Multi-Armed Bandits with Generalized Temporally-Partitioned Rewards

Ronald C. van den Broek, Rik Litjens, Tobias Sagis, Nina Verbeeke, and Pratik Gajane

Eindhoven University of Technology
{r.c.v.d.broek, r.litjens, t.g.m.sagis, n.c.verbeeke}@student.tue.nl
p.gajane@tue.nl

Abstract. Decision-making problems of sequential nature, where decisions made in the past may have an impact on the future, are used to model many practically important applications. In some real-world applications, feedback about a decision is delayed and may arrive via partial rewards that are observed with different delays. Motivated by such scenarios, we propose a novel problem formulation called multi-armed bandits with generalized temporally-partitioned rewards. To formalize how feedback about a decision is partitioned across several time steps, we introduce *β-spread property*. We derive a lower bound on the performance of any uniformly efficient algorithm for the considered problem. Moreover, we provide an algorithm called `TP-UCB-FR-G` and prove an upper bound on its performance measure. In some scenarios, our upper bound improves upon the state of the art. We provide experimental results validating the proposed algorithm and our theoretical results.

Keywords: Multi-armed bandits · Delayed rewards · Temporally-partitioned rewards.

1 Introduction

The classical multi-armed bandit (MAB, or simply bandit) problem is a framework to model sequential decision-making [5]. In a MAB problem, the learning agent is faced with a finite set of K decisions or *arms*, and a decision taken by the agent is symbolized by pulling an arm. Feedback about the decisions taken is available to the agent via numerical rewards. Multi-arm bandit literature typically focuses on scenarios where rewards are assumed to arrive immediately after pulling an arm. In contrast, the works on delayed-feedback bandits (e.g., [8,10]) assume a delay between pulling an arm and the observation of its corresponding reward. In those studies, the reward is assumed to be concentrated in a single round that is delayed. This setting can be extended by allowing the reward to be partitioned into partial rewards that are observed with different delays. This type of bandit problem, known as MAB with Temporally-Partitioned Rewards (TP-MAB), was introduced by [11].

In the TP-MAB setting, an agent receives subsets of the reward over multiple rounds. The complete reward of an arm is the sum of the partial rewards

obtained by pulling the arm. [11] present α-smoothness to characterize the reward structure. The α-smoothness property states that the maximum reward in a group of consecutive partial rewards cannot exceed a fraction of the maximum reward (precise definition given in Definition 2). However, the assumption of α-smoothness does not fit well if the cumulative reward is not uniformly spread. In this article, we introduce a more generalized way of formulating how an arm's delayed cumulative reward is spread across several rounds.

As a motivating application, consider websites (e.g., Coursera, Khan Academy, edX) that provide Massive Open Online Courses (MOOCs). Such websites aim to provide users with useful recommendations for courses. This problem can be modeled as a TP-MAP problem. A course, which consists of a series of video lectures, might be thought of as an arm. A course can be recommended to a user by an agent, which corresponds to pulling an arm. When the student follows a course, the agent can observe partial rewards (e.g., by checking the watch time retention). In this setting, α-smoothness rarely captures the actual cumulative reward distribution. Many students watch the video lectures at the beginning of a course but never finish the last few lectures, making the spread of partial rewards non-uniform. As a result, the existing work on delayed-feedback bandits and the algorithms proposed by [11] may fail to recommend courses that are relevant for the user. Motivated by such scenarios, we investigate a more generalized way of formulating the reward structure.

Our Contributions

1. We introduce a novel MAB formulation with a generalized way of describing how an arm's delayed cumulative reward is distributed across rounds.
2. We prove a lower bound on the performance measure of any uniformly efficient algorithm for the considered problem.
3. We devise an algorithm `TP-UCB-FR-G` and prove an upper bound on its performance measure. The proven upper bounds are tighter than the state of the art in some scenarios.
4. We provide experimental results that validate the correctness of our theoretical results and the effectiveness of our proposed algorithm.

2 Background and Related Work

The *non-anonymous* delayed feedback bandit problem was considered in [8], where it is assumed that knowledge of which action resulted in a specific delayed reward is available. Recently, a variety of delayed-feedback scenarios were studied in MAB settings different from ours, such as linear and contextual bandits [1,15,13], non-stationary bandits [14].

The majority of past research on the delayed MAB setting assumes that the entire reward of an arm is observed at once, either after some bounded delay [8,10] or after random delays from an unbounded distribution with finite expectation [7,12]. Our article studies the setting in which the reward for an arm

is spread over an interval with a finite maximum delay value. This is consistent with the applications that we aim to model, such as MOOC providers mentioned in Section 1. To the best of our knowledge, this setting was first analyzed in [11]. They introduced the Multi-Armed Bandit with Temporally-Partitioned Rewards (TP-MAB) setting. In the TP-MAB setting, a stochastic reward that is received by pulling an arm is partitioned over partial rewards observed during a finite number of rounds followed by the pull. In [11], it is assumed that the arm rewards follow α-smoothness property (precise definition given in Definition 2).

While the study by [11] provides promising results, it is based on the strong assumption that the α-smoothness property holds. As a result, their proposed solutions are not suitable for a broader variety of scenarios where rewards are partitioned non-uniformly. As a remedy, we propose to use general distributions that can more accurately characterize how the received reward is partitioned. Consider a scenario (e.g., as described Section 1) in which additional information is available about how the cumulative reward is spread over the rounds. By generalizing the reward structure, our approach is able to handle partitioned rewards in which the maximum reward per round is not partitioned uniformly across rounds, such as those shown on the right side of Figure 1.

Two novel algorithms, `TP-UCB-EW` and `TP-UCB-FR`, leveraging α-smoothness property are introduced in [11]. The setup of `TP-UCB-FR` is most suitable for leveraging assumed distribution in a generalized setting. Subsequently, we use `TP-UCB-FR` as a baseline for our proposed algorithm.

3 Problem Formulation

Consider a MAB problem with K arms over a time horizon of T rounds, where $K, T \in \mathbb{N}$. At every round $t \in \{1, 2, \ldots, T\}$ an arm from the set of arms $\{1, 2, \ldots, K\}$ is pulled. The reward for an arm i is drawn from an unknown distribution on $[0, 1]$ with mean μ_i. The performance of an algorithm $\mathfrak{A}$ after T time steps is measured using expected *regret* denoted as $\mathcal{R}_T(\mathfrak{A})$.

Definition 1 (Regret). *The regret of an algorithm $\mathfrak{A}$ after T time steps is $\mathcal{R}_T(\mathfrak{A}) \coloneqq \mu^* T - \sum_{i=1}^{K} \mu_i \cdot \mathbb{E}[N_i(T)]$, where $\mu^* \coloneqq \max_{1 \leq i \leq K} \mu_i$ and $N_i(T) =$ number of times an arm i is selected till time t.*

The total reward is temporally partitioned over a set of rounds $T' = \{1, 2, ..., \tau_{\max}\}$. Let $x^i_{t,m} (m \in T')$ denote the partitioned reward that the learner receives at round m, after pulling the arm i at round t. It is known to the agent which arm pull produced this reward. The cumulative reward is completely collected by the learner after a delay of at most $\tau_{\max}$. Each per-round reward $x^i_{t,m}$ is the realization of a random variable $X^i_{t,m}$ with support in $[0, \overline{X}^i_m]$. The cumulative reward collected by the learner from pulling arm i at round t is denoted by r^i_t and it is the realization of a random variable R^i_t such that $R^i_t := \sum_{n=1}^{\tau_{\max}} X^i_{t,n}$ with support $[0, \overline{R}^i]$. Straightforwardly, we observe that $\overline{R}^i := \sum_{n=1}^{\tau_{\max}} \overline{X}^i_n$.

It is shown in [11] that, in practice, per-round rewards for an arm provide information on the cumulative reward of the arm. They introduce α-smoothness

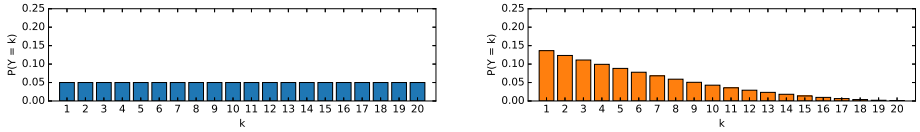

Fig. 1: α-smoothness reward distribution for the MOOC setting (left) and a near-perfect approximation of the reward distribution using β-spread (right)

property (defined in Definition 2) that partitions the rewards such that each partition corresponds to the sum of a set of consecutive per-round rewards. Formally, let $\alpha \in T'$ be such that α is a factor of $\tau_{\max}$. The cardinality of each partition, where we refer to partition as 'z-group', is denoted by $\phi := \frac{\tau_{\max}}{\alpha}$ with $\phi \in \mathbf{N}$. We can now define each z-group $z^i_{t,k}, k \in \{1, 2, ..., \alpha\}$ as the realization of a random variable $Z^i_{t,k}$, with support $[0, \overline{Z}^i_{\alpha,k}]$, such that for every k:

$$Z^i_{t,k} := \sum_{n=t+(k-1)\phi}^{t+k\phi-1} X^i_{t,n} \tag{1}$$

Definition 2 (α-smoothness). *For $\alpha \in \{1, ..., \tau_{max}\}$, the reward is α-smooth iff $\frac{\tau_{max}}{\alpha} \in \mathbb{N}$ and for each $i \in \{1, ..., K\}$ and $k \in \{1, 2, ..., \alpha\}$ the random variables $Z^i_{t,k}$ are independent and s.t. $\overline{Z}^i_{\alpha,k} = \overline{Z}^i_{\alpha} = \frac{\overline{R}^i}{\alpha}$.*

The α-smoothness property ensures that all temporally-partitioned rewards contribute towards bounding the values of future rewards within the same window. If the α-smoothness holds, then the maximum cumulative reward in a z-group $\overline{Z}^i_{\alpha,k}$ is equal for all z-groups $k \in \{1, 2, ..., \alpha\}$. Therefore, we can say that $\forall_k \in \{1, 2, ..., \alpha\}$, $\overline{Z}^i_{\alpha,k} = \overline{Z}^i_{\alpha}$. The assumption of α-smoothness is unsuitable for scenarios in which the cumulative reward is not uniformly partitioned across rounds. The goal of this article is to generalize the spread of the rewards across z-groups. To that end, one has to eliminate the assumption that every z-group has an equal probability of attaining a partial reward. To accomplish this, we replace α-smoothness with *β-spread property* that allows for modeling scenarios in which the cumulative reward is distributed non-uniformly across rounds i.e., a property that allows $\overline{Z}^i_{\alpha,k}$ to differ across z-groups.

Our Solution approach: β-spread property

Definition 3 (β-spread). *For $\alpha \in \{1, ..., \tau_{max}\}$, the reward is β-smooth if and only if*

1. *$\frac{\tau_{max}}{\alpha} = \phi$ with $\phi \in \mathbb{N}$,*
2. *the reward distribution can be described by a distribution $\mathcal{D}$ on a finite integer domain $\{1, 2, ..., \alpha\}$ with probability mass function $P_{\mathcal{D}}(k)$, and*

3. *for each* $i \in \{1, ..., K\}$ *and* $k \in \{1, 2, ..., \alpha\}$ *the random variables* $Z^i_{t,k}$ *are independent and s.t.* $\overline{Z}^i_{\alpha,k} = P_{\mathcal{D}}(k) \cdot \overline{R}^i$.

Based on prior information about how the cumulative reward is distributed over the rounds, the actual reward distribution can be approximated by a distribution $\widehat{\mathcal{D}}$ with corresponding probability mass function $P_{\widehat{\mathcal{D}}}(k)$, as long as it adheres to the definition of β-spread. We specify this, because the true reward distribution might not be known exactly in all cases. However, our solution approach requires at least some knowledge of the reward distribution.

As an example, consider the MOOC setting described at the end of Section 1. Consider the case where the watch time retention is considered a partial reward, and reduces linearly over time. The reward distribution under α-smoothness over the z-groups is illustrated in Figure 1 (left). Since we expect the partial reward to reduce linearly over time, the distribution of rewards under α-smoothness is inappropriate. Rather, we closely approximate the linear reduction with a Beta-binomial distribution with parameters $\alpha = 1$ and $\beta = 3$ (right), which will result in lower cumulative regret. In this article, we use Beta-binomial distributions frequently due to its capability to describe a wide variety of distributions.

4 Lower Bound on Regret

Using the β-spread property, we can derive the following lower bound for a uniformly efficient policy i.e., any policy with regret in $\mathcal{O}(T^x)$ with $x < 1$.

Theorem 1. *The regret of any uniformly efficient policy* $\mathfrak{U}$ *applied to a TP-MAB problem with the* β*-spread property after* T *time steps is lower bounded as*

$$\lim_{T\to+\infty} \inf \frac{\mathcal{R}_T(\mathfrak{U})}{\ln T} \geq \sum_{i:\mu_i<\mu^*} \frac{2}{(\alpha+1)} \sum_{\mathrm{k}=1}^{\alpha} \mathrm{kP}_{\mathcal{D}}(\mathrm{k}) \cdot \alpha \sum_{\mathrm{k}=1}^{\alpha} \left(\mathrm{P}_{\mathcal{D}}(\mathrm{k})\right)^2 \frac{\Delta_i}{\alpha \mathcal{KL}\left(\frac{\mu_i}{R_{\max}}, \frac{\mu^*}{R_{\max}}\right)}$$

where $\Delta_i \coloneqq \mu^* - \mu_i$ *and* $\mathcal{KL}(p, q) \coloneqq$ *Kullback-Leibler divergence between Bernoulli random variables with means* p *and* q *[9].*

Comparison with the Lower Bound given by [11] By assuming α-smoothness, the following lower bound for TP-MAB was proved in [11]:

$$\lim_{T\to+\infty} \inf \frac{\mathcal{R}_T(\mathfrak{U})}{\ln T} \geq \sum_{i:\mu_i<\mu^*} \frac{\Delta_i}{\alpha \mathcal{KL}\left(\frac{\mu_i}{R_{\max}}, \frac{\mu^*}{R_{\max}}\right)}. \tag{2}$$

Notice that the difference between the lower bound with the β-spread property and the lower bound derived in [11] lies in two factors. The first factor, $\frac{2}{(\alpha+1)} \sum_{\mathrm{k}=1}^{\alpha} \mathrm{kP}_{\mathcal{D}}(\mathrm{k})$ is equal to the normalized expected value of our assumed reward spread distribution. $\sum_{\mathrm{k}=1}^{\alpha} \mathrm{kP}_{\mathcal{D}}(\mathrm{k})$ calculates the expected value for the

chosen discrete distribution. $\frac{(\alpha+1)}{2}$ is the expected value when the chosen distribution is the uniform distribution. Hence, its inverse can be seen as a normalization term. The second factor, $\alpha \sum_{\mathrm{k}=1}^{\alpha} \left(\mathrm{P}_{\mathcal{D}}(\mathrm{k})\right)^2$, can be seen as a normalized approximation of the *index of coincidence* [6] between rewards. The index of coincidence, $\sum_{\mathrm{k}=1}^{\alpha} \left(\mathrm{P}_{\mathcal{D}}(\mathrm{k})\right)^2$ determines the probability of two reward points being observed in the same z-group. Its minimal value equals $\frac{1}{\alpha}$ and occurs when the α-smoothness property holds (uniform distribution). The value is maximal and equal to 1 if all rewards fall into one z-group. Multiplying the index of coincidence with α gives this factor more weight in the lower bound and extends the domain from $[\frac{1}{\alpha}, 1]$ to $[1, \alpha]$. This essentially means that it is 'harder' for algorithms to perform well when the rewards come in bulk, rather than over the course of multiple rounds. The lower bound given in Theorem 1 resolves to the lower bound given by [11] in Eq.(2) in case of α-smoothness. However, our lower bound for the considered problem setting is tighter when $\frac{2}{(\alpha+1)} \sum_{\mathrm{k}=1}^{\alpha} \mathrm{kP}_{\mathcal{D}}(\mathrm{k}) \cdot \alpha \sum_{\mathrm{k}=1}^{\alpha} \left(\mathrm{P}_{\mathcal{D}}(\mathrm{k})\right)^2 > 1$. This means that rewards that are expected to be observed late, or rewards that come all at the same time, contribute negatively to the performance of an algorithm in the described setting. Rewards that are expected to be observed early or are more spread out contribute positively.

Proof Sketch for Theorem 1 We start by constructing two MAB problem instances that call for different behaviors from the algorithm attempting to solve them. Then, we use the change-of-distribution argument to show that any uniformly efficient algorithm cannot efficiently distinguish between these instances. Please consult the extended version of this article given in [3] for the complete proof of Theorem 1.

5 Proposed Algorithm and Regret Upper Bound

In this section, we propose an algorithm that makes use of the β-spread property in the TP-MAB setting and prove an upper bound on its regret.

5.1 Proposed Algorithm: `TP-UCB-FR-G`

Our proposed algorithm `TP-UCB-FR-G` given below is a non-trivial extension of `TP-UCB-FR` [11]. In `TP-UCB-FR-G`, the most significant modification is the confidence interval c_{t-1}^{i} which is rigorously built to suit the β-spread property.

As input, the algorithm takes a smoothness constant $\alpha \in [\tau_{max}]$, a maximum delay τ_{max} and a probability mass function $P_{\widehat{D}}$. The algorithm uses $P_{\widehat{D}}$ to be able to give a proper judgment of an arm before all the delayed partial rewards are observed. This is realized by replacing the not yet received partial rewards with fictitious realizations, or in other words, the expected estimated rewards. At round t, the fictitious reward vectors are associated with each arm pulled in the span $H := \{t - \tau_{max} + 1, ..., t - 1\}$. These fictitious rewards are denoted by $\tilde{\boldsymbol{x}}_h^i = \left[\tilde{x}_{h,1}^i, \ldots, \tilde{x}_{h,\tau_{\max}}^i\right]$ with $h \in H$, where $\tilde{x}_{h,j}^i := x_{h,j}^i$, if $h + j \leq t$ (the reward

Algorithm 1 TP-UCB-FR-G

1: **Input:** $\alpha \in [\tau_{max}], \tau_{max} \in \mathbb{N}^*, P_{\widehat{\mathcal{D}}}$
2: **for** $t \in \{1, ..., K\}$ **do**
3: Pull an arm $i_t \leftarrow t$
4: **for** $t \in \{K+1, ..., T\}$ **do**
5: **for** $i \in \{1, ..., K\}$ **do**
6: Compute $\hat{R}^i_{t-1}$ and c^i_{t-1} as in (3) and (4)
7: $u^i_{t-1} \leftarrow \hat{R}^i_{t-1} + c^i_{t-1}$
8: Pull arm $i_t \leftarrow z = \text{argmax}_{i \in [K]}\, u^i_{t-1}$
9: Observe $x^{i_h}_{h,t-h+1}$ for $h \in \{t - \tau_{max} + 1, ..., t\}$

has already been seen), and $\tilde{x}^i_{h,j} = 0$, if $h + j > t$ (the reward will be seen in the future). The corresponding fictitious cumulative reward is $\tilde{r}^i_h := \sum_{j=1}^{\tau_{\max}} \tilde{x}^i_{h,j}$. In the initialization phase of the algorithm (lines 2-3), each arm is pulled once. Later, at each time step t, the upper confidence bounds u^i_{t-1} are determined for each arm i by computing the estimated expected reward $\hat{R}^i_{t-1}$ and confidence interval c^i_{t-1} using Eq. (3) and (4) respectively.

$$\hat{R}^i_{t-1} := \frac{1}{N_i(t-1)} \left(\sum_{h=1}^{t-\tau_{\max}} r^i_h \mathbb{1}_{\{i_h = i\}} + \sum_{h \in H} \tilde{r}^i_h \mathbb{1}_{\{i_h = i\}} \right), \tag{3}$$

$$c^i_{t-1} := \frac{\phi \bar{R}^i}{N_i(t-1)} \sum_{k=1}^{\alpha} k P_{\widehat{\mathcal{D}}(k)} + \bar{R}^i \sqrt{\frac{2 \ln(t-1) \sum_{k=1}^{\alpha} \left(P_{\widehat{\mathcal{D}}(k)} \right)^2}{N_i(t-1)}}, \tag{4}$$

where $N_i(t-1)$ is the number of times arm i has been pulled up to $t-1$ and i_h represents the arm that was pulled at time h. The algorithm then pulls the arm i with the highest upper confidence bound u^i_{t-1} and observes its rewards.

5.2 Regret Upper Bound of TP-UCB-FR-G

Theorem 2. *In the TP-MAB setting with β-spread reward, the regret of TP-UCB-FR-G after T time steps with $P_{\widehat{\mathcal{D}}}(k)$ matching $P_{\mathcal{D}}(k)$ is upper bounded as*

$$\begin{aligned}
&\mathcal{R}_T(\textit{TP-UCB-FR-G}) \\
&\leq \sum_{i:\mu_i < \mu^*} \frac{4 \ln T \, (\bar{\mathrm{R}}^{\mathrm{i}})^2 \sum_{\mathrm{k}=1}^{\alpha} \left(\mathrm{P}_{\mathcal{D}}(\mathrm{k}) \right)^2}{\Delta_i} \cdot \left(1 + \sqrt{1 + \frac{\Delta_i \phi \sum_{\mathrm{k}=1}^{\alpha} \mathrm{k} \mathrm{P}_{\mathcal{D}}(\mathrm{k})}{\bar{R}^i \ln T \sum_{k=1}^{\alpha} \left(P_{\mathcal{D}}(k) \right)^2}} \right) \\
&+ 2\phi \sum_{\mathrm{k}=1}^{\alpha} \mathrm{k} \mathrm{P}_{\mathcal{D}}(\mathrm{k}) \sum_{i:\mu_i < \mu^*} \bar{R}^i + \left(1 + \frac{\pi^2}{3} \right) \sum_{i:\mu_i < \mu^*} \Delta_i.
\end{aligned}$$

Observe that $\sum_{\mathrm{k}=1}^{\alpha} \mathrm{k} \mathrm{P}_{\mathcal{D}}(\mathrm{k})$ is equal to the expected value of our assumed reward spread distribution, similar to the factor in the lower bound but not normalized. The other factor is the index of coincidence $\sum_{\mathrm{k}=1}^{\alpha} \left(\mathrm{P}_{\mathcal{D}}(\mathrm{k}) \right)^2$, which also occurs in the lower bound but is not weighted for the upper bound.

Table 1: Parameter values for Beta-Binomial distributions

Distribution name	α	β	Distribution name	α	β	Distribution name	α	β
`extreme_begin`	1	100	`begin_middle`	2	4	`end`	8	2
`very_begin`	1	16	`middle`	5	5	`very_end`	16	1
`begin`	2	8	`middle_end`	4	2			

Comparison with the Upper Bound given in [11] Let us compare our upper bound with the upper bound given in [11]. For the latter bound to hold, the α estimate given as input to their algorithm has to match the α of the real reward distribution as well. Note that $\sum_{\mathrm{k}=1}^{\alpha} \mathrm{kP}_{\mathcal{D}}(\mathrm{k}) = \frac{\alpha+1}{2}$ in case of α-smoothness. For other assumed distributions with $\sum_{\mathrm{k}=1}^{\alpha} \mathrm{kP}_{\mathcal{D}}(\mathrm{k}) < \frac{\alpha+1}{2}$ our upper bound on the regret is lower. Furthermore, choosing a β-spread distribution as input with a low mean and a low index of coincidence will result in a better upper bound, by Theorem 2, compared to choosing $\widehat{\mathcal{D}}$ with rewards centered towards the end (high mean) and not spread out (high index of coincidence).

Proof Sketch of Theorem 2 Here we provide a proof sketch for Theorem 2. Please refer to the extended version of the article given in [3] for the complete proof. The approach can be divided into three steps. Firstly, we show that the probability that an optimal arm is estimated significantly lower than its mean is bounded by t^{-4}. Secondly, we show the probability of a suboptimal arm being estimated significantly higher than its mean is bounded by t^{-4}. Finally, we assess the algorithm's ability to differentiate between optimal and suboptimal arms.

6 Experimental Results

We compare our proposed algorithm `TP-UCB-FR-G` with `TP-UCB-FR` [11], `UCB1` [2], and `Delayed-UCB1` [8]. We use two experimental settings – a synthetically generated environment and a real-world playlist recommendation scenario. In these settings, we inherit learners used in the provided experiments in [11], and create new learner configurations using `TP-UCB-FR-G`. As input distributions for the new learners, we use Beta-Binomial distributions with unique parameter values for each learner. The Beta-Binomial distribution gives us the opportunity to model extreme scenarios, which should result in more insightful experimental results. We observe that other distributions do not grant the flexibility of a Beta-Binomial distribution, as demonstrated in experiments deferred to the appendix given with the extended version of this article [3]. In the plots under this section, we use the notation `TP-UCB-FR-G`(α, `dist_name`) to denote a learner for our algorithm, where `dist_name` is the name of the Beta-Binomial distribution for which the exact parameters are shown in Table 1. Further details about the used Beta-Binomial distributions and experimental settings are given in the appendix of the extended version of this article [3].

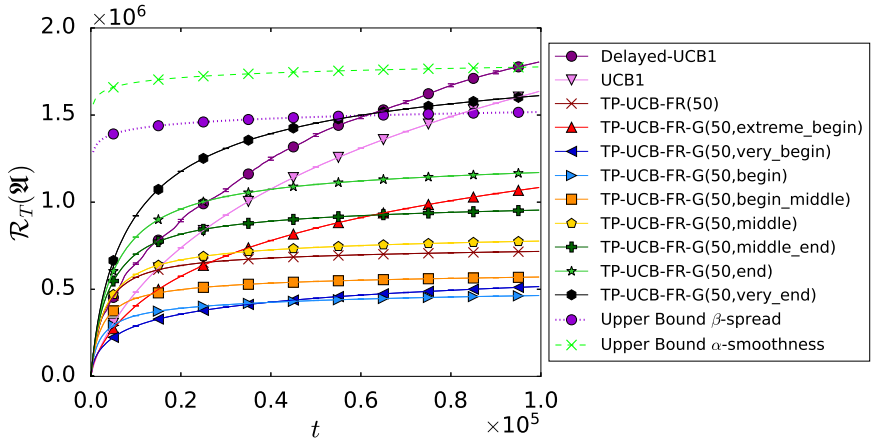

Fig. 2: Regret against time for Setting 1.2 with $\tau_{\max} = 100$ and $\alpha_{est} = 50$

6.1 Setting 1: Synthetic Environment

The distribution of rewards in this setting are s.t. $Z^i_{t,k} \sim \frac{\overline{R}^i}{\alpha} Beta[a^i_k, b^i_k]$ where $Beta$ is a Beta distribution with a, b s.t. rewards are distributed according to the spread of the corresponding setting. Again, we model $K = 10$ arms, an α-smoothness constant of $\alpha = 20$ and a maximum reward s.t. convergence to an optimal arm takes longer. That is, $\overline{R}^i = 100\zeta^i$ with $\zeta \in \{1, 3, 6, 9, 12, 15, 18, 21, 22, 23\}$. However, there is a difference in the $\tau_{\max}$, α_{est} and the parameters used for the assumed Beta distribution by the learners. The exact configurations can be found in the appendix given with the extended version of this article [3]. In general, there are 8 combinations consisting of 4 configurations with 2 scenarios each. The configurations differ in $\tau_{\max}$ and α_{est}, whereas the scenarios differ in distribution parameters. Generally, there is one where the rewards are observed late after the pull (Setting 1.1), and one where the results are observed just after the pull (Setting 1.2). We use learners with the distributions given in Table 1).

Results Let us denote $\Delta(s_1, s_2)$ for $s_1, s_2 \in \{\texttt{Setting 1.1}, \texttt{Setting 1.2}\}$ as the absolute difference in cumulative regret between Settings s_1 and s_2. The pairwise differences in cumulative regret observed between settings are marginal. As an example, $\Delta(\texttt{Setting 1.1}, \texttt{Setting 1.2}) \approx 4.8{\times}10^3$ for $\texttt{TP-UCB-FR-G}(50, \texttt{very\_end})$ which is the highest difference in average regret observed across all compared settings. Since the regret of $\texttt{TP-UCB-FR-G}(50, \texttt{very\_end})$ averaged over T is $\approx 1.61 \times 10^6$, the observed change of $\approx 0.3\%$ is negligible. Furthermore, the same experiment performed with different values for both $\tau_{\max}$ and α_{est} seems to confirm the same marginal change. For example, Setting 1 for $\tau_{\max} = 200$ and $\alpha_{est} = 20$ results in a maximum change in average regret of only $\approx 0.5\%$. These findings indicate that the performance of `TP-UCB-FR-G` learners in a uniformly distributed aggregate rewards setting is indistinguishable from that in a non-uniformly distributed aggregate rewards setting. Therefore, we can state that

TP-UCB-FR-G(α, begin) delivers a significant performance increase compared to the learner proposed by [11]. The gain that we observe for the mentioned settings is as high as $\approx 48.2\%$. An extensive performance summary is deferred to the appendix given with the extended version of this article [3].

In Figure 2, the theoretical upper bound of TP-UCB-FR-G as well as the upper bound of the TP-UCB-FR algorithm is plotted on top of the results for the Setting 1.2. The figure shows that the upper bound proposed in this article is tighter in this setting. Note that the theoretical upper bounds for TP-UCB-FR-G and TP-UCB-FR only hold for specific learners that assume the data generating distribution precisely and that the 'very end' learner exceeds the β-spread upper bound. This shows another reason to estimate the assumed distribution optimistically.

6.2 Setting 2: Spotify Playlists

We evaluate our algorithm on real-world data by addressing the user recommendation problem introduced by [11], using the Spotify dataset [4]. We select the $K = 6$ most played playlists as the arms to be recommended. Each time a playlist i is selected, the corresponding rewards x_t^i for the first $N = 20$ songs are sampled from the dataset. In this setting, the α-smoothness is $\alpha = 20$, the maximum delay $\tau_{\max} = 4N = 80$ and the results are averaged over 100 independent runs.

Results In Figure 3, we observe that optimistic learners significantly outperform the baseline TP-UCB-FR(20). We focus on TP-UCB-FR-G(20, begin), since it is by far the best-performing learner. This learner achieves a decrease of $\approx 26.3\%$ in regret averaged over time horizon T. Table 2 summarizes the performance gains of TP-UCB-FR-G learners in the Spotify setting.

We also observe that overly optimistic learners perform worse than TP-UCB-FR(20). However, TP-UCB-FR-G(20, begin) outperforms TP-UCB-FR(20) for larger t, making it a better option for playlist recommendations.

7 Concluding Remarks and Future Work

In this paper, we model sequential decision-making problems with delayed feedback using a novel formulation called multi-armed bandits with generalized temporally-partitioned rewards. To generalize delayed reward distributions, we introduce the β-spread property. We establish a tighter lower bound for the

Table 2: TP-UCB-FR-G learners and their decrease in regret

Learner TP-UCB-FR-G($\alpha_{est} = 20$)	Regret ($\times 10^4$)	Decrease (%)
extreme_begin	4.40	≈ -72.5
very_begin	2.40	≈ 5.9
begin	**1.88**	$\approx$ **26.3**
begin_middle	2.11	≈ 17.2

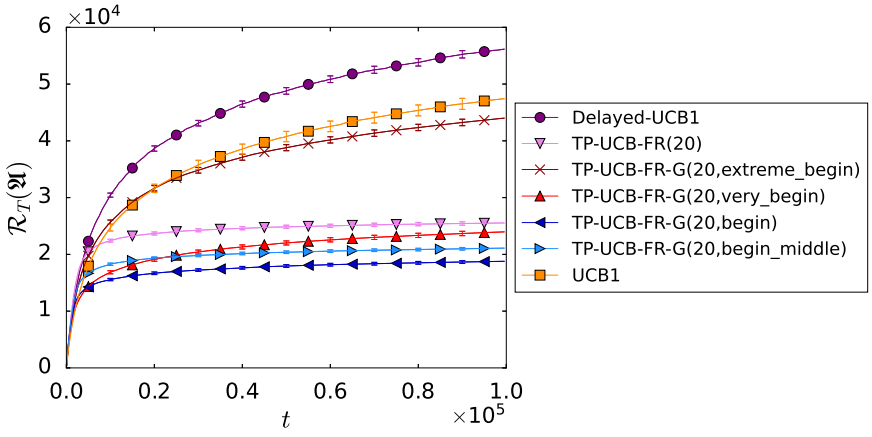

Fig. 3: Regret against time for begin-oriented learners in the Spotify setting

TP-MAB setting with the β-spread property compared to the TP-MAB setting with the α-smoothness property. We also introduce the `TP-UCB-FR-G` algorithm, which exploits the β-spread property. We demonstrate that in certain scenarios, the upper bound of this algorithm can be lower than that of the `TP-UCB-FR` algorithm, thus surpassing the upper bounds of the classical UCB1 and Delayed-UCB1 algorithms as well. Finally, we demonstrate that our algorithm outperforms `TP-UCB-FR` and other UCB algorithms in diverse experiments using synthetic and real-world data, achieving a remarkable 26.3% decrease in regret compared to the state-of-the-art `TP-UCB-FR` algorithm.

A possible future research direction is to explore removing the restriction of the β-spread property to discrete probability distributions bounded by a finite domain of size α. This can enhance the algorithm's flexibility and broaden its practical applications. Additionally, a valuable extension involves considering scenarios where arms are treated as subsets, each assigned distinct α-values and distributions. Moreover, an intriguing area of exploration involves studying scenarios where the partitioned reward time span, denoted as $\tau_{\max}$, varies.

Acknowledgements

This work is supported by the Dutch Research Council (NWO) in the framework of the TEPAIV research project (project number 612.001.752).

References

1. Arya, S., Yang, Y.: Randomized allocation with nonparametric estimation for contextual multi-armed bandits with delayed rewards. Statistics & Probability Letters **164**, 108818 (2020). https://doi.org/https://doi.org/10.1016/j.spl.2020.108818

2. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine Learning 2002 47:2 **47**, 235–256 (5 2002). https://doi.org/10.1023/A:1013689704352, https://link.springer.com/article/10.1023/A:1013689704352
3. van den Broek, R.C., Litjens, R., Sagis, T., Siecker, L., Verbeeke, N., Gajane, P.: Multi-armed bandits with generalized temporally-partitioned rewards (2023), https://arxiv.org/abs/2303.00620
4. Brost, B., Mehrotra, R., Jehan, T.: The music streaming sessions dataset. CoRR **abs/1901.09851** (2019), http://arxiv.org/abs/1901.09851
5. Bubeck, S., Cesa-Bianchi, N.: Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends in Machine Learning **5**(1), 1–122 (2012). https://doi.org/10.1561/2200000024
6. Friedman, W.F.: The index of coincidence and its applications in cryptanalysis, vol. 49. Aegean Park Press California (1987)
7. Gael, M.A., Vernade, C., Carpentier, A., Valko, M.: Stochastic bandits with arm-dependent delays. In: Proceedings of the 37th International Conference on Machine Learning. vol. 119, pp. 3348–3356 (13–18 Jul 2020), https://proceedings.mlr.press/v119/gael20a.html
8. Joulani, P., György, A., Szepesvári, C.: Online learning under delayed feedback. In: Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28. ICML'13 (2013)
9. Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics **22**(1), 79–86 (1951)
10. Mandel, T., Liu, Y.E., Brunskill, E., Popović, Z.: The queue method: Handling delay, heuristics, prior data, and evaluation in bandits. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
11. Romano, G., Agostini, A., Trovò, F., Gatti, N., Restelli, M.: Multi-armed bandit problem with temporally-partitioned rewards: When partial feedback counts. Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (2022). https://doi.org/10.24963/ijcai.2022/472
12. Vernade, C., Cappé, O., Perchet, V.: Stochastic Bandit Models for Delayed Conversions. In: Conference on Uncertainty in Artificial Intelligence (Aug 2017), https://hal.science/hal-01545667
13. Vernade, C., Carpentier, A., Lattimore, T., Zappella, G., Ermis, B., Brückner, M.: Linear bandits with stochastic delayed feedback. In: Proceedings of the 37th International Conference on Machine Learning. vol. 119, pp. 9712–9721 (13–18 Jul 2020), https://proceedings.mlr.press/v119/vernade20a.html
14. Vernade, C., György, A., Mann, T.A.: Non-stationary delayed bandits with intermediate observations. In: Proceedings of the 37th International Conference on Machine Learning. ICML'20 (2020)
15. Zhou, Z., Xu, R., Blanchet4, J.: Learning in Generalized Linear Contextual Bandits with Stochastic Delays. Curran Associates Inc. (2019)