# Multi-Armed Bandit Problem with Temporally-Partitioned Rewards: When Partial Feedback Counts

**Giulia Romano** , **Andrea Agostini** , **Francesco Trovò** , **Nicola Gatti** and **Marcello Restelli**
Politecnico di Milano, Piazza Leonardo da Vinci 32, I-20133, Milan, Italy
{giulia.romano, francesco1.trovo, nicola.gatti, marcello.restelli}@polimi.it,
andrea1.agostini@mail.polimi.it

## Abstract

There is a rising interest in industrial online applications where data becomes available sequentially. Inspired by the recommendation of playlists to users where their preferences can be collected during the listening of the entire playlist, we study a novel bandit setting, namely *Multi-Armed Bandit with Temporally-Partitioned Rewards* (TP-MAB), in which the stochastic reward associated with the pull of an arm is partitioned over a finite number of consecutive rounds following the pull. This setting, unexplored so far to the best of our knowledge, is a natural extension of delayed-feedback bandits to the case in which rewards may be dilated over a finite-time span after the pull instead of being fully disclosed in a single, potentially delayed round. We provide two algorithms to address TP-MAB problems, namely, `TP-UCB-FR` and `TP-UCB-EW`, which exploit the partial information disclosed by the reward collected over time. We show that our algorithms provide better asymptotical regret upper bounds than delayed-feedback bandit algorithms when a property characterizing a broad set of reward structures of practical interest, namely $\alpha$-*smoothness*, holds. We also empirically evaluate their performance across a wide range of settings, both synthetically generated and from a real-world media recommendation problem.

## 1 Introduction

Sequential decision-making occurs in many real-world scenarios such as clinical trials, recommender systems, web advertising, and e-commerce. Inspired by these applications, many different flavours of the multi-armed bandit (MAB) setting have been investigated. A crucial role is played by the time the reward is observed. In many cases, the reward is subject to a *delay*, and such a delay, if not sufficiently short, can prevent the design of algorithms that are effective in practice. Online learning with delayed feedback has received considerable attention in recent years, and several results are available in the literature, *e.g.*, see the seminal work by Joulani *et al.* [2013]. A major distinction in MABs with delayed feedback concerns the nature of the rewards, which may be stochastic [Mandel *et al.*, 2015; Cella and Cesa-Bianchi, 2020] or adversarial [Bistritz *et al.*, 2019; Thune *et al.*, 2019; van der Hoeven and Cesa-Bianchi, 2021].

Our work focuses on a special class of bandit problems with stochastic and delayed rewards, in which we can get partial feedback over time. More precisely, we study a novel setting, namely MAB with Temporally-Partitioned Rewards (TP-MAB), in which the reward associated with an action, a.k.a. *arm*, chosen at a given round is collected during a finite number of rounds following the choice, according to an unknown probability distribution. In classical delayed-feedback bandits (see, *e.g.*, Joulani *et al.* [2013]), the reward is concentrated in a single round that is (stochastically) delayed w.r.t. the round in which the learner pulled the corresponding arm. TP-MABs naturally extend this setting by allowing the reward to be partitioned into multiple elements that are collected with different delays. We call arm's *per-round reward* the partial reward observed by the learner in a single round, which is assumed to be the realization of a random variable with an unknown probability distribution. We call arm's *cumulative reward* the random variable given by the sum of all the per-round rewards obtained by pulling an arm. While the per-round reward can be observed round by round, the cumulative reward is revealed only at the end. Notice that, in a single round, the learner observes a per-round reward for each previously pulled arm whose cumulative reward is not terminated yet. Our goal is to find a policy to maximize the cumulative reward, exploiting the per-round rewards as intermediate signals on the arm performance.

**Motivating applications.** A motivating example for TP-MABs is recommending media content and, in particular, song playlists to a class of users (*i.e.*, users sharing similar characteristics). In this setting, each arm corresponds to a playlist. The reward is measured in listening time (proportional to the user's appreciation). The goal is to find the playlist that maximizes the reward. The recommendation system suggests a playlist to a new user at each round, whose appreciation is revealed through multiple steps. In particular, every partial observation corresponds to a song in the playlist, and the associated reward is positive if the user listens to that song and non-positive otherwise. The cumulative reward provided by recommending a playlist to a single user corresponds to the sum of the reward terms from all the playlist songs. Notice that the playlist cannot be trivially modeled as

a collection of independent songs, as their order in the playlist affects the user's behavior. In the classical delayed-feedback bandit setting, the feedback on the recommended playlist is obtained only once the user finishes listening to the entire playlist. However, the platform monitors whether every song is listened to or skipped by the user. Therefore, clues on the performances of the recommended arm can be exploited *before* the user finishes the playlist.

Another scenario captured by the TP-MAB framework is the evaluation of medical treatments taking place over a long period of time. In this setting, the per-round reward corresponds to the patient's state of health at each daily/weekly medical check, and the goal is to find the treatment providing the greatest overall benefit to the patient. In the case of severe pathologies, such as cancer, this type of *partial information* would span several months if not years, providing valuable insights that would be otherwise ignored. Applying a standard delayed-MAB approach to this scenario, *i.e.*, taking decisions only at the end of each treatment cycle, could negatively affect the time required to select an effective medical treatment. In this type of setting, we argue that the partial information provided by patients in periodic medical checks should be used to speed up the learning process.

**Original Contributions.** Initially, we focus on the lower bound of TP-MABs, showing that the TP-MAB setting has the same regret lower bound of the standard delayed MAB setting when there is no further assumption about how the rewards are partitioned over time. Since in many practical applications of interest the cumulative reward of each arm does not concentrate excessively in a short sub-range of rounds, we introduce a property describing how the maximum per-round reward distributes. We call this property $\alpha$-*smoothness* where $\alpha \geq 1$. In particular, the minimum value of $\alpha = 1$ corresponds to the case in which there is no structure and, therefore, the maximum per-round reward can be the entire cumulative reward. On the other hand, the maximum value of $\alpha$ is equal to the maximum delay and corresponds to the case in which the cumulative reward distributes evenly over time. Thus, the maximum per-round reward decreases as the value of $\alpha$ increases. We show that the lower bound of this setting is of a factor $1/\alpha$ smaller than that when $\alpha$-smoothness does not hold. Then, we design two novel algorithms, namely `TP-UCB-FR` and `TP-UCB-EW`, suited for the TP-MAB setting, which exploit partial feedback and the $\alpha$-smoothness property. We show that the regret of `TP-UCB-FR` is $\mathcal{O}(\ln T/\alpha)$, where $T$ is the time horizon of the learning process, and the regret of `TP-UCB-EW` is $\mathcal{O}(\ln T)$. A comprehensive analysis the regret bounds of our and state-of-the-art algorithms in various settings can be found in Table 3 (in Appendix A for reasons of space). Finally, we experimentally show that our algorithms outperform the state of the art over synthetically generated and a real-world playlist recommendation scenario.

**Related Works.** To the best of our knowledge, ours is the first work addressing a bandit problem in which the reward from a pull is partitioned across multiple rounds. The most related works concern the Delayed-MAB setting, such as the seminal paper by Joulani *et al.* [2013], which summarizes the known results on the regret upper bounds of online learning algorithms. They also provide a modification of the well-known UCB1 algorithm from Auer *et al.* [2002] for the delayed-feedback setting, called Delayed-UCB1. More recently, a variety of delayed-feedback scenarios were studied investigating directions different from ours, such as linear and contextual (Arya and Yang [2020], Vernade *et al.* [2020a], Zhou *et al.* [2019]), non-stationary (Vernade *et al.* [2020b]) bandits under delayed feedback. Pike-Burke *et al.* [2018] and Cesa-Bianchi *et al.* [2018] also analyze the case of delayed, aggregated, and anonymous feedback. For clarity, we remark that, in our work, per-round rewards corresponding to different pulls can be received in the same round, and it is known from which arm they were generated. Many works apply bandits to practical scenarios, *e.g.*, scheduling [Cayci *et al.*, 2019], advertising [Nuara *et al.*, 2018; Castiglioni *et al.*, 2022; Nuara *et al.*, 2022], pricing [Trovò *et al.*, 2018], and delayed feedback settings [Vernade *et al.*, 2017].

Works from the bandit literature, such as the ones by Dudik *et al.* [2011], Desautels *et al.* [2014], Neu *et al.* [2013], rely on known constant delays or maximum delay values. Similarly, in our work, we assume a maximum finite delay equal to $\tau_{\max}$, which is compliant with the real-world scenarios we aim at modeling, *e.g.*, in the above example of playlist recommendations, an infinite $\tau_{\max}$ would correspond to a playlist of an infinite number of songs. According to the terminology used in the delayed-MAB literature, our setting is *uncensored*, meaning that the reward provided by a given action is eventually observed after a finite maximum delay. Conversely, many works in the field, such as, *e.g.*, Manegueu *et al.* [2020] and Vernade *et al.* [2017], deals with random delays from an unbounded distribution with finite expectation.

## 2 Problem Formulation

Consider a MAB problem with $K \in \mathbb{N}^*$ arms, over a time horizon of $T \in \mathbb{N}^*$ rounds. At every round $t \in [T]$, the learner pulls an arm $i \in \mathcal{A} = [K]$ and, from the pull of that arm, gets a *per-round reward* $x^i_{t,m-t+1}$ at every round $m \in \{t, \dots, t+\tau_{\max}-1\}$, where $\tau_{\max} \in \mathbb{N}^*$ is the time span over which the reward is partitioned.[1] In particular, $\tau_{\max} - 1$ is the maximum delay affecting the observation of a per-round reward, whose value is known to the learner. Therefore, at round $t+\tau_{\max}-1$, the cumulative reward from pulling arm $i$ at round $t$ is completely collected by the learner. Furthermore, we denote by $\boldsymbol{x}^i_t = [x^i_{t,1}, \dots, x^i_{t,\tau_{\max}}]$ the vector of per-round rewards collected from pulling arm $i$ at round $t$. For every $j \in [\tau_{\max}]$, the per-round reward $x^i_{t,j}$ is a realization of a random variable $X^i_{t,j}$ with support $[\underline{X}^i_j, \overline{X}^i_j]$. The cumulative reward collected from pulling arm $i$ at round $t$ is denoted by $r^i_t$, and it is the realization of the random variable $R^i_t := \sum_{j=1}^{\tau_{\max}} X^i_{t,j}$, with support $[\underline{R}^i, \overline{R}^i]$, where $\underline{R}^i := \sum_{j=1}^{\tau_{\max}} \underline{X}^i_j$, and $\overline{R}^i := \sum_{j=1}^{\tau_{\max}} \overline{X}^i_j$. For every $i \in \mathcal{A}$ and $t \in [T]$, we assume that the

[1] We denote by $[n]$ the set $\{1, \dots, n\}$

variables $R_t^i$ are independent with mean $\mu_i := \mathbb{E}[R_t^i]$.[2]

A policy $\mathfrak{U}$ is an algorithm that at each round $t$ chooses an arm $i_t \in [K]$ . The performance of a policy $\mathfrak{U}$ is evaluated in terms of *pseudo-regret*, defined as the cumulative loss due to playing suboptimal arms during the time horizon $T$, formally:

$$\mathcal{R}_T(\mathfrak{U}) = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{i_t}\right],$$

where $\mu^* = \max_{i\in\mathcal{A}}\{\mu_i\}$ is the expected reward of the optimal arm $i^*$, and the expectation is taken w.r.t. the stochasticity of the policy $\mathfrak{U}$. Notice that we adopt the concept of pseudo-regret as for standard bandits, unlike what is done by Vernade *et al.* [2017], since our choice allows for a direct comparison with the vast prior work on delayed bandits.

In what follows, we cast the playlist recommendation problem, described in the introduction, in the TP-MAB setting.

**Example 1** (Playlist Recommendation)**.** *At each round $t$, a new user enters the platform, which provides a playlist suggestion. The different arms $i$ are the available playlists to suggest, each composed of $N$ songs. Songs are characterized by $4$ listening levels (from "skipped" to "complete"), each associated with a different Bernoulli random variable representing the corresponding per-round reward. The vector of realized per-round rewards of song $k \in [N]$ is $[x^i_{t,4(k-1)+1}, x^i_{t,4(k-1)+2}, x^i_{t,4(k-1)+3}, x^i_{t,4(k-1)+4}]$. Each variable assumes a value of $1$ if the user reaches the corresponding level, and a value of $0$ if the user stops listening to the song before that level. The cumulative reward $R_t^i$ for pulling arm $i$ at round $t$ is the sum of the rewards from the songs in the playlist, and the time span over which the platform observes the reward is $\tau_{\max} = 4N$.*

We show that the TP-MAB problem has a lower-bound on the regret of the same order of the delayed-feedback bandit problem. The rationale is that no better lower bound is possible as delayed-feedback MABs with a finite delay are a subclass of TP-MABs whose reward vector $\boldsymbol{x}_t^i$ has a single non-zero element for each $i \in \mathcal{A}$ and $t \in [T]$. Most interestingly, the worst-case instance for the regret lower bound in the TP-MAB setting is the delayed-feedback bandit.[3]

**Theorem 1.** *The regret of any uniformly efficient policy $\mathfrak{U}$ applied to the TP-MAB problem is bounded from below by:*

$$\lim\inf_{T\to+\infty} \frac{\mathcal{R}_T(\mathfrak{U})}{\ln T} \geq \sum_{i:\mu_i<\mu^*} \frac{\Delta_i}{KL\left(\frac{\mu_i}{\overline{R}_{\max}}, \frac{\mu^*}{\overline{R}_{\max}}\right)}, \quad (1)$$

*where $\Delta_i := \mu^* - \mu_i$ is the expected loss suffered by the learner if the arm $i$ is chosen instead of the optimal one $i^*$, $\overline{R}_{\max} := \max_{i\in[K]} \overline{R}^i$, and $KL(p, q)$ is the Kullback-Leibler divergence between Bernoulli r.v. with means $p$ and $q$.*[4]

Notice that the lower bound holds for general TP-MAB problems. In the following section, we show that focusing on a broad subset of instances of practical interest, we can design algorithms with a better regret upper bound.

[2]W.l.o.g., we assume $\underline{X}^i_j = 0, \forall i \in [K], \forall j \in [\tau_{\max}]$.

[3]All the proofs are deferred to Appendix B for space reasons. See https://trovo.faculty.polimi.it/01papers/romano2022multi.pdf.

[4]An uniformly efficient policy chooses the suboptimal arms on average $o(t^a)$ times $(0 < a < 1)$ over $t$ rounds.

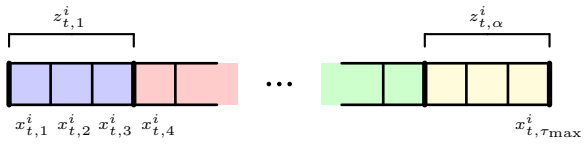


Figure 1: Example of $\alpha$-smooth reward with $\phi = 3$.

## 3 $\alpha$-Smoothness Property

From Theorem 1, we know that we cannot design algorithms with regret upper bounds better than those of the algorithms for the delayed-feedback bandit setting. Nonetheless, in practice, collecting per-round rewards can provide useful information on the cumulative reward of an arm. However, as already pointed out by Manegueu *et al.* [2020] for the standard delayed-feedback setting, zero rewards are ambiguous since they do not give any information on future rewards. In the general setting, small per-round rewards observed in the first rounds after the pull are not much informative to bound the values of future ones. To avoid this, we focus on those problems in which the maximum reward realized over a few rounds cannot exceed a fraction of the maximum reward $\overline{R}^i$.

Let us consider $\alpha \in [\tau_{\max}]$ s.t. $\alpha$ is a factor of $\tau_{\max}$, *i.e.*, $\frac{\tau_{\max}}{\alpha} =: \phi$ and $\phi \in \mathbb{N}$.[5] Let us define the vector $\boldsymbol{Z}^i_{t,\alpha} := \left[Z^i_{t,1}, \ldots, Z^i_{t,\alpha}\right]$ whose element $Z^i_{t,k}$ is the random variable corresponding to the sum of a set of consecutive per-round rewards of cardinality $\phi$. Formally, for every $k \in [\alpha]$:

$$Z^i_{t,k} := \sum_{j=(k-1)\phi+1}^{k\phi} X^i_{t,j}. \quad (2)$$

The support of $Z^i_{t,k}$ is denoted by $[\underline{Z}^i_{\alpha,k}, \overline{Z}^i_{\alpha,k}]$, where $\underline{Z}^i_{\alpha,k} := \sum_{j=(k-1)\phi+1}^{k\phi} \underline{X}^i_j$, and $\overline{Z}^i_{\alpha,k} := \sum_{j=(k-1)\phi+1}^{k\phi} \overline{X}^i_j$. Intuitively, the $\alpha$-smoothness property states that the elements in $\boldsymbol{Z}^i_{t,\alpha}$ are independent and that, when $\alpha > 1$, the maximum reward $\overline{R}^i$ of a pull cannot be realized in a single time span corresponding to a $Z^i_{t,k}$ element. Formally:

**Definition 1** ($\alpha$-smoothness)**.** *In the TP-MAB setting, for $\alpha \in [\tau_{\max}]$, we say that the reward is* $\alpha$-smooth *if and only if $\frac{\tau_{\max}}{\alpha} = \phi$, with $\phi \in \mathbb{N}$, and, for each $k \in [\alpha]$, the random variables $Z^i_{t,k}$ are independent and s.t. $\overline{Z}^i_{\alpha,k} = \overline{Z}^i_{\alpha} = \frac{\overline{R}^i}{\alpha}$.*

An example of $\alpha$-smooth environment with $\phi = 3$ is presented in Figure 1, where colors denote the elements $z^i_{t,k}$ that are the realizations of the variables $Z^i_{t,k}$.

Consider the extreme values of parameter $\alpha$. When $\alpha = 1$, the reward has no constraint on how it distributes over time. This scenario includes the delayed-feedback bandit setting in which the cumulative reward provided by the arm pulled at $t$ is entirely collected at a single round (including the last possible round $t+\tau_{\max}-1$). Note that, in this case, at each round before $t+\tau_{\max}-1$, the sum of the future per-round rewards is in the range $[0, \overline{R}^i]$. Conversely, when $\alpha = \tau_{\max}$, the vector of aggregated rewards coincides with the vector of per-round

[5]We assume $\alpha$ is a factor of $\tau_{\max}$ for the sake of presentation. The following results also hold for generic $\alpha \in [\tau_{\max}]$.

**Algorithm 1** TP−UCB−FR

1: **Input**: $\alpha \in [\tau_{\max}]$, $\tau_{\max} \in \mathbb{N}^*$
2: **for** $t \in \{1, \dots, K\}$ **do** ▷ init phase
3: Pull arm $i_t = t$
4: **for** $t \in \{K+1, \dots, T\}$ **do** ▷ loop phase
5: **for** $i \in \{1, \dots, K\}$ **do**
6: Compute $\hat{R}^i_{t-1}$ and $c^i_{t-1}$ as in Eq.s (4)-(5)
7: $u^i_{t-1} \leftarrow \hat{R}^i_{t-1} + c^i_{t-1}$
8: Pull arm $i_t = \arg\max_{i \in [K]} u^i_{t-1}$
9: Observe $x^{i_h}_{h,t-h+1}$ for $h \in \{t - \tau_{\max} + 1, \dots, t\}$

rewards, *i.e.*, $\boldsymbol{Z}^i_{t,\tau_{\max}} = \boldsymbol{X}^i_t$, and each per-round reward is at most $\overline{X}^i_j = \overline{R}^i/\tau_{\max}$. Thus, observing low rewards in the first rounds after the pull provides useful information on the actual cumulative reward. In particular, after observing the first $n < \tau_{\max}$ per-round rewards, we know that the cumulative reward achievable in the following rounds is in the range $[0, \frac{\tau_{\max}-n}{\tau_{\max}}\overline{R}^i]$. This information dramatically reduces the uncertainty on the future rewards w.r.t. a setting without smooth rewards (*e.g.*, $\alpha = 1$). The $\alpha$-smoothness property characterizes those setting where not gaining much in the first rounds precludes the possibility of achieving the maximum possible reward over the entire interval.

Consider the playlist recommendation problem in Example 1. Since the reward corresponding to a song is composed of 4 Bernoulli variables and has a maximum of $\overline{Z}^i_\alpha = 4$, $\alpha$-smoothness holds with $\alpha = \frac{\overline{R}^i}{\overline{Z}^i_\alpha} = \frac{4N}{4} = N$.

Assuming $\alpha$-smoothness, we have a lower bound of:

**Theorem 2.** *The regret of any uniformly efficient policy $\mathfrak{U}$ applied to the TP-MAB problem with the $\alpha$-smoothness property is bounded from below by:*

$$\liminf_{T\to+\infty} \frac{\mathcal{R}_T(\mathfrak{U})}{\ln T} \geq \sum_{i:\mu_i<\mu^*} \frac{\Delta_i}{\alpha KL\left(\frac{\mu_i}{\overline{R}_{\max}}, \frac{\mu^*}{\overline{R}_{\max}}\right)}. \quad (3)$$

We remark that this bound is tighter than the one provided in Theorem 1 by a multiplicative factor of $1/\alpha$.

## 4 Algorithms for the TP-MAB Setting

We propose two novel algorithms, namely Temporally-Partitioned rewards UCB with Fictitious Realizations (TP−UCB−FR) and Temporally-Partitioned rewards Element-Wise UCB (TP−UCB−EW), for the TP-MAB problem, which aim at maximizing the cumulative reward and exploit the $\alpha$-smoothness property to do that. From now on, we denote the two corresponding policies by $\mathfrak{U}_{\text{FR}}$ and $\mathfrak{U}_{\text{EW}}$, respectively.

### 4.1 The TP−UCB−FR Algorithm

The pseudo-code of TP−UCB−FR is provided in Algorithm 1. The rationale is to use the rewards coming from not fully-realized reward vectors by replacing the missing elements with fictitious realizations. At round $t$, fictitious reward vectors are associated to each arm pulled in the time span $H := \{t - \tau_{\max} + 1, \dots, t-1\}$. We denote them by $\tilde{\boldsymbol{x}}^i_h = [\tilde{x}^i_{h,1}, \dots, \tilde{x}^i_{h,\tau_{\max}}]$ with $h \in H$, where $\tilde{x}^i_{h,j} := x^i_{h,j}$, if $h + j \leq t$, and $\tilde{x}^i_{h,j} = 0$, if $h + j > t$. The corresponding fictitious cumulative reward is $\tilde{r}^i_h := \sum_{j=1}^{\tau_{\max}} \tilde{x}^i_{h,j}$. The algorithm takes as input the smoothness $\alpha \in [\tau_{\max}]$, and the maximum delay $\tau_{\max}$.[6] During the initialization phase, all arms are pulled once (Line 3). After that, at each round $t$, it computes the estimated expected reward for each arm $i$:

$$\hat{R}^i_{t-1} := \frac{1}{n^i_{t-1}} \left( \sum_{h=1}^{t-\tau_{\max}} r^i_h \mathbb{1}_{\{i_h=i\}} + \sum_{h\in H} \tilde{r}^i_h \mathbb{1}_{\{i_h=i\}} \right), \quad (4)$$

where $n^i_{t-1} := \sum_{h=1}^{t-1} \mathbb{1}_{\{i_h=i\}}$ is the number of times arm $i$ has been pulled by the policy up to round $t-1$, and the confidence interval:

$$c^i_{t-1} := \overline{R}^i \sqrt{\frac{2\ln(t-1)}{\alpha n^i_{t-1}}} + \frac{\phi(\alpha+1)\overline{R}^i}{2n^i_{t-1}}. \quad (5)$$

Finally, it pulls the arm with the largest upper confidence bound $u^i_{t-1}$ (Line 8), and observes its reward (Line 9).

We provide the following upper bound on the regret:

**Theorem 3.** *In the TP-MAB setting with $\alpha$-smooth reward, the pseudo-regret of* TP−UCB−FR *after $T$ rounds is:*

$$\mathcal{R}_T(\mathfrak{U}_{\text{FR}}) \leq \sum_{i:\mu_i<\mu^*} \frac{4(\overline{R}^i)^2 \ln T}{\alpha\Delta_i} \left(1 + \sqrt{1 + \frac{\alpha(\alpha+1)\phi\Delta_i}{2\overline{R}^i \ln T}}\right) + (\alpha+1)\phi \sum_{i:\mu_i<\mu^*} \overline{R}^i + \left(1 + \frac{\pi^2}{3}\right) \sum_{i:\mu_i<\mu^*} \Delta_i.$$

We observe that the dominant term in $T$ has the order of $\mathcal{O}\left(\sum_{i:\mu_i<\mu^*} \frac{\overline{R}^2_{\max} \ln T}{\alpha\Delta_i}\right)$, where $\overline{R}_{\max} = \max_i \overline{R}^i$. When $\alpha = 1$, the upper bound scales as the one of classical MAB algorithms in stochastic settings. Notice that the pseudo-regret indirectly depends on $\tau_{\max}$ since $\overline{R}^i$ represents the cumulative reward obtained over $\tau_{\max}$ rounds. Let us compare this result with the one provided in Theorem 1 for general TP-MAB problems. Applying to Theorem 1 the inequality $KL(p,q) \leq \frac{(p-q)^2}{q(1-q)}$, where for $p, q \in [0,1]$, derived using the fact that $\ln x \leq x - 1$, we get:

$$\liminf_{T\to+\infty} \frac{\mathcal{R}_T(\mathfrak{U})}{\ln T} \geq \sum_{i:\mu_i<\mu^*} \frac{\beta}{\Delta_i}, \quad (6)$$

where $\beta = \frac{\mu^*}{\overline{R}_{\max}}\left(1 - \frac{\mu^*}{\overline{R}_{\max}}\right)$.

For $\alpha > 4(\overline{R}^i)^2/\beta$, the multiplicative factor in the dominant term of the upper bound provided in Theorem 3 is better than that in the lower bound in Theorem 1. This suggests that exploiting the $\alpha$-smoothness provides an improvement over the classical and delayed-feedback MABs.

### 4.2 The TP−UCB−EW Algorithm

The pseudo-code of TP−UCB−EW is provided in Algorithm 2. The key idea is to compute an upper confidence bound for the average of each set of $k$-th realized aggregated rewards $z^i_{t,k}$ from arm $i$ and use them to build an upper bound on the

[6] If these information are not available one should use $\alpha = 1$, meaning we are not assuming any structure over the reward, and use as $\tau_{\max}$ the largest delay observed so far.

**Algorithm 2** TP-UCB-EW

1: **Input**: $\alpha \in [\tau_{\max}]$, $\tau_{\max} \in \mathbb{N}^*$
2: **for** $t \in \{1, \dots, K\}$ **do** ▷ init phase
3: Pull arm $i_t = t$
4: **for** $t \in \{K+1, \dots, T\}$ **do** ▷ loop phase
5: **for** $i \in \{1, \dots, K\}$ **do**
6: **for** $k \in \{1, \dots, \alpha\}$ **do**
7: Compute $\hat{Z}^i_{t-1,k}$ and $c^i_{t-1,k}$ as in Eq.s (7)-(8)
8: $u^i_{t-1} \leftarrow \sum_{k=1}^{\alpha} \left( \hat{Z}^i_{t-1,k} + c^i_{t-1,k} \right)$
9: Pull arm $i_t \in \arg\max_{i \in [K]} u^i_{t-1}$
10: Observe $x^{i_h}_{h,t-h+1}$ for $h \in \{t - \tau_{\max} + 1, \dots, t\}$

overall average reward $R^i_t$. It takes as input the smoothness parameter $\alpha$, and the maximum delay parameter $\tau_{\max}$. At first, it pulls each arm once (Line 3), while, in the following rounds, it computes the empirical mean:

$$\hat{Z}^i_{t-1,k} := \frac{\sum_{h=1}^{t-k\phi} z^i_{h,k} \mathbb{1}_{\{i_h = i\}}}{n^i_{t-1,k}}, \tag{7}$$

where $n^i_{t-1,k} := \sum_{h=1}^{t-k\phi} \mathbb{1}_{\{i_h=i\}}$ is the cardinality of the rewards observed up to round $t-1$ for the $k$-th element of $\boldsymbol{Z}^i_{t-1,\alpha}$, and the confidence bound:

$$c^i_{t-1,k} := \frac{\overline{R}^i}{\alpha} \sqrt{\frac{2\ln(t-1)}{n^i_{t-1,k}}}. \tag{8}$$

We remark that $\hat{Z}^i_{t-1,k} + c^i_{t-1,k}$ is an upper confidence bound for the $k$-th element of $\boldsymbol{Z}^i_{t-1,\alpha}$. Finally, the algorithm computes the upper bound $u^i_{t-1}$, summing the bounds above (Line 8), selects the arm $i$ choosing the largest $u^i_{t-1}$ (Line 9), and observes its reward (Line 10).

We provide the following upper bound on the regret:

**Theorem 4.** *In the TP-MAB setting with $\alpha$-smooth reward, the pseudo-regret of* TP-UCB-EW *after $T$ rounds is:*

$$\mathcal{R}_T(\mathfrak{U}_{\texttt{EW}}) \leq \sum_{i:\mu_i<\mu^*} \frac{8(\overline{R}^i)^2 \ln T}{\Delta_i} + \alpha\left(\phi + \frac{\pi^2}{3}\right) \sum_{i:\mu_i<\mu^*} \Delta_i.$$

Focusing on the dominant term in $T$ of the regret bound, we do not have an explicit improvement over the classical and delayed-feedback MAB algorithms. Therefore, in this case, the structure provided by the $\alpha$-smoothness seems not to affect the regret bound. Hence, from an asymptotic point of view, there is not a clear advantage from having $\alpha$-smooth rewards. However, the constant term is significantly smaller than that of TP-UCB-FR and allows TP-UCB-EW to be much more effective than TP-UCB-FR to tackle TP-MAB problems with a short time horizon.

## 5 Empirical Evaluation

We compare TP-UCB-FR and TP-UCB-EW algorithms with the UCB1 algorithm by Auer *et al.* [2002] and the Delayed-UCB1 algorithm by Joulani *et al.* [2013] in $\alpha$-smooth TP-MAB environments. Appendix A provides details on the adaptation of these two state-of-the-art algorithms to the TP-MAB problem. Notice that, for UCB1, we assume to immediately get the cumulative reward of a pull. Therefore, it represents a *clairvoyant* algorithm observing $R^i_t$ at round $t$. We compare the algorithms in three settings: two synthetically-generated environments and a real-world playlist recommendation scenario.[7]

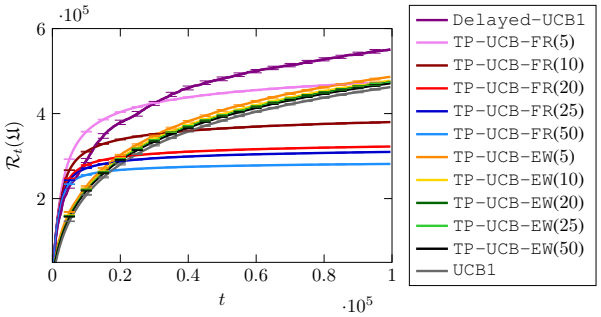


Figure 2: Pseudo-regret over time for Experimental Setting #1.

**Setting #1.** At first, we evaluate the influence of the parameter $\alpha$. We model $K = 10$ arms, whose maximum reward is s.t. $\overline{R}^i = 100i$. The reward is collected over $\tau_{\max} = 100$ rounds, the smoothness parameter is $\alpha = 20$, and the aggregated rewards are s.t. $Z^i_{t,k} \sim \frac{\overline{R}^i}{\alpha} \mathrm{U}([0,1])$, for each $k \in [\alpha]$. We run the algorithms over a time horizon of $T = 10^5$ and average the results over 50 independent runs. In the results, TP-UCB-FR($\eta$) and TP-UCB-EW($\eta$) are s.t. the value of $\alpha$ taken as input is $\eta$, with $\eta \in \{5, 10, 20, 25, 50\}$.

*Results.* Figure 2 shows the pseudo-regret $\mathcal{R}_t(\mathfrak{U})$ over the time horizon and the vertical bars represent the 95% confidence intervals for the mean value. Let us focus on TP-UCB-FR(20) and TP-UCB-EW(20), for which $\eta$ is equal to the $\alpha$ of the environment. TP-UCB-EW(20) provides better results than Delayed-UCB1 over the entire time horizon, while TP-UCB-FR(20) is better than Delayed-UCB1 for $t > 10^4$ and better than TP-UCB-EW(20) for $t > 2{\cdot}10^4$. This suggests that TP-UCB-FR(20) is more suitable for longer time horizons, and this behavior is confirmed by the asymptotic order of Theorem 3. Notice that UCB1 obtains the reward as soon as an arm has been pulled, but it does not exploit the $\alpha$-smoothness property. *Vice versa*, our algorithms incorporates this information that, in some specific situations, allows us to beat even the non-delayed baseline.

During rounds $t \in [1, 7000]$, the Delayed-UCB1 algorithm outperforms TP-UCB-FR, since, during the initial rounds, incomplete samples may be far different from the corresponding unseen realizations, and, therefore, TP-UCB-FR initially pulls the suboptimal arms more often than Delayed-UCB1. Nonetheless, TP-UCB-FR outperforms Delayed-UCB1 over longer time horizons, as expected given the result in Theorem 3. TP-UCB-EW has a similar asymptotic behavior of those of UCB1 and Delayed-UCB1, *i.e.*, the regret curves becomes parallel after $\approx 4000$ rounds. This is because the overall exploration term of the three algorithms is of the same order in $t$ and $\alpha$,

[7] More details about the experiments are deferred to Appendix C.

| $\tau_{\max}$ | $\alpha$ | $\mathcal{R}_T^{(\%)}(\mathfrak{U}_{\texttt{FR}})$ | $\mathcal{R}_T^{(\%)}(\mathfrak{U}_{\texttt{EW}})$ |
|---|---|---|---|
| 100 | 10 | 68.06% (0.26%) | 86.03% (0.59%) |
| 200 | 20 | 95.42% (0.15%) | 80.38% (0.34%) |
| 100 | 50 | 50.84% (0.11%) | 85.36% (0.33%) |
| 200 | 100 | 81.55% (0.10%) | 78.70% (0.24%) |

Table 1: $\mathcal{R}_T^{(\%)}(\mathfrak{U})$ for Experimental Setting #2.

and therefore the advantages of TP-UCB-EW are mainly experienced in the early stages of the learning process. Summarily, for short-time horizons, TP-UCB-EW is preferable to TP-UCB-FR, while TP-UCB-FR shows better performance over long periods.

Let us focus on the results obtained with TP-UCB-FR($\eta$). Setting $\eta < \alpha$, *i.e.*, underestimating the value of $\alpha$, provides worse results in terms of regret, while $\eta > \alpha$ seems to improve the performance of the algorithm without compromising the convergence properties. This suggests that if the $\alpha$ parameter is unknown, one should use an optimistic (large) value in the algorithm. Notice that the regret varies of $\approx 40\%$ w.r.t. the different versions of TP-UCB-FR changing the value of $\eta$, which suggests that TP-UCB-FR is strongly influenced by a mis-specification of the parameter $\eta$. Focusing on TP-UCB-EW($\eta$), we have a behaviour similar to the one observed for TP-UCB-FR($\eta$), showing how larger values for $\eta$ provide better results. Conversely, the performance of TP-UCB-EW present a lower variability by changing the parameter $\eta$, and the gap in terms of regret among the different versions of TP-UCB-EW is of $\approx 3\%$.

**Setting #2.** We study the behavior of our algorithms in settings with different maximum delay $\tau_{\max}$ and smoothness $\alpha$. The scenario is the same presented in Setting #1 except that the maximum reward for the arm $i$ is $\overline{R}^i = \tau_{\max} \cdot i$.[8] We evaluate the algorithms in terms of percentage of the regret w.r.t. the one provided by Delayed-UCB1, whose policy is denoted by $\mathfrak{U}_{\texttt{D}}$, formally $\mathcal{R}_T^{(\%)}(\mathfrak{U}) := \mathcal{R}_T(\mathfrak{U})/\mathcal{R}_T(\mathfrak{U}_{\texttt{D}}) \cdot 100$. We average the results over 50 independent experiments.

*Results.* Table 1 provides the values of $\mathcal{R}_T^{(\%)}(\mathfrak{U})$ for our algorithms (95% CI in brackets). In all the scenarios, the proposed algorithms outperform the Delayed-UCB1 algorithm, providing a regret smaller than 95.5% of the Delayed-UCB1 one. Comparing the results with the same maximum delay $\tau_{\max}$ we notice that a larger value for $\alpha$ provides better performance. This was expected since larger values for $\alpha$ imply that the TP-UCB-FR and TP-UCB-EW algorithms can better exploit the reward structure. By comparing the settings with maximum delay $\tau_{\max} = 100$ and $\tau_{\max} = 200$, the two algorithms behave in opposite ways: the performance of TP-UCB-EW improves by more than 6%, while the regret of TP-UCB-FR increases of more than 30%. This is due to the fact that, with larger $\tau_{\max}$, TP-UCB-FR shows its better behaviour for larger time horizons.

[8] In Appendix C, we also report experiments in scenarios differing in how the aggregated rewards are distributed over the $\phi$ elements composing $Z_{t,k}^i$, which confirm what is shown in this section.

| | $\mathcal{R}_T(\mathfrak{U})$ |
|---|---|
| Delayed-UCB1 | 56473 (805) |
| TP-UCB-FR | 25367 (369) |
| TP-UCB-EW | 55000 (951) |
| UCB1 | 47368 (1289) |

Table 2: Pseudo-regret for the Spotify experimental setting.

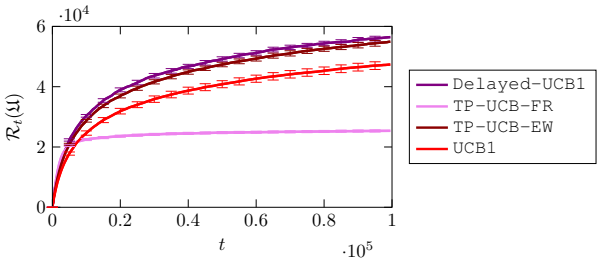


Figure 3: Pseudo-regret over time for the Spotify setting.

**Spotify Setting.** We apply the TP-MAB approach to solve the user recommendation problem presented in Example 1, using a dataset by Spotify [Brost *et al.*, 2019]. We select the $K = 6$ most played playlist as the arms to be recommended, and each time a playlist $i$ is selected, the corresponding reward realizations $\boldsymbol{x}_t^i$ for the first $N = 20$ songs is sampled from the listening sessions of that playlist contained in the dataset. We recall that, in this setting, the maximum delay is $\tau_{\max} = 4N = 80$, and the smoothness parameter is $\alpha = 20$. More details on the setting and the distributions of the reward for each playlist are provided in Appendix C. We average the results over 50 independent runs.

*Results.* Table 2 shows that the TP-UCB-FR algorithm provides the best performance among the analysed algorithms, outperforming UCB1 thanks to the exploitation of the $\alpha$-smoothness property. The regret over time in Figure 3 shows that the TP-UCB-FR provides worse performance than TP-UCB-EW only for a limited amount of rounds ($t < 4000$). This suggests that, in this specific scenario, the TP-UCB-FR algorithm represents a good candidate to provide playlist recommendations.

## 6 Conclusion and Future Works

This paper introduces the novel TP-MAB setting, which generalizes the delayed-feedback bandit setting with bounded delay. First, we show that the lower bound of the TP-MAB problem is the same of that of the standard delayed MAB problem. Then, we characterize a broad set of reward structures, by defining the $\alpha$-smoothness property, for which we provide a tighter lower bound. We design the TP-UCB-FR and the TP-UCB-EW algorithms, suited for the TP-MAB setting, which exploit the partial rewards collected over time and the $\alpha$-smoothness property. We show that the upper bounds on the regret for these algorithms are $\mathcal{O}(\ln T/\alpha)$ and $\mathcal{O}(\ln T)$, respectively. Finally, we empirically show that our algorithms outperforms the state of the art over a wide range of settings generated from synthetic and real-world data.

An interesting future extension would be to consider generic functions regulating the relationship between the cumulative and delayed rewards.

## References


[Arya and Yang, 2020] Sakshi Arya and Yuhong Yang. Randomized allocation with nonparametric estimation for contextual multi-armed bandits with delayed rewards. *Statistics & Probability Letters*, 164:108818, 2020.

[Auer *et al.*, 2002] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[Bistritz *et al.*, 2019] Ilai Bistritz, Zhengyuan Zhou, Xi Chen, Nicholas Bambos, and Jose Blanchet. Exp3 learning in adversarial bandits with delayed feedback. *NeurIPS*, 2019.

[Brost *et al.*, 2019] Brian Brost, Rishabh Mehrotra, and Tristan Jehan. The music streaming sessions dataset. In *WWW*. ACM, 2019.

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

[Castiglioni *et al.*, 2022] Matteo Castiglioni, Alessandro Nuara, Giulia Romano, Giorgio Spadaro, Francesco Trovò, and Nicola Gatti. Safe online bid optimization with return-on-investment and budget constraints subject to uncertainty. *arXiv preprint arXiv:2201.07139*, 2022.

[Cayci *et al.*, 2019] Semih Cayci, Atilla Eryilmaz, and Rayadurgam Srikant. Learning to control renewal processes with bandit feedback. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(2):1–32, 2019.

[Cella and Cesa-Bianchi, 2020] Leonardo Cella and Nicolò Cesa-Bianchi. Stochastic bandits with delay-dependent payoffs. In *AISTATS*, pages 1168–1177, 2020.

[Cesa-Bianchi *et al.*, 2018] Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Nonstochastic bandits with composite anonymous feedback. In *COLT*, pages 750–773, 2018.

[Desautels *et al.*, 2014] Thomas Desautels, Andreas Krause, and Joel W. Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *Journal of Machine Learning Research*, 15(119):4053–4103, 2014.

[Dudik *et al.*, 2011] Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. *arXiv preprint arXiv:1106.2369*, 2011.

[Joulani *et al.*, 2013] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *ICML*, pages 1453–1461, 2013.

[Mandel *et al.*, 2015] Travis Mandel, Yun-En Liu, Emma Brunskill, and Zoran Popović. The queue method: Handling delay, heuristics, prior data, and evaluation in bandits. In *AAAI*, volume 29, 2015.

[Manegueu *et al.*, 2020] Anne Gael Manegueu, Claire Vernade, Alexandra Carpentier, and Michal Valko. Stochastic bandits with arm-dependent delays. In *ICML*, pages 3348–3356, 2020.

[Neu *et al.*, 2013] Gergely Neu, András György, Csaba Szepesvari, and Andras Antos. Online markov decision processes under bandit feedback. *IEEE Transactions on Automatic Control*, 59(3):676–691, 2013.

[Nuara *et al.*, 2018] Alessandro Nuara, Francesco Trovò, Nicola Gatti, and Marcello Restelli. A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns. In *AAAI*, volume 32, 2018.

[Nuara *et al.*, 2022] Alessandro Nuara, Francesco Trovò, Nicola Gatti, and Marcello Restelli. Online joint bid/daily budget optimization of internet advertising campaigns. *Artificial Intelligence*, page 103663, 2022.

[Pike-Burke *et al.*, 2018] Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvari, and Steffen Grunewalder. Bandits with delayed, aggregated anonymous feedback. In *ICML*, pages 4105–4113, 2018.

[Thune *et al.*, 2019] Tobias Sommer Thune, Nicolò Cesa-Bianchi, and Yevgeny Seldin. Nonstochastic multiarmed bandits with unrestricted delays. In *NeurIPS*, 2019.

[Trovò *et al.*, 2018] Francesco Trovò, Stefano Paladino, Marcello Restelli, and Nicola Gatti. Improving multi-armed bandit algorithms in online pricing settings. *International Journal of Approximate Reasoning*, 98:196–235, 2018.

[van der Hoeven and Cesa-Bianchi, 2021] Dirk van der Hoeven and Nicolò Cesa-Bianchi. Nonstochastic bandits and experts with arm-dependent delays. *arXiv preprint arXiv:2111.01589*, 2021.

[Vernade *et al.*, 2017] Claire Vernade, Olivier Cappé, and Vianney Perchet. Stochastic bandit models for delayed conversions. In *UAI*, 2017.

[Vernade *et al.*, 2020a] Claire Vernade, Alexandra Carpentier, Tor Lattimore, Giovanni Zappella, Beyza Ermis, and Michael Brueckner. Linear bandits with stochastic delayed feedback. In *ICML*, pages 9712–9721, 2020.

[Vernade *et al.*, 2020b] Claire Vernade, Andras Gyorgy, and Timothy Mann. Non-stationary delayed bandits with intermediate observations. In *ICML*, pages 9722–9732, 2020.

[Zhou *et al.*, 2019] Zhengyuan Zhou, Renyuan Xu, and Jose Blanchet. Learning in generalized linear contextual bandits with stochastic delays. In *NeurIPS*, volume 32, 2019.