcal{E}_t^c\}\mathbf{1}\{n_t(a_j) \geq \eta_{m,j}\}. \end{aligned}$$

The second term is zero because, under the good event, once $n_t(a_j) \geq \eta_{m,j}$, arm j cannot appear in any of the first m positions.

Since arm j can only be pulled at most once (in one position) per round, we have $\sum_{i=1}^{m} \mathcal{A}_t^{ij} \leq 1$. Further, whenever an arm is pulled in any position, $n_t(a_j)$ increases. Using these observations in the first term, we obtain

$$\sum_{i=1}^{m} Z_T^{(i)}(a_j) = \sum_{t=1}^{T} \left(\sum_{i=1}^{m} \mathcal{A}_t^{ij} \right) \mathbf{1}\{n_t(a_j) < \eta_{m,j}\} \leq \eta_{m,j}.$$

A.4 Proof of Lemma 4

Fix any arm j. We aim to upper bound:

$$\sum_{i=1}^{j-1} (K-i)\Delta_{i,j} \cdot \mathbb{E}[Z_T^{(i)}(a_j)].$$

Let us define the cumulative sums:

$$S_m := \sum_{i=1}^{m} \mathbb{E}[Z_T^{(i)}(a_j)].$$

From Lemma 3, we have the bound

$$S_m \leq \eta_{m,j} := \frac{6 \log T}{\Delta_{m,j}^2}, \quad \forall m < j.$$

Observe that the weights $w_i := (K-i)\Delta_{i,j}$ in $\sum_{i=1}^{j-1} w_i \, \mathbb{E}[Z_T^{(i)}(a_j)]$ form a non-increasing sequence in i. Among all the possible solutions satisfying the cumulative constraints $\sum_{i=1}^{m} \mathbb{E}[Z_T^{(i)}(a_j)] \leq \eta_{m,j}$, the maximum of the weighted sum is attained by the *greedy allocation* that saturates each constraint sequentially, i.e.,

$$\mathbb{E}[Z_T^{(i)}(a_j)] = \eta_{i,j} - \eta_{i-1,j}, \quad \text{with } \eta_{0,j} := 0.$$

Substituting into the original expression yields

$$\sum_{i=1}^{j-1} (K-i)\Delta_{i,j} \cdot \mathbb{E}[Z_T^{(i)}(a_j)] \leq \sum_{i=1}^{j-1} (K-i)\Delta_{i,j} \cdot (\eta_{i,j} - \eta_{i-1,j}),$$

as claimed.

A.5 Two Additional lemmas required for Theorem 1 and Theorem 2

Lemma 5 (Regret due to Bad Events)**.** *Let the bad event at time t be defined as $\mathcal{E}_t = \left\{ \exists j : |\bar{p}_t(a_j) - p_j| > \sqrt{\frac{3}{2} \cdot \frac{\ln t}{n_t(a_j)}} \right\}$. Then, the regret due to these bad events satisfies*

$$\mathbb{E}\left[\sum_{t=1}^{T} \Delta(\sigma(t)) 1\{\mathcal{E}_t\} \right] \leq \frac{K^2 \pi^2}{3}.$$

Proof. Using the union bound over arms, we get

$$\mathbb{P}(\mathcal{E}_t) \leq \sum_{j=1}^{K} \mathbb{P}\left(|\bar{p}_t(a_j) - p_j| > \sqrt{\frac{3}{2} \cdot \frac{\ln t}{n_t(a_j)}} \right).$$

We now condition on the number of times arm j has been pulled up to time t, denoted $n_t(a_j) = s$, and apply the law of total probability to bound as:

$$\mathbb{P}(\mathcal{E}_t) \leq \sum_{j=1}^{K} \sum_{s=1}^{t} \mathbb{P}\left(\left|\bar{p}_{a_j,s} - p_j\right| > \sqrt{\frac{3}{2} \cdot \frac{\ln t}{s}} \right),$$

where $\bar{p}_{a_j,s}$ is the sample mean of the first s samples of arm j.

Using Hoeffding's inequality for bounded rewards in $[0, 1]$, we have

$$\mathbb{P}\left(\left|\bar{p}_{a_j,s} - p_j\right| > \sqrt{\frac{3}{2} \cdot \frac{\ln t}{s}} \right) \leq 2t^{-3}.$$

Substituting into the previous expression:

$$\mathbb{P}(\mathcal{E}_t) \leq \sum_{j=1}^{K} \sum_{s=1}^{t} 2t^{-3} = 2Kt^{-2}.$$

Summing over $t \geq 1$, we obtain:

$$\sum_{t=1}^{\infty} \mathbb{P}(\mathcal{E}_t) \leq \sum_{t=1}^{\infty} 2K \cdot \frac{1}{t^2} = 2K \cdot \frac{\pi^2}{6} = \frac{K\pi^2}{3}.$$

Using the fact that the maximum regret per round is K, i.e., $\Delta(\sigma) \leq K$, completes the proof. □

Lemma 6. *[Regret due to Good Events] Let the bad event at time t be defined as*

$$\mathcal{E}_t = \left\{ \exists j : |\bar{p}_t(a_j) - p_j| > \sqrt{\frac{3}{2} \cdot \frac{\ln t}{n_t(a_j)}} \right\}.$$

Let $\mathcal{A}_t^{ij}$ denote the indicator that arm j is sampled *at position i in round t. Then*

$$\mathbb{E}\left[\sum_{t=1}^{T} \Delta(\sigma(t)) 1\{\mathcal{E}_t^c\} \right] \leq \sum_{j=2}^{K} \sum_{i=1}^{j-1} (K - i)\Delta_{i,j} \mathbb{E}\left[Z_T^{(i)}(a_j) \right],$$

where $Z_T^{(i)}(a_j) := \sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}_t^{ij}\}\mathbf{1}\{\mathcal{E}_t^c\}$.

Proof. The proof is similar to the proof of Lemma 2. □

A.6 Proof of Theorem 1

The regret can be decomposed into the contributions from good and bad events as:

$$\begin{aligned} R_{\nu,u} &= \mathbb{E}\left[\sum_{t=1}^{T} \Delta(\sigma(t)) \right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \Delta(\sigma(t)) \mathbf{1}\{\mathcal{E}_t\} \right] + \mathbb{E}\left[\sum_{t=1}^{T} \Delta(\sigma(t)) \mathbf{1}\{\mathcal{E}_t^c\} \right]. \end{aligned}$$

From Lemma 5, the first term is bounded by

$$\mathbb{E}\left[\sum_{t=1}^{T} \Delta(\sigma(t)) \mathbf{1}\{\mathcal{E}_t\} \right] \leq \frac{K^2\pi^2}{3}.$$

From Lemma 6, the second term is bounded by

$$\mathbb{E}\left[\sum_{t=1}^{T}\Delta(\sigma(t))\mathbf{1}\{\mathcal{E}_t^c\}\right] \leq \sum_{j=2}^{K}\sum_{i=1}^{j-1}(K-i)\Delta_{i,j}\mathbb{E}\left[Z_T^{(i)}(a_j)\right].$$

Now using Lemma 4 to bound $\mathbb{E}\left[Z_T^{(i)}(a_j)\right]$ completes the proof.

A.7 A Corollary of Theorem 1 based on Remark 1

The following Corollary formalizes the $O(K^2)$ regret bound for UCB discussed in Remark 1.

Corollary 3. *The regret of the O-UCB algorithm satisfies*

$$R_{v,u} \leq \sum_{j=3}^{K}\sum_{i=1}^{j-2}\frac{12\ln T}{\Delta_{i,j}} + \sum_{j=2}^{K}(K-j+1)\frac{12\ln T}{\Delta_{j-1,j}} + \frac{K^2\pi^2}{3} + K^2.$$

Proof. We begin with the regret bound from Theorem 1:

$$R_{v,u} \leq 6\ln T\sum_{j=2}^{K}\sum_{i=1}^{j-1}(K-i)\Delta_{i,j}\left(\frac{1}{\Delta_{i,j}^2} - \frac{1}{\Delta_{i-1,j}^2}\right) + \frac{K^2\pi^2}{3} + K^2$$

We now rewrite the weight using the identity

$$(K-i) = \sum_{m=i}^{K-1} 1.$$

Thus,

$$\begin{aligned}
&\sum_{i=1}^{j-1}(K-i)\Delta_{i,j}\left(\frac{1}{\Delta_{i,j}^2} - \frac{1}{\Delta_{i-1,j}^2}\right) \\
&= \sum_{i=1}^{j-1}\sum_{m=i}^{K-1}\Delta_{i,j}\left(\frac{1}{\Delta_{i,j}^2} - \frac{1}{\Delta_{i-1,j}^2}\right) \\
&= \sum_{m=1}^{K-1}\sum_{i=1}^{\min(m,j-1)}\Delta_{i,j}\left(\frac{1}{\Delta_{i,j}^2} - \frac{1}{\Delta_{i-1,j}^2}\right).
\end{aligned}$$

We now state the following claim, which will be used to bound the inner sum:

Claim 2. *Let $x_1 \geq x_2 \geq \cdots \geq x_m > 0$ be a non-increasing sequence of positive numbers. Then,*

$$\sum_{i=1}^{m} x_i\left(\frac{1}{x_i^2} - \frac{1}{x_{i-1}^2}\right) \leq \frac{2}{x_m},$$

where we define $x_0 = \infty$ so that $1/x_0^2 = 0$.

This claim follows directly from Lemma 3 of Kveton et al. (2014). Applying the claim to each inner sum, we obtain:

$$\sum_{i=1}^{\min(m,j-1)}\Delta_{i,j}\left(\frac{1}{\Delta_{i,j}^2} - \frac{1}{\Delta_{i-1,j}^2}\right) \leq \frac{2}{\Delta_{\min(m,j-1),\,j}}.$$

Therefore,

$$\sum_{i=1}^{j-1}(K-i)\Delta_{i,j}\left(\frac{1}{\Delta_{i,j}^2}-\frac{1}{\Delta_{i-1,j}^2}\right)\leq\sum_{m=1}^{j-2}\frac{2}{\Delta_{m,j}}+\frac{2(K-j+1)}{\Delta_{j-1,j}}.$$

Using the above inequality to bound the inner summation of Theorem 1 statement completes the proof. □

A.8 Proof of Theorem 2

As done in the proof of Theorem 1, decomposing the regret into contributions from bad and good events using Lemma 5 and Lemma 6 gives:

$$\begin{aligned}R_{\nu,u}&=\mathbb{E}\left[\sum_{t=1}^{T}\Delta(\sigma(t))\right]\\&\leq\frac{K^2\pi^2}{3}+\sum_{j=2}^{K}\sum_{i=1}^{j-1}(K-i)\Delta_{i,j}\mathbb{E}\left[Z_T^{(i)}(a_j)\right].\end{aligned}\tag{3}$$

We now proceed to bound the term $\mathbb{E}\left[Z_T^{(i)}(a_j)\right]$. In any round t, under the good event $\mathcal{E}_t^c$, the UCB algorithm places arm j in position i only if

$$n_t(a_j)\leq\frac{6\ln t}{\Delta_{i,j}^2}.$$

Hence,

$$\begin{aligned}\mathbb{E}\left[Z_T^{(i)}(a_j)\right]&=\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\{\mathcal{A}_t^{ij}\}\mathbf{1}\{\mathcal{E}_t^c\}\right]\\&\leq\sum_{t=1}^{T}\mathbb{P}\left(n_t(a_j)\leq\frac{6\ln t}{\Delta_{i,j}^2}\right).\end{aligned}\tag{4}$$

From the definition of p_{all}, each arm j is pulled in any round with probability at least p_{all}. Let $B_t\sim\text{Binomial}(t,\,p_{\text{all}})$. Then $n_t(a_j)\succeq B_t$, and hence:

$$\mathbb{P}\left(n_t(a_j)\leq\frac{6\ln t}{\Delta_{i,j}^2}\right)\leq\mathbb{P}\left(B_t\leq\frac{6\ln t}{\Delta_{i,j}^2}\right).\tag{5}$$

Define the threshold time t_0 as:

$$t_0:=\inf\left\{t\geq1:\frac{6\ln t}{\Delta_{i,j}^2}\leq\frac{1}{2}t\cdot p_{\text{all}}\right\}.$$

We can verify that

$$t_0\leq\frac{12}{\Delta_{i,j}^2p_{\text{all}}}\ln\left(\frac{12}{\Delta_{i,j}^2p_{\text{all}}}\right).\tag{6}$$

Now, for all $t>t_0$, we have:

$$\frac{6\ln t}{\Delta_{i,j}^2}\leq\frac{1}{2}\,\mathbb{E}[B_t].$$

By applying the Chernoff bound, we get

$$\mathbb{P}\left(B_t \leq \frac{6\ln t}{\Delta_{i,j}^2}\right) \leq \mathbb{P}\left(B_t \leq \frac{1}{2}\cdot\mathbb{E}[B_t]\right) \leq \exp\left(-\frac{1}{8}p_{\text{all}}t\right). \tag{7}$$

Therefore, from (4), (5), and (7), we obtain:

$$\begin{aligned}\mathbb{E}\left[Z_T^{(i)}(a_j)\right] &\leq \sum_{t=1}^{T}\mathbb{P}\left(n_t(a_j) \leq \frac{6\ln t}{\Delta_{i,j}^2}\right)\\ &\leq \sum_{t=1}^{t_0} 1 + \sum_{t=t_0+1}^{T}\exp\left(-\frac{1}{8}p_{\text{all}}t\right)\\ &\leq t_0 + \sum_{t=t_0+1}^{\infty}\exp\left(-\frac{1}{8}p_{\text{all}}t\right).\end{aligned}$$

Using the bound on t_0 given in (6), and bounding the tail of the exponential series, $\sum_{t=t_0+1}^{\infty}\exp\left(-\frac{1}{8}p_{\text{all}}t\right) \leq \frac{1}{\exp(p_{\text{all}}/8)-1}$, we conclude:

$$\mathbb{E}\left[Z_T^{(i)}(a_j)\right] \leq \frac{12}{\Delta_{i,j}^2 p_{\text{all}}}\ln\left(\frac{12}{\Delta_{i,j}^2 p_{\text{all}}}\right) + \frac{1}{\exp(p_{\text{all}}/8)-1}. \tag{8}$$

Substituting (8) into (3) completes the proof.

A.9 Proof of Theorem 3

Recall that $R_{\nu,g}(T)$ denotes the expected regret of the *O-Greedy* algorithm and let $c_j = \frac{p_j + p_{j+1}}{2}$ denote the average of the success probabilities of arms j and $j+1$. Let $\mathcal{E}_G(t)$ denote the event that the *O-Greedy* algorithm pulls each arm at least $tp_{\text{all}}/2$ times by round t. The total regret can be decomposed into contributions from the events $\mathcal{E}_G(t)$ and their complements $\mathcal{E}_G^c(t)$.

$$\begin{aligned}R_{\nu,g}(T) &\leq K\sum_{t=1}^{T}\mathbb{P}\big(\boldsymbol{\sigma}(t) \neq \boldsymbol{\sigma}^*\big)\\ &= K\sum_{t=1}^{T}\mathbb{P}\big(\boldsymbol{\sigma}(t) \neq \boldsymbol{\sigma}^*, \mathcal{E}_G(t)\big) + K\sum_{t=1}^{T}\mathbb{P}\big(\boldsymbol{\sigma}(t) \neq \boldsymbol{\sigma}^*, \mathcal{E}_G^c(t)\big).\end{aligned} \tag{9}$$

First, we focus on the term $\sum_{t=1}^{T}\mathbb{P}\big(\boldsymbol{\sigma}(t) \neq \boldsymbol{\sigma}^*, +\mathcal{E}_G^c(t)\big)$.

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{P}\big(\boldsymbol{\sigma}(t) \neq \boldsymbol{\sigma}^*, +\mathcal{E}_G^c(t)\big) &\leq \sum_{t=1}^{T}\mathbb{P}\big(\mathcal{E}_G^c(t)\big)\\ &= \sum_{t=1}^{T}\mathbb{P}\big(\exists j : n_t(a_j) \leq tp_{\text{all}}/2\big)\\ &\leq \sum_{t=1}^{T}\sum_{j=1}^{K}\mathbb{P}\big(n_t(a_j) \leq tp_{\text{all}}/2\big)\\ &\leq \sum_{t=1}^{T}\sum_{j=1}^{K} e^{-tp_{\text{all}}/8} \leq \frac{8K}{p_{\text{all}}}.\end{aligned} \tag{10}$$

Now, we focus on the term $\sum_{t=1}^{T} \mathbb{P}\big(\boldsymbol{\sigma}(t) \neq \boldsymbol{\sigma}^*, \mathcal{E}_G(t)\big)$. For the *O-Greedy* algorithm, if $\boldsymbol{\sigma}(t) \neq \boldsymbol{\sigma}^*$, then there exists an index $j \in \{1, \ldots, K-1\}$ such that $\bar{p}_t(a_j) \leq \bar{p}_t(a_{j+1})$. Therefore,

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{P}\big(\boldsymbol{\sigma}(t) \neq \boldsymbol{\sigma}^*, \mathcal{E}_G(t)\big) &\leq \sum_{t=1}^{T} \mathbb{P}\big(\exists j : \bar{p}_t(a_j) \leq \bar{p}_t(a_{j+1}), \mathcal{E}_G(t)\big) \\
&\leq \sum_{t=1}^{T} \sum_{j=1}^{K-1} \mathbb{P}\big(\bar{p}_t(a_j) \leq \bar{p}_t(a_{j+1}), \mathcal{E}_G(t)\big) \\
&\leq \sum_{t=1}^{T} \sum_{j=1}^{K-1} \mathbb{P}\big(\bar{p}_t(a_j) \leq c_j, \mathcal{E}_G(t)\big) + \sum_{t=1}^{T} \sum_{j=1}^{K} \mathbb{P}\big(\bar{p}_t(a_{j+1}) > c_j, \mathcal{E}_G(t)\big)
\end{aligned}$$

$$\begin{aligned}
&\leq \sum_{t=1}^{T} \sum_{j=1}^{K-1} \mathbb{P}\big(\bar{p}_t(a_j) \leq c_j, n_t(a_j) \geq t p_{\text{all}}/2\big) + \sum_{t=1}^{T} \sum_{j=1}^{K-1} \mathbb{P}\big(\bar{p}_t(a_{j+1}) > c_j, n_t(a_{j+1}) \geq t p_{\text{all}}/2\big) \\
&\leq 2 \sum_{t=1}^{T} \sum_{j=1}^{K-1} e^{-t p_{\text{all}} \Delta_{j,j+1}^2/8} \leq \frac{16}{p_{\text{all}}} \sum_{j=1}^{K-1} \frac{1}{\Delta_{j,j+1}^2}. \qquad (11)
\end{aligned}$$

By substituting (10) and (11) in (9), we get

$$R_{\nu,g}(T) \leq \frac{16K}{p_{\text{all}}} \sum_{j=1}^{K-1} \frac{1}{\Delta_{j,j+1}^2} + \frac{8K^2}{p_{\text{all}}} + K^2.$$

B Additional Simulations

This appendix section contains extended experimental details, including pseudocode for the proposed algorithms, additional simulation results on both synthetic and real-world datasets, and a description of the Yandex dataset used in our evaluation.

B.1 Description of the Baseline Algorithms

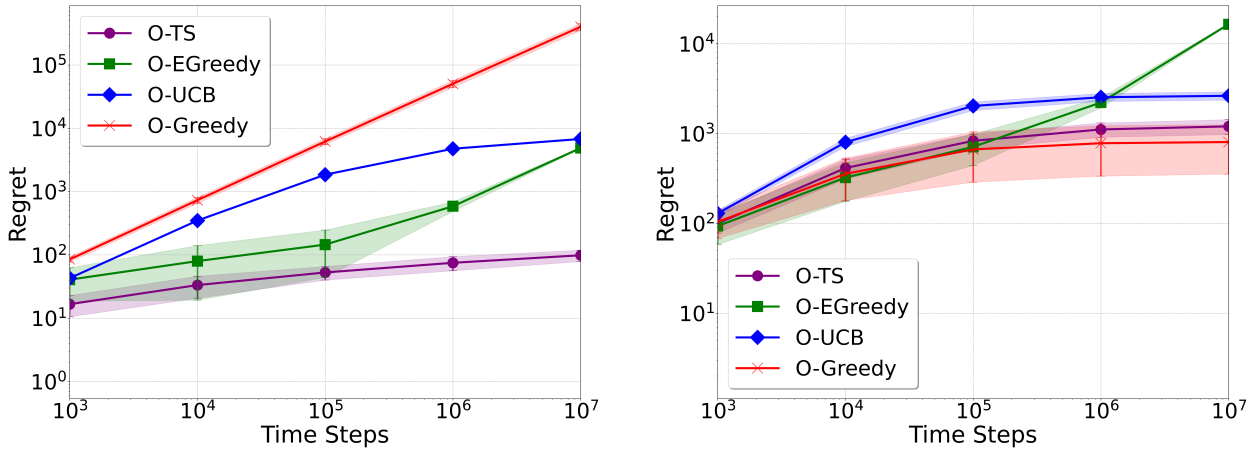
Here we are providing the pseudocode for the algorithms *O-EGreedy* and *O-TS*, which we introduced as baseline algorithms in Section 4 of the main paper.

Algorithm 2 *O-EGreedy Algorithm*

Input: Total rounds T, Number of arms K, exploration probability ϵ
Initialization: For each arm a_i, set estimated success probability $\bar{p}_0(a_i) = 0$, and count $n_0(a_i) = 0$
for $t = 1$ to T **do**
  With probability ϵ, select a random permutation $\boldsymbol{\sigma}(t)$ of the arms
  With probability $1 - \epsilon$, sort arms in decreasing order of $\bar{p}_{t-1}(a_i)$ to obtain $\boldsymbol{\sigma}(t)$
  Play arms in the order of $\boldsymbol{\sigma}(t)$ until a success is observed or all K arms are played
  Let j be the index of the first successful arm in $\boldsymbol{\sigma}(t)$; if no success, set $j = K$
  For each arm a_i in the top-j positions of $\boldsymbol{\sigma}(t)$:
    Observe outcome $r_t(a_i) \in \{0, 1\}$
    Update: $n_t(a_i) \leftarrow n_{t-1}(a_i) + 1$
    Update: $\bar{p}_t(a_i) \leftarrow \bar{p}_{t-1}(a_i) + \frac{1}{n_t(a_i)}(r_t(a_i) - \bar{p}_{t-1}(a_i))$
end for

Algorithm 3 *O-TS Algorithm*

Input: Total rounds T, Number of arms K
Initialization: For each arm a_i, set successes $s_0(a_i) = 0$, failures $f_0(a_i) = 0$
for $t = 1$ to T **do**
  For each arm a_i, sample $\theta_i \sim \text{Beta}(s_{t-1}(a_i) + 1, f_{t-1}(a_i) + 1)$
  Sort arms in decreasing order of sampled values θ_i to obtain $\boldsymbol{\sigma}(t)$
  Play arms in the order of $\boldsymbol{\sigma}(t)$ until a success is observed or all K arms are played
  Let j be the index of the first successful arm in $\boldsymbol{\sigma}(t)$; if no success, set $j = K$
  For each arm a_i in the top-j positions of $\boldsymbol{\sigma}(t)$:
    Observe outcome $r_t(a_i) \in \{0, 1\}$
    If $r_t(a_i) = 1$, update $s_t(a_i) \leftarrow s_{t-1}(a_i) + 1$
    If $r_t(a_i) = 0$, update $f_t(a_i) \leftarrow f_{t-1}(a_i) + 1$
end for



(a) Low p_{all} (b) High p_{all}

Figure 5: Regret and standard deviation vs. time horizon T in synthetic environments with low and high p_{all}.

B.2 Synthetic Data: Regret Variability Analysis

We revisit the two synthetic environments from the main paper—one characterized by low inherent exploration (i.e., low p_{all}) and the other by high inherent exploration (high p_{all}). To assess the robustness of our results, we rerun the simulations with 100 independent trials and report the mean regret with shaded standard deviation. As shown in Figure 5, the qualitative trends and relative performance of the algorithms remain consistent, confirming the stability of our conclusions.

B.3 Real-World Data: Yandex Dataset

The Yandex dataset Yandex (2014) comprises over 31,000 unique URLs and over 6,000 unique URL ID sets, each of length 10. A histogram of the empirical click-through probabilities is presented in Figure 6. The majority of items exhibit low click probabilities, with over 80% having empirical click-through rates below 0.2. This corresponds to a setting with high inherent exploration (i.e., large p_{all}).

From this pool of URL ID sets, we select six instances that appeared most frequently in the dataset to ensure sufficient interaction data for evaluation. In each instance, the 10 URLs are treated as arms in our cascading bandit setting. The specific URL IDs and their empirical success probabilities (click-through rates) of these arms are shown in Table 1.

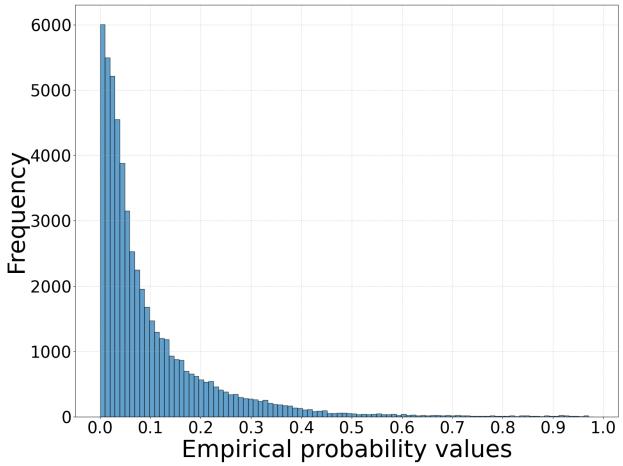


Figure 6: Histogram of empirical probability values

Instance 1	
URL ID	p
1045409	0.0006
7920148	0.0002
11352247	0.0056
26202930	0.0007
32363511	0.0002
34033472	0.0328
34175267	0.9588
35491056	0.0001
43630531	0.0007
60010091	0.0002

Instance 2	
URL ID	p
15370141	0.0014
26202930	0.0004
29902424	0.0015
31342884	0.0017
34033472	0.1668
34171511	0.0021
34175267	0.8175
35444452	0.0050
43630531	0.0028
67773505	0.0007

Instance 3	
URL ID	p
24789609	0.0164
29276408	0.0104
29302552	0.0893
29328434	0.1358
29408102	0.3131
29451601	0.0325
29464699	0.2907
29516474	0.0626
29530804	0.0359
63371986	0.0132

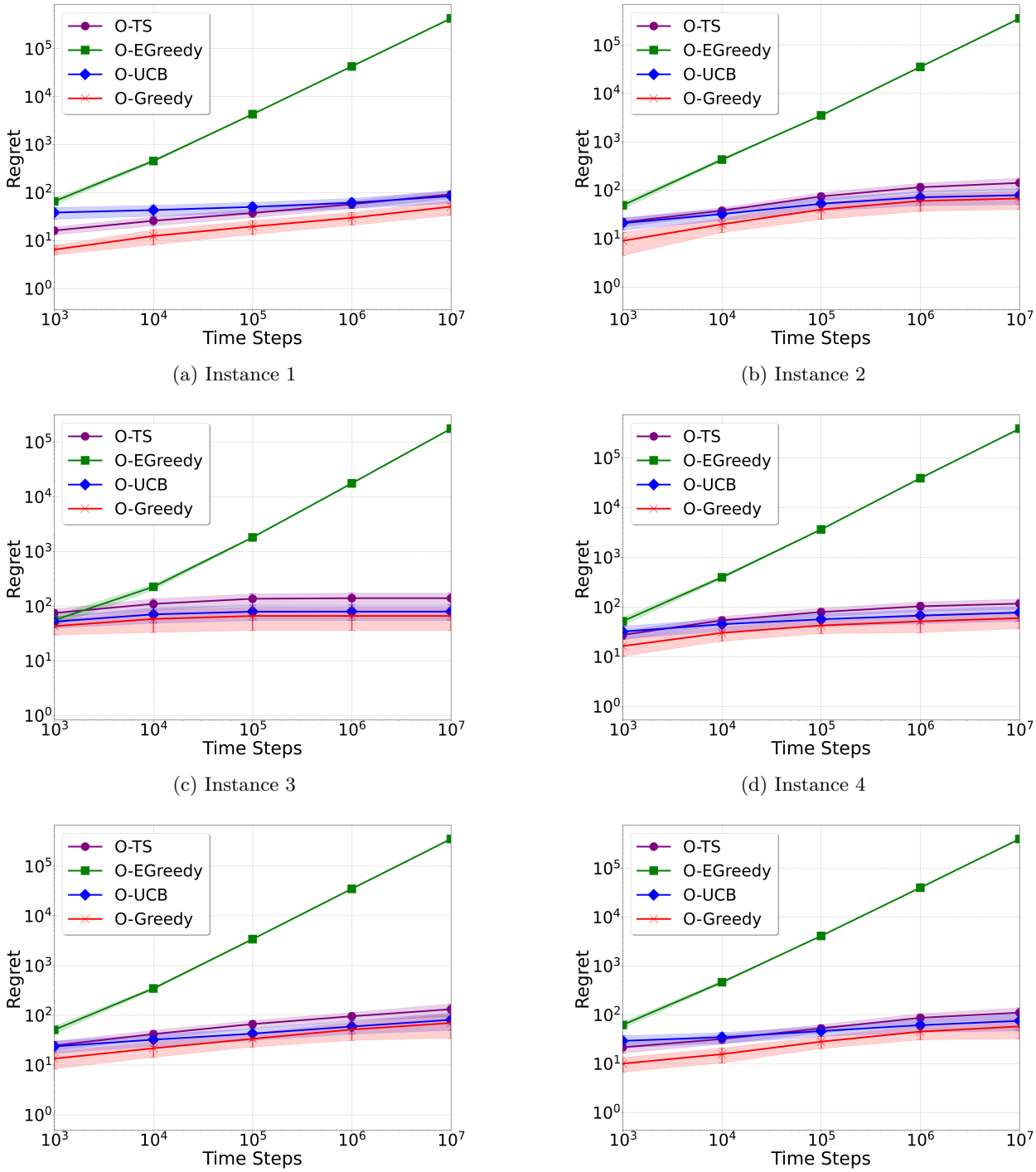
Instance 4	
URL ID	p
2522212	0.0533
2522317	0.0039
21250468	0.0039
27221002	0.0094
27398472	0.0208
29220341	0.0114
34192218	0.0028
36612684	0.0024
39061378	0.8913
69693878	0.0008

Instance 5	
URL ID	p
11352247	0.0043
26202930	0.0005
29902424	0.0018
34033472	0.1672
34171062	0.0009
34175267	0.8101
34853233	0.0007
35444452	0.0112
39016998	0.0016
43630531	0.0018

Instance 6	
URL ID	p
1045409	0.0002
11352247	0.0014
19715603	0.0008
31455963	0.0089
34033472	0.0760
34175267	0.9092
39008410	0.0003
43630531	0.0015
52754664	0.0011
60010091	0.0004

Table 1: Empirical success probabilities (click-through rates) for six Yandex instances. Each table lists the 10 URL IDs used as arms in that instance. Here, p represents empirical success probabilities.

We simulate each instance over a time horizon from 10^3 to 10^7, averaging regret over 100 independent runs. The resulting regret curves, along with shaded standard deviation, are shown in Figure 7. Due to the high inherent exploration (p_{all}) in these instances, the *O-Greedy* algorithm performs best by effectively exploiting the order structure. *O-TS* and *O-UCB* also achieve low regret, while *O-EGreedy* consistently incurs the highest regret.



(a) Instance 1

(b) Instance 2

(c) Instance 3

(d) Instance 4

(e) Instance 5

(f) Instance 6

Figure 7: Regret vs. time horizon T with standard deviation shaded for six selected Yandex instances.

B.4 Simulation Tool and Code Details

All simulations and plots were implemented using standard Python libraries, including `NumPy`, `Matplotlib`, and `Pandas`. The code is uploaded as part of the supplementary materials.