Delay as Payoff in MAB

Ofir Schlisselberg[*1], **Ido Cohen**[*1], **Tal Lancewicki**[1], **Yishay Mansour**[1,2]
[1]Tel Aviv University
[2]Google Research
{ofirs4, idoc, lancewicki}@mail.tau.ac.il, mansour.yishay@gmail.com

Abstract

In this paper, we investigate a variant of the classical stochastic Multi-armed Bandit (MAB) problem, where the payoff received by an agent (either cost or reward) is both delayed, and directly corresponds to the magnitude of the delay. This setting models faithfully many real world scenarios such as the time it takes for a data packet to traverse a network given a choice of route (where delay serves as the agent's cost); or a user's time spent on a web page given a choice of content (where delay serves as the agent's reward). Our main contributions are tight upper and lower bounds for both the cost and reward settings. For the case that delays serve as costs, which we are the first to consider, we prove optimal regret that scales as $\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + d^*$, where T is the maximal number of steps, Δ_i are the sub-optimality gaps and d^* is the *minimal* expected delay amongst arms. For the case that delays serves as rewards, we show optimal regret of $\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + \bar{d}$, where $\bar{d}$ is the second *maximal* expected delay. These improve over the regret in the general delay-dependent payoff setting, which scales as $\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + D$, where D is the maximum possible delay. Our regret bounds highlight the difference between the cost and reward scenarios, showing that the improvement in the cost scenario is more significant than for the reward. Finally, we accompany our theoretical results with an empirical evaluation.

1 Introduction

Classical stochastic Multi-armed Bandit (MAB) is a well studied theoretical framework for sequential decision making, where at every step an agent chooses an action and immediately receives some payoff, be it reward or cost. A natural generalization of this framework considers the situation where the payoff is only received after a certain delay. This is known as the stochastic MAB problem with randomized delays (Joulani, Gyorgy, and Szepesvári 2013), and was extensively researched in previous work under various variants (Vernade, Cappé, and Perchet 2017; Pike-Burke et al. 2018; Zhou, Xu, and Blanchet 2019; Manegueu et al. 2020; Vernade et al. 2020; Wu and Wager 2022; Howson, Pike-Burke, and Filippi 2023; Shi, Wang, and Wu 2023). In all these works the delay was considered reward-independent, namely, the reward and delay are sampled from independent distributions (more on this in the following sections). Later, Lancewicki et al. (2021) introduced reward-dependent delay, where the delay and reward are sampled from a joint distribution. This model is more challenging as it introduces selection bias into the observed payoffs. Tang, Wang, and Zheng (2024) consider a special case of this they call strongly-dependent, where for all intents and purposes the delay is exactly the reward that we are trying to maximize. We second their motivation to study this case, and additionally generalize this to delay as cost that we are trying to minimize. This motivation is twofold. First, it models well many real world scenarios. Second, from a performance perspective, it offers a significant gain in the regret. We show that the *delay as payoff* scenario is actually "simpler" than the general payoff-dependent setting by providing a tighter upper bound compared to (Tang, Wang, and Zheng 2024), and a significantly better bound for the cost setting.

To motivate the cost scenario, consider a communication network where we route packets from node a to node b, and would like to do it in the fastest way possible. We can model this as a stochastic MAB, where every route $a \to b$ is an arm (action) and the time it takes the packet to arrive (formally known as Round Trip Time (RTT), see Postel (1981)) is our payoff. For routing we want to minimize the *RTT* and so we call the payoff *"cost"*. To motivate the reward scenario, consider a web page with some dynamic content. We wish to capture the viewer's attention for as long as possible, by choosing the content wisely. A common metric used in advertising is Average Time on Page (ATP), used, for example, by Google Analytics. In this case we want to maximize the *ATP* and so we call the payoff *"reward"*. In both scenarios the payoff (*RTT* or *ATP*) is the time elapsed from choosing an action until its payoff is final, hence it can be modeled as delay.

As far as performance, an immediate observation is that while the agent is waiting for an action's final payoff, it gains partial knowledge about its payoff as time progresses. More specifically, if at time t_1 an arm is played and by time t_2 the payoff has not been revealed, we can learn that the delay (hence the payoff) is at least $t_2 - t_1$. This is a crucial observation that we use. Taking advantage of this knowledge, we can improve the regret bounds. Notice that this knowledge is one-sided in the sense that every time-step that passes pro-

[*] These authors contributed equally.

vides an improved lower bound on the delay, but the same cannot be said for an upper bound. This is a challenging limitation that is the key difference between *cost* and *reward*, and explains how a better regret can be achieved for *cost*. (This issue is further discussed in Section 1.2.)

1.1 Our Contributions

We study both reward and cost as payoff in the special case of payoff-dependent delay where delay serves as payoff. This setting presents an opportunity to use the partial knowledge accumulated while waiting for the payoff, to achieve better regret bounds. In order to conform with the literature, we normalize the payoff to be in $[0, 1]$, by setting it to the actual delay divided by the maximum delay. This has no implications on the analysis, and is aimed to be inline with the existing regret bounds. Our main contributions are the following:

1. In the case of cost, we offer tight lower and upper bounds that scale as $\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + \min\{d^*, D\Delta_{max}\}$, where d^* is the *minimal* expected delay.
2. In the case of rewards, we offer tight lower and upper bounds that scale as $\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + \min\{\bar{d}, D\Delta_{max}\}$, where $\bar{d}$ is the second *maximal* expected delay. Note that the cost regret bound can be significantly smaller than the reward regret bound, on the same problem instance.
3. We complement our theoretical results in an experimental evaluation.

Our main results, along with a concise comparison to previous work, are presented in Table 1. The two bounds provided, improve both on the general delay-dependent payoff setting which scales as $\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + D$ (Lancewicki et al. 2021)[1] and $\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + D \sum_{i:\Delta_i>0} \Delta_i$ by (Tang, Wang, and Zheng 2024). Note that $D\Delta_{max}$ is already an improvement of factor K, but more significantly if $\bar{d} \ll D\Delta_{max}$ our bound is substantially lower. In the cost case this becomes even more clear as d^* is potentially much smaller than $\bar{d}$.

1.2 Cost vs Reward – Intuition

The goal of a player in a MAB environment can be either to maximize his total payoff, in which case the payoff is called the “reward”, or to minimize his total payoff, then the payoff is called the “cost”. Normally when considering a MAB setting without delay, the choice of using cost or reward is interchangeable by simply changing the sign of the payoff, and most algorithms would be oblivious to the change. We argue that in the case of delayed payoff, where the payoff is the delay, cost and reward are very different. Specifically, when minimizing cost we can make better use of partial knowledge that we gain while waiting for payoff feedback.

We provide here some informal intuition for this, and we will make it more formal by providing lower bounds for both cases in Sections 4.3 and 5.3. Consider a scenario where we have K arms with constant delays (and thus, payoff) sorted from low to high $\{d_i\}_{i=1}^{K}$. If we maximize reward the best arm is i_K with delay d_K. No matter how we play, arm i_{K-1} and arm i_K are indistinguishable until after d_{K-1} time steps, simply because no feedback is received from either arm before d_{K-1}. So the number of times we play sub-optimal arms depends on the second highest delay. In comparison, when minimizing cost, the best arm is i_1 with delay d_1. After d_1 time steps we can already start getting some information about the cost of i_1, and so we can hope to stop playing sub-optimal arms as early as $O(d_1)$.

Paper organization The rest of the paper is organized as follows. In Section 2 we discuss related work and in Section 3 we formally present our settings. In Section 4 we present our main algorithm and analysis for the *delay as cost* setting. In Section 5 we present our algorithm and results for the *delay as reward* setting. In Section 6 we present empirical evaluation of our algorithms compared to previous related works. Section 7 is a discussion. Most of the proofs are deferred to the the full version of the paper (Schlisselberg et al. 2024).

2 Related Work

The delayed payoff in MAB has recently gained significant attention. Most previous works have been devoted to *payoff-independent* delays, often treating them as some unknown distribution. This line of work started with (Dudik et al. 2011) who introduced a constant delay, and offered a regret bound with linear dependence on the delay. Joulani, Gyorgy, and Szepesvári (2013) extended this to stochastic, yet bounded, delay. Later variations include Zhou, Xu, and Blanchet (2019), who made a distinction between *arm-dependent* and *arm-independent* delay. And Pike-Burke et al. (2018), who consider an aggregated rewards model.

Delayed payoff was also studied from an adversarial perspective, where both the delay and rewards are adversarial. This includes works such as (Cesa-Bianchi, Gentile, and Mansour 2019; Thune, Cesa-Bianchi, and Seldin 2019; Bistritz et al. 2019; Gyorgy and Joulani 2021). Masoudian, Zimmert, and Seldin (2022, 2023) presented a “best-of-both-worlds” algorithm for the delayed setting, which is a modification of Zimmert and Seldin (2020). Interestingly, in the adversarial setting, if the delay is adversarial and arm dependant as in (Van Der Hoeven and Cesa-Bianchi 2022), the adversary can correlate the payoff and the delay, thus the payoff also depends on the delay. While this resembles the payoff-dependant setting, the resulting regret bounds are very different. In particular, the delay has a multiplicative effect on the regret, while in the stochastic case we only suffer an additive term.

Only few works considered regret in the *reward-dependent* setting. Lancewicki et al. (2021) considers the case where the delay and reward are sampled from a joint distribution. In their work, there is no assumption on the delay distribution, in particular it may be unbounded. However the reward is still bounded in $[0, 1]$. Note that in our special case, where the payoff directly corresponds to the magnitude of the delay, this bounded payoff implies that the delay is also bounded. Their regret bound has an additive term that

[1] In (Lancewicki et al. 2021) the additional additive term is formally the $(1-\Delta_{\min}/4)$-quantile, but in general this can be as large as D - see discussion in section 2.

	Reward	Cost
(Lancewicki et al. 2021)	$\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + D$	$\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + D$
(Tang, Wang, and Zheng 2024)	$\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + D \sum_{i:\Delta_i>0} \Delta_i$	N/A
This work	$\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + \min\{\bar{d}, D\Delta_{max}\}$	$\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i} + \min\{d^*, D\Delta_{max}\}$

Table 1: Pseudo regret comparison for works on delay-dependent payoff. T is the total number of steps, Δ_i is the sub-optimality gap of arm i and $\Delta_{max} = \max_i \Delta_i$, D is the maximum possible delay, $\bar{d}$ is the second *maximal* expected delay (across arms), and d^* is *minimal* expected delay. Bounds shown hide logarithmic factors that are independent of T.

scales as the $(1-\Delta_{\min}/4)$-quantile of the delay distribution. In general, this can easily be as large as the maximal delay D. For instance, when there is an arm with Bernoulli payoffs with $\mu(i) > 1/4$. Lancewicki et al. (2021) have also considered the case where the delay and reward are independent, which falls outside the scope of the reward-dependent case we consider here. Later, Tang, Wang, and Zheng (2024), consider the setting where the delay equals the reward. They consider general distributions, and their bound scales with a complex quantity dependent on these distributions. In the case where the distribution is bounded by D, they establish a regret bound with an additive term of $D\sum_i \Delta_i$. For the same setting we show an additive regret of $\bar{d}$, which we can also improve to $\min(\bar{d}, D\max_i \Delta_i)$.

3 Problem Setup

Our *delay-as-payoff* model is as follows. There is a set $[K]$ of K arms. Each arm $i \in [K]$ has a distribution $\mathcal{D}_i$ with support $[D] \cup \{0\}$, where D is the maximum delay. In each step $t = 1, 2, ..., T$, the agent chooses arm $i_t \in [K]$, and incurs a delay $d_t \sim \mathcal{D}_{i_t}$. The agent observes the payoff of i_t at time $t + d_t$. The payoff is d_t/D, which we denote by r_t for rewards, or c_t for cost. Thus, the average payoff is $E_{X\sim\mathcal{D}_i}[X/D]$ denoted by $\mu(i)$. Until step $t+d_t$, we refer to the payoff of arm i_t as *missing*, since we do not know its actual delay, and thus payoff, yet. At step $t+d_t$, d_t is revealed, and thus the agent observes the payoff. The interaction protocol is in Algorithm 1.

Algorithm 1 Protocol1

```
for $t \in [T]$ do
    Agent picks an action $i_t \in [K]$
    Environment samples $d_t \sim \mathcal{D}_{i_t}$
    Agent observes feedback $\{d_s : t = s + d_s\}$
```

The performance of the agent is measured by the *expected pseudo regret*, which is the difference between the algorithm's cumulative expected payoff and the best expected payoff of any fixed arm. In the case of reward this will be:

$$\mathcal{R}_T = \max_{i\in[K]} T\mu(i) - \mathbb{E}\left[\sum_{t=1}^{T} r_t\right] = \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{i_t}\right]$$

And in the case of cost:

$$\mathcal{R}_T = \mathbb{E}\left[\sum_{t=1}^{T} c_t\right] - \min_{i\in[K]} T\mu(i) = \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{i_t}\right]$$

where Δ_i is the sub-optimality gap of arm i, i.e., $\Delta_i = |\mu(i) - \mu^*|$ and $\mu^* = \max_{i\in[K]} \mu(i)$, for rewards, and $\mu^* = \min_{i\in[K]} \mu(i)$, for cost. Respectively we define $i^* = i \in [K]\ s.t.\ \mu(i) = \mu^*$. Which, without loss of generality, we assume to be single. For readability, we make use of the following additional notations; $d(i) = D\mu(i)$ is the mean delay of arm $i \in [K]$ and $\Delta_{max} = \max_{i\in[K]} \Delta_i$.

4 Delay As Cost

In this section, we consider the case where the cost is proportional to the agent's delay. We introduce our main algorithm, Bounded Doubling Successive Elimination (BDSE, Algorithm 3), and its associated subroutine, Cost Successive Elimination (CSE, Algorithm 2). CSE builds on the well-known Successive Elimination (SE) algorithm (Even-Dar, Mannor, and Mansour 2006), and as discussed in Section 4.1, it introduces an improved lower confidence bound (LCB). This LCB leverages not only the observed payoff but also takes into account the number of missing observations and their current duration. As we discuss later in this section, a similar improvement cannot be obtained for an upper confidence bound. Instead, BDSE employs a doubling scheme that upper bounds μ^*. This combination is a key component that enables us to achieve our optimal bounds. In the following subsections, we expand on these algorithms and their regret guarantees.

4.1 CSE Algorithm

Much like standard SE, CSE maintains a set of active arms, where initially all arms are active. The algorithm works in rounds, where in each round each active arm is selected once. Unlike standard SE, which eliminates an arm only when there is confidence that it is suboptimal, CSE also eliminates an arm when there is confidence that it is worse than a specific threshold parameter B. In our following definitions we distinguish between the three following groups (Note that they are not mutually exclusive):

1. $M_t(i)$ are the time steps with chosen arm i that have not returned feedback by time t. Formally, $M_t(i) =$

$\{s \in [t] | \, i_s = i \wedge s + d_s \geq t\}$). We denote the size of this group $m_t(i) = |M_t(i)|$.

2. $\mathcal{O}_t(i)$ are time steps with chosen arm i that have returned feedback by time t. Formally, $\mathcal{O}_t(i) = \{s \in [t] | \, i_s = i \wedge s + d_s < t\}$.
3. $F_t(i)$ are the time steps with chosen arm i that are at least D time steps ago, hence their feedback must have returned. Formally, $F_t(i) = \{s \in [t] | \, s \leq t - D\}$
4. Additionally, $n_t(i) = |\{i_s = i | 1 \leq s \leq t\}|$ is the number of all plays of arm $i \in [K]$ before step $t \in [T]$.

Our lower-confidence-bound comprises three terms, $LCB_t(i) = \max\{L_t^1(i), L_t^2(i), L_t^3(i)\}$; each bounds the expected cost with high probability:

1. $L_t^1(i)$ incorporates both observed and unobserved samples, optimistically assuming that the payoff of unobserved samples will be received in the next round. Formally,

$$L_t^1 = \hat{\mu}_t^-(i) - \sqrt{\frac{2\log T}{n_t(i)}}, \tag{1}$$

where $\hat{\mu}_t^-(i) = \frac{1}{n_t(i)}(\sum_{s \in M_t(i)} \frac{t-s}{D} + \sum_{s \in \mathcal{O}_t(i)} c_s)$.

2. $L_t^2(i)$ uses only observed samples that were played up to time $t - D$:

$$L_t^2(i) = \hat{\mu}_t^F(i) - \sqrt{\frac{2\log T}{|F_t(i)| \vee 1}} \tag{2}$$

where $\hat{\mu}_t^F(i) = \frac{1}{|F_t(i)| \vee 1}(\sum_{s \in F_t(i)} c_s)$ is the empirical average of those samples ($\vee$ indicates $\max$). We take maximum in the denominator for the case that some arm was not played by $t - D$, this can occur until $t = D + K$.

3. $L_t^3(i)$ directly leverages the fact the cost corresponds to the magnitude of delay. In particular, as we establish in Lemma 4.1, the number of missing samples can't be much larger than a factor of the expected delay. We define,

$$L_t^3(i) = \frac{|S_t|}{D}\left(\frac{m_t(i)}{2} - 8\log T - 1\right) \tag{3}$$

where S_t is the set of active arms at time t. With the use of Lemma 4.1, $L_t^3(i)$ serves as a valid lower-confidence bound for $\mu(i)$.

For the elimination step we require an upper confidence bound. We use a similar bound as in L_t^2:

$$UCB_t(i) = \hat{\mu}_t^F(i) + \sqrt{\frac{2\log T}{|F_t(i)| \vee 1}} \tag{4}$$

The CSE algorithm is formally described in Algorithm 2 and as a full pseudo code in Algorithm 6 in the the full version of the paper (Schlisselberg et al. 2024).

Note that if the delays were deterministic, then we would have $m_t(i) \leq d(i)/R$, for every arm i. The following lemma handles the case that the delays are stochastic with expectation $d(i)$.

Algorithm 2 Cost Successive Elimination (CSE)

```
Input: number of rounds $T$, number of arms $K$, maximum delay $D$, Elimination Threshold $B$.
Initialization: $t \leftarrow 1$, $S \leftarrow [K]$
Output : Status (either Success or Fail) and $t$ number of time steps performed.
while $t < T$ do
    Play each arm $i \in S$
    Observe any incoming feedback
    Set $t \leftarrow t + |S|$
    for $i \in S$ do
        $LCB_t(i) \leftarrow \max\{L_t^1(i), L_t^2(i), L_t^3(i)\}$
        as defined in Equations (1) to (3)
        Update $UCB_t(i)$ as defined in Equation (4)

    ▷ Elimination step
    Remove from $S$ any arm $i$ if there exists $j$ such that
    $\min\{UCB_t(j), B\} < LCB_t(i)$
    if $S = \emptyset$ then
        Return (Fail,$t$)
Return (Success,$t$)
```

Lemma 4.1 *For every step t, if the last* $\min\{D, t\}$ *steps was played with a round robin of a set of size at least R:*

$$Pr\left[m_t(i) \leq \frac{2d(i)}{R} + 16\log T + 2\right] \geq 1 - \frac{1}{T^2}$$

Note that using the missing plays to upper bound the mean delay results in a significantly weaker bound, and thus unhelpful. With that in hand, and standard concentration bounds, we can define an event G that happens with high probability.

Definition 4.2 *Assume that the actions were played in a round robin manner. Denote R_t to be minimum size of the round robin by time $t \in [T]$. Let G be the event that for every $t \in [T]$ and $i \in [K]$:*

$$m_t(i) \leq \frac{2d(i)}{R_t} + 16\log T + 2$$

$$|\mu(i) - \hat{\mu}_t(i)| \leq \sqrt{\frac{2\log T}{n_t(i)}} \tag{5}$$

where, $\hat{\mu}_t(i) = \frac{1}{n_t(i)}\sum_{s \in \{1 \leq s \leq t | i_s = i\}} r_s$ *is the empirical average of payoff of time steps with chosen arm i. Note that due to missing plays, this is likely unknown to the algorithm at time t.*

We show that G holds with high probability.

Lemma 4.3 *The event G holds with probability* $1 - 3/T^2$.

As previously mentioned, CSE adopts a less conservative elimination rule than standard SE, as it also eliminates arms that perform worse than a specified threshold B. Consequently, it might eliminate all arms, in which case it would return a Fail. For this reason we have a main program BDSE that call CSE with the threshold B. When CSE return a Fail back to BDSE, then BDSE doubles the threshold B and calls CSE with the new threshold.

The following theorem shows that CSE will not eliminate the optimal arm, if $B \geq \mu^*$.

Lemma 4.4 (Safe Elimination) *Assuming G holds and $B \geq \mu^*$, the procedure CSE will not fail and i^* will not be eliminated.*

The following theorem bounds the regret suffered in one call to the procedure CSE.

Theorem 4.5 *The regret of CSE (Algorithm 2) with elimination threshold B is bounded by,*

$$\sum_{i:\Delta>0} \frac{129 \log T}{\Delta_i} + 8D \min\{B, \Delta_{max}\} \log K$$

The first term in the regret scales as optimal instance-dependent, non-delayed MAB. The second term scales with the magnitude of B. Note that, by Lemma 4.4, B will remain smaller than $2\mu^*$ and thus the second term is at most $\tilde{O}(d^*)$.

Proof sketch: For the sake of simplicity we provide the proof sketch only for the B term in the *min*. Assume the good event G holds. Let S_t be the set S at time t. Fix any sub-optimal arm $i \in [K]$, and let τ_i be the last elimination step which arm i remained active.

Recall that L^1_t is computed with an optimistic empirical average $\hat{\mu}^-_t(i)$. That is, any missing sample is assumed to be observed in the next round. At worse, such missing sample eventually would return after D steps and would have cost of 1. Thus, the difference between $\mu^-_t(i)$ and the actual empirical mean $\hat{\mu}_t(i)$ is at most $m_t(i)/n_t$. Since the good event G holds, and the arm i was not yet eliminated at time τ_i,

$$\begin{aligned}
\mu(i) &\leq LCB_{\tau_i}(i) + 2\sqrt{\frac{2\log T}{n_{\tau_i}(i)}} + \frac{m_{\tau_i}(i)}{n_{\tau_i}(i)} \\
&\leq B + 2\sqrt{\frac{2\log T}{n_{\tau_i}(i)}} + \frac{m_{\tau_i}(i)}{n_{\tau_i}(i)} \\
&\approx B + 2\sqrt{\frac{2\log T}{n_{\tau_i}(i)}} + \frac{2DL^3_{\tau_i}(i)}{n_{\tau_i}(i)|S_{\tau_i}|} \\
&\leq B + 2\sqrt{\frac{2\log T}{n_{\tau_i}(i)}} + \frac{2DB}{n_{\tau_i}(i)|S_{\tau_i}|} \qquad (6)
\end{aligned}$$

We consider three cases: (i) $\mu^* < B$ and $\mu(i) < 2B$, (ii) $B \leq \mu^*$ and (iii) $2B \leq \mu(i)$.

case (i): By Lemma 4.4, we know that i^* will not be eliminated (since $\mu^* < B$). Since i was not eliminated at time τ_i, $L^2_{\tau_i}(i) \leq UCB_{\tau_i}(i)$. Using standard arguments this implies that $\Delta_i n^F_{t_{\tau_i}}(i) \leq O(\log(T)/\Delta_i)$. Recall that $n^F_{t_{\tau_i}}(i)$ is the number of times we played i until time $\tau_i - D$. In the last D plays i was played approximately $D/|S_{\tau_i}|$ times due to the round-robin, which causes additional regret of $\Delta_i \frac{D}{|S_{\tau_i}|} \leq O\big(\frac{DB}{|S_{\tau_i}|}\big)$. This accumulates to a total regret of $O\big(\frac{\log T}{\Delta_i} + \frac{DB}{|S_{\tau_i}|}\big)$.

case (ii): We use Equation (6) to show that $\Delta_i \leq 2\sqrt{\frac{2\log T}{n_{\tau_i}(i)}} + \frac{2DB}{n_{\tau_i}(i)|S_{\tau_i}|}$. This implies that either $\Delta_i \leq 4\sqrt{\frac{2\log T}{n_{\tau_i}(i)}}$ or $\Delta_i \leq \frac{4DB}{n_{\tau_i}(i)|S_{\tau_i}|}$. In the first case we get that the regret from arm i is bounded by $O(\log(T)/\Delta_i)$. Similarly, in the second case the regret is bounded by $O((DB)/|S_{\tau_i}|)$.

Algorithm 3 Bounded Doubling Successive Elimination

```
Input: number of rounds $T$, number of arms $K$, maximum delay $D$.
Initialization: $B \leftarrow 1/D$
while $t < T$ do
    Run (ret, $\tau$) $\leftarrow$ CSE($T - t, K, D, B$)
    if ret=Fail then
        $B \leftarrow 2B$
        $t \leftarrow t + \tau$
```

case (iii): The third case assumes that $B \leq \mu(i)/2$. By rearranging the terms of Equation (6), we have that $\Delta_i \leq \mu(i) \leq O\Big(\sqrt{\frac{\log T}{n_{\tau_i}(i)}} + \frac{DB}{n|S_{\tau_i}|}\Big)$. Similarly to case (ii), the total regret is bounded by $O\big(\frac{\log T}{\Delta_i} + \frac{DB}{|S_{\tau_i}|}\big)$.

Now we can sum over all sub-optimal arms i and bound the regret by $O\big(\sum_{i:\Delta>0} \frac{\log T}{\Delta_i} + DB \log K\big)$. Note that the regret when G does not hold is in expectation only $O(1)$. ■

4.2 Bounded Doubling Successive Elimination

CSE (Algorithm 2) demands a parameters B, which is not available for the agent. In this algorithm we estimate B using the ”doubling” technique.

Corollary 4.6 *Algorithm BDSE has a regret of at most,*

$$\sum_{i:\Delta>0} \frac{129 \log T \log d^*}{\Delta_i} + 8 \min\{d^*, D\Delta_{max} \log d^*\} \log K$$

Proof: From Lemma 4.4 we know that if $B \geq \mu^*$ then CSE will not fail, which means that the number of calls to CSE is at most $\log d^*$. Notice that on the j's call of BSE, $DB = 2^j$. The total regret will be at most

$$\begin{aligned}
&\sum_{j=0}^{\log d^*} \left(\sum_{i:\Delta>0} \frac{129 \log T}{\Delta_i} + 8 \cdot \min\left\{2^j, D\Delta_{max}\right\} \log K \right) \\
&= \sum_{i:\Delta>0} \frac{129 \log T \log d^*}{\Delta_i} \\
&\qquad + 8 \min\{d^*, D\Delta_{max} \log d^*\} \log K
\end{aligned}$$

■

4.3 Lower Bound

In this section, we show two lower bounds for the cost setting. The first is a general lower bound which nearly matches the regret bound of our algorithm. And the second, a lower bound for classical SE algorithms. The main challenge is to understand the impact of d^* on the regret. We focus on the second term of the upper bound, as the first term, $\sum_{i:\Delta>0} \frac{\log T}{\Delta_i}$, is a well-known instance-dependent bound even when there are no delays (Bubeck and Cesa-Bianchi 2012).

Theorem 4.7 *In the cost scenario, for every choice of $d^* \leq D/2$, there is an instance for which any algorithm will have a regret of $\Omega(d^*)$*

Proof: We consider two arms with deterministic delays, one is d^* (and cost $\mu^* = d^*/D \leq 1/2$) and the other is D (with cost $\mu = 1$). We select at random which arm has delay d^* and which D. Until time d^* both arms are indistinguishable, and hence the regret is $(1-\mu^*)d^*$. Since $\mu^* \leq 1/2$ we have a regret of at least $d^*/2$. ■

Conservative SE algorithms: We show, for a natural class of SE algorithms, which are also conservative (w.h.p. do not eliminate the optimal action), a lower bound of $\sqrt{Dd^*}$. Interestingly, this bound is also tight, as we show in the full version of the paper (Schlisselberg et al. 2024), a conservative SE algorithm which attains it.

For this impossibility result we use the following two problem instances. In the first problem instance we have the delay of arm 1 to be $\sqrt{Dd^*}/2$ w.p. $2\sqrt{d^*/D}$ and otherwise 0. For arm 2 the delay is deterministic D. In the second problem instance we have the delay of arm 1 to be D w.p. $2\sqrt{d^*/D}$ and otherwise 0. For arm 2 the delay is deterministic $\sqrt{Dd^*}$. In the first instance the best arm is 1 while in the second it is arm 2. Until time $\sqrt{Dd^*}$ we cannot distinguish between the two instances, so a conservative SE algorithm will keep playing both arms in a round-robin manner, and have a regret of $\Omega(\sqrt{Dd^*})$ in the first instance.

It is worth observing why our BDSE overcomes those two problem instances. Due to the doubling scheme, every time the number of missing plays reaches (roughly) the current threshold, both arms will be eliminated until the threshold surpasses μ^*. In the first instance, this will happen after d^* steps, at which point only the optimal arm will remain, and thus the regret is at most $d^*\Delta = d^*$. Similarly, in the second instance, the threshold will surpass μ^* after $\sqrt{Dd^*}$ steps. The Δ here is $\sqrt{d^*/D}$ and so the regret is at most d^*.

5 Delay As Reward

In this section, we consider the case where the delay corresponds to the agent's reward. Similarly to cost, we have a main program Bounded Halving Successive Elimination algorithm (BHSE, Algorithm 5), and an associated subroutine Reward Successive Elimination (RSE). Besides the transition from minimization to maximization, the main difference is that the missing feedbacks at time t should be interpreted differently. In the following subsections, we include the details of these algorithms and their regret guarantees.

5.1 RSE Algorithm

As in the cost scenario, we start with a Reward Successive Elimination algorithm. Since we consider rewards, we would like the threshold B to decrease with time (rather than increase, as was done in the cost scenario). Eventually, RSE expects $B \leq \mu^*$ to guarantee success. As in the CSE algorithm, we will eliminate arms based on suboptimality, in comparison to other arms, or when there is confidence that they are worse than the parameter B.

Algorithm 4 Reward Successive Elimination

```
Input: number of rounds $T$, number of arms $K$, maximum delay $D$, Elimination Threshold $B$.
Initialization: $t \leftarrow 1$, $S \leftarrow [K]$
Output : Status (either Success or Fail) and $t$ number of time steps performed.
while $t < T$ do
    Play each arm $i \in S$
    Observe incoming payoff from $\{s : s + d_s = t\}$
    Set $t \leftarrow t + |S|$
    for $i \in S$ do
        $UCB_t(i) \leftarrow \min\{U_t^1(i), U_t^2(i)\}$
        as defined in Equations (7) and (8)
        Update $LCB_t(i)$ as defined in Equation (9)

    ▷ Elimination step
    Remove from $S$ any arm $i$ if there exists $j$ such that
    $\max\{LCB_t(j), B\} > UCB_t(i)$
    if $S = \emptyset$ then
        Return (Fail, $t$)
Return (Success, $t$)
```

Our upper-confidence-bound comprises only two terms, $UCB_t(i) = \min\{U_t^1(i), U_t^2(i)\}$; which are analogues to L_1 and L_2 in the cost case. Formally,

$$U_t^1 = \hat{\mu}_t^+(i) + \sqrt{\frac{2\log T}{n_t(i)}}, \tag{7}$$

where $\hat{\mu}_t^+(i) = \frac{1}{n_t(i)}(\sum_{s\in M_t(i)} 1 + \sum_{s\in\mathcal{O}_t(i)} r_s)$ is an optimistic estimate of $\mu(i)$. (Recall that $r_s = d_s/D$.) Similarly, U_2 as well as LCB_t are defined by,

$$U_t^2(i) = \hat{\mu}_t^F(i) + \sqrt{\frac{2\log T}{|F_t(i)| \vee 1}}, \tag{8}$$

For the LCB we have,

$$LCB_t(i) = \hat{\mu}_t^F(i) - \sqrt{\frac{2\log T}{|F_t(i)| \vee 1}} \tag{9}$$

where $\hat{\mu}_t^F(i) = \frac{1}{|F_t(i)|\vee 1}(\sum_{s\in F_t(i)} r_s)$.

We use the same good event G as defined in Definition 4.2 in the previous section which also holds here w.h.p.

Theorem 5.1 (Safe Elimination) *Assuming G holds and $B \leq \mu^*$, the procedure RSE will not return Fail and i^* will not be eliminated.*

The following theorem bounds RSE's regret.

Theorem 5.2 *Assume $B \geq \frac{\mu^*}{2}$. The regret of RSE (Algorithm 4) with elimination threshold B is bounded by,*

$$\sum_{i:\Delta>0} \frac{289\log T}{\Delta_i} + 12\min\left\{\bar{d}, D\Delta_{max}\right\}\log K,$$

where $\bar{d}$ is the second highest expected delay.

The assumption that $B \geq \mu^*/2$ is satisfied under the main program BHSE (Algorithm 5) due to Theorem 5.1.

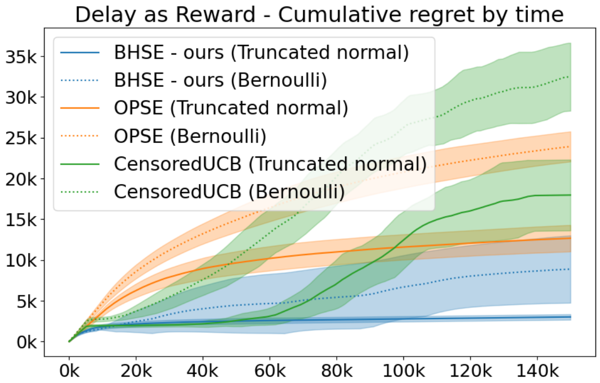

Figure 1: This graph shows results of experiments on different algorithms (color) and different distributions (line style).

5.2 Bounded Halving Successive Elimination

Similar to the main program in the cost case, `BHSE` estimates a lower bound for μ^*. It starts with an over-estimation of $B = 1$ and this time *halves* it by 2 whenever `RSE` returns `Fail`.

Algorithm 5 Bounded Halving Successive Elimination (`BHSE`)

```
Input: number of rounds $T$, number of arms $K$, maximum delay $D$.
Initialization: $B \leftarrow 1$
while $t < T$ do
    Run $(\texttt{ret}, \tau) = \texttt{RSE}(T - t, K, D, B)$
    if ret = Fail then
        $B \leftarrow B/2$ ; $t \leftarrow t + \tau$
```

Corollary 5.3 *Algorithm* `BHSE` *has regret of at most,*

$$\left(\sum_{i:\Delta>0} \frac{289 \log T}{\Delta_i} + 12 \min\left\{\bar{d}, D\Delta_{max}\right\} \log K \right) \log \frac{1}{\mu^*}$$

Proof: From Theorem 5.1 we know that if $B \leq \mu^*$ `BSE` will not fail, which means that the loop will run a maximum of $\log(1/\mu^*)$ times. This also means that $B \geq \mu^*/2$, as needed. Therefore, the total regret will be $\left(\sum_{i:\Delta>0} \frac{289 \log T}{\Delta_i} + 12 \min\{\bar{d}, D\Delta_{max}\} \log K\right) \log \frac{1}{\mu^*}$ ■

Note that without loss of generality we can assume that $\mu^* \geq 1/T$, since otherwise $\Delta_i \leq 1/T$ for all arms and the regret is trivially bounded by 1. Therefore the term $\log(1/\mu^*)$ would be at most $\log T$.

Notice that unlike Corollary 4.6, the regret bound depends on $\bar{d}$. On the one hand, this is better than the delay of the best arm (which has the maximal expected delay). On the other hand, this can be much larger than the regret in the cost scenario, which depends on the minimal expected delay.

5.3 Lower Bound

In this section we show a lower bound for the reward setting, which nearly matches our regret bound.

Theorem 5.4 *In the reward scenario, for every choice of* $\bar{d} \leq D/2$*, there is an instance for which any algorithm will have a regret of* $\Omega\left(\bar{d}\right)$

Proof: Consider the case of K arms. We have one arm with constant delay D, one arm with constant delay $\bar{d} = D/2$, and the remaining arms have delay 0 (and hence reward 0). We select at random the identities of the arms. This implies that for any sub-optimal arm i we have $\Delta_i \geq 1/2$. Clearly, until time $\bar{d}$ the best two arms are indistinguishable hence the regret is at least of order of $\bar{d} \min_i \Delta_i \geq \bar{d}/2$. ■

Conservative SE algorithms: Using similar arguments as in the cost case we can show here a lower bound of $\sqrt{D\bar{d}}$.

6 Experiments

We conducted synthetic experiments for both the *cost* and *reward* settings, using the algorithms in Table 1 as baselines. We show results on two representative distributions: Truncated Normal (bounded in $[0, D]$) and Bernoulli. Due to space constraints, we defer the cost experiments to the the full version of the paper (Schlisselberg et al. 2024). All experiments use T=$150,000$, K=30 and D=5000. For the truncated Normal we sample K means and standard deviations (std), and adjust them to get a truncated version. Since our additive term in the regret is $\min\{\bar{d}, D\Delta_{max}\}$, our contribution is mainly for instances where $\bar{d} < D\Delta_{max}$. Hence, we show the result on such instance, by using an exponential distribution to sample the means of the arms, creating sparsity in the higher regime. The stds are sampled uniformly in $[0, D]$. For the Bernoulli distribution, we sample K probabilities p_i uniformly in $[0, 1]$, so that arm i gets 0 with probability p_i and D with probability $1 - p_i$. Figure 1 shows the average cumulative regret over 10 runs. The shaded region is the std of these runs. `BHSE` (our algorithm) outperforms `OPSE` (from Lancewicki et al. (2021)) and `CensoredUCB` (from Tang, Wang, and Zheng (2024)) in both distributions. Bernoulli distribution is more challenging, resulting in higher regret and std.

7 Discussion

In this paper, we explored a variant of the classical MAB problem, where the payoff is both delayed and directly corresponds to the magnitude of the delay. For the delay as reward setting we introduced tighter upper and lower regret bounds compared to those established in previous works. We are the first to generalize also to cost, highlighting the inherent difference between cost and reward in this setting.

There are several interesting future directions. First, as our motivation for the reward setting is the online advertising, it is a natural question to ask if we can expect similar results in the a *delay as payoff* contextual bandit setting or other variants of the MAB problem. Furthermore, it remains unclear whether our results can be generalized for more general delay distributions which are potentially unbounded (but have a bounded expectation). Finally, adopting this perspective in the adversarial setting, where delays serves as payoffs, is a challenging new problem.

Acknowledgments

We thank Batya Berzack and Yoav Nagel for their input and work during the initiation of this paper and Ran Darshan for his constructive feedback. OS is supported by the Darshan lab. IC is supported by the Israeli Science Foundation (grant number 1186/18). TL and YM are supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 882396), by the Israel Science Foundation and the Yandex Initiative for Machine Learning at Tel Aviv University. IC, TL and YM are also supported by a grant from the Tel Aviv University Center for AI and Data Science (TAD).

References

Bistritz, I.; Zhou, Z.; Chen, X.; Bambos, N.; and Blanchet, J. 2019. Online EXP3 Learning in Adversarial Bandits with Delayed Feedback. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Bubeck, S.; and Cesa-Bianchi, N. 2012. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. arXiv:1204.5721.

Cesa-Bianchi, N.; Gentile, C.; and Mansour, Y. 2019. Delay and cooperation in nonstochastic bandits. *Journal of Machine Learning Research*, 20(17): 1–38.

Cohen, A.; Efroni, Y.; Mansour, Y.; and Rosenberg, A. 2021. Minimax regret for stochastic shortest path. *Advances in Neural Information Processing Systems*, 34.

Dann, C.; Lattimore, T.; and Brunskill, E. 2017. Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 30.

Dudik, M.; Hsu, D.; Kale, S.; Karampatziakis, N.; Langford, J.; Reyzin, L.; and Zhang, T. 2011. Efficient optimal learning for contextual bandits. *arXiv preprint arXiv:1106.2369*.

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

Gyorgy, A.; and Joulani, P. 2021. Adapting to delays and data in adversarial multi-armed bandits. In *International Conference on Machine Learning*, 3988–3997. PMLR.

Howson, B.; Pike-Burke, C.; and Filippi, S. 2023. Delayed feedback in generalised linear bandits revisited. In *International Conference on Artificial Intelligence and Statistics*, 6095–6119. PMLR.

Joulani, P.; Gyorgy, A.; and Szepesvári, C. 2013. Online learning under delayed feedback. In *International conference on machine learning*, 1453–1461. PMLR.

Lancewicki, T.; Segal, S.; Koren, T.; and Mansour, Y. 2021. Stochastic multi-armed bandits with unrestricted delay distributions. In *International Conference on Machine Learning*, 5969–5978. PMLR.

Manegueu, A. G.; Vernade, C.; Carpentier, A.; and Valko, M. 2020. Stochastic bandits with arm-dependent delays. arXiv:2006.10459.

Masoudian, S.; Zimmert, J.; and Seldin, Y. 2022. A best-of-both-worlds algorithm for bandits with delayed feedback. *Advances in Neural Information Processing Systems*, 35: 11752–11762.

Masoudian, S.; Zimmert, J.; and Seldin, Y. 2023. An Improved Best-of-both-worlds Algorithm for Bandits with Delayed Feedback. *arXiv preprint arXiv:2308.10675*.

Pike-Burke, C.; Agrawal, S.; Szepesvari, C.; and Grunewalder, S. 2018. Bandits with delayed, aggregated anonymous feedback. In *International Conference on Machine Learning*, 4105–4113. PMLR.

Postel, J. 1981. Transmission Control Protocol. In *RFC 793, Internet Engineering Task Force (IETF)*.

Schlisselberg, O.; Cohen, I.; Lancewicki, T.; and Mansour, Y. 2024. Delay as Payoff in MAB. *arXiv preprint arXiv:2408.15158*.

Shi, L.; Wang, J.; and Wu, T. 2023. Statistical inference on multi-armed bandits with delayed feedback. In *International Conference on Machine Learning*, 31328–31352. PMLR.

Slivkins, A. 2024. Introduction to Multi-Armed Bandits. arXiv:1904.07272.

Tang, Y.; Wang, Y.; and Zheng, Z. 2024. Stochastic Multi-Armed Bandits with Strongly Reward-Dependent Delays. In *International Conference on Artificial Intelligence and Statistics*, 3043–3051. PMLR.

Thune, T. S.; Cesa-Bianchi, N.; and Seldin, Y. 2019. Nonstochastic Multiarmed Bandits with Unrestricted Delays. arXiv:1906.00670.

Van Der Hoeven, D.; and Cesa-Bianchi, N. 2022. Nonstochastic Bandits and Experts with Arm-Dependent Delays. In Camps-Valls, G.; Ruiz, F. J. R.; and Valera, I., eds., *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, 2022–2044. PMLR.

Vernade, C.; Cappé, O.; and Perchet, V. 2017. Stochastic Bandit Models for Delayed Conversions. arXiv:1706.09186.

Vernade, C.; Carpentier, A.; Lattimore, T.; Zappella, G.; Ermis, B.; and Brueckner, M. 2020. Linear bandits with stochastic delayed feedback. In *International Conference on Machine Learning*, 9712–9721. PMLR.

Wu, H.; and Wager, S. 2022. Thompson sampling with unrestricted delays. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, 937–955.

Zhou, Z.; Xu, R.; and Blanchet, J. 2019. Learning in generalized linear contextual bandits with stochastic delays. *Advances in Neural Information Processing Systems*, 32.

Zimmert, J.; and Seldin, Y. 2020. An Optimal Algorithm for Adversarial Bandits with Arbitrary Delays. arXiv:1910.06054.