

Stochastic Contextual Bandits with Known Reward Functions

Pranav Sakulkar and Bhaskar Krishnamachari

Ming Hsieh Department of Electrical Engineering

Viterbi School of Engineering

University of Southern California, Los Angeles, CA, USA

{sakulkar, bkrishna}@usc.edu

Abstract

Many sequential decision-making problems in communication networks such as power allocation in energy harvesting communications, mobile computational offloading, and dynamic channel selection can be modeled as contextual bandit problems which are natural extensions of the well-known multi-armed bandit problem. In these problems, each resource allocation or selection decision can make use of available side-information such as harvested power, specifications of the jobs to be offloaded, or the number of packets to be transmitted. In contextual bandit problems, at each step of a sequence of trials, an agent observes the side information or context, pulls one arm and receives the reward for that arm. We consider the stochastic formulation where the context-reward tuples are independently drawn from an unknown distribution in each trial. The goal is to design strategies for minimizing the expected cumulative regret or reward loss compared to a distribution-aware genie. We analyze a setting where the reward is a known non-linear function of the context and the chosen arm's current state which is the case with many networking applications. This knowledge of the reward function enables us to exploit the obtained reward information to learn about rewards for other possible contexts. We first consider the case of discrete and finite context-spaces and propose DCB(ϵ), an algorithm that yields regret which scales logarithmically in time and linearly in the number of arms that are not optimal for any context. This is in contrast with existing algorithms where the regret scales linearly in the total number of arms. Also, the storage requirements of DCB(ϵ) do not grow with time. DCB(ϵ) is an extension of the UCB1 policy for the standard multi-armed bandits to contextual bandits using sophisticated proof techniques for regret analysis. We then study continuous context-spaces with Lipschitz reward functions and propose CCB(ϵ, δ), an algorithm that uses DCB(ϵ) as a subroutine. CCB(ϵ, δ) reveals a novel regret-storage trade-off that is parametrized by δ . Tuning δ to the time horizon allows us to obtain sub-linear regret bounds, while requiring sub-linear storage. Joint learning for all the contexts results in regret bounds that are unachievable by any existing contextual bandit algorithm for continuous context-spaces. Similar performance bounds are also shown to hold for unknown horizon case by employing a doubling trick.

Index Terms

Contextual bandits, multi-armed bandits (MABs), online learning.

I. INTRODUCTION

Many problems in networking such as dynamic spectrum allocation in cognitive radio networks involve sequential decision making under the face of uncertainty. The multi-armed bandit problem (MAB, see [1], [2], [3], [4]), a fundamental online learning framework, has been previously applied in networks for multi-user channel allocation [5] and distributed opportunistic spectrum access [6]. These problems also appear widely in online advertising and clinical trials. In the standard MAB version, an agent is presented with a sequence of trials where in each trial it has to choose an arm from a set of K arms, each providing stochastic rewards over time with unknown distributions. The agent receives a payoff or reward based on its action in each trial. The goal is to maximize the total expected reward over time. On one hand, different arms need to be explored often enough to learn their rewards and on the other hand, prior observations need to be exploited to maximize the immediate reward. MAB problems, therefore, capture the fundamental trade-off between exploration and exploitation that appear widely in these decision making problems.

The contextual bandit problem considered in this paper is a natural extension of the basic MAB problem where the agent can see a context or some hint about the rewards in the current trial that can be used to make a better decision. This makes contextual bandits more general and more suitable to various practical applications such as ad placements [7], [8] and personalized news recommendations [9], since settings with no context information are rare in practice. As we show in this work, contextual bandits are also applicable to many problems arising in wireless communications and networking.

In this paper, we consider contextual bandit problems where the relationship between the contexts and the rewards is known, which we argue is the case with many network applications. In this case, the reward information revealed after an arm-pull for a certain context can also be used to learn about rewards for other contexts as well, since the reward function is already known. The arm-pulls for one context thus help in reducing the exploration requirements of other contexts. Performance of any bandit algorithm is measured in terms of its regret, defined as the difference between the expected cumulative rewards of the Bayes optimal algorithm and that of the agent. Since the bulk of the regret in bandit problems is due to the exploration of different arms, the knowledge of reward function helps in reducing the regret substantially. As previous bandit algorithms cannot exploit this knowledge, we develop new contextual bandit algorithms for this setting. Also, the previously studied contextual bandit problems either assume linear reward mapping [9], [10], [11], [12] or do not make explicit assumptions about the exact structure of reward functions [7], [13], [14], [8], [15]. It must, however, be noted that the network applications may often involve non-linear reward functions and thus warrant a new approach. The goal of this paper is to study these contextual bandits and develop efficient algorithms for them.

A. Motivating Examples

We present some examples of problems arising in communication networks where a contextual bandit formulation is applicable. In each case, there are a few key ingredients: a set of *arms* (typically corresponding to resources such as channels or servers) with associated *attributes* or performance measures (such as throughput capacity, channel

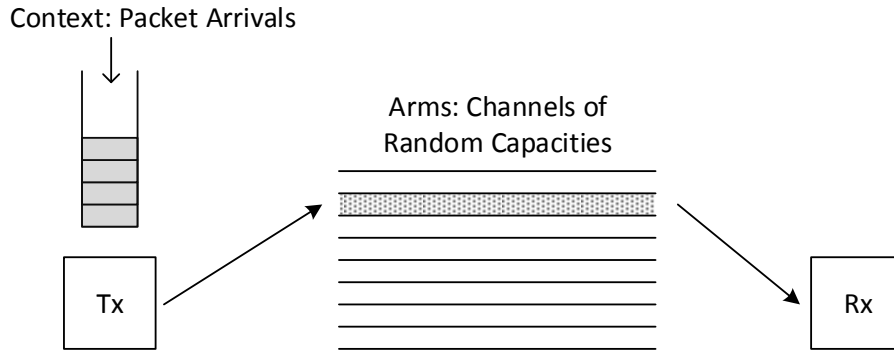


Fig. 1: Dynamic channel selection problem.

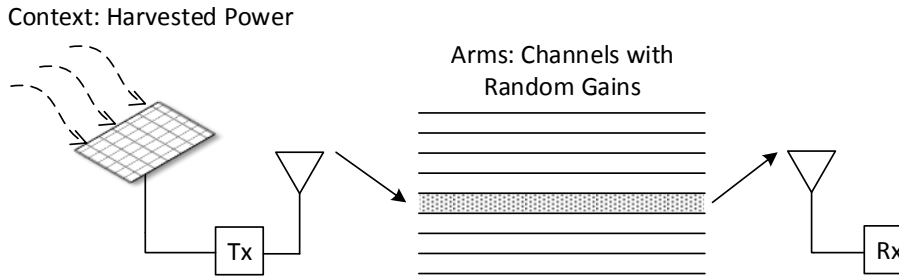


Fig. 2: Power-aware channel selection in energy harvesting communications.

gain, processing rate) evolving stochastically with an unknown distribution, and a *context* known to the user, such that the *reward* obtained at each step is a known (possibly non-linear) function of the context and the random attribute of the arm selected in that step.

1) *Dynamic Channel Selection*: The problem of selecting the best one of several capacity-limited channels every slot based on the available number data packets can be modeled as a contextual bandit problem as shown in figure 1. Here, arms correspond to the different channels available, arm-values or attributes are the instantaneous capacities of these channels which can be assumed to vary stochastically over time (e.g., due to fading or interference in the case of wireless networks) and the context is the number of data-bits that need to be transmitted during the slot. The reward representing the number of successfully transmitted bits over the chosen channel can be expressed as

$$g(y, x) = \min \{y, x\}, \quad (1)$$

where y is the number of data-bits to be transmitted, x is the capacity of the channel. In this problem, channels are to be selected sequentially over the trials based on the context information. Assuming no prior information

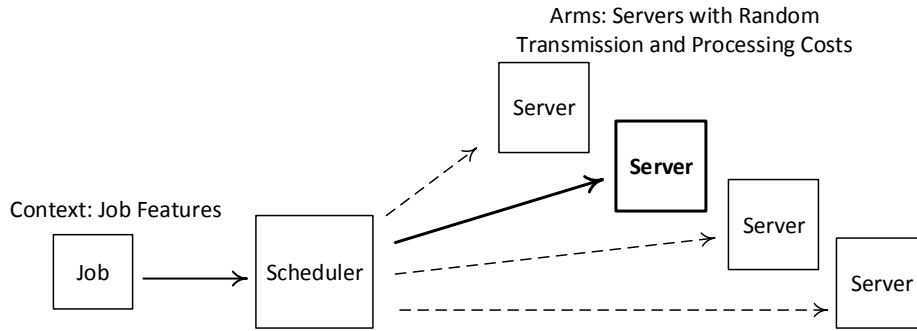


Fig. 3: Job-aware computational offloading problem.

about the channels, the goal in this problem is to maximize the number of successfully transmitted bits over time. When the number of bits to be sent at each time are finite, this represents the case of discrete contextual bandits considered in this paper.

2) *Energy Harvesting Communications*: Consider a power-aware channel selection problem in energy harvesting communications shown in figure 2. In every slot, the current harvested power available for transmission p is known to the transmitter as context. Based on this context information, a channel is to be selected sequentially in each slot for transmission. The reward function in this case is the power-rate function, which can be assumed to be proportional to the AWGN channel capacity $\log(1 + px)$, where x denotes the instantaneous channel gain-to-noise ratio (the arm attribute) varying stochastically over time. The goal of this problem is to maximize the sum-rate over time. Note that the harvested power p can even be a continuous random variable, and $\log(1 + px)$ is Lipschitz continuous in p for all $x \geq 0$. This represents the case of continuous contextual bandits considered in the paper.

3) *Multi-Objective Optimization in Computational Offloading*: Computational offloading involves transferring job-specific data to a remote server and further computations on the server. These problems involve rewards that could be functions of multiple objectives such as transmission latency and processing time required on the server. For each server, the reward would depend on various attributes such as the data-rate of the transmission link to the server and the current effective processing rate of the server, which could be unknown stochastic processes, as well as specific features of each job such as the data transmission and processing requirements which act as the context for this problem. As shown in figure 3, this problem of sequential server allocation for various jobs to maximize the cumulative reward over time can thus be formulated as a suitable contextual bandit problem.

B. Contributions

We first analyze the case of discrete and finite context-spaces and propose a policy called discrete contextual bandits or DCB(ϵ) that requires $O(MK)$ storage, where M is the number of distinct contexts and K is the number of arms. DCB(ϵ) uses upper confidence bounds similar to UCB1, a standard stochastic MAB algorithm from [3] and

yields a regret that grows logarithmically in time and linearly in the number of arms not optimal for any context. A key step in our analysis to use a novel proof technique to prove a constant upper bound on the regret contributions of the arms which are optimal for some context. In doing so, we also prove a high probability bound for UCB1 on the number of pulls of the optimal arm in the standard MAB problem. This high probability bound can be independently used in other bandit settings as well. Further, we use the standard MAB asymptotic lower bound results from [1] to show the order optimality of DCB(ϵ). Note that DCB(ϵ) outperforms UCB1 and Multi-UCB, a discretized version of the Query-ad-clustering algorithm from [8]. Regret of UCB1 scales linearly with time as it ignores the context information completely. Multi-UCB uses the context information to run separate instances of UCB1 for each context, but is unable to exploit the reward function knowledge. Regret of Multi-UCB, therefore, grows logarithmically in time, but unlike DCB(ϵ) it scales linearly in the total number of arms.

For continuous context-spaces with Lipschitz rewards, we propose a novel algorithm called continuous contextual bandits or CCB(ϵ, δ) that quantizes the context-space and uses DCB(ϵ) over the quantized space as a subroutine. Our analysis of CCB(ϵ, δ) reveals an interesting trade-off between the storage requirement and the regret, where desired performance can be achieved by tuning the parameter δ to the time horizon. Decreasing the quantization step-size increases the storage requirements while decreasing the regret. Thus by exploiting the reward-function knowledge for joint learning across contexts, CCB(ϵ, δ) is able to obtain regrets even smaller than the lower bound of $\Omega(T^{2/3})$ from [8] for continuous contextual bandits without such knowledge. For the case of unknown time horizon, we employ a doubling technique that provides similar regret guarantees while using the same amount of storage. CCB(ϵ, δ), therefore, empowers the system designer by giving control over δ to tune the regret performance.

C. Organization

This paper is organized as follows. We first formulate the general contextual bandit problem in section III. In section IV, we discuss various algorithmic approaches to discrete contextual bandits and propose our DCB(ϵ) algorithm. Regret analysis of this algorithm is performed in section V followed by a discussion on the regret lower bound for the discrete case. We extend the algorithm to continuous contexts and analyze the storage vs regret trade-off in section VI. Results of numerical simulations are presented in section VII. Finally, we summarize our contributions and discuss some avenues for future work in section VIII. The detailed proofs of the theorems are provided in appendices at the end of this paper.

II. RELATED WORK

Lai and Robbins [1] wrote one of the earliest papers on the stochastic MAB problem and provided an asymptotic lower bound of $\Omega(K \log T)$ on the expected regret for any bandit algorithm. In [2], sample-mean based upper confidence bound (UCB) policies are presented that achieve the logarithmic regret asymptotically. In [3], several variants of the UCB based policies, including UCB1, are presented and are proved to achieve the logarithmic regret bound uniformly over time for arm-distributions with finite support.

Contextual bandits extend the standard MAB setting by providing some additional information to the agent about each trial. In the contextual bandits studied in [9], [10], [11], [12], the agent observes feature vectors corresponding to each arm. The expected reward of the chosen arm is assumed to be a linear function of its corresponding context vector. LinRel [10] and LinUCB [11] use the context features to estimate the mean reward and its corresponding confidence interval for each arm. The arm that achieves the highest UCB which is the sum of mean reward and its confidence interval is pulled. These confidence intervals are defined in such a way that the expected reward lies in the confidence interval around its estimated mean with high probability. In LinRel and LinUCB, mean and covariance estimates of the linear function parameters are used to estimate the upper confidence bounds for the arm rewards. LinUCB algorithm has also been empirically evaluated on Yahoo!’s news recommendation database in [9]. An algorithm based on Thompson sampling is proposed in [12] for bandits with linear rewards. In this Bayesian approach, the agent plays an arm based on its posterior probability of having the best reward. The posterior probabilities are then updated according to the reward obtained after each pull. LinRel and LinUCB achieve a regret bound of $\tilde{O}(\sqrt{T})$ while the Thompson sampling algorithm achieves a bound of $\tilde{O}(\sqrt{T^{1+\epsilon}})$ for any $\epsilon > 0$, where T is the time horizon. Since these algorithms consider linear functions and store estimates of the parameters characterizing the rewards, their storage requirements do not increase with time.

More general contextual bandits that do not assume any specific relation between the context and reward vectors are studied in [7], [13], [14]. The epoch-greedy algorithm [7] separates the exploration and exploitation steps and partitions the trials in different epochs. Each epoch starts with one exploration step followed by several exploitation steps. It stores the history of contexts, actions and rewards for the exploration steps and requires $O(T^{2/3})$ storage. Its regret is also upper bounded by $\tilde{O}(T^{2/3})$. RandomizedUCB [13], however, does not make any distinction between exploration and exploitation. At each trial it solves an optimization problem to obtain a distribution over all context-action mappings and samples an arm as the action for the trial. Its regret bound is $\tilde{O}(\sqrt{T})$, but it needs $O(T)$ storage as it stores the history of all trials. ILTCB [14] is a more efficient algorithm that solves an optimization problem only on a pre-specified set of trials. However, it still needs $O(T)$ storage. Since these general contextual bandit algorithms do not make any assumption on the context-reward mapping, they store the raw history of trials and do not summarize the information into meaningful statistics. Storage requirements for these approaches, therefore, increase with time.

Contextual bandits in general metric spaces are studied in [8], [15] under the assumption of Lipschitz reward function. This formulation is also able to handle the case of continuous contexts unlike the previously discussed general contextual bandit problems. Query-ad-clustering algorithm [8] partitions the metric spaces in uniform clusters, whereas the meta-algorithm from [15] considers adaptive partitions and uses a general bandit algorithm as a subroutine. Upper confidence indexes are evaluated knowing the corresponding clusters of incoming contexts, the clusters-wise empirical means of previously obtained rewards and the number of pulls for each cluster. Regret bounds for these approaches involve the packing and covering dimensions of the context and arm spaces.

Recently, contextual bandits with budget and time constraints have been studied in [16], [17]. Resourceful contextual bandits from [16] consider a general setting with random costs associated with the arms, a continuous

context-space and multiple budget constraints. An algorithm called Mixture_Elimination that achieves $O(\sqrt{T})$ regret is proposed for this problem. A simplified model with fixed costs, discrete and finite contexts, exactly one time constraint and one budget constraint is considered in [17]. For this model, an algorithm called UCB_ALP achieving $O(\log T)$ regret is proposed. In these problem formulations, time and budget constraints also affect the regular exploration-exploitation trade-off.

III. PROBLEM FORMULATION

In the general stochastic contextual bandit problem, there is a distribution D over (\mathbf{y}, \mathbf{r}) , where $\mathbf{y} \in \mathcal{Y}$ is a context vector in context space \mathcal{Y} and $\mathbf{r} \in \mathbb{R}^K$ is the reward vector containing entries corresponding to each arm. The problem is a repeated game such that at each trial t , an independent sample $(\mathbf{y}_t, \mathbf{r}_t)$ is drawn from D , context vector \mathbf{y}_t is presented to the agent. Once the agent chooses an arm $a_t \in \{1, \dots, K\}$, it receives a reward $r_{a,t}$ which is the a_t -th element of the vector. A contextual bandit algorithm \mathcal{A} chooses an arm a_t to pull at each trial t , based on the previous history of contexts, actions, their corresponding rewards: $(\mathbf{y}_1, a_1, r_{a,1}), \dots, (\mathbf{y}_{t-1}, a_{t-1}, r_{a,t-1})$ and the current context \mathbf{y}_t . The goal is to maximize the total expected reward $\sum_{t=1}^T \mathbb{E}_{(\mathbf{y}_t, \mathbf{r}_t) \sim D} [r_{a,t}]$.

Consider a set \mathcal{H} consisting of hypotheses $h : \mathcal{Y} \rightarrow \{1, \dots, K\}$. Each hypothesis maps the context \mathbf{y} to an arm a . The expected reward of a hypothesis h is expressed as

$$R(h) = \mathbb{E}_{(\mathbf{y}, \mathbf{r}) \sim D} [r_{h(\mathbf{y})}]. \quad (2)$$

The Bayes optimal hypothesis h^* refers to the hypothesis with maximum expected reward $h^* = \arg \max_{h \in \mathcal{H}} R(h)$ and maps contexts as $h^*(\mathbf{y}) = \arg \max_{a \in \{1, \dots, K\}} \mathbb{E} [r_a | \mathbf{y}]$. Agent's goal is to choose arms sequentially and compete with the hypothesis h^* . Regret of the agent's algorithm \mathcal{A} is defined as the difference in expected reward accumulated by the best hypothesis h^* and that by its algorithm over different trials, which can be expressed as

$$\mathfrak{R}_{\mathcal{A}}(n) = nR(h^*) - \mathbb{E} \left[\sum_{t=1}^n r_{\mathcal{A}(t),t} \right], \quad (3)$$

where $\mathcal{A}(t)$ denote the arm pulled by algorithm \mathcal{A} in t -th trial. Note that the expectation in (3) is over the sequence of random samples from D and any internal randomization potentially used by \mathcal{A} . The reward maximization problem is equivalently formulated as a regret minimization problem. Smaller the regret $\mathfrak{R}_{\mathcal{A}}(n)$, better is the policy. We would like the regret to be sublinear with respect to time n so that the time-averaged regret will converge to zero. These policies are asymptotically Bayes optimal, since $\lim_{n \rightarrow \infty} \frac{\mathfrak{R}_{\mathcal{A}}(n)}{n} = 0$.

Let $x_{a,t}$ denote the state or the value of a -th arm at t -th slot and \mathcal{X} denote the set of all possible states of an arm. We assume that the reward is a known bounded function of the selected arm's state and the context represented as

$$r_{a,t} = g(\mathbf{y}_t, x_{a,t}), \quad (4)$$

where $|g(\mathbf{y}, x)| \leq B$ for some $B > 0$ and $\forall \mathbf{y} \in \mathcal{Y}, x \in \mathcal{X}$. Note that, in the standard MAB problems, reward of an arm is same as its state. We assume semi-bandit feedback, which means only the state of the agent's chosen arm is revealed. If the reward is an injective function of the arm state for all contexts, then the arm state can be

inferred without ambiguity from the reward observation, since the current context is already known to the user. Thus, from now on we assume that the chosen arm's state is either revealed by semi-bandit feedback or inferred from the reward observation. In this problem formulation, the context space \mathcal{Y} can be either discrete or continuous. We first develop an algorithm called DCB(ϵ) for discrete context spaces in section IV and later propose its extension CCB(ϵ, δ) for continuous context spaces in section VI assuming that the reward function is Lipschitz with respect to the context variable.

IV. DISCRETE CONTEXTS

In this section, we consider discrete and finite context space $\mathcal{Y} = \{\mathbf{y}_{(1)}, \dots, \mathbf{y}_{(M)}\}$ with M elements. As a convention, we use j and k to index the arms and i to index the discrete contexts.

A. Naive Approach

In the standard MAB setting, the reward of an arm is same as its value and all trials are stochastically identical. UCB1 policy, proposed by Auer *et al.* in [3], pulls the arm that maximizes $\bar{X}_k + \sqrt{2 \ln n / m_k}$ at n -th slot, where \bar{X}_k is the mean observed value of the k -th arm and m_k is the number of pulls of k -th arm. The second term is the size of the one-sided confidence interval for the empirical mean within which the actual expected reward value falls with high probability. Applying such standard MAB algorithms directly to our problem will result in poor performance, as they ignore the context-reward relationship. Playing the arm with highest mean value is not necessarily a wise policy, since the trials are not all identical. When every context in \mathcal{Y} appears with a non-zero probability, playing the same arm over different trials implies that a suboptimal arm is pulled with a non-zero probability at each trial. The expected regret, therefore, grows linearly over time. Although the naive approach is not directly applicable, it serves as one of the benchmarks to compare our results against.

B. Multi-UCB

Main drawback of the plain UCB1 policy is that it ignores the contexts completely. One way to get around this problem is to run a separate instance of UCB1 for each context. Each UCB1 can be tailored to a particular context, where the goal is to learn the arm-rewards for that context by trying different arms over time. UCB1 instance for context \mathbf{y}_i picks the arm that maximizes $\bar{g}_{i,k} + \sqrt{2 \ln n_i / m_{i,k}}$, where $\bar{g}_{i,k}$ denotes the empirical mean of the rewards obtained from previous $m_{i,k}$ pulls of k -th arm for context \mathbf{y}_i and n_i denotes the number of occurrences of that context till the n -th trial. Since each instance incurs a regret that is logarithmic in time, the overall regret of Multi-UCB1 is also logarithmic in time. Compared to the naive approach where the regret is linear in time, this is a big improvement. Notice that Multi-UCB learns the rewards independently for each context and does not exploit the information revealed by the pulls for other contexts. Unless a single arm is optimal for all contexts, every arm's contribution to the regret is logarithmic in time, since they are explored independently for all contexts. Hence the regret scales as $O(MK \log n)$.

C. New Policy

Let $\theta_{i,j}$ be the expected reward obtained by pulling the j -th arm for context $\mathbf{y}_{(i)}$ which is evaluated as

$$\theta_{i,j} = \mathbb{E}[r_j | \mathbf{y}_{(i)}] = \mathbb{E}[g(\mathbf{y}_{(i)}, x_j)]. \quad (5)$$

Let the corresponding optimal expected reward be $\theta_i^* = \theta_{i,h^*(\mathbf{y}_{(i)})}$. We define G_i , the maximum deviation in the rewards, as

$$G_i = \sup_{x \in \mathcal{X}} g(\mathbf{y}_{(i)}, x) - \inf_{x \in \mathcal{X}} g(\mathbf{y}_{(i)}, x). \quad (6)$$

Since the reward function is assumed to be bounded over the domain of arm values, $G_i \leq 2B$.

Similar to Multi-UCB, the idea behind our policy is to store an estimator of $\theta_{i,j}$ for every context-arm pair. For these statistics, we compute upper confidence bounds similar to UCB1 [3] depending on the number of previous pulls of each arm and use them to decide current action. Since the selected arm can be observed or inferred irrespective of the context, it provides information about possible rewards for other contexts as well. This allows joint learning of arm-rewards for all contexts. Intuitively, this policy should be at least as good as Multi-UCB. Our policy, shown in algorithm 1, is called as Discrete Contextual Bandit or DCB(ϵ). In table I, we summarize the notations used.

Algorithm 1 DCB(ϵ)

- 1: **Parameters:** $\epsilon > 0$.
- 2: **Initialization:** $(\hat{\theta}_{i,j})_{M \times K} = (0)_{M \times K}$, $(m_j)_{1 \times K} = (0)_{1 \times K}$.
- 3: **for** $n = 1$ to K **do**
- 4: Set $j = n$ and pull j -th arm;
- 5: Update $\hat{\theta}_{i,j}$, $\forall i : 1 \leq i \leq M$, as $\hat{\theta}_{i,j} = g(\mathbf{y}_{(i)}, x_{j,n})$ and m_j as $m_j = 1$;
- 6: **end for**
- 7: // MAIN LOOP
- 8: **while** 1 **do**
- 9: $n = n + 1$;
- 10: Given the context $\mathbf{y}_n = \mathbf{y}_{(i)}$, pull the arm $\mathcal{A}(n)$ that solves the maximization problem

$$\mathcal{A}(n) = \arg \max_{j \in \mathcal{S}} \left\{ \hat{\theta}_{i,j} + G_i \sqrt{\frac{(2 + \epsilon) \ln n}{m_j}} \right\}; \quad (7)$$

- 11: Update $\hat{\theta}_{i,j}$ and m_j , $\forall i : 1 \leq i \leq M$ as

$$\hat{\theta}_{i,j}(n) = \begin{cases} \frac{\hat{\theta}_{i,j}(n-1)m_j(n-1) + g(\mathbf{y}_{(i)}, x_{j,n})}{m_j(n-1) + 1}, & \text{if } j = \mathcal{A}(n) \\ \hat{\theta}_{i,j}(n-1), & \text{else} \end{cases} \quad (8)$$

$$m_j(n) = \begin{cases} m_j(n-1) + 1, & \text{if } j = \mathcal{A}(n) \\ m_j(n-1); & \text{else} \end{cases} \quad (9)$$

- 12: **end while**
-

TABLE I: Notations for Algorithm Analysis

K	:	number of arms
M	:	number of distinct contexts (for discrete contexts)
\mathcal{Y}	:	set of all contexts
\mathcal{X}	:	support set of the arm-values
*	:	index to indicate the contextually optimal arm
p_i	:	probability of the context being $\mathbf{y}_{(i)}$ (for discrete contexts)
\mathcal{Y}_j	:	set of all contexts for which j -th arm is optimal
q_j	:	sum probability of all contexts in \mathcal{Y}_j
\mathcal{S}	:	index set of all arms
\mathcal{O}	:	index set of all optimal arms
$\overline{\mathcal{O}}$:	index set of all non-optimal arms
$T_j^{\mathcal{O}}(n)$:	number of <i>optimal</i> pulls of j -th arm in n trials
$T_j^{\mathcal{N}}(n)$:	number of <i>non-optimal</i> pulls of j -th arm in n trials
$T_j(n)$:	number of pulls of j -th arm in n trials (Note: $T_j(n) = T_j^{\mathcal{O}}(n) + T_j^{\mathcal{N}}(n)$)
$\mathcal{A}(n)$:	index of the arm pulled in n -th trial by DCB(ϵ)

Algorithmically, DCB(ϵ) is similar in spirit to UCB1. It differs from UCB1 in updating multiple reward averages after a single pull. Also, the reward-ranges G_i for various contexts are in general different and better regret constants are obtained by scaling the confidence bounds accordingly. The major difference, however, comes from the parameter ϵ which needs to be strictly positive. This condition is essential for bounding the number of pulls of the arms that are optimal for some context. The analysis for these arms is sophisticated and differs significantly from the standard analysis of UCB1, and is therefore one of the main contributions of our paper.

DCB(ϵ) uses an $M \times K$ matrix $(\hat{\theta}_{i,j})_{M \times K}$ to store the reward information obtained from previous pulls. $\hat{\theta}_{i,j}$ is the sample mean of all rewards corresponding to context $\mathbf{y}_{(i)}$ and the observed x_j values. In addition, it uses a length K vector $(m_j)_{1 \times K}$ to store the number of pulls of j -th arm up to the current trial. At the n -th trial, $\mathcal{A}(n)$ -th arm is pulled and $x_{\mathcal{A}(n),n}$ is revealed or inferred from the reward. Based on this, $(\hat{\theta}_{i,j})_{M \times K}$ and $(m_i)_{1 \times K}$ are updated. It should be noted that the time indexes in (8) and (9) are only for notational clarity. It is not necessary to store the previous matrices while running the algorithm. Storage required by the DCB(ϵ) policy is, therefore, only $\Theta(MK)$ and does not grow with time. In section V, we analyze and upper bound the cumulative regret of our policy and show that it scales logarithmically in time and linearly in the number of arms not optimal for any context.

V. REGRET ANALYSIS OF DCB(ϵ)

In the standard MAB problems, regret arises when the user pulls the non-optimal arms. Hence the regret upper bounds can be derived by analyzing the expected number of pulls of each non-optimal arm. In our problem, however, there can be multiple contextually optimal arms. It is, therefore, important to analyze not only the number of times an arm is pulled but also the contexts it is pulled for.

Let $\mathcal{S} = \{1, 2, \dots, K\}$ denote the index set of all arms. Let us define an optimal set \mathcal{O} and a non-optimal set $\overline{\mathcal{O}}$ as

$$\begin{aligned}\mathcal{O} &= \{j \mid \exists \mathbf{y}_{(i)} \in \mathcal{Y} : h^*(\mathbf{y}_{(i)}) = j\}, \\ \overline{\mathcal{O}} &= \mathcal{S} \setminus \mathcal{O}.\end{aligned}\tag{10}$$

Note that the arms in the non-optimal set $\overline{\mathcal{O}}$ are not optimal for any context. This means that every time an arm in $\overline{\mathcal{O}}$ is pulled, it contributes to the expected regret. However, an arm in \mathcal{O} contributes to the expected regret only when it is pulled for contexts with different optimal arms. When these optimal arms are pulled for the contexts for which they are optimal, they don't contribute to the expected regret. We analyze these two cases separately and provide upper bounds on the number of pulls of each arm for our policy.

Theorem 1 (Bound for non-optimal arms). *For all $j \in \overline{\mathcal{O}}$, under the DCB(ϵ) policy*

$$\mathbb{E}[T_j(n)] \leq \frac{4(2 + \epsilon) \ln n}{\min_{1 \leq i \leq M} (\Delta_j^{(i)})^2} + 1 + \frac{\pi^2}{3},\tag{11}$$

where $\Delta_j^{(i)} = \theta_i^* - \theta_{i,j}$.

Theorem 2 (Bound for optimal arms). *For all $j \in \mathcal{O}$, under the DCB(ϵ) policy*

$$\mathbb{E}[T_j^N(n)] \leq n_o + \left(\sum_{\mathbf{y}_{(i)} \in \mathcal{Y}_j} \frac{2}{p_i^2} \right) + 2^{3+2\epsilon} K^{4+2\epsilon} c_\epsilon \left(\sum_{\mathbf{y}_{(i)} \in \mathcal{Y}_i} \frac{1}{(p_i)^{2+2\epsilon}} \right) + \frac{\pi^2}{3} \left(\sum_{\mathbf{y}_{(i)} \notin \mathcal{Y}_j} p_i \right),\tag{12}$$

where $c_\epsilon = \sum_{n=1}^{\infty} \frac{1}{n^{1+2\epsilon}}$ and n_o is the minimum value of n satisfying

$$\left\lfloor \frac{p_o n}{2K} \right\rfloor > \left\lceil \frac{4(2 + \epsilon) \ln n}{(\Delta_o)^2} \right\rceil,\tag{13}$$

with $p_o = \min_{1 \leq i \leq M} p_i$, and $\Delta_o = \min_{\forall i, j: h^*(i) \neq j} \Delta_j^{(i)}$.

Theorem 1, whose proof can be found in appendix B, provides an upper bound on the number of pulls of any non-optimal arm that scales logarithmically in time. As discussed previously, all such pulls contribute to the regret. Theorem 2 states that the number of non-optimal pulls of any optimal arm is bounded above by a constant. Regret contribution of the optimal arms is, therefore, bounded. We sketch the proof of theorem 2 to emphasize the basic ideas of our novel proof technique and provide the detailed proof in appendix D.

In order to bound the number of pulls of the optimal arms in DCB(ϵ), we require a high probability bound on the optimal pulls by UCB1 [3] in the standard MAB problem. Since we are not aware of any such existing result, we derive one ourselves. Lemma 1 shows a bound for the generalized UCB1 policy called UCB1(ϵ), proof of which can be found in appendix C. Note that the confidence interval in UCB1(ϵ) is $\sqrt{\frac{(2+\epsilon) \ln n}{m_k}}$, which reduces to that of UCB1 for $\epsilon = 0$. In this context, μ_j denotes the mean value of j -th machine and $\mu^* = \max_{1 \leq j \leq K} \mu_j$.

Lemma 1 (High probability bound for UCB1(ϵ)). *In the standard stochastic MAB problem, if the UCB1(ϵ) policy is run on K machines having arbitrary reward distributions with support in $[0, 1]$, then*

$$\Pr \left\{ T^*(n) < \frac{n}{K} \right\} < \frac{2K^{4+2\epsilon}}{n^{2+2\epsilon}}, \quad (14)$$

for all n such that

$$\left\lfloor \frac{n}{K} \right\rfloor > \frac{4(2 + \epsilon) \ln n}{\min_{j \neq j^*} (\mu^* - \mu_j)^2}. \quad (15)$$

Note that $\frac{n}{\ln n}$ is an increasing sequence in n and there exists some $n' \in \mathbb{N}$ such that condition (15) is satisfied for every $n \geq n'$. n' is a constant whose actual value depends on the true mean values for the arms.

Proof sketch of theorem 2: For every optimal arm, there exists a non-zero probability of appearance of the contexts for which it is optimal. Over successive pulls, DCB(ϵ) leads us to pull the optimal arms with very high probability for their corresponding contexts. Thus, these arms are pulled at least a constant fraction of time with high probability. Since the constant fraction of time is much more than the logarithmic exploration requirement, these optimal arms need not be explored during other contexts. The idea is to show that the expected number of these non-optimal pulls is bounded.

Let us define K mutually exclusive and collectively exhaustive context-sets \mathcal{Y}_j for $j \in \mathcal{S}$ as

$$\mathcal{Y}_j = \{\mathbf{y}_{(i)} \in \mathcal{Y} \mid h^*(i) = j\}.$$

Further, let $N_i(n)$ denote the number of occurrences of context $\mathbf{y}_{(i)}$ till the n -th trial. At the n -th trial, if an optimal arm j is pulled for a context $\mathbf{y}_n = \mathbf{y}_{(i)} \notin \mathcal{Y}_j$, this happens due to one of the following reasons:

- 1) The number of pulls of j -th arm till the n -th trial is small and we err due to lack of sufficient observations of j -th arm's values, which means the arm is *under-played*.
- 2) We err in spite of having previously pulled j -th arm enough number of times.

We bound the probability of occurrence for these two cases by functions of n , whose infinite series over n is convergent. These reasons can be analyzed in three mutually exclusive cases as follows

a) Under-realized contexts: In this case, the number of occurrences of contexts in \mathcal{Y}_j till the n -th trial is small. If all the contexts in \mathcal{Y}_j are under-realized, this could lead to j -th arm being under-explored at trial n . We use Hoeffding's inequality (see appendix A) to bound the expected number of occurrences of this event.

b) Under-exploited arm: We assume that no context $\mathbf{y}_{(i)} \in \mathcal{Y}_j$ is under-realized and yet j -th arm is not pulled in enough trials. For these contexts j -th arm is optimal, but we still don't pull it often enough. In order to bound the expected number of occurrences of this event, we use the high probability bound from lemma 1.

c) Dominant confidence bounds: This case corresponds to the event where no context in \mathcal{Y}_j is under-realized and j -th arm is also pulled sufficiently often, yet it is pulled non-optimally. This occurs when the upper confidence bound for some other arm dominates the corresponding DCB(ϵ) index.

We prove the theorem by upper bounding the expected number of occurrences of these events over all trials. ■

A. Upper Bound on Regret

Upper bounds on the number of pulls of all arms in theorems 1 and 2 lead us to the regret bound in theorem 3. It states a regret upper bound that grows linearly in the number of non-optimal arms and logarithmically in time, i.e. $O(|\overline{\mathcal{O}}| \ln n)$.

Theorem 3 (Regret bound for DCB(ϵ)). *The expected regret under the DCB(ϵ) policy till trial n is at most*

$$\frac{4(2 + \epsilon)\Delta_{\max}}{(\Delta_o)^2} |\overline{\mathcal{O}}| \ln n + O(1), \quad (16)$$

where $\Delta_{\max} = \max_{\forall i,j} \Delta_j^{(i)}$.

Proof of theorem 3: Since the expected number of pulls of optimal arms is bounded (from theorem 2), their regret contribution is also bounded. Note that this regret bound is still a function of K , but a constant in terms of n . Using theorem 1 and the definition of Δ_o , total number of pulls of any non-optimal arm till n -th trial is upper bounded by $\frac{4(2+\epsilon)\Delta_{\max} \ln n}{(\Delta_o)^2} + O(1)$. As the contribution to expected regret of any non-optimal pull is upper bounded by Δ_{\max} , considering all non-optimal arms proves the theorem. ■

Remark 1. *Note that in order for theorem 2 to hold, DCB(ϵ) must be run with a strictly positive ϵ . For $\epsilon = 0$, the confidence bound looks similar to the one from UCB1 [3]. It remains to be seen if the condition $\epsilon > 0$ is an artifact of our proof technique or a stringent requirement for constant regret bound in theorem 2.*

Remark 2. *If the reward function is not dependent on the context, this formulation reduces to the standard MAB problem. In this special case, our DCB(ϵ) algorithm reduces to UCB1. Thus, our results generalize the results from [3].*

Remark 3. *Strategies like deterministic sequencing of exploration and exploitation (DSEE) [4], ϵ -greedy [3] and epoch greedy [7] separate exploration from exploitation. All the arms are explored either deterministically equally in DSEE or uniformly at random in others. Using these strategies for our problem leads to logarithmically scaling regret contributions for all the arms. Since the optimal arms are not known a priori, these strategies do not reduce the regret contribution from optimal arms.*

B. Asymptotic Lower bound

For the standard MAB problem, Lai and Robbins provided an asymptotic lower bound on the regret in [1]. For every non-optimal arm j and the families of arm-distributions parametrized by a single real number, they showed:

$$\liminf_{n \rightarrow \infty} \frac{\mathbb{E}\{T_j(n)\}}{\ln n} \geq D_j, \quad (17)$$

where $D_j > 0$ is a function of the Kullback-Leibler divergence between the reward distribution of j -th arm and that of the optimal arm. This result was extended by Burnetas and Katehakis to distributions indexed by multiple parameters in [18]. It is important to note that the lower bound holds only for consistent policies. A policy π is

said to be consistent if $\mathfrak{R}_\pi(n) = o(n^\alpha)$ for all $\alpha > 0$ as $n \rightarrow \infty$. Any consistent policy, therefore, pulls every non-optimal arm at least $\Omega(\ln n)$ times asymptotically.

The contextual bandit problem can be visualized as an interleaved version of several MAB problems, each corresponding to a distinct context. These MAB instances are, however, not independent. When the reward function is known, information gained from an arm pull for any context can be used to learn about the rewards for other contexts as well. In terms of learning, there is no distinction among the pulls of a specific arm for distinct contexts. The burden of exploration, thus, gets shared by all the contexts. Let $N_i(n)$ denote the number of occurrences of $\mathbf{y}_{(i)}$ till the n -th trial. Under the conditions from [1], for a non-optimal arm j under a consistent policy we get

$$\liminf_{n \rightarrow \infty} \frac{\mathbb{E}\{T_j(n)\}}{\ln N_i(n)} \geq D_{i,j}, \quad (18)$$

where $D_{i,j} > 0$ depends on the KL divergence between the conditional reward distribution of j -th arm and that of the contextually optimal arm. By the law of large numbers, $N_i(n) \rightarrow p_i n$ as $n \rightarrow \infty$. Thus, we write

$$\lim_{n \rightarrow \infty} \frac{\ln N_i(n)}{\ln n} = 1 + \lim_{n \rightarrow \infty} \frac{\ln(N_i(n)/n)}{\ln n} = 1.$$

Condition (18), therefore, reduces to

$$\liminf_{n \rightarrow \infty} \frac{\mathbb{E}\{T_j(n)\}}{\ln n} \geq D_{i,j}.$$

Note that there is one such condition corresponding to each context and non-optimal arm pair. Combining these conditions for a non-optimal arm j , we get

$$\liminf_{n \rightarrow \infty} \frac{\mathbb{E}\{T_j(n)\}}{\ln n} \geq \max_{1 \leq i \leq M} D_{i,j} > 0. \quad (19)$$

This gives an asymptotic lower bound of $\Omega(\ln n)$ on the number of pulls of a non-optimal arm under a consistent policy. Note that this bound matches the upper bound of $O(\ln n)$ for DCB(ϵ) policy proving its order optimality.

VI. CONTINUOUS CONTEXTS

In this section, we extend our algorithm for discrete contexts to the continuous case. For simplicity, we assume that $\mathcal{Y} \subseteq \mathbb{R}$. This can be easily extended to general metric spaces similar to [8], [15]. We additionally assume the reward function to be L -Lipschitz in context, which means

$$|g(y, x) - g(y', x)| \leq L |y - y'|, \quad \forall y, y' \in \mathcal{Y}, x \in \mathcal{X}. \quad (20)$$

Our main idea is to partition the support of context space into intervals of size δ . We learn and store the statistics from the discrete-context algorithms for the center points of each interval. We quantize each incoming context value y_t into \hat{y}_t and use DCB(ϵ) as a subroutine to perform successive pulls. Since the context-distribution is not known, we resort to uniform quantization. Algorithm 2 presents our continuous contextual bandit algorithm based on these ideas. Theorem 4 states its regret upper bound in terms of the time horizon T .

Theorem 4 (Regret bound for CCB(ϵ, δ)). *The expected regret under CCB(ϵ, δ) policy is upper bounded by $\delta LT + O(\log T)$, where T is the time horizon.*

Algorithm 2 CCB(ϵ, δ)

- 1: **Parameters:** $\epsilon, \delta > 0$.
 - 2: Partition \mathcal{Y} into intervals of size δ .
 - 3: Set up a DCB(ϵ) instance with the set $\hat{\mathcal{Y}}$ containing the center points of each partition as the context-set.
 - 4: **for** $t = 1$ to T **do**
 - 5: Quantize y_t into \hat{y}_t ;
 - 6: Feed \hat{y}_t as the context to DCB(ϵ) and pull the arm it recommends;
 - 7: Update the internal statistics of DCB(ϵ);
 - 8: **end for**
-

Proof: Let the instance of DCB(ϵ) algorithm working with the quantized contexts \hat{y}_t be \mathcal{A} and its expected regret be $\mathfrak{R}_{\mathcal{A}}(T)$. Similarly, let \mathcal{B} denote the instance of CCB(ϵ, δ) algorithm and $\mathfrak{R}_{\mathcal{B}}(T)$ its expected regret. The Bayes optimal policies on \mathcal{Y} and $\hat{\mathcal{Y}}$ are denoted by h^* and \hat{h}^* , respectively. It is important to note that $\hat{h}^*(\hat{y}) = h^*(\hat{y})$ for all $\hat{y} \in \hat{\mathcal{Y}}$ and $\mathcal{A}(t) = \mathcal{B}(t)$ since \mathcal{B} follows the recommendations of \mathcal{A} . The main difference is the input context for \mathcal{A} and \mathcal{B} , which changes the optimal rewards. The regret of \mathcal{A} , in terms of these notations, is

$$\mathfrak{R}_{\mathcal{A}}(T) = \sum_{t=1}^T \mathbb{E} \left[g(\hat{y}_t, x_{\hat{h}^*(t),t}) - g(\hat{y}_t, x_{\mathcal{A}(t),t}) \right]. \quad (21)$$

Thus, the regret of \mathcal{B} is described as

$$\begin{aligned} \mathfrak{R}_{\mathcal{B}}(T) &= \sum_{t=1}^T \mathbb{E} \left[g(y_t, x_{h^*(t),t}) - g(y_t, x_{\mathcal{B}(t),t}) \right] \\ &= \sum_{t=1}^T \mathbb{E} \left[g(y_t, x_{h^*(t),t}) - g(\hat{y}_t, x_{h^*(t),t}) + g(\hat{y}_t, x_{h^*(t),t}) - g(\hat{y}_t, x_{\mathcal{B}(t),t}) + g(\hat{y}_t, x_{\mathcal{B}(t),t}) - g(y_t, x_{\mathcal{B}(t),t}) \right] \\ &\leq \sum_{t=1}^T \mathbb{E} \left[L \frac{\delta}{2} \right] + \sum_{t=1}^T \mathbb{E} \left[g(\hat{y}_t, x_{h^*(t),t}) - g(\hat{y}_t, x_{\mathcal{B}(t),t}) \right] + \sum_{t=1}^T \mathbb{E} \left[L \frac{\delta}{2} \right] \quad (\text{L-Lipschitz condition}) \\ &\leq \delta LT + \sum_{t=1}^T \mathbb{E} \left[g(\hat{y}_t, x_{\hat{h}^*(t),t}) - g(\hat{y}_t, x_{\mathcal{A}(t),t}) \right] \\ &\leq \delta LT + \mathfrak{R}_{\mathcal{A}}(T). \end{aligned} \quad (22)$$

From theorem 3, we know that $\mathfrak{R}_{\mathcal{A}}(T)$ is at most $O(\log T)$. Substituting this in (22) yields the result. \blacksquare

Note that the choice of δ is dependent on the time horizon T . The regret upper bound is, therefore, not linear in T as it might appear from theorem 4. In the following subsections, we discuss how δ can be tuned to T in order to obtain the desired storage and regret guarantees. Hereafter, we will use δ_T to denote the tuned parameter δ .

A. Known Time Horizon

The CCB(ϵ, δ_T) regret bound of $O(\delta_T LT)$ is largely controlled by the parameter δ_T . Apart from the regret, another important concern for online learning algorithms is the required storage. Since there are $O(\frac{1}{\delta_T})$ context

intervals, the storage required is $O(\frac{K}{\delta_T})$. This manifests a peculiar storage vs regret trade-off. As δ_T increases, its storage decreases, while the regret increases linearly. This trade-off arises due to the structure of our contextual bandit setting that knows the context-reward mapping function. If the time horizon is fixed and known in advance, δ_T can be tuned based on the performance requirements.

Note that the regret bounds of $\text{CCB}(\epsilon, \delta)$ are only possible due to joint learning for all context intervals. Multi-UCB can also be extended to handle continuous contexts by using this quantization technique. This essentially generalizes the Query-ad-clustering algorithm from [8] with tunable δ . A similar algorithm employing adaptive quantization intervals appears in [15]. Both of these algorithms cannot exploit the reward function knowledge and they thus have to perform independent exploration of arms for different context intervals. As δ decreases, the number of context intervals increases which also increases the number of trials spent in exploring different arms. These algorithms achieve minimum regret bounds of $O(T^{2/3+\epsilon})$ for any $\epsilon > 0$ and $O(T^{2/3} \log T)$ respectively. Furthermore a lower bound of $\Omega(T^{2/3})$ on all continuous contextual bandit algorithms is proved in [8]. Since we assume the reward function knowledge, the agent can obtain sub-linear regret bounds as small as $\Theta(\log T)$ by appropriately choosing δ_T in $\text{CCB}(\epsilon, \delta)$. This is a substantial improvement over the existing contextual bandit algorithms. Exploiting the knowledge of the reward function, therefore, helps in drastically reducing the regret for the continuous contextual bandits.

B. Unknown Time Horizon

Even when the time horizon is unknown, similar performance can be obtained using the so-called doubling trick from [8], [15]. It converts a bandit algorithm with a known time horizon into one that runs without that knowledge with essentially same performance guarantees. Let \mathcal{B}_T denote the $\text{CCB}(\epsilon, \delta_T)$ instance for a fixed horizon T with δ_T tuned for regret bounds of $O(LT^\alpha)$ for $\alpha \in [0, 1]$. The new algorithm \mathcal{C} runs in phases $m = 1, 2, \dots$ of 2^m trials each, such that each phase m runs a new instance of \mathcal{B}_{2^m} . Following recurrence relationship relates the regret of \mathcal{C} to that of \mathcal{B}_T

$$\mathfrak{R}_{\mathcal{C}}(T) = \mathfrak{R}_{\mathcal{C}}(T/2) + \mathfrak{R}_{\mathcal{B}_{T/2}}(T/2). \quad (23)$$

Hence we get a regret bound of $O(LT^\alpha \log T)$, while using the storage of same order $O(KT^{1-\alpha})$. Note that during the execution of \mathcal{C} , previously learned values are discarded in each phase and a fresh instance of \mathcal{B} is run. Using those stored values for next phases may help in decreasing the regret further.

The system designer can analyze the regret-storage trade-off and tune δ_T based on the hardware specifications and performance requirements. This empowers the designer with more control over the algorithmic implementation than any of the existing contextual bandit algorithms.

VII. NUMERICAL SIMULATION RESULTS

In this section, we present the results of numerical simulations for the channel selection problem from I-A1. Consider the case of $M = 4$ contexts uniformly distributed over the set $\mathcal{Y} = \{1, 2, 3, 4\}$. Let there be $K = 7$ arms whose arm-values are scaled Bernoulli random variables. For every $1 \leq j \leq 7$, the arm values are distributed as

$\Pr\{X_j = j\} = \frac{8-j}{10}$, $\Pr\{X_j = 0\} = \frac{2+j}{10}$. Since the genie knows these distributions, it figures out the optimal arms for all the contexts based on the expected reward matrix $(\theta_{i,j})$ given by

$$(\theta_{i,j}) = \begin{pmatrix} \boxed{0.7} & 0.6 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 \\ 0.7 & \boxed{1.2} & 1.0 & 0.8 & 0.6 & 0.4 & 0.2 \\ 0.7 & 1.2 & \boxed{1.5} & 1.2 & 0.9 & 0.6 & 0.3 \\ 0.7 & 1.2 & 1.5 & \boxed{1.6} & 1.2 & 0.8 & 0.4 \end{pmatrix},$$

where the components in the box are the optimal arms. Hence, $h^*(i) = i$ for all i . Figure 4a plots the cumulative regret over the logarithm of trial index. We observe that $\frac{\mathfrak{R}(n)}{\log n}$ converges to a constant for the DCB(10^{-2}) policy, whereas for the UCB1 policy it continues to increase rapidly, since the regret for UCB1 policy grows linearly for contextual bandits. It must be noted that the logarithmic regret of DCB(ϵ) is due to the presence of 3 non-optimal arms. Multi-UCB regret is also logarithmic in time albeit with higher constants due to independent exploration of arms for each context.

If we reduce the number of arms to 4, by removing the non-optimal arms 5, 6 and 7, then the expected reward matrix for the channel selection problem shrinks to

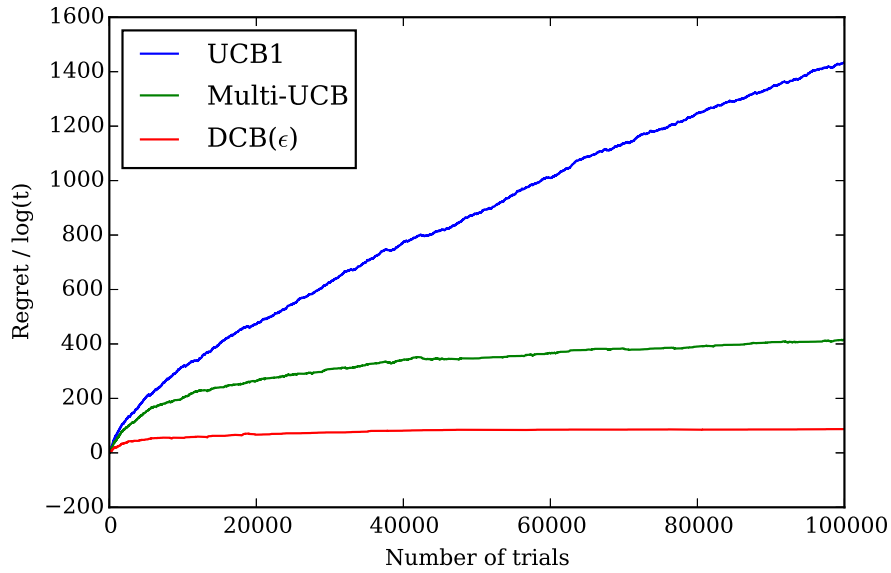
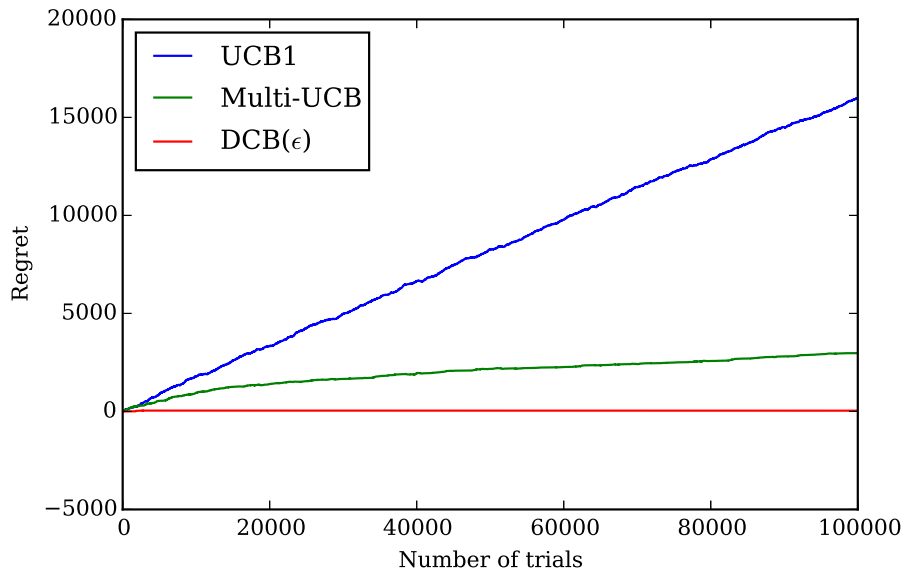
$$(\theta_{i,j}) = \begin{pmatrix} \boxed{0.7} & 0.6 & 0.5 & 0.4 \\ 0.7 & \boxed{1.2} & 1.0 & 0.8 \\ 0.7 & 1.2 & \boxed{1.5} & 1.2 \\ 0.7 & 1.2 & 1.5 & \boxed{1.6} \end{pmatrix}.$$

Regret performance for this case is plotted in figure 4b, which shows that the regret growth stops after some trials for DCB(10^{-2}). It must be noted that we plot the net regret in this case and not the regret divided by $\log n$. Bounded regret is expected, since all arms are optimal and the regret due to non-optimal arms is logarithmic in time, which do not exist as $|\overline{\mathcal{O}}| = 0$. Since Multi-UCB is unable to exploit the reward-function knowledge, its regret still grows logarithmically in time. Such a case demonstrates the significance of our policy, as it reduced the regret drastically by jointly learning the arms-rewards for all the contexts. Table II compares the regret for UCB1, Multi-UCB and DCB(10^{-2}) when $T = 10^5$ for both channel selection examples. We see that the regret reduction using DCB(ϵ) is substantial, especially when the non-optimal arm set is empty.

TABLE II: Regret for the channel selection problem when $T = 10^5$

	UCB1	Multi-UCB	DCB(10^{-2})
$M = 4, K = 7, \overline{\mathcal{O}} = 3$	17262	4893	1294
$M = 4, K = 4, \overline{\mathcal{O}} = 0$	15688	3278	28

We compare the performance of continuous contextual bandit algorithms for the power-aware channel selection problem in energy harvesting communications from section I-A2. We use the same $K = 4$ arms from the previous example. The context is assumed to be uniformly distributed in $(0, 1)$. Note that arms 3 and 4 are the optimal

(a) $K = 7$ and $|\bar{\mathcal{O}}| = 3$.(b) $K = 4$ and $|\bar{\mathcal{O}}| = 0$.Fig. 4: Simulation results for the channel selection problem with $\epsilon = 10^{-2}$.

arms in this case. We use $\delta = \frac{1}{\sqrt{T}}$ with $T = 10^6$ trials for $\text{CCB}(\epsilon, \delta)$ with known time horizon. According to the theoretical results from section VI, $\text{CCB}(\epsilon, \delta)$ yields regret bounds of $O(\sqrt{T})$ and $O(\sqrt{T} \log T)$ in this setting with and without the knowledge of the horizon respectively. We also run a Multi-UCB instance with the same

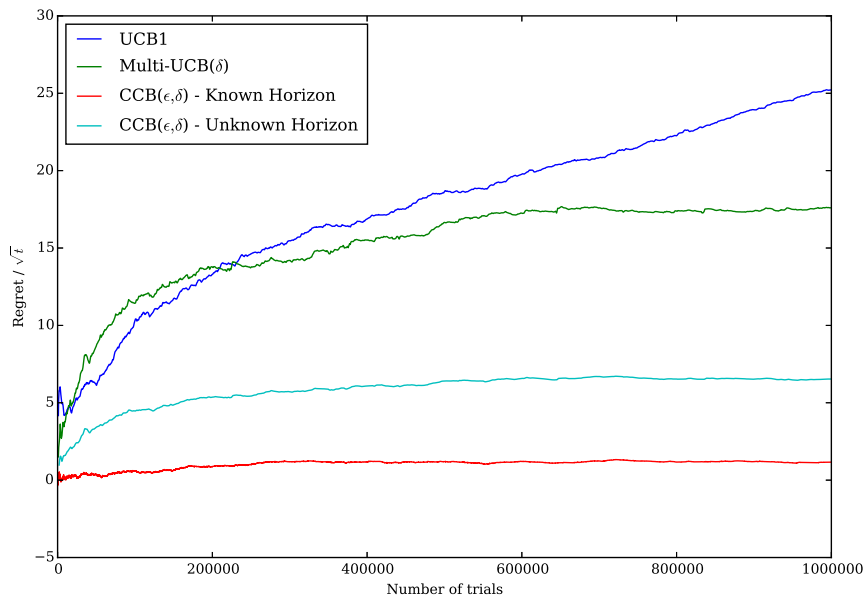


Fig. 5: Simulation results for the energy harvesting communications problem with $K = 4$, $\epsilon = 10^{-2}$ and $\delta = \frac{1}{\sqrt{T}}$ for tuned algorithms.

TABLE III: Regret for the power-aware channel selection problem when $T = 10^6$

	$\delta = T^{-1/3}$	$\delta = T^{-1/2}$	$\delta = T^{-2/3}$
Multi-UCB(δ)	15535.8	17583.9	23117.2
CCB(ϵ, δ) - Unknown T	8645.7	6533.0	1476.2
CCB(ϵ, δ) - Known T	3010.5	1163.4	481.8
UCB1	25201.5		

quantization intervals. Figure 5 plots $\frac{\mathfrak{R}(n)}{\sqrt{n}}$ for UCB1, Multi-UCB with quantized contexts, and CCB(ϵ, δ) with known and unknown horizon. The curve corresponding to CCB(ϵ, δ) with known horizon converges to a constant, while that with unknown horizon continues to grow slowly. Both of these algorithms outperform Multi-UCB which spends a large number of trials exploring the arms independently for each quantized context. Table III shows the regret at the horizon for the algorithms tuned with different values of δ . We notice that the regret of CCB(ϵ, δ) decreases as δ decreases. Even when the time horizon is unknown, CCB(ϵ, δ) outperforms Multi-UCB and UCB1. The numerical regret results for Multi-UCB concur with the theoretical result from [8] that the quantization intervals of size $\Theta(T^{-1/3})$ yield minimum regret. Thus we observe that reducing δ does not necessarily reduce the regret of Multi-UCB as it does for CCB(ϵ, δ).

VIII. CONCLUSION

Multi-armed bandits have been previously used to model many networking applications such as channel selection and routing. In many sequential decision making problems in networks such as channel selection, power allocation and server selection, however, agent knows some side-information such as number of packets to be transmitted, transmit power available, features about the job to scheduled. Motivated by these applications, we have considered stochastic contextual bandit problems in this paper. In our formulation, the agent also knows the reward functions, i.e. the relationship between the context and the reward.

For the case of discrete and finite context spaces, we have proposed a UCB-like algorithm called DCB(ϵ). It exploits the knowledge of reward functions for updating reward information for all contexts. We proposed a novel proof technique to bound the number of non-optimal pulls of the optimal arms by constant. This helped us obtain a regret bound that grows logarithmically in time and linearly in the number of non-optimal arms. This regret is shown to be order optimal by a natural extension of the lower bound result for standard multi-armed bandits. This regret performance is an improvement over bandit algorithms unaware of reward functions where regret grows linearly with the number of arms. For the proposed DCB(ϵ) policy, the non-optimal pulls of the optimal arms are shown to be bounded for $\epsilon > 0$. It remains to be seen if such a guarantee can be provided for $\epsilon = 0$. While proving the regret results for DCB(ϵ, δ), we also proved a high probability bound on the number of optimal pulls by UCB1 in the standard MAB setting. This result could independently have potential applications in other bandit problems.

Further contributions involve extending DCB(ϵ) to continuous context spaces. We proposed an algorithm called CCB(ϵ, δ) for Lipschitz reward functions that uses DCB(ϵ) as a subroutine. Regret analysis of CCB(ϵ, δ) uncovered an interesting regret vs storage trade-off parameterized by δ where the regret can be reduced by using larger storage. System designers can obtain sub-linear regrets by tuning δ based on the time horizon and storage requirements. Even when the time horizon is unknown, similar performance is guaranteed by the proposed epoch-based implementation of CCB(ϵ, δ). The joint learning of arm-rewards in CCB(ϵ, δ) yields regret bounds that are unachievable by any of the existing contextual bandit algorithms for continuous context spaces.

In the current setting, we assumed no queuing of data packets or the harvested energy. When there exist data queues or batteries at the transmitter, the agent can decide to send some of those data-bits or use some of that energy in the current slot and potentially store the rest for later slots. Such a setting with this additional layer of decision making is a non-trivial extension that warrants further investigation.

REFERENCES

- [1] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.
- [2] R. Agrawal, "Sample mean based index policies with $o(\log n)$ regret for the multi-armed bandit problem," *Advances in Applied Probability*, pp. 1054–1078, 1995.
- [3] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.
- [4] S. Vakili, K. Liu, and Q. Zhao, "Deterministic sequencing of exploration and exploitation for multi-armed bandit problems," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7, no. 5, pp. 759–767, 2013.

- [5] Y. Gai, B. Krishnamachari, and R. Jain, “Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations,” *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 5, pp. 1466–1478, 2012.
- [6] Y. Gai and B. Krishnamachari, “Distributed stochastic online learning policies for opportunistic spectrum access,” *Signal Processing, IEEE Transactions on*, vol. 62, no. 23, pp. 6184–6193, 2014.
- [7] J. Langford and T. Zhang, “The epoch-greedy algorithm for multi-armed bandits with side information,” in *Advances in neural information processing systems*, pp. 817–824, 2008.
- [8] T. Lu, D. Pál, and M. Pál, “Contextual multi-armed bandits,” in *International Conference on Artificial Intelligence and Statistics*, pp. 485–492, 2010.
- [9] L. Li, W. Chu, J. Langford, and R. E. Schapire, “A contextual-bandit approach to personalized news article recommendation,” in *Proceedings of the 19th international conference on World wide web*, pp. 661–670, ACM, 2010.
- [10] P. Auer, “Using confidence bounds for exploitation-exploration trade-offs,” *The Journal of Machine Learning Research*, vol. 3, pp. 397–422, 2003.
- [11] W. Chu, L. Li, L. Reyzin, and R. E. Schapire, “Contextual bandits with linear payoff functions,” in *International Conference on Artificial Intelligence and Statistics*, pp. 208–214, 2011.
- [12] S. Agrawal and N. Goyal, “Thompson sampling for contextual bandits with linear payoffs,” in *International Conference on Machine Learning*, 2013.
- [13] M. Dudik, D. Hsu, S. Kale, N. Karampatziakis, J. Langford, L. Reyzin, and T. Zhang, “Efficient optimal learning for contextual bandits,” in *Conference on Uncertainty in Artificial Intelligence*, 2011.
- [14] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. E. Schapire, “Taming the monster: A fast and simple algorithm for contextual bandits,” in *International Conference on Machine Learning*, pp. 1638–1646, 2014.
- [15] A. Slivkins, “Contextual bandits with similarity information,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2533–2568, 2014.
- [16] A. Badanidiyuru, J. Langford, and A. Slivkins, “Resourceful contextual bandits,” in *Conference on Learning Theory*, 2014.
- [17] H. Wu, R. Srikant, X. Liu, and C. Jiang, “Algorithms with logarithmic or sublinear regret for constrained contextual bandits,” in *Advances in Neural Information Processing Systems*, pp. 433–441, 2015.
- [18] A. N. Burnetas and M. N. Katehakis, “Optimal adaptive policies for markov decision processes,” *Mathematics of Operations Research*, vol. 22, no. 1, pp. 222–255, 1997.
- [19] W. Hoeffding, “Probability inequalities for sums of bounded random variables,” *Journal of the American statistical association*, vol. 58, no. 301, pp. 13–30, 1963.

APPENDIX A

SUM OF BOUNDED RANDOM VARIABLES

We use following version of Hoeffding’s inequality from [19].

Lemma 2 (Hoeffding’s Inequality). *Let Y_1, \dots, Y_n be i.i.d. random variables with mean μ and range $[0, 1]$. Let $S_n = \sum_{t=1}^n Y_t$. Then for all $\alpha \geq 0$*

$$\Pr\{S_n \geq n\mu + \alpha\} \leq e^{-2\alpha^2/n}$$

$$\Pr\{S_n \leq n\mu - \alpha\} \leq e^{-2\alpha^2/n}.$$

APPENDIX B

PROOF OF THEOREM 1: BOUND FOR NON-OPTIMAL ARMS

For any $j \in \overline{\mathcal{O}}$, we write

$$\begin{aligned} T_j(n) &= 1 + \sum_{t=K+1}^n \mathbb{I}\{\mathcal{A}(t) = j\} \\ &\leq l + \sum_{t=K+1}^n \mathbb{I}\{\mathcal{A}(t) = j, T_j(t-1) \geq l\}, \end{aligned} \quad (24)$$

where $\mathbb{I}(x)$ is the indicator function defined to be 1 when the predicate x is true and 0 otherwise, and l is an arbitrary positive integer. Let $C_{t,s} = \sqrt{\frac{(2+\epsilon)\ln t}{s}}$. Let $\hat{\theta}_{i,j,s}$ denote $\hat{\theta}_{i,j}$ when j -th arm has been pulled s times. We use a subscript $*$ to refer to optimal arms' statistics. For example we write $\hat{\theta}_{i,s}^*$ and $T_i^*(t-1)$ to respectively denote $\hat{\theta}_{i,j^*,s}$ and $T_{j^*}(t-1)$ for the optimal arm: $j = h^*(\mathbf{y}_{(i)})$. Our idea is to upper bound the probability of the indicator function of the event $\{\mathcal{A}(t) = j, T_j(t-1) \geq l\}$ and bound the number of pulls as

$$\begin{aligned} \mathbb{E}[T_j(n)] &\leq l + \sum_{t=K+1}^n \Pr\{\mathcal{A}(t) = j, T_j(t-1) \geq l\} \\ &= l + \sum_{t=K+1}^n \sum_{i=1}^M \Pr\{\mathcal{A}(t) = j, T_j(t-1) \geq l, \mathbf{y}_t = \mathbf{y}_{(i)}\} \\ &= l + \sum_{i=1}^M p_i \sum_{t=K+1}^n \Pr\{\mathcal{A}(t) = j, T_j(t-1) \geq l \mid \mathbf{y}_t = \mathbf{y}_{(i)}\} \\ &\leq l + \sum_{i=1}^M p_i \sum_{t=K+1}^n \Pr\left\{\hat{\theta}_{i,T_i^*(t-1)}^* + G_i C_{t-1, T_i^*(t-1)} \leq \hat{\theta}_{i,j, T_j(t-1)} + G_i C_{t-1, T_j(t-1)}, T_j(t-1) \geq l\right\} \\ &\leq l + \sum_{i=1}^M p_i \sum_{t=K+1}^n \Pr\left\{\min_{1 \leq s < t} \left(\hat{\theta}_{i,s}^* + G_i C_{t-1,s}\right) \leq \max_{l \leq s_j < t} \left(\hat{\theta}_{i,j,s_j} + G_i C_{t-1,s_j}\right)\right\} \\ &\leq l + \sum_{i=1}^M p_i \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_j=l}^{t-1} \Pr\left\{\hat{\theta}_{i,s}^* + G_i C_{t,s} \leq \hat{\theta}_{i,j,s_j} + G_i C_{t,s_j}\right\}. \end{aligned} \quad (25)$$

Observing that $\hat{\theta}_{i,s}^* + G_i C_{t,s} \leq \hat{\theta}_{i,j,s_j} + G_i C_{t,s_j}$ cannot hold, unless at least one of the following conditions hold

$$\hat{\theta}_{i,s}^* \leq \theta_i^* - G_i C_{t,s}, \quad (26)$$

$$\hat{\theta}_{i,j,s_j} \geq \theta_{i,j} + G_i C_{t,s_j}, \quad (27)$$

$$\theta_i^* \leq \theta_{i,j} + 2G_i C_{t,s_j}. \quad (28)$$

Using Hoeffding's inequality (see appendix A) on (26) and (27), we get

$$\Pr\{\hat{\theta}_{i,s}^* \leq \theta_i^* - G_i C_{t,s}\} \leq e^{-2(2+\epsilon)\ln t} = t^{-2(2+\epsilon)},$$

$$\Pr\{\hat{\theta}_{i,j,s_j} \geq \theta_{i,j} + G_i C_{t,s_j}\} \leq e^{-2(2+\epsilon)\ln t} = t^{-2(2+\epsilon)}.$$

For $s_j \geq l = \left\lceil \frac{4(2+\epsilon) \ln n}{\min_{1 \leq i \leq M} (\Delta_j^{(i)})^2} \right\rceil$, we get

$$\begin{aligned} \theta_{i,j_i^*} - \theta_{i,j} - 2G_i C_{t,s_j} &= \theta_i^* - \theta_{i,j} - 2G_i \sqrt{\frac{(2+\epsilon) \ln t}{s_j}} \\ &\geq \Delta_j^{(i)} - \min_{1 \leq i \leq M} \Delta_j^{(i)} \\ &\geq 0. \end{aligned}$$

Hence, the condition (25) reduces to

$$\begin{aligned} \mathbb{E}[T_j(n)] &\leq \left\lceil \frac{4(2+\epsilon) \ln n}{\min_{1 \leq i \leq M} (\Delta_j^{(i)})^2} \right\rceil + \sum_{i=1}^M p_i \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_j=l}^{t-1} \left(\Pr \left\{ \hat{\theta}_{i,s}^* \leq \theta_i^* - G_i C_{t,s} \right\} + \Pr \left\{ \hat{\theta}_{i,j,s_j} \geq \theta_{i,j} + G_i C_{t,s_j} \right\} \right) \\ &\leq \left\lceil \frac{4(2+\epsilon) \ln n}{\min_{1 \leq i \leq M} (\Delta_j^{(i)})^2} \right\rceil + \sum_{i=1}^M p_i \sum_{t=1}^{\infty} \sum_{s=1}^t \sum_{s_j=l}^t 2t^{-2(2+\epsilon)} \\ &\leq \left\lceil \frac{4(2+\epsilon) \ln n}{\min_{1 \leq i \leq M} (\Delta_j^{(i)})^2} \right\rceil + 2 \sum_{i=1}^M p_i \sum_{t=1}^{\infty} t^{-2(1+\epsilon)} \\ &\leq \left\lceil \frac{4(2+\epsilon) \ln n}{\min_{1 \leq i \leq M} (\Delta_j^{(i)})^2} \right\rceil + \frac{\pi^2}{3} \sum_{i=1}^M p_i \\ &\leq \frac{4(2+\epsilon) \ln n}{\min_{1 \leq i \leq M} (\Delta_j^{(i)})^2} + 1 + \frac{\pi^2}{3}, \end{aligned}$$

which concludes the proof. ■

APPENDIX C

PROOF OF LEMMA 1: HIGH PROBABILITY BOUND FOR UCB1(ϵ)

Let π denote the instance of UCB1(ϵ) in the standard MAB problem and j^* the index of optimal arm. UCB1(ϵ) stores the empirical means of arm-rewards. We use $\hat{X}_{j,s}$ and \hat{X}_s^* to respectively denote the averages of arm rewards for j -th arm and the optimal arm when they have been pulled for s trials.

If the optimal arm gets pulled less than $\frac{n}{K}$ times during the first n trials, then according to the pigeonhole principle there must exist an arm that gets pulled more than $\frac{n}{K}$ times.

$$\begin{aligned} \Pr \left\{ T^*(n) < \frac{n}{K} \right\} &\leq \Pr \left\{ \exists j \in \mathcal{S} \setminus \{j^*\} : T_j(n) > \frac{n}{K} \right\} \\ &\leq \sum_{j \in \mathcal{S} \setminus \{j^*\}} \Pr \left\{ T_j(n) > \frac{n}{K} \right\}. \end{aligned} \tag{29}$$

For the j -th arm, if $T_j(n) \geq \lfloor \frac{n}{K} \rfloor + 1$, then at some $\lfloor \frac{n}{K} \rfloor + 1 \leq t \leq n$ it must have been pulled for the $(\lfloor \frac{n}{K} \rfloor + 1)$ -th time. We track this trial-index t and bound the probabilities for non-optimal arms.

$$\begin{aligned}
\Pr \left\{ T_j(n) > \frac{n}{K} \right\} &\leq \sum_{t=\lfloor \frac{n}{K} \rfloor + 1}^n \Pr \left\{ \boldsymbol{\pi}(t) = j, T_j(t-1) = \lfloor \frac{n}{K} \rfloor \right\} \\
&\leq \sum_{t=\lfloor \frac{n}{K} \rfloor + 1}^n \Pr \left\{ \hat{X}_{t^*(t-1)}^* + C_{t-1, T^*(t-1)} \leq \hat{X}_{j, \lfloor \frac{n}{K} \rfloor} + C_{t-1, \lfloor \frac{n}{K} \rfloor} \right\} \\
&\leq \sum_{t=\lfloor \frac{n}{K} \rfloor + 1}^n \Pr \left\{ \min_{1 \leq s < t} \hat{X}_s^* + C_{t-1, s} \leq \hat{X}_{j, \lfloor \frac{n}{K} \rfloor} + C_{t-1, \lfloor \frac{n}{K} \rfloor} \right\} \\
&\leq \sum_{t=\lfloor \frac{n}{K} \rfloor + 1}^n \sum_{s=1}^{t-1} \Pr \left\{ \hat{X}_s^* + C_{t-1, s} \leq \hat{X}_{j, \lfloor \frac{n}{K} \rfloor} + C_{t-1, \lfloor \frac{n}{K} \rfloor} \right\}. \tag{30}
\end{aligned}$$

Note that $\hat{X}_s^* + C_{t-1, s} \leq \hat{X}_{j, \lfloor \frac{n}{K} \rfloor} + C_{t-1, \lfloor \frac{n}{K} \rfloor}$ implies that at least one of the following must hold

$$\hat{X}_s^* \leq \mu^* - C_{t, s} \tag{31}$$

$$\hat{X}_{j, \lfloor \frac{n}{K} \rfloor} \geq \mu_j - C_{t, \lfloor \frac{n}{K} \rfloor} \tag{32}$$

$$\mu^* < \mu_j + 2C_{t, \lfloor \frac{n}{K} \rfloor} \tag{33}$$

Using Hoeffding's inequality we bound the probability of events (31) and (32) as

$$\begin{aligned}
\Pr \left\{ \hat{X}_s^* \leq \mu^* - C_{t, s} \right\} &\leq e^{-2(2+\epsilon) \ln t} = t^{-2(2+\epsilon)}, \\
\Pr \left\{ \hat{X}_{j, \lfloor \frac{n}{K} \rfloor} \geq \mu_j - C_{t, \lfloor \frac{n}{K} \rfloor} \right\} &\leq e^{-2(2+\epsilon) \ln t} = t^{-2(2+\epsilon)}.
\end{aligned}$$

As stated in (15) for $\lfloor \frac{n}{K} \rfloor > \frac{4(2+\epsilon) \ln n}{\min_{j \neq j^*} (\mu^* - \mu_j)^2}$,

$$\begin{aligned}
\mu^* - \mu_j - 2C_{t, \lfloor \frac{n}{K} \rfloor} &= \mu^* - \mu_j - 2\sqrt{\frac{(2+\epsilon) \ln t}{\lfloor \frac{n}{K} \rfloor}} \\
&> \mu^* - \mu_j - 2\sqrt{\frac{(2+\epsilon) \ln n}{\lfloor \frac{n}{K} \rfloor}} \\
&> \mu^* - \mu_j - \min_{1 \leq j \leq K} (\mu^* - \mu_j) \\
&\geq 0. \tag{34}
\end{aligned}$$

Thus, condition (33) is always false when $\lfloor \frac{n}{K} \rfloor > \frac{4(2+\epsilon) \ln n}{\min_{j \neq j^*} (\mu^* - \mu_j)^2}$.

Using union bound on (30) for every non-optimal arm, we get

$$\begin{aligned}
\Pr \left\{ T_j(n) > \frac{n}{K} \right\} &\leq \sum_{t=\lfloor \frac{n}{K} \rfloor + 1}^n \sum_{s=1}^{t-1} \left(\Pr \left\{ \hat{X}_s^* \leq \mu^* - C_{t, s} \right\} + \Pr \left\{ \hat{X}_{j, \lfloor \frac{n}{K} \rfloor} \geq \mu_j - C_{t, \lfloor \frac{n}{K} \rfloor} \right\} \right) \\
&\leq \sum_{t=\lfloor \frac{n}{K} \rfloor + 1}^n \sum_{s=1}^{t-1} 2t^{-2(2+\epsilon)}
\end{aligned}$$

$$\begin{aligned}
&\leq 2 \sum_{t=\lfloor \frac{n}{K} \rfloor + 1}^n t^{-3-2\epsilon} \\
&< 2 \sum_{t=\lfloor \frac{n}{K} \rfloor + 1}^n \left(\frac{n}{K}\right)^{-3-2\epsilon} \\
&< 2n \left(\frac{n}{K}\right)^{-3-2\epsilon} \\
&= \frac{2K^{3+2\epsilon}}{n^{2+2\epsilon}}.
\end{aligned} \tag{35}$$

Substituting this in (29), for all n satisfying condition (15), we get

$$\begin{aligned}
\Pr \left\{ T^*(n) < \frac{n}{K} \right\} &< \sum_{j \in \mathcal{S} \setminus \{j^*\}} \frac{2K^{3+2\epsilon}}{n^{2+2\epsilon}} \\
&< \frac{2K^{4+2\epsilon}}{n^{2+2\epsilon}}.
\end{aligned}$$

■

APPENDIX D

PROOF OF THEOREM 2: BOUND FOR OPTIMAL ARMS

Fix $j \in \mathcal{O}$. Let $\mathcal{E}_{j,t}$ denote an event where j -th arm is pulled non-optimally at t -th trial. Total number of non-optimal pulls can, therefore, be written as $T_j^N(n) = \sum_{t=1}^n \mathbb{I}\{\mathcal{E}_{j,t}\}$. Let $\mathcal{E}_{j,t}^1$ denote an event of at least one of the contexts $\mathbf{y}_{(i)} \in \mathcal{Y}_j$ not having occurred even half the number of its expected occurrences till the t -th trial. Additionally, let $\mathcal{E}_{j,t}^2$ be the event of j -arm not having been pulled at least $\frac{1}{K}$ fraction of such pulls by the optimal hypothesis till the t -th trial. In terms of these events, we write the following bound on expected number of non-optimal pulls

$$\begin{aligned}
\mathbb{E}[T_j^N(n)] &= \sum_{t=1}^n \Pr\{\mathcal{E}_{j,t}\} \\
&\leq \sum_{t=1}^n \left(\Pr\{\mathcal{E}_{j,t}^1\} + \Pr\{\mathcal{E}_{j,t}^2 \cap \overline{\mathcal{E}_{j,t}^1}\} + \Pr\{\mathcal{E}_{j,t} \cap \overline{\mathcal{E}_{j,t}^1} \cap \overline{\mathcal{E}_{j,t}^2}\} \right).
\end{aligned} \tag{36}$$

a) Under-realized contexts: Let $N_i(n)$ denote the number occurrences of $\mathbf{y}_{(i)}$ till the n -th trial. Thus, $\mathcal{E}_{j,t}^1$ corresponds to the event where $N_i(t) \leq p_i \frac{t}{2}$ for at least one context $\mathbf{y}_{(i)} \in \mathcal{Y}_j$.

In terms of indicators, $N_i(n) = \sum_{t=1}^n \mathbb{I}\{\mathbf{y}_t = \mathbf{y}_{(i)}\}$ and $\mathbb{E}[N_i(n)] = p_i n$. These indicator random variables are i.i.d. and thus, we use Hoeffding's inequality (appendix A) with $\alpha = p_i n/2$ to get

$$\Pr \left\{ N_i(n) \leq \frac{p_i}{2} n \right\} \leq \exp \left\{ -\frac{p_i^2}{2} n \right\}. \tag{37}$$

The exponential bound is important, since it helps us bound the number of occurrences of context under-realization $N(\mathcal{E}_j^1)$ by a constant. Note that $N(\mathcal{E}_j^1) = \sum_{n=1}^{\infty} \mathbb{I}\{\mathcal{E}_{j,n}^1\}$. We obtain a bound on its expectation, which also bounds the first term in equation (36), as follows

$$\mathbb{E} [N(\mathcal{E}_j^1)] = \sum_{n=1}^{\infty} \Pr\{\mathcal{E}_{j,n}^1\}$$

$$\begin{aligned}
&\leq \sum_{n=1}^{\infty} \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_j} \exp \left\{ -\frac{p_i^2}{2} n \right\} && \text{(Union bound)} \\
&\leq \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_j} \frac{\exp \left\{ -\frac{p_i^2}{2} \right\}}{1 - \exp \left\{ -\frac{p_i^2}{2} \right\}} \\
&\leq \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_j} \frac{2}{p_i^2}. && (38)
\end{aligned}$$

b) Under-exploited arm: We assume that no context $\mathbf{y}^{(i)} \in \mathcal{Y}_j$ is under-realized and yet j -th arm is not pulled in enough number of trials. For these contexts j -th arm is optimal and we don't end up pulling it enough nevertheless. We define an event \mathcal{E}_j^2 corresponding to the existence of a context $\mathbf{y}^{(i)} \in \mathcal{Y}_j$ for which the arm is under-exploited. We now bound the probability that for some context $\mathbf{y}^{(i)} \in \mathcal{Y}_j$, $T_{j,i}^O(n) < \frac{N_i(n)}{K}n$, where $T_{j,i}^O(n)$ denotes the number of pulls of j -th arm for context $\mathbf{y}^{(i)}$. Here, we use a high probability bound (see lemma 1) on the optimal arm for UCBI(ϵ) which we prove in appendix C.

We upper bound the second term from equation (36) as follows:

$$\begin{aligned}
\mathbb{E}\{N(\mathcal{E}_j^2 \mid \overline{\mathcal{E}_j^1})\} &= \sum_{n=1}^{\infty} \Pr \left\{ \mathcal{E}_{j,n}^2 \cap \overline{\mathcal{E}_{j,n}^1} \right\} \\
&\stackrel{(a)}{\leq} n_o + \sum_{n=n_o}^{\infty} \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_i} \Pr \left\{ T_{j,i}^O(n) < \frac{N_i(n)}{K}, N_i(n) \geq \frac{p_i n}{2} \right\} \\
&\leq n_o + \sum_{n=n_o}^{\infty} \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_i} \sum_{m=p_i n/2}^n \Pr \left\{ T_{j,i}^O(n) < \frac{N_i(n)}{K}, N_i(n) = m \right\} \\
&= n_o + \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_i} \sum_{n=n_o}^{\infty} \sum_{m=p_i n/2}^n \Pr \left\{ T_{j,i}^O(n) < \frac{N_i(n)}{K} \mid N_i(n) = m \right\} \Pr \{ N_i(n) = m \} \\
&\leq n_o + \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_i} \sum_{n=n_o}^{\infty} \sum_{m=p_i n/2}^n \Pr \left\{ T_{j,i}^O(n) < \frac{m}{K} \right\} \\
&\stackrel{(b)}{\leq} n_o + \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_i} \sum_{n=n_o}^{\infty} \sum_{m=p_i n/2}^n \frac{2K^{4+2\epsilon}}{m^{2+2\epsilon}} && \text{(Lemma 1)} \\
&\leq n_o + \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_i} \sum_{n=n_o}^{\infty} \sum_{m=p_i n/2}^n \frac{2K^{4+2\epsilon}}{(p_i n/2)^{2+2\epsilon}} \\
&\leq n_o + \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_i} \frac{2^{3+2\epsilon} K^{4+2\epsilon}}{(p_i)^{2+2\epsilon}} \sum_{n=n_o}^{\infty} \frac{1}{n^{1+2\epsilon}} \\
&\leq n_o + 2^{3+2\epsilon} K^{4+2\epsilon} c_{\epsilon} \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_i} \frac{1}{(p_i)^{2+2\epsilon}}, && (39)
\end{aligned}$$

where $c_{\epsilon} = \sum_{n=1}^{\infty} \frac{1}{n^{1+2\epsilon}} < \infty$, since we know that the series converges for $\epsilon > 0$. Thus, we need $\epsilon > 0$ in order to provide theoretical performance guarantees. Note that (a) holds as we bound the probabilities for the first n_o terms by 1, and (b) holds since the pulls for one context can only decrease the exploration requirements of other contexts. For (b) to hold, condition of lemma 1 needs to be satisfied, which is $\left[\frac{p_i n}{2K} \right] > \frac{4(2+\epsilon) \ln n}{(\Delta_j^{(i)})^2}$ for all $n \geq n_o$. In order

to make n_o independent of i and j , we choose n_o as the minimum value of n satisfying following inequality:

$$\left\lfloor \frac{p_o n}{2K} \right\rfloor > \left\lceil \frac{4(2+\epsilon) \ln n}{(\Delta_o)^2} \right\rceil, \quad (40)$$

where $p_o = \min_{1 \leq i \leq M} p_i$ and $\Delta_o = \min_{\forall i, j: h^*(i) \neq j} \Delta_j^{(i)}$.

Note that all terms in the upper bound in (39) are constant, showing that the number of occurrences of under-exploited optimal arms is bounded.

c) Dominant confidence bounds: This case corresponds to the event where no context is under-realized and the optimal arms are also pulled sufficiently often, yet an optimal arm is pulled non-optimally. This means that j -th arm is pulled for a context $\mathbf{y}^{(i)} \notin \mathcal{Y}_j$ at trial n , while $N_i(n) \geq \frac{p_i}{2}n$ and $T_{j,i}^O(n) \geq \frac{N_i(n)}{K}n$ for all $\mathbf{y}^{(i)} \in \mathcal{Y}_j$. Let $q_j = \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_j} p_i$. Note that we only need to analyze $n \geq n_o$, since we already upper bound the probabilities of non-optimal pulls by 1 as shown in previous case. Thus, we have

$$\begin{aligned} T_j(n) &\geq \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_j} T_{j,i}^O(n) \\ &> \sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_j} \frac{N_i(n)}{2} \\ &> \frac{nq_j}{2K}. \end{aligned}$$

Expected number of events $N(\mathcal{E}_j \cap \overline{\mathcal{E}_j^1} \cap \overline{\mathcal{E}_j^2})$ gets upper bounded as follows:

$$\begin{aligned} N(\mathcal{E}_j \cap \overline{\mathcal{E}_j^1} \cap \overline{\mathcal{E}_j^2}) &= \sum_{n=n_o}^{\infty} \Pr\{\mathcal{E}_{j,n} \cap \overline{\mathcal{E}_{j,n}^1} \cap \overline{\mathcal{E}_{j,n}^2}\} \\ &= \sum_{n=n_o}^{\infty} \sum_{\mathbf{y}^{(i)} \notin \mathcal{Y}_j} \Pr\left\{\mathcal{A}(n) = j, T_j(n) \geq \frac{nq_j}{2K}, \mathbf{y}_n = \mathbf{y}^{(i)}\right\} \\ &\leq \sum_{n=n_o}^{\infty} \sum_{\mathbf{y}^{(i)} \notin \mathcal{Y}_j} p_i \Pr\left\{\mathcal{A}(n) = j, T_j(n) \geq \frac{nq_j}{2K} \mid \mathbf{y}_n = \mathbf{y}^{(i)}\right\}. \end{aligned} \quad (41)$$

Following the line of argument from the proof of theorem 1 in appendix B, we get

$$\begin{aligned} N(\mathcal{E}_j \cap \overline{\mathcal{E}_j^1} \cap \overline{\mathcal{E}_j^2}) &\leq 2 \sum_{\mathbf{y}^{(i)} \notin \mathcal{Y}_j} p_i \sum_{n=n_o}^{\infty} n^{-2(1+\epsilon)} \\ &\leq \frac{\pi^2}{3} \sum_{\mathbf{y}^{(i)} \notin \mathcal{Y}_j} p_i \\ &= \frac{\pi^2}{3} (1 - q_j). \end{aligned} \quad (42)$$

Note that the arguments of theorem 1 hold, since we only consider trials $n \geq n_o$.

Combining the results form (38), (39) and (42), we get

$$\mathbb{E} [T_j^N(n)] \leq n_o + \left(\sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_j} \frac{2}{p_i^2} \right) + 2^{3+2\epsilon} K^{4+2\epsilon} c_\epsilon \left(\sum_{\mathbf{y}^{(i)} \in \mathcal{Y}_i} \frac{1}{(p_i)^{2+2\epsilon}} \right) + \frac{\pi^2}{3} (1 - q_j),$$

which concludes the proof. ■