A New Framework: Short-Term and Long-Term Returns in Stochastic Multi-Armed Bandit

Abdalaziz Sawwan and Jie Wu
Department of Computer and Information Sciences, Temple University

***Abstract*—Stochastic Multi-Armed Bandit (MAB) has recently been studied widely due to its vast range of applications. The classic model considers the reward of a pulled arm to be observed after a time delay that is sampled from a random distribution assigned for each arm. In this paper, we propose an extended framework in which pulling an arm gives both an instant (short-term) reward and a delayed (long-term) reward at the same time. The distributions of reward values for short-term and long-term rewards are related with a previously known relationship. The distribution of time delay for an arm is independent of the reward distributions of the arm. In our work, we devise three UCB-based algorithms, where two of them are near-optimal-regret algorithms for this new model, with the corresponding regret analysis for each one of them. Additionally, the random distributions for time delay values are allowed to yield infinite time, which corresponds to a case where the arm only gives a short-term reward. Finally, we evaluate our algorithms and compare this paradigm with previously known models on both a synthetic data set and a real data set that would reflect one of the potential applications of this model.**

***Index Terms*—Delayed feedback, learning theory, multi-armed bandit, upper-confidence bound.**

I. Introduction

The decision-making process is a crucial area to study under uncertainty in many applications in computer science. Reinforcement learning is the most prominent example of having uncertain rewards for decisions. Furthermore, those rewards are not often observed instantly in real settings [1, 2]. Some reinforcement learning models could have two related feedback returns observed at two different times. The entire problem of delayed rewards is one of the most challenging in reinforcement learning [2, 3]. Many other applications are reduced to a model in which rewards are observed at different times, such as in some recommendation systems where some feedback is observed instantly with the click of the customer alongside a related delayed feedback that is considered for events that reflect user retention. In addition, the time delay until the long-term reward is observed would vary from one advertisement to another. Hence, the algorithm running the recommendation system has to account for those various delays as well as the instant feedback [4–9].

This wide range of applications makes it imperative to study the stochastic Multi-Armed Bandit (MAB) with delayed feedback in which each arm has its own distribution for time delay for the long-term reward where the distribution can yield a value of an infinite delay. Moreover, there is an instant reward value that is observed from a distribution that is related to the distribution of the long-term feedback according to a previously known relationship. Another simple example can be shown through internship and training programs in society: if a company has different pools of candidates (those pools are modeled as arms) to hire from for its internship and training program, the revenue the intern produces for the company during the internship period can be modeled as a short-term reward, while if the intern decides to apply for a full-time job and obtains it at the company, that would be reflected as a delayed long-term reward that may never come. There is obviously a relationship between the short-term and long-term rewards as they can be considered to be sampled independently from two reward distributions with a predetermined relationship. The objective of the company is to maximize the total rewards considering both the instantaneous and delayed rewards, where the amount of delay can be infinite for the long-term rewards.

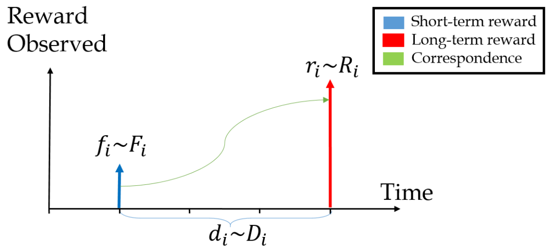

Fig. 1: Rewards of a pulled arm with index i. The short-term reward f_i is from a distribution F_i, the long-term reward r_i is from a distribution R_i, and the time delay d_i is from a distribution D_i. F_i and R_i are related.

This setup introduces many challenges in order to devise a good strategy: the novel challenge is how to appropriately account for both the short-term and long-term rewards given the relationship of the two reward distributions they are sampled from. Another challenge is how to tackle the problem of the missing information due to delayed feedback on the run while actively learning from both the instantaneous feedback returns and observed delayed ones. The challenge of the possibility of not getting a long-term reward at all is possible as well.

In this paper, we set up a more general framework and provide a comparison between the classic Upper-Confidence Bound (UCB) algorithm performing under it, and two other UCB-based algorithms that are modified to address the deterioration of the classic UCB strategy when run under the circumstances of delayed rewards with the existence of short-term rewards. This means that our work lies the bridge

This research was supported in part by NSF grants CPS 2128378, CNS 2107014, CNS 2150152, CNS 1824440, CNS 1828363, and CNS 1757533.

between the typical MAB model in which all rewards are observed instantly, and the typical stochastic MAB model in which rewards are observed after a delay. Figure 1 shows an illustration of the general framework. As seen there, pulling an arm (indexed i) produces two independent feedback returns, where one is instantaneous sampled from a distribution F_i, and the other is delayed sampled from a distribution R_i. The amount of time delay is sampled from the independent distribution D_i (to void redundancy, we call random variables by distributions here).

From another perspective, our framework is a generalization that would provide the ability to knob-tune the two previous MAB models, which are the classic one with a short-term only reward, and the delayed stochastic version in which rewards are observed after a time delay from pulling the arm. Our paradigm introduces a scaling factor κ that is tunable and can determine how dominant each one of the two aspects is with respect to the other one. This tunable parameter will play a role in the regret bound guaranteed by the different strategies shown in this work.

Lastly, it is worth mentioning how the previous model that represents a stochastic MAB with delay times [10] is a special case of the general framework introduced in this paper. On the other hand, the classic MAB model [11] that includes only instant rewards can be derived from this framework indirectly by setting the tunable parameter κ to zero and setting the delay distribution for all arms to be a simple Dirac delta function at zero. In this case, since the strategies will be designed to account for the learning of the delay distribution for each one of the arms in the process, the regret is expected to suffer more than the typical strategies for instant-reward models like UCB. However, the general strategies designed for this model do guarantee a reasonably-bounded regret in the case of having all arms with a fixed zero delay.

Our new results in this paper can be summarized as follows:

- We propose a general framework that would include typically-used settings whether with delayed feedback or instantaneous feedback given that the relationship of the distributions of the two feedback returns is known.
- We study the performance of different algorithms and prove the near-optimality of two UCB-based algorithms by providing the regret analysis for each one of them.
- We run extensive simulations on both synthetic and real-world data in order to compare the performance of the algorithms under this novel general model.

II. Related Work

Classical MAB is thoroughly studied as it represents the basic abstraction of sequential decision-making processes under uncertainty [12–14]. In the classic MAB, the observer pulls a single arm at each round and gets a feedback return sampled from an unknown distribution [13]. The natural exploration-exploitation trade-off arises naturally in this problem, and many basic strategies for it were introduced in the literature to address this trade-off. Some examples of those strategies are UCB [11] and Thompson sampling [13, 15].

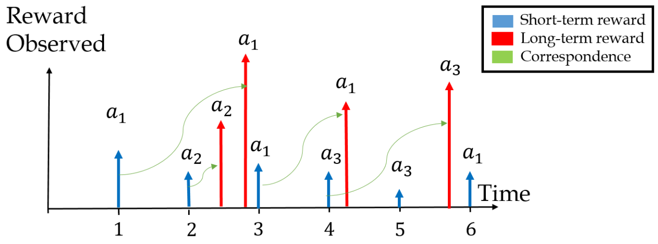

Fig. 2: The illustration of the problem. Blue arrows represent the instantaneous rewards. Red arrows represent the long-term rewards.

Delayed feedback with uncertainty in the process of learning is a widely-known problem with various applications, including some applications in advertising, e-commerce, and finance [16–18]. Hence, different variations of the model of the problem were extensively studied under many assumptions and cases of delays in the setting of delayed stochastic MAB.

The delayed stochastic MAB problem has been introduced initially by Dudik *et al.* [19] with a strong assumption on the delay of the arms. They studied the problem where all arms yield the same fixed delay and they derived a bounded regret under this setting. Furthermore, Joulani *et al.* [20] surprisingly were able to reduce the delayed version of the problem with the assumption of fixed delays applied to the classic non-delayed MAB setting. An example for an additional layer of hardness added to the stochastic MAB with delays is to make the observer get feedback of exactly one value, which is the total sum of rewards that arrive at the same round. This model was introduced by Pike-Burke *et al.* [21].

To the best of our knowledge, the settings in which the observer does not have the ability to know whether they have not received any reward at all or they received a reward of 0 (typically happens in the case of Bernoulli MAB settings) were introduced for the first time by Vernade *et al.* [22]. On the other hand, their model provides the distribution of the time delay for each one of the arms while restricting this distribution with certain conditions. Some of those conditions and assumptions on the distributions of time delays were removed by Gael *et al.* [23] who added other soft assumptions on those distributions that exclude the case where delays are fixed. However, their model accounts for the case where the delay distribution can yield an infinite time. Our case is different as the observer distinguishes between not receiving a feedback at all and receiving a feedback return of value 0.

In contrast to our work here, all those models put certain restrictions on the time delay of the arms. Some other restrictions are introduced to the model in order to solve it in other variations [24, 25]. However, the first to introduce a framework that considers any time distribution without restriction with the possibility of yielding an infinite time were Lancewicki *et al.* [10]. Their paradigm is comprehensive as it includes different unrestricted delay distributions for the arms that can be dependent on the reward observed from the arm. The difference of our more general framework here is the

TABLE I: Description of Commonly-Used Notation

Variable	Description
K	The number of arms.
T	The time horizon.
a_i	The arm with index i.
$\hat{\mu}_t(i)$	Observed empirical average for arm i until t.
$n_t(i)$	The number of observed returns from arm i until t.
$m_t(i)$	The number of times arm i was pulled until t.
$UCB_t(i)$	Upper confidence bound of arm i at time t.
$LCB_t(i)$	Lower confidence bound of arm i at time t.
$f_t(a_t)$	Short-time reward observed at time t from a_t.
$r_t(a_t)$	Long-time reward observed at time $t + d_t(a_t)$.
$d_t(a_t)$	Time delay for long-term reward from a_t.
$d_i(q)$	Quantile function of arm i's delay distribution.
κ	Scaling factor from long-term to short-term.
$\mathbb{E}[.]$	The expected value.

fact that an arm yields two reward values that are sampled independently from two distributions that are related.

Lastly, it is worth mentioning that although each one of the arms in our model has two reward distributions for the short-term and the long-term rewards, our model is completely different from the various multi-objective MAB settings that study the problem of designing algorithms that optimize over more than one objective [26–30].

III. Background and Problem Formulation

Our novel framework considers a variant of the classic stochastic MAB problem with delays. At each round $t = 1, 2, \ldots, T$, the observer pulls an arm $a_t \in \{1, 2, \ldots, K\}$ and observes an instant reward $f_t(a_t)$ and generates a delayed reward $r_t(a_t)$ that will be observed after $d_t(a_t)$ rounds, or more specifically, the delayed reward of pulling a_t is observed at round $t + d_t(a_t)$. In addition, when the long-term reward is observed, we know which arm it is from. However, we do not need to know which exact pull (at which round) the long-term reward was generated. Hence, the value of $d_t(a_t)$ is not observed alongside $r_t(a_t)$ when it arrives at round $t + d_t(a_t)$.

When an arm is pulled, the environment independently generates three values: the long-term reward $r_t(i)$ that is sampled from the distribution R_i, the short-term reward $f_t(i)$ that is sampled from the distribution F_i, and the time delay $d_t(i)$ that is sampled from the distribution D_i. The relationship between the long-term and short-term reward distributions is that one is a linear transformation of the other, with a transformation factor of κ, where $\kappa \in [0, 1]$ is the long-term to short-term scaling factor that makes the rewards observed from an arm reasonably related. This makes $r_t(i) \in [0, 1]$, $f_t(i) \in [0, \kappa]$. Regarding the delay $d_t(i)$, its domain is $\mathbb{N} \cup \{\infty\}$, which means that the long-term reward $r_t(i)$ will never be observed when the value of $d_t(i)$ is infinite. Further, μ_i denotes the mean value of the long-term reward distribution of the arm with index i, so the mean value of the short-term reward distribution of the arm will be $\kappa\mu_i$. Figure 2 shows a basic example of how the rewards in the model are observed.

Regarding a good metric to measure the performance of the strategy employed by the observer, we consider an extended metric of the regret derived from the one used by Gael *et al.* [23] and Lancewicki *et al.* [10], unlike the one proposed by Vernade et al. [22]. Our simple extended measure is the difference between the strategy's expected cumulative reward and the expected total reward of the arm with the highest μ_i. This arm is called the optimal arm and is denoted by i^*. This measure is extended from the expected pseudo-regret. It is defined formally as follows.

$$\mathcal{R}_T = \max_i \mathbb{E}[\Sigma_{t=1}^T (r_t(i) + f_t(i))] - \mathbb{E}[\Sigma_{t=1}^T r_t(a_t) + f_t(a_t)]$$
$$= (1 + \kappa) \times (T\mu_{i^*} - \mathbb{E}[\Sigma_{t=1}^T \mu_{a_t}]) = (1 + \kappa) \times \mathbb{E}[\Sigma_{t=1}^T \Delta_{a_t}],$$

where $\Delta_i = \mu_{i^*} - \mu_i \; \forall i \in [1, K]$. From now on in the paper, we use the word regret interchangeably with pseudo-regret, and we may abuse the notation to so that $r_t(a_t)$ is denoted by r_t and $d_t(a_t)$ is denoted by d_t. Table 1 gives a description of the commonly-used notation.

Furthermore, $m_t(i)$ denotes the number of times an algorithm pulls an arm i before round t. In addition, $n_t(i)$ represents the number of times a feedback was observed from an arm i before round t, where $m_t(i)$ and $n_t(i)$ would not be the same due to the delayed feedback returns. We set $n_t(i)$ set to equal 1 both at the beginning and when the first feedback is observed, after that, it would increment to reflect the number of observations.

Now, consider $\hat{\mu}_t(i)$ to be the observed empirical mean of the long-term rewards observed from arm i before round t, defined as follows.

$$\hat{\mu}_t(i) = \begin{cases} \frac{1}{n_t(i)} \Sigma_{\tau : t > \tau + d_\tau} \mathbb{I}\{a_\tau = i\}(r_\tau + \frac{f_\tau}{\kappa}), & \kappa \neq 0 \\ \frac{1}{n_t(i)} \Sigma_{\tau : t > \tau + d_\tau} \mathbb{I}\{a_\tau = i\} r_\tau, & \kappa = 0 \end{cases},$$

where $\mathbb{I}\{X\}$ is 1 if $X = true$, and 0 if $X = false$. Moreover, we introduce the typical [10] quantile function derived from the delay distribution of arm i, D_i, and represent it by $d_i(q)$. This means that if $\hat{d}_i$ is the actual delay that arm i yields, then the quantile function of the delay is defined as follows.

$$d_i(q) = \min\{\beta \in \mathbb{N} \mid \Pr[\hat{d}_i \leq \beta] \geq q\}.$$

IV. Solutions of The Problem

To reiterate, we study the case in which time delays of the rewards are sampled independently from the rewards given by the pulled arm. We start now with introducing a slightly adjusted version of both UCB [11] and Successive Elimination (SE) [31] to handle the two feedback returns and the time delay associated for the second reward. Moreover, we demonstrate the regret analysis of those modified versions of the two strategies. Afterwards, a third strategy is introduced and its regret analysis. This algorithm is derived from the classic Phased Successive Elimination (PSE) [10].

A. The Classic UCB Algorithm

As widely known, the classic UCB algorithm adopts the upper confidence bound of the arms as the sole criterion to decide which arm to pull. The UCB of an arm is the value that upper-bounds the actual mean with a high probability derived from the Chernoff-Hoeffding inequality. This makes the algorithm operate with optimism under uncertainty. The algorithm is simple; it only pulls the arm with the highest UCB value at each round.

Algorithm 1 UCB for Short-Term and Long-Term Rewards

```
Input: $T$, $K$. //Number of rounds and number of arms.
Output: The set of pulled arms $a_t$ s.t. $t \in [1, T]$.
Initialization: $t \leftarrow 1$. //Start from the first round.
    Pull each arm $i \in [1, K]$ one time.
    Observe any incoming reward.
    Let $t \leftarrow t + K$.
While $t < T$ do
    for $i \in [1, K]$ do
        $n_t(i) \leftarrow \Sigma_{\tau:t>\tau+d_\tau} \mathbb{I}\{a_\tau = i\}$.
        $\hat{\mu}_t(i) \leftarrow \frac{1}{n_t(i)} \Sigma_{\tau:t>\tau+d_\tau} \mathbb{I}\{a_\tau = i\}(r_\tau + \frac{f_\tau}{\kappa})$.
        $UCB_t(i) \leftarrow \hat{\mu}_t(i) + \sqrt{\frac{2\log(T)}{n_t(i)}}$.
    Pull arm $a_t = \arg\max_i UCB_t(i)$.
    Observe reward.
    Let $t \leftarrow t + 1$.
```

We will first simply extend this algorithm to work under our general framework considering the given relationship between the short-term and long-term reward distributions. Afterward, we will prove a lower bound for the regret that this algorithm would suffer under a specific type of setting. The UCB algorithm for our extended framework considers the distribution of the long-term rewards and chooses the arm with the highest corresponding UCB. However, the development of the two terms of the upper confidence bound will be typically faster as the short-term rewards would contribute in the learning process. We were able to include both of the feedback returns in the calculation of the UCB since the relationship between the two distributions is linear. The extension of the UCB algorithm for the general framework is shown in Algorithm 1.

In contrast to the classic setting of having an instant reward only, UCB is not optimal when some (or all) feedback is delayed. We can see that clearly if we consider the simple case of setting $\kappa = 0$, and the delay distributions of all the arms to be simple Dirac delta function at a fixed time delay d_f, that is $D_i = \delta(t - d_f)\ \forall i \in [1, K]$. In this simple case, Joulani *et al.* [20] show how the regret of UCB is bounded by $O(\mathcal{R}_T^{MAB} + Kd_f)$. Lancewicki *et al.* [10] showed that the stochastic version with long-term delays only will guarantee to suffer a regret of $\Omega(Kd_f)$ in a specific case with fixed delay d_f. We will extend the argument of their proof to include the general model. Consider the case in which the optimal arm suffers a fixed delay of d_{f_1}, and all the other arms suffer a fixed delay d_{f_2} such that $d_{f_1} \geq K + d_{f_2}$. In this case, there will be an additional minimum regret for Algorithm 1 of amount $\Omega((d_{f_2} - d_{f_1})\Sigma_{i \neq i^*}\Delta_i)$. This is due to the fact that the UCB strategy persistently pulls the arm with the highest UCB at the time in a naive way so that it may take some time delay $(d_{f_2} - d_{f_1})$ for it to update properly to determine the best arm. The following theorem shows this novel property that is exclusive for our general framework.

Theorem 1. *There exists an instance in which Algorithm 1 gives a regret of* $\Omega((d_{f_2} - d_{f_1})\Sigma_{i \neq i^*}\Delta_i)$ *in our model.*

Proof. Consider the example in which κ is substantially very small and where all the arms' reward distributions follow the Bernoulli distribution such that the average of the optimal arm is $\mu_{i^*} = 1$, and for the other arm is $\mu_i = 0.5$. The delay distribution for the optimal arm gives a fixed delay $d_{f_1} = d_{f_2} + K + 1$, where d_{f_2} is the fixed delay of all the other arms. We consider the case in which the index of the optimal arm i^* is the largest, which means $i^* = K$. Now, we know that UCB would initiate by going in a round-robin fashion over all the arms. Let $\hat{\mu}_1(i)$ be the first observation of the long-term reward from an arm i. Now, with a fixed probability, the long-term reward of at least 0.25 of arms $i < i^*$ is exactly 1. Chernoff inequality would yield

$$\Pr(\Sigma_{i \leq i^*}\hat{\mu}_1(i) \geq K/4) \geq 1 - e^{-K/8} \geq 1 - e^{-1/8}.$$

In other words, when the UCB value of the arms is evaluated given no more that one long-term reward is observed, there will be at least $K/4$ arms with a UCB that is higher than the optimal arm. Without loss of generality, we can consider that those $K/4$ arms are the ones indexed starting from the beginning of the arms.

Now, because $K < (d_{f_2} - d_{f_1})$, until time $K + (d_{f_2} - d_{f_1})$, there will be two options, there will be either no observation of any long-term reward until time $(d_{f_2} - d_{f_1}) + 1$, or some returns will be observed in a way that would typically be shaped in a round-robin way from time $(d_{f_2} - d_{f_1}) + 1$ to $K + (d_{f_2} - d_{f_1})$. Now, as the first long-term reward of arm 1 is 1, it will be the only arm pulled until time $(d_{f_2} - d_{f_1}) + K$. That is because it has the maximum UCB for that time and lowest index (we let UCB choose the arm with lowest index when two arms have the same UCB value). Now, at time $K + (d_{f_2} - d_{f_1}) + 1$, a second long-term reward is observed from arm 1, which would decrease the second term of its UCB. Meanwhile, arm 2 will have the lowest index that has the maximum UCB. Afterward, we pull it $(d_{f_2} - d_{f_1})$ times until time $K + 2(d_{f_2} - d_{f_1})$, while there is no new long-term rewards observed from any arm (beside arm 1 that was already pulled multiple times before and that has lower UCB). After that, we observe a second long-term reward from arm 2 at round $K + 2(d_{f_2} - d_{f_1}) + 1$. Then, let arm 3 will be the next to be pulled. This pattern is repeated as we pull $(d_{f_2} - d_{f_1})$ times each one of arms $i_2, \ldots, i_{K/4}$ in this order. Hence, the regret of the UCB algorithm for this example would be

$$\mathcal{R}_T \geq \frac{d_{f_2} - d_{f_1}}{4 \times 2 \times (1 - e^{-1/8})}\Sigma_{i \neq i^*}\Delta_i = \Omega(\Sigma_{i \neq i^*}\Delta_i(d_{f_2} - d_{f_1})),$$

which concludes the proof. □

B. Successive Elimination

SE keeps a set of active arms such that at the beginning, all the arms are in the set of active arms. This strategy would outperform UCB when there is a delay in some of the feedback observations after pulling the arms. This is due to the reason of giving more weight to more arms whose feedback was not observed yet. The assignment of this weight follows a uniform way over the active arms as it pulls all of them equally. Moreover, an arm can be excluded from the set of active

Algorithm 2 SE for Short-Term and Long-Term Rewards

```
Input: $T$, $K$. //Number of rounds and number of arms.
Output: The set of pulled arms $a_t$ s.t. $t \in [1, T]$.
Initialization: $t \leftarrow 1, S \leftarrow [1, K]$. //Start from the first round.
While $t < T$ do
    Pull each arm $i \in S$.
    Observe all incoming feedback.
    Set $t \leftarrow t + |S|$.
    for $i \in [1, K]$ do
        $n_t(i) \leftarrow \Sigma_{\tau:t>\tau+d_\tau}\mathbb{I}\{a_\tau = i\}$.
        $\hat{\mu}_t(i) \leftarrow \frac{1}{n_t(i)}\Sigma_{\tau:t>\tau+d_\tau}\mathbb{I}\{a_\tau = i\}(r_\tau + \frac{f_\tau}{\kappa})$.
        $UCB_t(i) \leftarrow \hat{\mu}_t(i) + \sqrt{\frac{2\log(T)}{n_t(i)}}$.
        $ULB_t(i) \leftarrow \hat{\mu}_t(i) - \sqrt{\frac{2\log(T)}{n_t(i)}}$.
    Update $S$ by including all arms except all arms $i$ such
        that there exists $j$ with $UCB_t(i) < LCB_t(j)$.
```

arms when we can say that it is suboptimal with confidence. Lastly, it is worth mentioning that an arm can be excluded from an active set and then returned back to the set in some extreme cases in which the high confidence that the arm is suboptimal is significantly reduced. Algorithm 2 shows the detailed pseudo-code of SE.

In addition, in order to pull S samples from K number of arms, SE would need $KS + d_f$ rounds, in contrast to the UCB algorithm, that would need $K(S + d_f)$ rounds in some scenarios like the one shown in the previous subsection. The total regret of SE is derived using a combination of the previously known result from the delayed stochastic MAB version [10], and the short-term regret [31], which we introduce. This combination is shown in the following important bound.

Theorem 2. *The regret of the strategy in Algorithm 2 is bounded under our model. The bound is given by*

$$\mathcal{R}_T \le \min_{\vec{q}\in(0,1]^K} \Sigma_{i\neq i^*} 40(\log T/\Delta_i)(1/q_i + 1/q_{i^*}) + \log(K)\max_{i\neq i^*}\{(d_i(q_i) + d_{i^*}(q_{i^*}))\Delta_i\}\} + \kappa\sqrt{KT\log T}. \tag{1}$$

Furthermore, we can get another incomparable different bound for the regret, which is given by

$$\mathcal{R}_T \le \min_{q\in(0,1]} \Sigma_{i\neq i^*} 325\frac{\log T}{q\Delta_i} + 4\max_i d_i(q) + \kappa\sqrt{KT\log T}. \tag{2}$$

Proof. The regret here is a combination of the regret from the short-term rewards, and the long-term rewards. We will start by using an argument similar to the one used in the delayed MAB version [10] for the long-term rewards and then combine it with the regret resulting from the short-term feedback. We can start by fixing the vector $\vec{q} \in (0, 1]^K$ and defining $d_{\max} = \max_{i\neq i^*} d_i(q_i)$, but we know that for arm i, with a high confidence, the actual average value for the reward distribution R_i lies within the confidence range, which is the interval $[LCB_t(i), UCB_t(i)]$. This is true at all rounds. However, as the time increases, the confidence increases. Considering this, we can conclude with high confidence that the algorithm may never eliminate the optimal arm. However, in the extreme case in which the optimal arm is eliminated, it will eventually be activated again. Now, for a suboptimal arm i that was not excluded by time t from the active arms set, it will hold for it that, $UCB_t(i) \ge LCB_t(i^*)$, from which we know with high confidence that

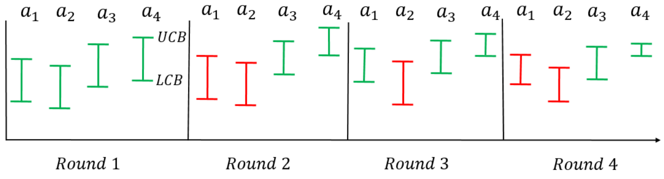

Fig. 3: Illustration of Successive Elimination Algorithm. Green colour represents active arms. Red colour represents inactive colour arms.

$$\frac{\Delta_i}{2} \le \sqrt{\frac{2\log(T)}{n_t(i)}} + \sqrt{\frac{2\log(T)}{n_t(i^*)}}.$$

Now, using Chernoff bound, we can demonstrate that the number of times a long-time feedback return is observed from arm j at time t can be expressed roughly as a fraction q_j from the number of times the arm was pulled until round $t - d_j(q_j)$, where $d_j(q_j)$ is the quantile function of arm j's delay distribution. We can now bound both $n_t(i^*)$ and $n_t(i)$ in order to get the following expression.

$$m_{t-d_{\max}}(i) = O(\frac{\log T}{\Delta_i^2}(1/q_i + 1/q_{i^*})).$$

To this point, we can formulate the total regret that results from the long-term rewards. So, given that t is the last round at which arm i was pulled, we get

$$\begin{aligned} m_t(i)\Delta_i &= m_{t-d_{\max}}(i)\Delta_i + (m_t(i) - m_{t-d_{\max}}(i))\Delta_i \\ &\le O(\frac{\log T}{\Delta_i}(1/q_i + 1/q_{i^*})) + m_t(i) - m_{t-d_{\max}}(i), \end{aligned}$$

where $m_t(i) - m_{t-d_{\max}}(i)$ represents the number of times arm i was pulled in the interval between round $t - d_{\max}$ and round t, and we can see that this is bounded by $d_{\max}$ as well. However, because of the round-robin selection of the arms in the active set S, we can divide this time by the number of total active arms. Furthermore, the number of active arms is K arms (all the arms) before the first time of elimination. The second time will have at most $K - 1$ active arms, and so on, until there are only two active arms before elimination. Hence, we can add up the regret from the long-term rewards at all of those stages of elimination to get the bound of

$$O(\Sigma_{i\neq i^*}\frac{\log T}{\Delta_i}(1/q_i + 1/q_{i^*})) + \log(K)\max_i d_i(q_i),$$

where we made use of the inequality $\log K \ge 1/K + 1/(K-1) + \cdots + 1/2$. By that, we successfully demonstrated the first two terms of regret shown in Equation 1. The third term comes from a known result about SE [31] that the instant rewards have a regret bounded by $\sqrt{KT\log T}$, and given the aforementioned definition of the pseudo-regret, we are able to derive the total regret from the regret produced by the long-term and the regret produced by short-term rewards to get

$$\mathcal{R}_T = O(\Sigma_{i\neq i^*}\frac{\log T}{\Delta_i}(1/q_i + 1/q_{i^*})) \\ + \log(K)\max_i d_i(q_i) + \kappa\sqrt{KT\log T},$$

The argument for Equation 2 would be similar but with restricting the choice of different quantiles for each arm to a single quantile for all the arms. □

A straightforward implication of Theorem 2 is that if all arms have fixed delay d_f for their long-term reward, the regret of the algorithm would add up to a result of $\mathcal{R}_T = O(\mathcal{R}_T^{MAB} + d_f) + \kappa\sqrt{KT\log T}$.

C. Phased Successive Elimination with Delays

We now introduce the slightly modified version of the Phased Successive Elimination presented by Lancewicki *et al.* [10] as well as the corresponding regret. This version of the Successive Elimination algorithm was motivated by the common use of phased versions of different algorithms [32]. PSE does not pull the arms in the naive round robin way as the SE algorithm, rather it dynamically keeps a balanced amount of long-term rewards to be observed at each phase. Hence, the dependency on the delay of the optimal arm is alleviated. However, the dependency on the delay of other arms remains present. This algorithm would typically outperform the normal SE algorithm because of the assignment of extra weight to pulled arms that did not have their feedback observed is a more dynamic way following a certain granularity that depends on the phases. This means that the near-uniform bias introduced in the SE algorithm would change over the phases to give less weight to the arms with long-delayed feedback.

PSE strategy is shown in detail in Algorithm 3. At each one of the phases of PSE, denoted by ℓ, the arms that were not deleted in a previous phase are first pulled in the same naive round-robin way similar to SE. However afterwards, if an arm has its long-term reward observed $16\log(T)/2^{-2\ell}$ times or more, we don't pull it anymore in the phase. On the other hand, the algorithm keeps pulling the other active arms. The process persists until enough long-term observations are made from all the active arms in S. After that, S is updated with the same rule of update in the SE algorithm. Then the next phase $\ell{+}1$ starts and the same process is applied again with the new set $|S|$. Similar to the approach shown in Theorem 2, we devise the following new regret bound for our bound by seamlessly combining the regret bound of the long-term rewards and short-term rewards. That is shown in the following Theorem.

Theorem 3. *The regret of the strategy in Algorithm 2 is bounded under our model. The bound is given by*

$$\mathcal{R}_T \leq \min_{\vec{q}\in(0,1]^K} \Sigma_{i\neq i^*} 290\log(T)/q_i\Delta_i \\ + \log(T)\log(K)\max_{i\neq i^*} d_i(q_i)\Delta_i + \kappa\sqrt{KT\log T}. \tag{3}$$

Algorithm 3 PSE for Short-Term and Long-Term Rewards

```
Input: $T$, $K$. //Number of rounds and number of arms.
Output: The set of pulled arms $a_t$ s.t. $t \in [1,T]$.
Initialization: $t \leftarrow 1, S \leftarrow [1,K], \ell \leftarrow 0$.
While $t < T$ do
  Let $S_\ell \leftarrow S$, $\ell \leftarrow \ell + 1$. //Phase counting.
  While $S_\ell \neq \emptyset$ do
    Pull each arm $i \in S_\ell$, observe incoming feedback.
    Set $t \leftarrow t + |S_\ell|$.
    for $i \in [1,K]$ do
      $n_t(i) \leftarrow \Sigma_{\tau:t>\tau+d_\tau}\mathbb{I}\{a_\tau = i\}$.
      $\hat{\mu}_t(i) \leftarrow \frac{1}{n_t(i)}\Sigma_{\tau:t>\tau+d_\tau}\mathbb{I}\{a_\tau = i\}(r_\tau + \frac{f_\tau}{\kappa})$.
      $UCB_t(i) \leftarrow \hat{\mu}_t(i) + \sqrt{\frac{2\log(T)}{n_t(i)}}$.
      $ULB_t(i) \leftarrow \hat{\mu}_t(i) - \sqrt{\frac{2\log(T)}{n_t(i)}}$.
    Eliminate all arms that were observed at least $\frac{\log(T)}{2^{-2\ell-4}}$
      times from $S_\ell$.
  Update $S$ by including all arms except all arms $i$ such
    that there exists $j$ with $UCB_t(i) < LCB_t(j)$.
```

Proof. Similar to the argument used in the proof of Theorem 2, we start by fixing a vector $\vec{q}\in(0,1]^K$. Then we follow an argument for the long-term rewards similar to the one in [10] with slight modification. To this end, we start by defining two failure events A_1 and A_2 in the following manner:

$$A_1 = \{\exists t, i : |\hat{\mu}_t(i) - \mu_i| > \sqrt{2\log(T)/n_t(i)}\},$$
$$A_2 = \{\exists t, i : m_t(i) \geq 32\log(T)/q_i \cap n_{t+d_i(q_i)}(i) < q_i/2m_t(i)\},$$

and then we define the negation of those failure events by defining $B = \neg A_1 \cap \neg A_2$. Directly from the Chernoff bound, we get $\Pr(B) \geq 1 - 3T^{-2}$.

Now, we denote the last round of phase ℓ by t_ℓ. We know that if an arm i is not eliminated by round t_ℓ, by definition of the algorithm, we will get

$$LCB_{t_\ell}(i^*) \leq UCB_{t_\ell}(i),$$

consider this arm i to be removed at round $t_{\ell+1}$. This positive event will give for the long-term rewards

$$\Delta_i = (\mu_{i^*} - \mu_i) \leq 2\sqrt{2\log(T)/n_{t_\ell}(i)} + 2\sqrt{2\log(T)/n_{t_\ell(i^*)}} \\ \leq 4\sqrt{2\log(T)/(16\log(T)/2^{-2\ell})} \leq \sqrt{2}2^{-\ell} = 2\sqrt{2}\times 2^{-1-\ell}, \tag{4}$$

such that the last inequality can be derived since the phase ℓ ends at the time when all arms i were observed at least $16\log(T)/2^{-2\ell}$. Now, define τ_i' to be the last round at which arm i was pulled. We get $n_{\tau_i'-1}(i) < 16\log(T)/2^{-1-\ell 2}$. Next, we assume that

$$m_{\tau_i'-d_i(q_i)-1}(i) > 32\log(T)/q_i, \tag{5}$$

so now given the positive event B occurs, we will get

$$m_{\tau_i'-d_i(q_i)-1}(i) \leq \frac{2n_{\tau_i'-1}(i)}{q_i} \leq \frac{32\log(T)}{q_i 2^{-2-2\ell}} \leq \frac{256\log(T)}{q_i\Delta_i^2},$$

which holds by Equation 4. On the other hand, that would hold in a trivial way given the condition in Equation 5 does not hold. Hence, we can conclude that the total regret from long-term rewards of arm i will reduce to the formula shown in [10], which is

$$\begin{aligned} m_{\tau_i'}(i)\Delta_i &= m_{\tau_i'-d_i(q_i)-1}(i)\Delta_i + (m_{\tau_i'}(i) - m_{\tau_i'-d_i(q_i)-1}(i))\Delta_i \\ &\leq 256\log(T)/q_i\Delta_i + (m_{\tau_i'}(i) - m_{\tau_i'-d_i(q_i)-1}(i))\Delta_i \\ &= 256\log(T)/q_i\Delta_i + \Sigma_{t=\tau_i'-d_i(q_i)}^{\tau_i'}\mathbb{I}\{a_t = i\}\Delta_i. \end{aligned}$$

To this end, we need to sum up over all the arms to get a total regret from the long-term rewards

$$\begin{aligned} &\Sigma_{i\neq i^*}256\frac{\log T}{q_i\Delta_i} + \Sigma_{i\neq i^*}\Sigma_{t=\tau_i'-d_i(q_i)}^{\tau_i'}\mathbb{I}\{a_t = i\}\Delta_i \\ &+ T\Pr(\neg B) \leq \Sigma_{i\neq i^*}257\log(T)/q_i\Delta_i \\ &+ \Sigma_{i\neq i^*}\Sigma_{t=\tau_i'-d_i(q_i)}^{\tau_i'}\mathbb{I}\{a_t = i\}\Delta_i = \Sigma_{i\neq i^*}257\frac{\log T}{q_i\Delta_i} \qquad (6) \\ &+ \Sigma_{i\neq i^*}257\log(T)/q_i\Delta_i + \Sigma_{i\neq i^*}\Sigma_{t=\tau_i'-d_i(q_i)}^{\tau_i'}\mathbb{I}\{a_t = i\}\Delta_i \\ &+ \Sigma_{\ell=1}^{L}\Sigma_{i=1}^{K}\Sigma_{t=1}^{T}\mathbb{I}\{a_t = i, t \in [\tau_i' - d_i(q_i), \tau_i'] \cap [t_{\ell-1}+1, t_\ell]\}\Delta_i, \end{aligned}$$

given that L is defined to be the total number of phases. Starting from S_ℓ to include all the arms i, such that some rounds in $[\tau_i' - d_i(q_i), \tau_i']$ are in phase ℓ. In other words

$$S_\ell = \{i \in [1, K] : [\tau_i' - d_i(q_i), \tau_i'] \cap [t_{\ell-1} + 1, t_\ell] \neq \emptyset\}.$$

Define $\phi_\ell(i)$ to be the number of arms active at phase ℓ by the round $\min\{\tau_i', t_\ell\}$. We will get

$$\begin{aligned} &\Sigma_{i=1}^{K}\Sigma_{t=1}^{T}\mathbb{I}\{a_t = i, t \in [\tau_i' - d_i(q_i), \tau_i'] \cap [t_{\ell-1} + 1, t_\ell]\}\Delta_i \\ &= \Sigma_{i\in S_\ell}\Sigma_{t=1}^{T}\mathbb{I}\{a_t = i, t \in [\tau_i' - d_i(q_i), \tau_i'] \cap [t_{\ell-1} + 1, t_\ell]\}\Delta_i \\ &\leq \Sigma_{i\in S_\ell}[(d_i(q_i) + 1)\Delta_i/(\phi_\ell(i)) + 1] \\ &\leq \Sigma_{i\in S_\ell}\max_{i\neq i^*}(d_i(q_i) + 1)\Delta_i/\phi_\ell(i) + |S_\ell| \\ &\leq (\log(K) + 1)\max_{i\neq i^*}d_i(q_i)\Delta_i + \log(K) + K. \qquad (7) \end{aligned}$$

Equation 7 holds because the summation of the indicators is zero when the arm $i \notin S_\ell$. In addition, the first inequality there can be reduced to $|[\tau_i' - d_i(q_i), \tau_i']| \leq d_i(q_i) + 1$ and that at least $\phi_\ell(i)$ arms are to be pulled in a round-robin fashion. The second inequality is derived from the fact that

$$\Sigma_{i\in S_\ell}\phi_\ell(i) \leq \frac{1}{|S_\ell|} + \frac{1}{|S_\ell| - 1} + \ldots + 1 \leq 1 + \log|S_\ell| \leq \log K + 1.$$

By substituting Equation 7 in Equation 6, and making use of the maximum possible number of phases in terms of the time horizon, which is $\log_2(T)$, we have for the long-term reward a regret bound of

$$\Sigma_{i\neq i^*}\frac{290\log(T)}{q_i\Delta_i} + \log(T)(\log(K) + 1)\max_{i\neq i^*}d_i(q_i)\Delta_i.$$

Combining that with the short-term regret bound the algorithm guarantee as it would work the same as SE for the short-term, which is $\kappa\sqrt{KT\log T}$, we get

$$\begin{aligned} \mathcal{R}_T \leq \Sigma_{i\neq i^*}\frac{290\log(T)}{q_i\Delta_i} &+ \log(T)(\log(K) + 1)\max_{i\neq i^*}d_i(q_i)\Delta_i \\ &+ \kappa\sqrt{KT\log T}, \end{aligned}$$

which concludes the proof. □

In other words, in a similar way of Theorem 2's proof, both Algorithms 2 and 3 exclude arm i roughly when $\sqrt{\log T/n_t(i)} + \sqrt{\log T/n_t(i^*)} = \Delta_i$.

From another perspective, when we consider the long-term rewards, PSE will keep minimizing the first two terms at the same pace evading the dependency on the time delay distribution, specifically on q_{i^*}, in the first term of Equation 3. On the other hand, in SE, the algorithm's regret is dependent on $\log(K)$ introduced by the long-term rewards part. That is due to the reason that the strategy keeps pulling all arms in S at the same rate, reducing a term scaling in the order of $\log K$. For PSE, this is not what happens as the pulling of the arms happens in the round-robin fashion. This pulling process would happen over all of the $(\log T)$ phases such that the number of pulls is $O(\Sigma_{i\in S_\ell}\phi_\ell(i)$, which is bounded by $O(\log K)$. This will give a term of order $O(\log T \log K)$ in the regret, which appears in the second term of Equation 3.

An example of a distribution of time delays for the arms in which PSE outperforms SE is when the long-term reward either arrives immediately with probability p_i or that the long-term reward never arrives with a probability of $(1 - p_i)$. Under those settings, the regret of SE will yield to a $O(\Sigma_{i\neq i^*}\log(T)/\Delta_i \times (1/p_i + 1/p_{i^*}))$ regret bound. However, PSE will have a better regret that is bounded by $O(\Sigma_{i\neq i^*}\log T/(\Delta_i p_i))$. Those regret bounds are comparable. Notice especially when p_{i^*} is close to zero, the PSE will significantly be better than SE in this case. That is because of two reasons, the first is how the short-term rewards favor PSE more in improving its performance. In addition, it is because PSE dynamically determines how much to pull arms that get enough feedback returns.

Regarding future work, it would include relaxing the condition of having a linear transformation between the reward distribution for the short-term and the long-term rewards. This relaxation would include a version in which κ is an unknown random variable that is being learned in the process. The model can be further modified so that it would include multiple long-term rewards for pulling an arm once, where each one of those observed rewards has its own distribution of reward and distribution of time delay.

Lastly, the characterization of the relationship between the reward values can be different so that it is represented by an unknown probability distribution that follows a certain class of distributions. In this case, the process of learning needs to learn an additional distribution, which is the one that represents the relationship between the short-term and long-term reward. Tackling this additional layer of hardness is a potential for future work as adding it can likely be embedded into our analysis in this paper.

V. Simulations

A. Experimental Settings

In our simulation, we use two different settings, the first one is a synthetic one that we comprise, the other one is

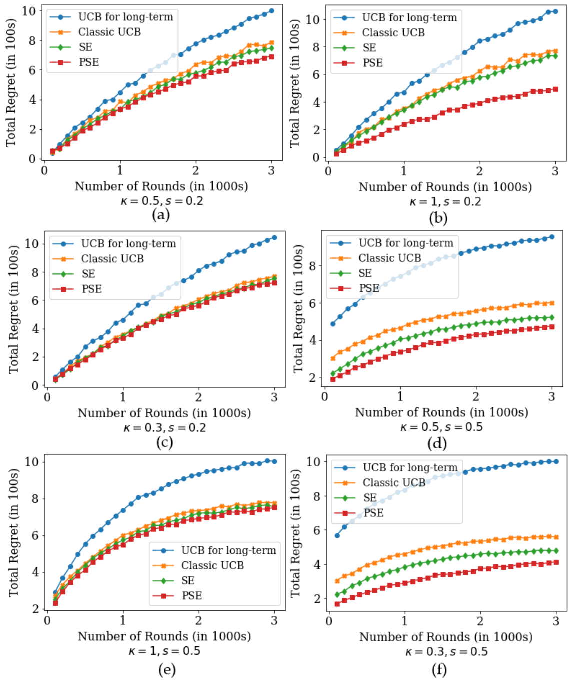

Fig. 4: The basic metrics for default values (synthetic data).

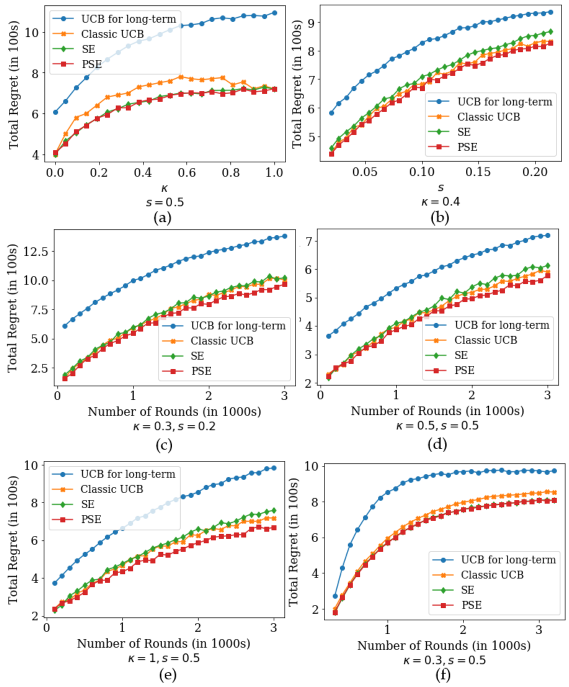

Fig. 5: The basic metrics for default values (real data).

from real-world data. The real-world data [33] represents an application of sparse learning of incomplete traffic speed data that is given. We consider some of those data to be missing where we try to learn them using our different strategies. We consider an instant reward to be accredited for a decision that is related to how close the guess is with the actual value. The long-term reward is observed whenever there is a collective of data and the accreditation of this collective reward is given proportionally to the decisions such that our model of the κ-scaling still holds. Maintaining this condition is sufficient. The traffic speed data represents vehicle mobility information, i.e., the GPS records of vehicles, which are collected from Beijing. This dataset considers a number of road segments for the traffic speed covered by more than $32,000$ taxis that generated the trajectories. On the other hand, the synthetic data represents $K = 20$ arms under Bernoulli settings. Pulling an arm would yield to either a zero reward or κ reward for the short-term feedback, and would yield to either a zero reward or 1 reward for the delayed feedback after a delay that is sampled from a distribution that follows the truncated exponential distribution with a mean delay of 100 (unless stated otherwise). The mean of the long-term reward distribution is randomly sampled from $[0.25s, 0.75s]$, where s is a tunable factor such that $s \in [0, 1]$.

B. Algorithm Comparison

Under both the synthetic and real-world data settings, we study how our three algorithms perform. That includes the classic UCB, SE, and PSE that are presented in Algorithms 1-3. Furthermore, we compare those results to the classic UCB when it operates under the same settings where only the long-term is observed. In order to make the comparison fair, we scale up this long-term reward by $(1+\kappa)$. By that, we are able to observe the advancement of the learning process itself given the short-term reward, and without this short-term reward, which comprises the novelty of our work in comparison to the existing frameworks. The different algorithms would perform slightly differently under both the synthetic and real-world data settings, we see how our three algorithms behave. Moreover, we contrast the results and performance of the classic UCB when it operates under the same settings where only the short-term is observed.

C. Experimental Results

The first observation that we can make is the order of the performance of the algorithms under various different parameters set. As shown in Figure 4, the best algorithm performing under our Bernoulli distribution is PSE as its regret is bounded following Equation 3. This regret grows following that formula that keeps performing better than SE.

In addition, Figure 4 (b) shows a case in which SE would perform badly at the beginning even after comparing it to the classic UCB algorithm performing by simply choosing the arm with highest UCB without consideration of delay. This is due to the fact that low s values, and hence lower values of rewards, are detrimental to the performance of SE initially as it starts by naively going through the arms in a round-robin fashion before the exclusion of some arms starts.

Figure 5 shows how the different algorithms perform differently under various settings. Plots (a) and (b) are particularly interesting as they show the regret over a span of ranges of the parameters κ and s. From there, we can see how the three algorithms perform under settings where the short-term rewards are dominant (κ is high) or the settings where the long-term rewards are dominant (κ is low).

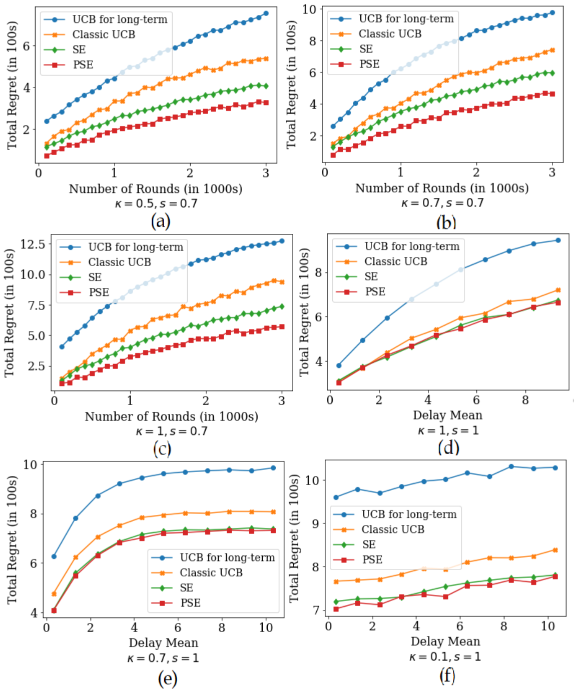

Fig. 6: The effect of various parameters on total regret (real data).

Figure 6 shows more how various time delays for the long-term reward affects the performance of the different algorithms. We can see clearly from plot (d) that the regret grows more steeply as the value of s is high to guarantee higher reward values for the arms. Interestingly enough, plots (e) and (f) of Figure 6 show how the delay means is crucial when the value of κ is high. However, when the value of κ is small, the average time will not have the same effect.

D. Simulation Summary

We were able to observe how SE and PSE were both robust against different time delays when the long-term reward is weighted more with respect to the short-term rewards. Furthermore, we saw how for relatively-low time delay, SE behaves in a superior way in this model, especially for low values of κ, which confirms our theoretical results. In addition, we saw how the regret of UCB increases in a near-linear way at the beginning, to be more specific, for roughly the first $K \times d_f$ rounds. On the other hand, SE does not kick off with a linear regret like the case of UCB that imitates the uniformly-random way. It only selects the set of active arms even though they are weighted uniformly for it in terms of importance. Hence, although SE can not avoid pulling each one of the non-optimal arms for $\frac{d_f}{K}$ times, it never exceeds the minimum bound of arm selection needed so that the strategy can eliminate sub-optimal arms. Figures 7-8 show the performance of the different algorithms under different configurations of parameters. From what we see from the plots in Figure 8, we notice how our SE algorithm outperforms the classic UCB by around 21% on average for different κ values. For different values of the reward distribution mean values, the PSE algorithm outperforms the classic UCB by around 37%.

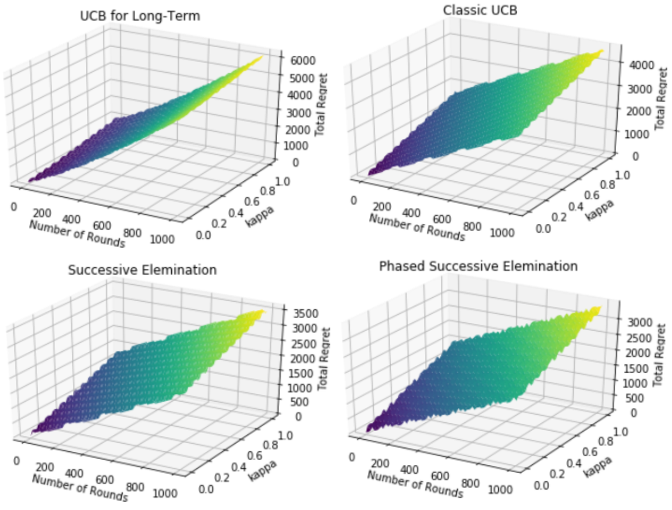

Fig. 7: The effect of various parameters on the total regret (synthetic data).

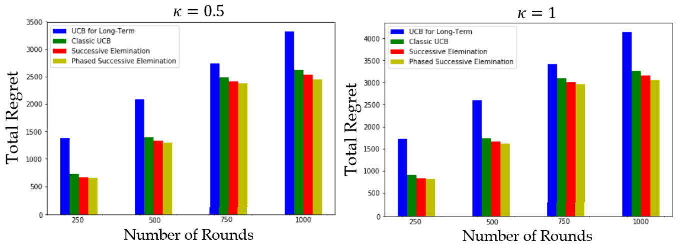

Fig. 8: The effect of κ on the total regret (synthetic data).

VI. Conclusion

The problem of stochastic Multi-Armed Bandit (MAB) is extensively studied due to its vast range of applications. The classic model of this problem considers the reward of a pulled arm to be observed after a time delay that is sampled from a random distribution assigned for each arm. In this paper, we studied a generalized framework in which pulling an arm gives both an instant (short-term) reward and a delayed (long-term) reward at the same time. The distributions of reward values for short-term and long-term rewards are related with a previously known relationship. The distribution of time delay for each one of the arms is independent from the reward distribution of the arm. This time can be infinite, which would mean that the reward is never observed. In our work, we devised two near-optimal-regret UCB-based algorithms after introducing the simple UCB algorithm that would work for this new framework. We provided the corresponding regret analysis for each one of the three discussed algorithms. Finally, we evaluated the performance of our algorithms and compared this model with previously known models on both a synthetic data set and a realistic data set that would reflect one of the potential applications of this model. The first UCB-based near-optimal algorithm was Successive Elimination (SE), which empirically outperformed the old algorithm by around 21% on average for different parameter values. Lastly, the last algorithm introduced, which is Phased Successive Elimination (PSE), outperforms the old algorithm by around 37% on average under this new model.

References

[1] Even-Dar, E., Mannor, S., Mansour, Y., Mahadevan, S. (2006). Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. Journal of machine learning research, 7(6).

[2] Sutton, R. S., Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

[3] Even-Dar, E., Mannor, S., Mansour, Y., Mahadevan, S. (2006). Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. Journal of machine learning research, 7(6).

[4] Silva, N., Werneck, H., Silva, T., Pereira, A. C., Rocha, L. (2022). Multi-Armed Bandits in Recommendation Systems: A survey of the state-of-the-art and future directions. Expert Systems with Applications, 197, 116669.

[5] Elena, G., Milos, K., Eugene, I. (2021). Survey of multiarmed bandit algorithms applied to recommendation systems. International Journal of Open Information Technologies, 9(4), 12-27.

[6] Bouneffouf, D., Rish, I., Aggarwal, C. (2020, July). Survey on applications of multi-armed and contextual bandits. In 2020 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-8). IEEE.

[7] Shi, C., Shen, C. (2021, May). Federated multi-armed bandits. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 11, pp. 9603-9611).

[8] Bubeck, S., Budzinski, T., Sellke, M. (2021, July). Cooperative and stochastic multi-player multi-armed bandit: Optimal regret with neither communication nor collisions. In Conference on Learning Theory (pp. 821-822). PMLR.

[9] Gyorgy, A., Joulani, P. (2021, July). Adapting to delays and data in adversarial multi-armed bandits. In International Conference on Machine Learning (pp. 3988-3997). PMLR.

[10] Lancewicki, T., Segal, S., Koren, T., Mansour, Y. (2021, July). Stochastic multi-armed bandits with unrestricted delay distributions. In International Conference on Machine Learning (pp. 5969-5978). PMLR.

[11] Auer, P., Cesa-Bianchi, N., Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. Machine learning, 47(2), 235-256.

[12] Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. SIAM journal on computing, 32(1), 48-77.

[13] Bubeck, S., Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends® in Machine Learning, 5(1), 1-122.

[14] Berry, D. A., Fristedt, B. (1985). Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability). London: Chapman and Hall, 5(71-87), 7-7.

[15] Kuleshov, V., Precup, D. (2014). Algorithms for multi-armed bandit problems. arXiv preprint arXiv:1402.6028.

[16] Chapelle, O. (2014, August). Modeling delayed feedback in display advertising. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1097-1105).

[17] Yoshikawa, Y., Imai, Y. (2018). A nonparametric delayed feedback model for conversion rate prediction. arXiv preprint arXiv:1802.00255.

[18] Cover, T. M. (2011). Universal portfolios. In The Kelly Capital Growth Investment Criterion: Theory and Practice (pp. 181-209).

[19] Dudik, M., Hsu, D., Kale, S., Karampatziakis, N., Langford, J., Reyzin, L., Zhang, T. (2011). Efficient optimal learning for contextual bandits. arXiv preprint arXiv:1106.2369.

[20] Joulani, P., Gyorgy, A., Szepesvári, C. (2013, May). Online learning under delayed feedback. In International Conference on Machine Learning (pp. 1453-1461). PMLR.

[21] Pike-Burke, C., Agrawal, S., Szepesvari, C., Grunewalder, S. (2018, July). Bandits with delayed, aggregated anonymous feedback. In International Conference on Machine Learning (pp. 4105-4113). PMLR.

[22] Vernade, C., Cappé, O., Perchet, V. (2017). Stochastic bandit models for delayed conversions. arXiv preprint arXiv:1706.09186.

[23] Gael, M. A., Vernade, C., Carpentier, A., Valko, M. (2020, November). Stochastic bandits with arm-dependent delays. In International Conference on Machine Learning (pp. 3348-3356). PMLR.

[24] Desautels, T., Krause, A., Burdick, J. W. (2014). Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. Journal of Machine Learning Research, 15, 3873-3923.

[25] Zhou, Z., Xu, R., Blanchet, J. (2019). Learning in generalized linear contextual bandits with stochastic delays. Advances in Neural Information Processing Systems, 32.

[26] Hüyük, A., Tekin, C. (2021). Multi-objective multi-armed bandit with lexicographically ordered and satisficing objectives. Machine Learning, 110(6), 1233-1266.

[27] Almeida, C. P., Gonçalves, R. A., Venske, S., Lüders, R., Delgado, M. (2020). Hyper-heuristics using multi-armed bandit models for multi-objective optimization. Applied Soft Computing, 95, 106520.

[28] Roijers, D. M., Zintgraf, L. M., Libin, P., Reymond, M., Bargiacchi, E., Nowé, A. (2020, September). Interactive multi-objective reinforcement learning in multi-armed bandits with gaussian process utility models. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 463-478). Springer, Cham.

[29] Almeida, C. P., Gonçalves, R. A., Venske, S., Lüders, R., Delgado, M. (2020). Hyper-heuristics using multi-armed bandit models for multi-objective optimization. Applied Soft Computing, 95, 106520.

[30] Shi, C., Shen, C., Yang, J. (2021, March). Federated multi-armed bandits with personalization. In International Conference on Artificial Intelligence and Statistics (pp. 2917-2925). PMLR.

[31] Even-Dar, E., Mannor, S., Mansour, Y., Mahadevan, S. (2006). Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. Journal of machine learning research, 7(6).

[32] Auer, P., Ortner, R. (2010). UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. Periodica Mathematica Hungarica, 61(1-2), 55-65.

[33] Available online on: https://www.microsoft.com/en-us/research/publication/inferring-gas-consumption-and-pollution-emission-of-vehicles-throughout-a-city