Budget-Constrained and Deadline-Driven Multi-Armed Bandits with Delays

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

***Abstract*—Many extensions of the Multi-Armed Bandit (MAB) problem were studied recently offering a strong theoretical basis for applications that require active learning. The Budget-Constrained and Deadline-Driven MAB with Delay (BD-MAB) variation introduces a novel scenario in which a player must pull from K arms, each associated with reward, delay, and cost distributions. To the best of our knowledge, this is the first work that combines budget constraints with time delays in an MAB problem. The model involves three phases. The first phase comprises pulling arms, while incurring random cost, and observing rewards. This phase ends when the budget is depleted. The second phase includes observing some of the delayed rewards. The third phase starts at a fixed termination deadline, marking the end of reward observation, where any rewards returned after this phase are not observed and are considered dead. We present a novel solution to this problem by developing a new Upper Confidence Bound (UCB)-based algorithm. The name of the algorithm is Budget-Constrained and Deadline-Driven UCB with Delay (BD-UCB) algorithm. We provide extensive regret analysis that confirms the efficiency of our approach in managing these complexities. Lastly, we provide numerical simulations which further demonstrate the effectiveness of our proposed solution.**

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

I. Introduction

The prominence of uncertain and delayed feedback in decision-making processes offers a significant shift in numerous computer science applications. In such applications, decision-making becomes an intricate process of assessing options, predicting outcomes, and adapting to changing circumstances. Among various decision-making models studied in computer science and machine learning, the Multi-Armed Bandit (MAB) problem is the fundamental abstraction of how to balance between exploration and exploitation under uncertainty [1–3]. These foundational elements of the MAB problem have led to its widespread application in numerous fields, from the placement of online advertisements to the allocation of resources in a dynamic environment [4–7].

However, real-world scenarios often present complexities that extend beyond the classical MAB model. One such complexity is the constraint of a limited budget, which leads to the formulation of the Budget-Constrained MAB (B-MAB) problem [8–10]. In this variation, each arm pull incurs some cost, whether random or deterministic, and the total cost cannot exceed a predetermined budget B. This added constraint makes the decision-making process harder, as one needs to optimize the reward while considering the cost of each action.

This research was supported in part by NSF grants CNS 2214940, CPS 2128378, CNS 2107014, and CNS 2150152.

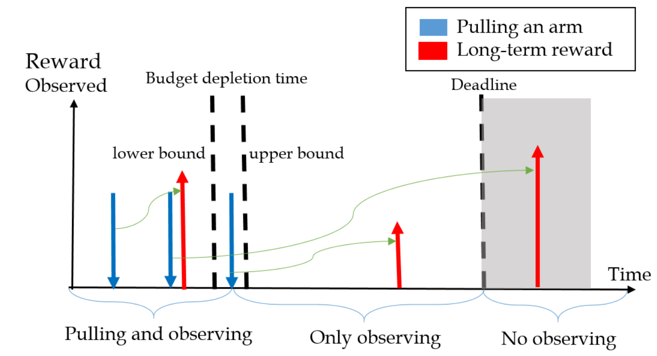

Fig. 1: A basic illustration of the model. The first phase has an arm pulled every round, and the second phase is where observing incoming rewards is allowed, but no pulling is possible because the budget had been depleted. The last phase is after the termination deadline where observation stops.

As many applications increase the need to model more complex real-world scenarios, we encounter additional constraints that need to be accounted for [11–14]. An often overlooked but important constraint is time, specifically, the delay in observing rewards associated with each action and the existence of a deadline, after which no further rewards are considered. This temporal aspect is particularly important in applications and situations where immediate feedback is not available and rewards can be observed only after a certain delay that is sampled from a stochastic process. Those rewards sometimes may never arrive, meaning the delay can be infinite.

Given these considerations, we introduce a new model: the Budget-Constrained and Deadline-Driven MAB with Delay (BD-MAB) problem. Our model captures the intricacies of decision-making under budget constraints, delays in reward observations, and the presence of a strict deadline. It can be viewed as a game with three distinct phases: the pulling and observing phase, the observing phase, and the deadline phase. Each phase presents unique challenges and together they add a layer of complexity unseen in existing MAB models.

Figure 1 shows the model of the BD-MAB problem. The first phase involves arm pulling, where each pull incurs a random cost and results in rewards that might not be observed immediately. This phase ends when the budget is depleted. The second phase involves the observation of delayed rewards from the actions taken in the first phase. Notably, the transition from the first phase to the second one is stochastic as it depends on both the budget and the random costs incurred while pulling the arms. The last phase begins at a fixed point in time, marking the end of the game. Any rewards arriving post this deadline are not observed.

This dynamic and temporally constrained problem presents unique challenges, like how to balance exploration and exploitation given the constraints of cost budget, delay, and termination deadline. The stochastic nature of the transition between the first and second phases adds another dimension of uncertainty to the problem because of the random cost incurred from pulling the arms. To tackle the BD-MAB problem, we introduce a novel solution: the Budget-Constrained and Deadline-Driven UCB with Delay (BD-UCB) algorithm.

This algorithm takes into account the reward distribution of each arm and it also considers the cost and delay distributions. Our UCB-based algorithm makes use of the constraints on the time delay distributions of the arms to dynamically estimate the amount of rewards that are expected to be eventually observed before the deadline round. The algorithm dynamically adjusts those estimates as the rewards are (or are not) observed. Moreover, we provide a theoretical analysis of our proposed algorithm, including regret bounds, which confirms the efficiency of our BD-UCB algorithm under this model, which combines budget constraints and stochastic delay.

The BD-MAB problem and the BD-UCB algorithm have widespread applicability in numerous real-world scenarios. For instance, in the domain of online advertising, ad placements could be managed effectively by considering each ad as an arm with associated costs, potential click-through rewards, and observation delays [15–18]. Similarly, in healthcare, treatment alternatives could be modeled as arms, each with its associated cost, expected health outcome (reward), and observation delay [19–21]. Another application is in cloud computing, where virtual machines could be allocated optimally, taking into account the cost of allocation, the expected efficiency gains, and the delay in observing these gains [22–24].

Our main contributions in this paper are as follows:

- We propose the BD-MAB model, a novel extension of the MAB problem, incorporating budget constraints, delayed reward observation, and a deadline constraint.
- We develop the BD-UCB algorithm, a unique solution to the BD-MAB problem, and we provide comprehensive regret analysis for the algorithm.
- We illustrate the efficiency and robustness of the proposed BD-UCB algorithm through numerical simulations, comparing its performance with standard methods.

II. Related Work

The MAB problem with delay, referred to as Delayed MAB, has been studied with a focus on various assumptions about the delay patterns. For instance, Joulani *et al.* [25] addressed the case where all arms yield a constant delay, successfully reducing the delayed MAB problem to the classical non-delayed MAB problem under this specific assumption, which shares the fact of having constraints on the delay distributions with our work. In contrast, our work in this paper introduces a new variant of MAB, namely BD-MAB, which incorporates both budget constraints and delay in rewards. This variant bears some similarity to the work by Tran-Thanh *et al.* [8, 9], where MAB problems with budget constraints were considered. However, these studies assumed fixed and known costs, differing from our model which assumes stochastic and unknown costs. Our model incorporates delay with some restrictions in receiving rewards, unlike the aforementioned works which consider the classic immediate feedback.

Our BD-MAB setting is also related to the work by Xia *et al.* [26, 27], where they studied the MAB problem with stochastic rewards and costs. Their work did not consider budget constraints and delays, which are the focus of our study. Similarly, the UCB-BV algorithm proposed by Ding *et al.* [10] solves the B-MAB problem in which cost values are random. However, the reward is immediate as no delay is considered. Regarding previous work on MAB with delays, both Gael *et al.* [28] and Zhou *et al.* [29] study the problem while imposing some restrictions on the delay distributions, similar to our work. However, Lancewicki *et al.* [30] and Sawwan *et al.* [31] do consider delay distributions with no constraints, but there is no budget constraint in their work. Our work uniquely extends MAB literature by combining budget constraints with delayed feedback, offering a more comprehensive model despite limited constraints on time delay.

III. Background and Problem Formulation

We present the formal model of the BD-MAB problem, detailing its three phases and the characteristics of the random distributions for time delay, reward value, and arm cost.

A. Environment and Constraints

We consider an environment consisting of K distinct arms, denoted by the set $1, 2, \dots, K$. Each arm i has associated with it three random variables that represent the reward ($\mathcal{R}(i)$), time delay ($\mathcal{D}(i)$), and cost ($\mathcal{C}(i)$). When arm i is pulled at time t, it yields a random reward $r_t(i)$, incurs a random cost $c_t(i)$, and the reward will be observed after a random delay $d_t(i)$, or more specifically, if a pulling algorithm a pulls an arm a_t at round t, the reward $r_t(a_t)$ is observed at round $t+d_t(a_t)$ if $t + d_t(a_t) \leq T_{deadline}$. In addition, whenever a reward $r_t(i)$ is observed, we know which arm it came from. However, we do not need to know which exact pull (at which round) it was generated. In other words, the value of $d_t(i)$ is not observed alongside $r_t(i)$ when it arrives at round $t + d_t(i)$.

Each reward $r_t(i)$ lies in $[0, 1]$. The cost values $c_t(i)$ can only be integer multiples of the unit cost, $1/w$, and are within the range $(0, 1]$. Regarding time delay $d_t(i)$, its domain is $\mathbb{N} \cup \{\infty\}$, which means that the reward $r_t(i)$ will never be observed when the value of $d_t(i)$ is infinite. In the BD-MAB problem, we consider two principal constraints on the time delay distributions of the arms. The first constraint mandates the presence of a Dirac delta function at $t = 0$ with a weight of at least κ, where $0 < \kappa \leq 1$. This means that, on average, at least κ of the rewards are instantaneously observed upon pulling the arm. We refer to the probability of instantaneous

TABLE I: Description of Commonly-Used Notation

Variable	Description
K	The total number of arms.
B	The budget amount.
T_{deadline}	The deadline round.
$r_t(i)$	Reward from pulling arm i at round t.
$\mu^r(i)$	Average for reward distribution of arm i.
$c_t(i)$	Cost from pulling arm i at round t.
$\mu^c(i)$	Average for cost distribution of arm i.
$d_t(i)$	Delay sampled from pulling arm i at round t.
$1/w$	The unit for the cost values.
$t_a(B)$	Budget depletion time of a pulling algorithm a given budget B.
$n_t(i)$	Number of times arm i's reward was observed before round t.
$m_t(i)$	Number of times arm i was pulled before round t.
σ	Lower bound for average cost values.
κ	Lower bound of the probability of instant returns.
γ	Upper bound of the probability of dead returns.
$\gamma(i)$	Probability of dead return for arm i.
$\mathcal{D}(i)$	Delay distribution for arm i.
$\mathcal{C}(i)$	Cost distribution for arm i.
$\mathcal{R}(i)$	Reward distribution for arm i.
BD-UCB$_t(i)$	Budget-Constrained and Deadline-Driven UCB with Delay score for arm i at round t.

returns for each arm i as $\kappa(i)$, which is lower-bounded by κ. In other words, for all arms i, we have $\int_{0^-}^{0^+} \mathcal{D}(i)dt = \kappa(i) \geq \kappa$.

The second constraint caps the expected proportion of rewards returning after the deadline round, T_{deadline}, at γ for all arms, such that $0 \leq \gamma < 1$. Mathematically, this implies that $\int_{T_{\text{deadline}}}^{\infty} \mathcal{D}(i)dt = \gamma(i) \leq \gamma$ for all arms i. For this constraint to hold, we need to assume that T_{deadline} is significantly large compared to the budget depletion time.

Other constraints considered in our model include that the total budget B exceeds the total number of arms K, i.e., $B > K$, allowing each arm to be pulled at least once during the initialization phase. We also place a lower bound σ on the expected value of cost distribution $\mathcal{C}(i)$ for each arm, ensuring that no cost distribution has an average cost lower than σ. In other words, It always holds that $\int_0^1 x\mathcal{C}(i)dx \geq \sigma$. Table 1 shows a description of the commonly-used notation.

B. Phases of the Problem

As discussed earlier, the BD-MAB problem encompasses three different phases with the following specifications:

- *Phase 1:* This phase starts at the beginning of the problem and continues until the budget is depleted at the budget depletion round, denoted by $t_a(B)$ for a given pulling algorithm a. During this phase, an arm is pulled at each round, and all arriving rewards are observed.
- *Phase 2:* This phase begins immediately after the depletion of the budget and lasts until the deadline round, T_{deadline}. In this phase, while no arms are pulled, we continue to observe any incoming rewards.
- *Phase 3:* This final phase follows T_{deadline} and continues indefinitely. No rewards are observed in this phase, and consequently, no arms are pulled. Any rewards that arrive during this phase are considered as dead returns.

The settings of the problem are shown in Algorithm 1. When an arm i is pulled, the environment independently generates three values: the reward $r_t(i)$ that is sampled from the probability distribution function (PDF) $\mathcal{R}(i)$, the cost $c_t(i)$ sampled from the PDF $\mathcal{C}(i)$, and the time delay $d_t(i)$ sampled from the PDF $D(i)$. Note that we abuse the notation in our use of the aforementioned PDFs as we omit the actual independent variable of the PDF (for example, $D(i)$ represents the PDF of the values of time delay when arm i is pulled). Furthermore, $\mu^r(i)$ denotes the mean value of the reward distribution of the arm i, and $\mu^c(i)$ denotes the mean value of the cost of arm i.

Algorithm 1 Settings of BD-MAB

```
for $t = 1, 2, \dots, T_{deadline}$ do
    if $B > 0$ then
        Player pulls an arm $a_t = i$ where $i \in \{1, 2, \dots, K\}$.
        MAB samples $c_t(a_t) \sim \mathcal{C}(a_t)$.
        if $B - c_t(a_t) \geq 0$ then
            MAB samples $(r_t(a_t), d_t(a_t)) \sim (\mathcal{R}(a_t), \mathcal{D}(a_t))$.
            $B \leftarrow B - c_t(a_t)$.
        else
            $B \leftarrow 0$.
    Observe arriving rewards $\{(a_s, r_s(a_s)) | s = t - d_s(a_s)\}$.
```

C. Expected Regret and Metrics

For a reliable criterion for the performance of the player's algorithm, we consider the metric of total regret which is the difference between the algorithm's expected cumulative reward and the expected total reward of the arm with the highest $\frac{(1-\gamma(i))\mu^r(i)}{\mu^c(i)}$. This arm is called the optimal arm (denoted by i^*). This measure is extended from the expected pseudo-regret [30, 31]. The expected regret is defined as:

$$Regret(B) = \max_i \mathbb{E}[\Sigma_{t=1}^{T_{deadline}}(r_t(i))] - \mathbb{E}[\Sigma_{t=1}^{T_{deadline}} r_t(a_t)]$$
$$=(1-\gamma(i^*))\frac{B}{\mu^c(i^*)}\mu^r(i^*) - \mathbb{E}[\Sigma_{t=1}^{t_a(B)}(1-\gamma(a_t))\mu^r(a_t)] = \mathbb{E}[\Sigma_{t=1}^{T_{deadline}}\Delta_{a_t}],$$

where $\Delta_i = \frac{(1-\gamma(i^*))\mu^r(i^*)}{\mu^c(i^*)} - \frac{(1-\gamma(i))\mu^r(i)}{\mu^c(i)}$ $\forall i \in [1, K]$.

In addition, we consider that $m_t(i)$ denotes the number of times an algorithm pulls an arm i before round t. Moreover, $n_t(i)$ represents the number of times a reward from arm i has been observed before round t. Furthermore, we consider $\hat{\mu}_t^r(i)$ to be the empirical average of the rewards observed from arm i before round t, defined as follows:

$$\hat{\mu}_t^r(i) = (1/n_t(i))\Sigma_{s:t>s+d_s(a_s)}\left(\mathbb{I}\{a_s = i\} \times r_s(i)\right),$$

where $\mathbb{I}\{P\}$ is the indicator function, which equals 1 if $P = true$, and 0 if $P = false$. Similarly, we define the empirical average of costs to be the following:

$$\hat{\mu}_t^c(i) = (1/m_t(i))\Sigma_{s=1}^{t-1}\left(\mathbb{I}\{a_s = i\} \times c_s(i)\right).$$

Lastly, before each round t, Our strategy needs to have an upper-bound estimate of the average percentage of rewards arriving after $T_{deadline}$, which is represented by $\hat{\gamma}_t(i)$, and can be evaluated based on the number of observed rewards before round t as follows:

$$\hat{\gamma}_t(i) = \min\{1 - n_t(i)/m_t(i), \gamma\}.$$

The subroutine shown in Algorithm 2 shows how the parameters are updated and will be used in our strategy.

Algorithm 2 Update Parameters(t)

```
Input: $(a_s, r_s(a_s), c_s(a_s), d_s(a_s)) \, \forall s \le t$, $K$, $\gamma$.
Output: NULL. // Just update the values.
for $i \in 1, 2, \ldots, K$ do
    $n_t(i) \leftarrow \Sigma_{s:t>s+d_s(a_s)} \mathbb{I}\{a_s = i\}$.
    $m_t(i) \leftarrow \Sigma_{s=1}^{t-1} \mathbb{I}\{a_s = i\}$.
    $\hat{\mu}_t^r(i) \leftarrow \frac{1}{n_t(i)} \Sigma_{s:t>s+d_s(a_s)} (\mathbb{I}\{a_s = i\} \times r_s(i))$.
    $\hat{\mu}_t^c(i) \leftarrow \frac{1}{m_t(i)} \Sigma_{s=1}^{t-1} (\mathbb{I}\{a_s = i\} \times c_s(i))$.
    if $1 - n_t(i)/m_t(i) \ge \gamma$ then
        $\hat{\gamma}_t(i) = \gamma$.
    else
        $\hat{\gamma}_t(i) = 1 - n_t(i)/m_t(i)$.
Return NULL.
```

Algorithm 3 BD-UCB

```
Input: $K$, $B$, $\kappa$, $\gamma$, $\sigma$, $w$, $T_{\text{deadline}}$.
Output: The set of pulled arms $a_t$ s.t. $t \in [1, t_a(B)]$.
Initialization: $t \leftarrow 1$.
    Pull each arm $i \in 1, 2, \ldots, K$ one time.
    $B \leftarrow B - \sum_{s=1}^{b} c_s(a_s)$.
    Observe $\{(a_s, r_s(a_s)) | s \le K - d_s(a_s), s \le K\}$.
    $t \leftarrow t + K$.
while $B > 0$ do
    Call Update Parameters($t$) subroutine from Algorithm 2.
    BD-UCB$_t(i) \leftarrow \frac{(1-\hat{\gamma}_t(i))\hat{\mu}_t^r(i)}{\hat{\mu}_t^c(i)} + \frac{\left(1+\frac{1}{\sigma}\right)\sqrt{(\log(t-1))/n_t(i)}}{\sigma - \sqrt{(\log(t-1))/n_t(i)}}$.
    Pull arm $a_t = \arg\max_i$ BD-UCB$_t(i)$.
    if $B - c_t(a_t) \ge 0$ then
        $B \leftarrow B - c_t(a_t)$.
        Observe rewards $\{(a_s, r_s(a_s)) | s = t - d_s(a_s)\}$.
        $t \leftarrow t + 1$.
    else
        $B \leftarrow 0$.
Observe $\{(a_s, r_s(a_s)) | t - d_s(a_s) \le s \le T_{\text{deadline}} - d_s(a_s), s < t\}$.
Return $\{a_1, a_2, \ldots a_{t-1}\}$.
```

IV. The Solution of the Problem

In this section, we present our solution to the BD-MAB problem, referred to as BD-UCB algorithm, and describe it in details. This strategy leverages the concept of the UCB while considering the delay and budget constraints associated with each arm. Our strategy is displayed in Algorithm 3.

Given the set of arms K, budget B, delay parameters κ and γ, the average cost lower bound σ, unit cost $1/w$, and deadline round T_{deadline}, the algorithm begins by initializing the variables and pulling each arm once. This provides the algorithm with initial cost, and perhaps some instant rewards, from each arm i. At each subsequent round t, while the budget B is still positive, the algorithm executes the following steps:

- *Parameter Update*: This step includes the invocation of a subroutine Update Parameters(t), which adjusts the empirical averages of rewards, costs, and delay characteristics for each arm based on the past observed values.
- *BD-UCB Score Calculation*: For each arm i, a BD-UCB score is calculated. This score is a modified version of the traditional UCB score, accounting for the delay and budget constraints. The BD-UCB score for an arm i is a trade-off between its empirical observed-reward-to-cost ratio and an exploration term that encapsulates the uncertainty around this ratio. The numerator of the exploitation term of the score is scaled by $(1 - \hat{\gamma}_t(i))$ to factor in the portion of rewards that are not dead returns, which makes the algorithm naturally avoid arms with high probabilities of dead returns.
- *Arm Selection*: The arm a_t with the maximum BD-UCB score is chosen for pulling at round t. This principle is based on optimism in the face of uncertainty, a strategy that aims to balance exploration (trying out arms that have not been pulled much to gain more information) and exploitation (choosing arms that currently seem optimal).
- *Budget Update and Reward Observation*: If pulling the chosen arm doesn't lead to a budget overrun, the cost of pulling the chosen arm is deducted from the budget, and the algorithm proceeds to observe rewards that have arrived during this round from previous arm pulls.

When the budget is depleted, the algorithm stops pulling arms but continues to observe arriving rewards until the deadline round. Finally, the algorithm outputs the sequence of pulled arms when round T_{deadline} comes. The factors we chose in our BD-UCB$_t(i)$ score are carefully chosen to guarantee a bounded expected regret, as shown in the next section.

V. Regret Analysis

Our algorithm extends classical UCB algorithms, but this extension introduces unique challenges. While one might assume the regret analysis for BD-UCB mirrors that of its predecessors, it is more complex due to several factors.

Stochastic reward arrivals add significant complexity. In traditional MAB problems, rewards are received instantly or at predictable intervals. However, with stochastic delays, rewards arrive unpredictably, disrupting the linear time progression assumed in classical bandit problems. This unpredictability complicates reward estimation and decision-making, necessitating new techniques to analyze BD-UCB's regret bound. The budget constraint further complicates matters. Unlike classical MAB, where the optimal policy involves consistently selecting the arm with the highest expected reward or reward-to-cost ratio, the budget constraint forces a more nuanced evaluation of potential rewards against costs, considering the unpredictable arrival times. This requires a more sophisticated strategy, making regret bound derivation more intricate.

The deadline constraint adds another layer of complexity by imposing a hard stop on reward observation beyond a certain point regardless of when rewards arrive. Unlike typical MAB scenarios, this forces the algorithm to optimize both reward-to-cost and time-to-reward ratios under stochastic delays further complicating regret analysis. The combination of stochastic delays, budget constraints, and deadlines requires a novel approach. We first examine the optimal policy's behavior and total reward when all distributions are known (Theorem 1). Next, we analyze the budget depletion time ($t_a(B)$) for BD-UCB and establish bounds for its expected value in Lemma 1. Finally, we present the regret bound for BD-UCB in Theorem 2, with a corollary addressing the case of deterministic costs.

We begin by deriving the upper bound of the optimal policy's expected reward, given that all underlying distributions are known. Theorem 1 provides this upper bound, where i^* represents the arm with the largest expected average-observed-reward-to-cost ratio:

$$i^* = \arg\max_i (1-\gamma(i))\mu^r(i)/\mu^c(i).$$

Theorem 1. *For the stochastic optimization problem:*

$$\max_a \mathbb{E}\left[\sum_{i=1}^{b} \sum_{s:T_{\textit{deadline}} \geq s+d_s(a_s)} \left(\mathbb{I}\{a_s = i\} \times r_s(i)\right)\right], \tag{1}$$

$$\textit{such that:} \quad \textstyle\sum_{t=1}^{t_a(B)} c_t(a_t) \leq B,$$

where $\forall t$, $(r_t(a_t), c_t(a_t), d_t(a_t))$ *are sampled independently from their corresponding random distributions* $(\mathcal{R}(a_t), \mathcal{C}(a_t), \mathcal{D}(a_t))$*, and where* $\int_{t=T_{\textit{deadline}}}^{\infty} \mathcal{D}(i) \leq \gamma$ $\forall i \in \{1, 2, \ldots, K\}$*, the maximum value is upper bounded by*

$$(1-\gamma)\mu^r(i^*)(B+1)/\mu^c(i^*).$$

Proof. Denote $R(B)$ as the maximum of the stochastic optimization problem in Equation 1. It is given that $B = \frac{b}{w} > 0$, given $\forall \ell < b$, and we get:

$$\mathbb{E}[R(\ell/w)] \leq (\ell/w + 1)\,(1-\gamma(i^*))\mu^r(i^*)/\mu^c(i^*),$$

where the term $(1-\gamma(i^*))\mu^r(i^*)$ is the expected overall average observed reward for the optimal arm i^* before T_{deadline}. We now need to prove that:

$$\mathbb{E}[R(b/w)] \leq (b/w + 1)(1-\gamma)\mu^r(i^*)/\mu^c(i^*).$$

In an analysis of the initial stage of the optimal algorithm, say that the algorithm pulls arm i with a probability $z(i)$ at the first round. Regardless of the actions of the optimal algorithm during this initial round, the total probability across all arms, denoted by $\sum_{i=1}^{K} z(i)$, equals 1. The variable $p(i,j)$ denotes the probability that the cost of arm i would be $\frac{j}{w}$. For arm i, the set of values $\{p(i,1), p(i,2), \ldots, p(i,w)\}$ show the cost random distribution $\mathcal{C}(i)$. In scenarios where the optimal algorithm pulls arm v at round 1, $z(v)$ would equal 1, and $z(i)$ would equal 0 $\forall i \neq v$. Now, since $\gamma \geq \gamma(i)$ $\forall i \in \{1, 2, \ldots, K\}$, this setup would lead to the inequalities:

$$\begin{aligned}
\mathbb{E}\left[R\left(\frac{b}{w}\right)\right] &\leq \sum_{i=1}^{b} z(i) \sum_{j=1}^{w} p(i,j)\left(\mathbb{E}\left[r(i) \,|\, c(i) = \frac{j}{w}\right] + R\left(\frac{b-j}{w}\right)\right) \\
&\leq \sum_{i=1}^{b} z(i)((1-\gamma(i))\mu^r(i)) \\
&+ \sum_{i=1}^{b} z(i)\left(((1-\gamma(i^*))\mu^r(i^*)/\mu^c(i^*)) \sum_{j=1}^{w} p(i,j)(1 + (b-j)/w)\right) \\
&\leq (1-\gamma)\mu^r(i^*)\,(1 + b/w)\,/\mu^c(i^*).
\end{aligned}$$

which concludes the proof. $\square$

Now, we need to show both the upper and lower bounds of the expected budget depletion time ($\mathbb{E}[t_a(B)]$) from our BD-UCB algorithm, denoted in this context as algorithm a. The following lemma shows the upper and lower bounds.

Lemma 1. *If for all arms i, the total number of times the arm is pulled before the budget depletion time is $m_{t_a(B)}(i)$. Furthermore, if for all suboptimal arms $i \neq i^*$, there exists strictly positive factors ζ_i and ψ_i such that it holds that $\mathbb{E}\left[m_{t_a(B)}(i)\right] \leq \zeta_i \log t_a(B) + \psi_i$, then we can guarantee:*

$$\mathbb{E}[t_a(B)] \leq \frac{B+1}{\mu^c(i^*)} + \zeta \log\left(\frac{2B+2}{\mu^c(i^*)} + 2\zeta\log(2\zeta) + 2\psi\right) + \psi, \tag{2}$$

$$\mathbb{E}[t_a(B)] > \frac{B-\psi}{\mu^c(i^*)} - \frac{\zeta}{\mu^c(i^*)}\log\left(\frac{2B+2}{\mu^c(i^*)} + 2\zeta\log(2\zeta) + 2\psi\right) - 1, \tag{3}$$

where $\psi = \sum_{i \neq i^*} \psi_i, \zeta = \sum_{i \neq i^*} \zeta_i$.

Proof. First, we consider the blind pulling algorithm that spends the whole budget B on pulling arm i. We denote this pulling algorithm as x_i, and its budget depletion time as $t_{x_i}(B)$. Now, we know that since pulling algorithm x_i only pulls arm i, the average value for $t_{x_i}(B)$ would be:

$$\mathbb{E}\left[t_{x_i}(B)\right] = \lfloor B/\mu^c(i) \rfloor, \tag{4}$$

which can be equivalently formulated as the set of bounds:

$$B/\mu^c(i) - 1 < \mathbb{E}\left[t_{x_i}(B)\right] \leq (B+1)/\mu^c(i). \tag{5}$$

To show this, first, we know that $B = b/w > 0$. Now, assuming Equation 5 holds for all $\ell < b$, then it is also valid for B. This validates the correctness of Equation 5 for $B = (b/w)$. After the first round of the algorithm, the residual budget would be $(b-j)/w$; the expected depletion time acquired by this remaining budget would be $\mathbb{E}[t_{x_i}((b-j)/w)]$. Hence, $t_{x_i}(B)$ can be formulated in the following recursive equation:

$$\mathbb{E}\left[t_{x_i}(B)\right] = \textstyle\sum_{j=1}^{w} p(i,j)\left(1 + \mathbb{E}\left[t_{x_i}((b-j)/w)\right]\right). \tag{6}$$

Now, since $b > m$, we get:

$$\begin{aligned}
\mathbb{E}\left[t_{x_i}(B)\right] &> \textstyle\sum_{j=1}^{w} (p(i,j)/\mu^c(i))\,(B - j/w) \\
&= \textstyle\sum_{j=1}^{w} (p(i,j)/\mu^c(i))B - \sum_{j=1}^{w} (p(i,j)/\mu^c(i))(j/w) \\
&= (B/\mu^c(i))\cancelto{1}{\textstyle\sum_{j=1}^{w} p(i,j)} - (1/\mu^c(i))\cancelto{\mu^c(i)}{\textstyle\sum_{j=1}^{w} p(i,j)\frac{j}{w}} \\
&= B/\mu^c(i) - 1.
\end{aligned} \tag{7}$$

In addition, we have the inequality:

$$\mathbb{E}\left[t_{x_i}(B)\right] \leq \sum_{j=1}^{w} p(i,j)\left(1 + \frac{B - \frac{j}{w} + 1}{\mu^c(i)}\right) = \frac{B+1}{\mu^c(i)}. \tag{8}$$

Proving the bounds in Equation 5. Now, we substitute Equations 7-8 with the conditions provided in the lemma. Therefore, we obtain the following two inequalities:

$$\begin{aligned}
\mathbb{E}[t_a(B)] &\leq \mathbb{E}\left[t_{x_{i^*}}(B)\right] + \textstyle\sum_{i \neq i^*} \zeta_i \mathbb{E}[\log t_a(B)] + \sum_{i \neq i^*} \psi_i \\
&\leq (B+1)/\mu^c(i^*) + \zeta\mathbb{E}[\log t_a(B)] + \psi.
\end{aligned} \tag{9}$$

$$\begin{aligned}
\mathbb{E}[t_a(B)] &\geq \mathbb{E}\left[t_{x_{i^*}}(B - \textstyle\sum_{i \neq i^*} \zeta_i \log t_a(B) - \sum_{i \neq i^*} \psi_i)\right] \\
&> (B - \zeta\mathbb{E}[\log t_a(B)] - \psi)/\mu^c(i^*) - 1.
\end{aligned} \tag{10}$$

But we have the following inequality:

$$\mathbb{E}[\log t_a(B)] \leq \mathbb{E}[t_a(B)]/(2\zeta) + \log(2\zeta) - 1. \tag{11}$$

Lastly, we substitute Equation 11 into Equations 9-10, which gives us the two inequalities shown in Equations 2-3. Doing that concludes the proof. □

To this end, we have everything set up to introduce the following regret bound of the BD-UCB algorithm.

Theorem 2. *The upper bound for the expected regret of the BD-UCB algorithm is:*

$$\begin{aligned}&\mathit{Regret}(B) \leq \Omega + (1-\gamma)\times\\&\left(\nu\log\!\left(\frac{B+1}{\mu^c(i^*)}+\zeta\log\!\left(\frac{2B+2}{\mu^c(i^*)}+2\zeta\log(2\zeta)+2\psi\right)+\psi\right)+\right.\\&\left.\frac{\mu^r(i^*)}{\mu^c(i^*)}\left(\zeta\log\!\left(\frac{2B+2}{\mu^c(i^*)}+2\zeta\log(2\zeta)+2\psi\right)+\psi+\mu^c(i^*)+1\right)\right),\end{aligned} \quad (12)$$

$$\textit{where}\begin{cases}\Delta_i = \frac{(1-\gamma(i^*))\mu^r(i^*)}{\mu^c(i^*)} - \frac{(1-\gamma(i))\mu^r(i)}{\mu^c(i)},\\ \sigma \leq \min_i \mu^c(i),\\ \psi_i = 2+\frac{2\pi^2}{3}, \quad \zeta_i = \left((2+\frac{2}{\sigma}+\Delta_i)/(\Delta_i\sigma)\right)^2,\\ \psi = \sum_{i:i\neq i^*}\psi_i, \quad \zeta = \sum_{i:i\neq i^*}\zeta_i,\\ G_i = \{i|(1-\gamma(i))\mu^r(i) < (1-\gamma(i^*))\mu^r(i^*)\},\\ \nu = \sum_{i\in G_i}\left(\zeta_i\mu^r(i^*) - \zeta_i\mu^r(i)\right),\\ \Omega = (1-\gamma)\sum_{i:i\neq i^*}\left(\psi_i\left(\mu^r(i) - \mu^r(i^*)\right)\right).\end{cases}$$

Proof. Before we start, we need to bound the number of times an arm $i \neq i^*$ is pulled by the time the budget is depleted. Hence, we get the upper-bound:

$$\mathbb{E}\left[m_{t_a(B)}(i)\right] \leq \left(\frac{2+\frac{2}{\sigma}+\Delta_i}{\Delta_i\sigma}\right)^2\log t_a(B) + 2 + \frac{2\pi^2}{3}. \quad (13)$$

Now, we introduce round $\tau_a(i,B)$ such that:

$$\tau_a(i,B) = \left((2+(2/\sigma)+\Delta_i)/(\Delta_i\sigma)\right)^2 \times \log t_a(B). \quad (14)$$

Given a budget depletion time $t_a(B)$ and its corresponding $\tau_a(i,B)$ value, the value of $m_{t_a(B)}(i)$ would be either smaller than $\tau_a(i,B)$, for which it directly follows:

$$\mathbb{E}\left[m_{t_a(B)}(i), m_{t_a(B)}(i) < \tau_a(i,B)\right] < \tau_a(i,B), \quad (15)$$

or greater than or equal to $\tau_a(i,B)$, for which we get:

$$\begin{aligned}&\mathbb{E}\left[m_{t_a(B)}(i), m_{t_a(B)}(i) \geq \tau_a(i,B)\right]\\&=1+\mathbb{E}\left[\textstyle\sum_{t=K+1}^{t_a(B)}\mathbb{I}\{a_t=i\}\,\middle|\, m_{t_a(B)}(i)\geq\tau_a(i,B)\right]\\&\leq\sum_{t=K+1}^{t_a(B)}P\Bigg(\frac{(1-\hat{\gamma_t}(i))\hat{\mu}^r_t(i)}{\hat{\mu}^c_t(i)}+L_t(i)\geq\frac{(1-\hat{\gamma_t}(i^*))\hat{\mu}^r_t(i^*)}{\hat{\mu}^c_t(i^*)}+L_t(i^*),\\&\quad m_t(i)\geq\tau_a(i,B)\Bigg)+\tau_a(i,B)\\&\leq P\left(\frac{(1-\hat{\gamma_t}(i))\hat{\mu}^r_t(i)}{\hat{\mu}^c_t(i)}\geq\frac{(1-\gamma(i))\mu^r(i)}{\mu^c(i)}+L_t(i)\right)\\&\quad+P\left(\frac{(1-\hat{\gamma_t}(i^*))\hat{\mu}^r_t(i^*)}{\hat{\mu}^c_t(i^*)}\leq\frac{(1-\gamma(i^*))\mu^r(i^*)}{\mu^c(i^*)}-L_t(i^*)\right)\\&\quad+P\Bigg(\frac{(1-\gamma(i^*))\mu^r(i^*)}{\mu^c(i^*)}<\frac{(1-\gamma(i))\mu^r(i)}{\mu^c(i)}+2L_t(i),\\&\quad m_t(i)\geq\tau_a(i,B)\Bigg)+\tau_a(i,B),\end{aligned} \quad (16)$$

where we define:

$$L_t(i) = \frac{\left(1+\frac{1}{\sigma}\right)\sqrt{(\log(t-1))/n_t(i)}}{\sigma - \sqrt{(\log(t-1))/n_t(i)}}. \quad (17)$$

Now, we know that if the inequality $\frac{(1-\hat{\gamma_t}(i))\hat{\mu}^r_t(i)}{\hat{\mu}^c_t(i)} \geq \frac{(1-\gamma(i))\mu^r(i)}{\mu^c(i)} + L_t(i)$ holds, event A_1 or event A_2, or both events would happen, such that:

$$\begin{aligned}&A_1 : (1-\hat{\gamma_t}(i))\hat{\mu}^r_t(i) \geq (1-\gamma(i))\mu^r(i) + \alpha_t(i),\\&A_2 : \hat{\mu}^c_t(i) \leq \mu^c(i) - \alpha_t(i),\end{aligned} \quad (18)$$

given that $\alpha_t(i) = \sqrt{\log t/n_t(i)}$. Otherwise, we would get:

$$\begin{aligned}&((1-\hat{\gamma_t}(i))\hat{\mu}^r_t(i))/\hat{\mu}^c_t(i) - ((1-\gamma(i))\mu^r(i))/\mu^c(i)\\&=((1-\hat{\gamma_t}(i))\hat{\mu}^r_t(i) - (1-\gamma(i))\mu^r(i))\,\mu^c(i)/(\hat{\mu}^c_t(i)\mu^c(i))\\&\quad+(\mu^c(i)-\hat{\mu}^c_t(i))\,(1-\gamma(i))\mu^r(i)/(\hat{\mu}^c_t(i)\mu^c(i))\\&<\alpha_t(i)/\hat{\mu}^c_t(i) + \alpha_t(i)(1-\gamma(i))\mu^r(i)/(\hat{\mu}^c_t(i)\mu^c(i))\\&\leq\alpha_t(i)/(\sigma-\alpha_t(i)) + \alpha_t(i)/((\sigma-\alpha_t(i))\,\sigma) = L_t(i).\end{aligned} \quad (19)$$

Applying Chernoff-Hoeffding inequality:

$$\begin{aligned}&P\left((1-\hat{\gamma_t}(i))\hat{\mu}^r_t(i) \geq (1-\gamma(i))\mu^r(i)+\alpha_t(i)\right)\\&\quad\leq\exp\left(-2(\alpha_t(i))^2 n_t(i)\right) = t^{-2},\\&P\left(\hat{\mu}^c_t(i)\leq\mu^c(i)-\alpha_t(i)\right)\\&\quad\leq\exp\left(-2(\alpha_t(i))^2 n_t(i)\right) = t^{-2}.\end{aligned} \quad (20)$$

Hence,

$$\begin{aligned}&P\left(\frac{(1-\hat{\gamma_t}(i))\hat{\mu}^r_t(i)}{\hat{\mu}^c_t(i)}\geq\frac{(1-\gamma(i))\mu^r(i)}{\mu^c(i)}+L_t(i)\right)\\&\leq\sum_{t=1}^{\infty}P\left((1-\hat{\gamma_t}(i))\hat{\mu}^r_t(i)\geq(1-\gamma(i))\mu^r(i)+\alpha_t(i)\right)\\&+\sum_{t=1}^{\infty}P\left(\hat{\mu}^c_t(i)\leq\mu^c(i)-\alpha_t(i)\right)\leq 2\sum_{t=1}^{\infty}t^{-2}\leq 1+\frac{\pi^2}{3}.\end{aligned} \quad (21)$$

Next, we can apply the Chernoff-Hoeffding inequality in the same manner to get:

$$P\left(\frac{(1-\hat{\gamma_t}(i))\hat{\mu}^r_t(i)}{\hat{\mu}^c_t(i)}\leq\frac{(1-\gamma(i))\mu^r(i)}{\mu^c(i)}-L_t(i^*)\right)\leq 1+\frac{\pi^2}{3}. \quad (22)$$

But $\forall n_t(i) \geq \tau_a(i,B)$, we can derive the following two inequalities:

$$\sqrt{\log t/\tau_a(i,B)} < \sigma,$$

$$L_t(i) \leq \frac{\left(1+\frac{1}{\sigma}\right)\sqrt{\log t/\tau_a(i,B)}}{\sigma-\sqrt{\log t/\tau_a(i,B)}} \leq \frac{\Delta_i}{2}.$$

Hence we get:

$$\begin{aligned}&P\Bigg(\frac{(1-\gamma(i^*))\mu^r(i^*)}{\mu^c(i^*)} < \frac{(1-\gamma(i))\mu^r(i)}{\mu^c(i)} + 2L_t(i),\\&\quad n_t(i)\geq\tau_a(i,B)\Bigg) = 0.\end{aligned} \quad (23)$$

Substituting Equations 21, 22, and 23 in Equation 16:

$$\mathbb{E}\left[m_{t_a(B)}(i), m_{t_a(B)}(i)\geq\tau_a(i,B)\right]\leq\tau_a(i,B)+2+\frac{2\pi^2}{3}. \quad (24)$$

We now substitute the bound achieved in Theorem 1 to derive the bound of the expected regret:

$$\begin{aligned}&\mathit{Regret}(B)\leq\\&\left(\frac{(1-\gamma(i^*))\mu^r(i^*)}{\mu^c(i^*)}(B+1)-(1-\gamma(i^*))\mu^r(i^*)\mathbb{E}\left[t_a(B)\right]\right)\\&+\left((1-\gamma(i^*))\mu^r(i^*)\mathbb{E}\left[t_a(B)\right]-\mathbb{E}\left[\textstyle\sum_{t=1}^{t_a(B)}r_t(a_t)\right]\right).\end{aligned} \quad (25)$$

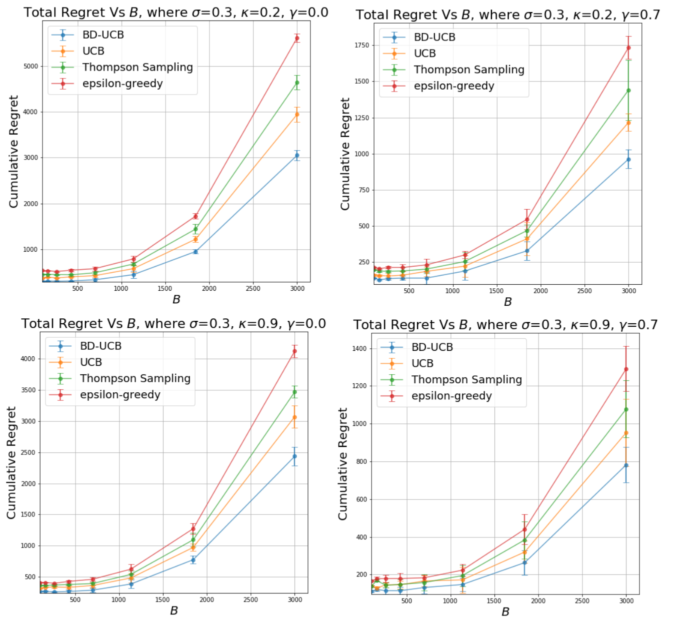

Fig. 2: Total regret versus the budget amount for the different algorithms under different parameters where $\sigma = 0.3$.

To this end, we make use of the bounds we got in Lemma 1 to bound the budget depletion time that shows up in the first term, while setting the specific values of ζ and ψ such that: $\zeta = \sum_{i \neq i^*} \left(\frac{2 + \frac{2}{\sigma} + \Delta_i}{\Delta_i \sigma} \right)^2$ and $\psi = 2(K-1)\left(1 + \frac{\pi^2}{3}\right)$.

For the second term, we notice that it is just a normalized version of the standard MAB settings such that:

$$(1-\gamma(i^*))\mu^r(i^*)\mathbb{E}\left[t_a(B)\right] - \mathbb{E}\left[\sum_{t=1}^{t_a(B)} r_t(a_t)\right] = \mathbb{E}\Big[\sum_{i \neq i^*} m_{t_a(B)}(i)\big((1-\gamma(i^*))\mu^r(i^*) - (1-\gamma(i))\mu^r(i)\big)\Big]. \quad (26)$$

This gives us the bound of the expected regret shown in Equation 12, which concludes the proof. $\square$

By Theorem 2, we showed how, on average, our BD-UCB algorithm guarantees a regret bounded by $O(\log B)$. The following Corollary shows the expected regret bound of BD-UCB if the cost values were fixed (i.e. deterministic).

Corollary 1. *The expected regret of BD-UCB for the case in which $\forall i \in \{1, 2, \dots, K\}$, the cost distribution $\mathcal{C}(i)$ is just one pulse at a specific value $c(i)$ and zero everywhere else, is bounded by $Regret(B)$ shown in Equation 12 with substituting $\zeta_i = \left(\frac{2 + \frac{2}{\min_k c(k)} + \Delta_i}{\Delta_i \min_k c(k)} \right)^2$ and $\psi_i = 2 + \frac{2\pi^2}{3}$.*

VI. Simulations

A. Experimental Settings

Now, we detail the experimental setup we used to validate the performance of the BD-UCB algorithm. Our aim is to illustrate the effectiveness of BD-UCB in navigating the complexities of the BD-MAB problem and compare it to existing algorithms under the same conditions. Some previous similar work regarding modeling delayed MAB problems can be found in the work done in [28–30].

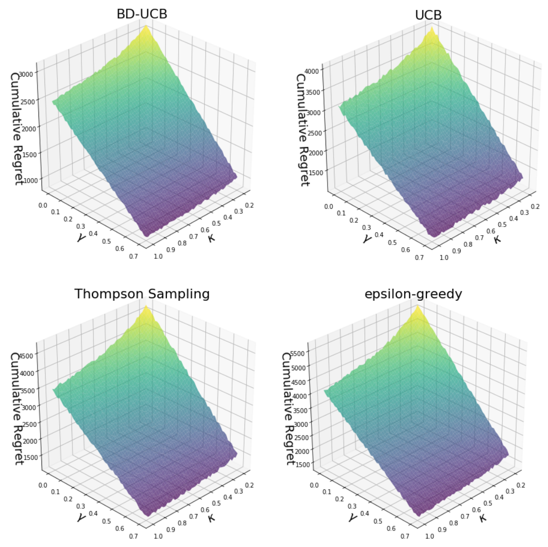

Fig. 3: Total regret versus the time delay distribution bounding parameters κ and σ. The other parameters are set so that $B = 3000$ and $\sigma = 0.3$.

1) Simulation Environment: We conduct our experiments in a synthesized data environment designed to mirror a BD-MAB scenario. The simulation environment includes $K = 30$ arms, each with its own unique reward, cost, and delay distributions. The reward distribution for each arm i is assumed to be a scaled and truncated Gaussian distribution with a mean of $\mu^r(i)$ and a standard deviation of $\sigma^r(i)$. This means the rewards $r_t(i) \sim \bar{\mathcal{N}}(\mu^r(i), (\sigma^r(i))^2)$. The value of the average reward for each arm i is sampled randomly from a uniform distribution that yields a value between 0 and 1. In other words $\mu^r(i) \sim \text{UNIFORM}(0,1)$. The standard deviation of the reward values $\sigma^r(i)$ are set to 0.2 for all arms.

The cost values, on the other hand, are sampled from a different cost distribution for each arm i. Those cost distributions $\mathcal{C}(i)$ follow a discretized, truncated, and scaled version of the Gaussian distribution with a mean value of $\mu^c(i)$ and a standard deviation of $\sigma^c(i)$ for each arm i. The unit cost $1/w$ is set to 0.05. This means that $c_t(i) \sim \bar{\mathcal{N}}_{0.05}(\mu^c(i), (\sigma^c(i))^2)$ for all arms. The values of $\mu^c(i)$ are randomly sampled from a uniform distribution that yields a value between σ and 1. Mathematically, $\mu^c(i) \sim \text{UNIFORM}(\sigma, 1)$ for all arms i. Lastly for the cost, the standard deviations of the cost values $\sigma^c(i)$ are set to 0.2 as well, for all arms i. The value of σ is fixed and takes a value in the range $[0.3, 0.6]$.

The player has a budget of B that is set to have a fixed value from the range of $[100, 300]$. The termination deadline round $T_{deadline}$ is set to a large value compared to the average budget depletion time for all pulling algorithms, more specifically $\lim_{B\to\infty}(B/\sigma)/T_{\text{deadline}} = 0$. Hence, we opt to set the value of $T_{deadline}$ to be $(B/\sigma)^2$ for all experiments. Regarding the delay distributions, they all have a Dirac delta function of weight $\kappa(i)$ at $t = 0$ and another Dirac delta function of weight $\gamma(i)$ at round $T_{\text{deadline}} + 1$. For each arm i, the value of $\kappa(i)$ is first sampled from a uniform random distribution that gives a value between κ and 1. Afterwards, the value of $\gamma(i)$ is sampled from a uniform random distribution that gives a

Algorithm	$\sigma = 0.3$		$\sigma = 0.4$		$\sigma = 0.5$		$\sigma = 0.6$	
	$B = 1000$	$B = 3000$	$B = 1000$	$B = 3000$	$B = 1000$	$B = 3000$	$B = 1000$	$B = 3000$
BD-UCB	205 $\pm$21	841 $\pm$87	319 $\pm$33	710 $\pm$70	205 $\pm$23	811 $\pm$79	215 $\pm$29	783 $\pm$83
UCB	248 $\pm$23	1152 $\pm$102	276 $\pm$28	990 $\pm$107	299 $\pm$32	968 $\pm$91	229 $\pm$33	990 $\pm$96
Thompson Samp.	237 $\pm$22	1209 $\pm$113	315 $\pm$33	1210 $\pm$117	291 $\pm$31	1238 $\pm$136	232 $\pm$34	1322 $\pm$120
Epsilon- greedy	242 $\pm$21	1310 $\pm$129	249 $\pm$26	1330 $\pm$145	249 $\pm$28	1380 $\pm$140	252 $\pm$38	1397 $\pm$138

Fig. 4: Total regret versus the cost distribution bounding parameter σ and budget B. The other parameters are set so that $\gamma = 0.7$, and $\kappa = 0.9$.

value between 0 and $\min\{\gamma, 1 - \kappa(i)\}$.

The value of κ is fixed and takes a value between 0.2 and 1, while the value of γ is fixed and takes a value between 0 and 0.7. Those settings reflect a diverse set of costs and delays that may arise in a real-world scenario. All the distributions are assumed to be stationary throughout the simulation period. The remainder parts of the delay distributions for each arm (i.e., a weight of $1 - \kappa(i) - \gamma(i)$) are modeled as truncated Gaussian distributions that cover the region $t \in [0, T_{\text{deadline}}]$ such that the average of this Gaussian distribution for each arm is sampled from another Gaussian distribution with a mean value of $t = 10000$, and a standard deviation of 2000. The standard deviation of the main truncated Gaussian distribution is set to 2000 as well. Simply adding this truncated Gaussian distribution with the two Dirac delta functions at $t = 0$ and $t = T_{\text{deadline}} + 1$ for each arm yields the final time delay distributions $\mathcal{D}(i)$ used in each instance of our simulations.

2) Replication and Randomization: To ensure the robustness of our results, each simulation scenario is repeated N times, with N set at 50. Between each replication, the simulation environment is randomized, i.e., the random values specified, as the mean reward, cost, and delay for each arm are re-drawn from their respective distributions.

B. Algorithm Comparison

Now, we present the algorithms we will use to compare with our proposed BD-UCB algorithm. Specifically, we consider the standard UCB, Thompson Sampling, and ϵ-greedy algorithms for comparison. These algorithms have been selected due to their wide application and success in addressing various MAB problems. Although not designed to handle both budget and delay constraints simultaneously, these algorithms would provide a reasonable benchmark to evaluate the effectiveness of the proposed BD-UCB algorithm for the BD-MAB.

1) Standard UCB [32]: The classical UCB algorithm balances exploration and exploitation by picking, every round, the arm with the highest UCB score, defined as:

$$\text{UCB}_t(i) = \hat{\mu}_t^r(i) + \sqrt{2 \log t / n_t(i)},$$

which is the optimal strategy for classic MAB settings.

2) Thompson Sampling [33]: Thompson Sampling is a Bayesian approach to the MAB problem that selects arms based on sampled expected rewards $\hat{\mu}_t^r(i)$. The algorithm has demonstrated strong empirical performance in various settings. However, like the standard UCB, it does not incorporate budget or delay constraints into the arm selection process.

3) ϵ-greedy [8]: The ϵ-greedy algorithm is another popular solution to the MAB problem. It explores uniform-randomly all arms with probability ϵ and exploits the best-known arm that has the highest $\hat{\mu}_t^r(i)$ with probability $(1-\epsilon)$ every round.

C. Simulation Results

Extensive simulations reveal a clear hierarchy in algorithm performance across various parameter sets. BD-UCB consistently outperforms other algorithms in BD-MAB settings. Figure 2 illustrates cumulative regret against the player's budget B, which also determines T_{deadline}. Our BD-UCB algorithm shows approximately 25% less regret than the second-best, classical UCB, depending on the problem specifics. As expected, epsilon greedy performs the worst. Thompson Sampling outperforms classical UCB at low σ values but suffers higher regret at high σ values.

The effect of the lower bound of average cost values σ on algorithm performance is shown in Figure 4. Thompson Sampling is notably sensitive to changes in σ, displaying drastic shifts in performance as σ varies, due to its reliance on frequent arm pulls. Interestingly, regret decreases with increasing σ for both UCB algorithms, while it increases for Thompson Sampling and epsilon greedy. This is because higher σ reduces the number of arm pulls, benefiting UCB algorithms due to their formulaic structure. Additionally, lower σ extends the period before T_{deadline}, favoring UCB algorithms by giving them more time to optimize rewards.

Figure 3 shows the impact of delay distribution parameters on algorithm behavior. For most algorithms, regret decreases linearly as γ increases, consistent with the theoretical bounds in Theorem 2. A higher γ means more rewards are accrued after T_{deadline}, lowering comparative regret since the optimal strategy x_{i^*} collects rewards more efficiently. When γ approaches zero, nearly all rewards are gathered post-deadline, driving cumulative regret to zero. Regarding the second delay parameter, κ, our results show a negative correlation with total regret across algorithms. This is due to a reduced fraction of immediate feedback, slowing the update process for estimated mean rewards $\hat{\mu}_t^r(i)$ and delaying the recalibration of metrics like UCB scores. Consequently, this leads to impaired algorithm performance, as reflected in increased regret. This analysis underscores the significant impact of time delay parameters on algorithm efficacy in dynamic environments.

VII. Conclusion

This paper presents a new extension of the Multi-Armed Bandit (MAB) problem, introducing the novel Budget-Constrained and Deadline-Driven MAB with Delay (BD-MAB) variation. By combining budget constraints and time delays in the MAB framework, this work addresses a previously unexplored scenario with practical implications for applications that require active learning with delays. We proposed a solution, named the Budget-Constrained and Deadline-Driven UCB with Delay (BD-UCB) algorithm, which efficiently manages the three distinct phases, time delay distributions, and budget and deadline constraints of the BD-MAB problem. Theoretical analysis of regret confirms the effectiveness of the BD-UCB algorithm. The numerical simulations prove that BD-UCB consistently outperforms other algorithms, showing a better regret than the regular UCB by around 25%. Future work would include tackling a version of the problem with looser constraints on the time delay distributions of the arm to make the problem more general.

References

[1] Slivkins, Aleksandrs. "Introduction to multi-armed bandits." Foundations and Trends® in Machine Learning 12.1-2 (2019).

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

[3] Mahajan, A., & Teneketzis, D. (2008). Multi-armed bandit problems. In Foundations and applications of sensor management (pp. 121-151). Boston, MA: Springer US.

[4] 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).

[5] Chen, W., Wang, Y., & Yuan, Y. (2013, February). Combinatorial multi-armed bandit: General framework and applications. In International conference on machine learning (pp. 151-159).

[6] Sawwan, A., & Wu, J. (2024). Diversity-based recruitment in crowdsensing by combinatorial multi-armed bandits. Tsinghua Science and Technology.

[7] Rajesh Chauhan, D., Unnikrishnan, A., & Boyles, S. D. (2022). Maximum Profit Facility Location and Dynamic Resource Allocation for Instant Delivery Logistics. Transportation Research Record, 2676(7), 697-710.

[8] Tran-Thanh, L., Chapman, A., De Cote, E. M., Rogers, A., & Jennings, N. R. (2010, July). Epsilon–first policies for budget–limited multi-armed bandits. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 24, No. 1).

[9] Tran-Thanh, L., Chapman, A., Rogers, A., & Jennings, N. (2012). Knapsack based optimal policies for budget–limited multi–armed bandits. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 26, No. 1, pp. 1134-1140).

[10] Ding, W., Qin, T., Zhang, X. D., & Liu, T. Y. (2013, June). Multi-armed bandit with budget constraint and variable costs. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 27, No. 1, pp. 232-238).

[11] Wan, Z., Zhang, Z., Li, T., Zhang, J., & Sun, X. (2023, June). Quantum multi-armed bandits and stochastic linear bandits enjoy logarithmic regrets. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 8, pp. 10087-10094).

[12] Meidani, K., Mirjalili, S., & Farimani, A. B. (2022). MAB-OS: multi-armed bandits metaheuristic optimizer selection. Applied Soft Computing, 128, 109452.

[13] Zhou, X., & Ji, B. (2022). On kernelized multi-armed bandits with constraints. Advances in Neural Information Processing Systems, 35, 14-26.

[14] Sawwan, A., & Wu, J. (2024, June). A Combinatorial Multi-Armed Bandit Approach for Stochastic Facility Allocation Problem. In Proceedings of the 2024 Workshop on Advanced Tools, Programming Languages, and PLatforms for Implementing and Evaluating algorithms for Distributed systems (pp. 1-10).

[15] Zhao, Q. (2022). Multi-armed bandits: Theory and applications to online learning in networks. Springer Nature.

[16] Schwartz, E. M., Bradlow, E. T., & Fader, P. S. (2017). Customer acquisition via display advertising using multi-armed bandit experiments. Marketing Science, 36(4), 500-522.

[17] Tang, Y., Wang, Y., & Zheng, Z. (2024, April). Stochastic Multi-Armed Bandits with Strongly Reward-Dependent Delays. In International Conference on Artificial Intelligence and Statistics.

[18] Song, Y., & Jin, H. (2021, May). Minimizing entropy for crowdsourcing with combinatorial multi-armed bandit. In IEEE INFOCOM 2021- Conference on Computer Communications.

[19] Zhou, Z., Wang, Y., Mamani, H., & Coffey, D. G. (2019). How do tumor cytogenetics inform cancer treatments? dynamic risk stratification and precision medicine using multi-armed bandits. Dynamic Risk Stratification and Precision Medicine Using Multi-armed Bandits (June 17, 2019).

[20] Shi, L., Wang, J., & Wu, T. (2023, July). Statistical inference on multi-armed bandits with delayed feedback. In International Conference on Machine Learning (pp. 31328-31352). PMLR.

[21] Zhou, T., Wang, Y., Yan, L., & Tan, Y. (2023). Spoiled for choice? Personalized recommendation for healthcare decisions: A multiarmed bandit approach. Information Systems Research, 34(4), 1493-1512.

[22] Rastegar, F., & Fooladi, M. D. T. (2019, February). Online virtual machine assignment using multi-armed bandit in cloud computing. In 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon).

[23] Wang, X., Ye, J., & Lui, J. C. (2022, May). Decentralized task offloading in edge computing: A multi-user multi-armed bandit approach. In IEEE INFOCOM 2022-IEEE Conference on Computer Communications (pp. 1199-1208). IEEE.

[24] Ouyang, T., Chen, X., Zhou, Z., Li, R., & Tang, X. (2021). Adaptive user-managed service placement for mobile edge computing via contextual multi-armed bandit learning. IEEE Transactions on Mobile Computing, 22(3), 1313-1326.

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

[26] Xia, Y., Ding, W., Zhang, X. D., Yu, N., & Qin, T. (2016, February). Budgeted bandit problems with continuous random costs. In Asian conference on machine learning (pp. 317-332).

[27] Xia, Y., Qin, T., Ding, W., Li, H., Zhang, X., Yu, N., & Liu, T. Y. (2017). Finite budget analysis of multi-armed bandit problems. Neurocomputing, 258, 13-29.

[28] 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).

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

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

[31] Sawwan, A., & Wu, J. (2023, May). A New Framework: Short-Term and Long-Term Returns in Stochastic Multi-Armed Bandit. In 42th IEEE International Conference on Computer Communications (IEEE INFOCOM 2023).

[32] Garivier, A., & Moulines, E. (2011, October). On upper-confidence bound policies for switching bandit problems. In International Conference on Algorithmic Learning Theory (pp. 174-188). Berlin, Heidelberg: Springer Berlin Heidelberg.

[33] Agrawal, S., & Goyal, N. (2012, June). Analysis of thompson sampling for the multi-armed bandit problem. In Conference on learning theory. JMLR Workshop and Conference Proceedings.