Adversarial Combinatorial Bandits with Switching Cost and Arm Selection Constraints

Yin Huang†, Qingsong Liu‡, Jie Xu†
†Department of Electrical and Computer Engineering, University of Miami.
‡Institute for Interdisciplinary Information Sciences, Tsinghua University
Email: yxh954@miami.edu, liu-qs19@mails.tsinghua.edu.cn, jiexu@miami.edu.

IEEE INFOCOM 2024 - IEEE Conference on Computer Communications | 979-8-3503-8350-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/INFOCOM52122.2024.10621364

***Abstract*—The multi-armed bandits (MAB) framework is widely used for sequential decision-making under uncertainty, finding applications in various domains, including computer and communication networks. To address the increasing complexity of real-world systems and their operational requirements, researchers have proposed and studied various extensions to the basic MAB framework. In this paper, we focus on an adversarial MAB problem inspired by real-world systems with combinatorial semi-bandit arms, switching costs, and anytime cumulative arm selection constraints. To tackle this challenging problem, we introduce the Block-structured Follow-the-Regularized-Leader (B-FTRL) algorithm. Our approach employs a hybrid Tsallis-Shannon entropy regularizer in arm selection and incorporates a block structure that divides time into blocks to minimize arm switching costs. The theoretical analysis shows that B-FTRL achieves a reward regret bound of $O(T^{\frac{2a-b+1}{1+a}} + T^{\frac{b}{1+a}})$ and a switching regret bound of $O(T^{\frac{1}{1+a}})$, where a and b are tunable algorithm parameters. By carefully selecting the values of a and b, we are able to limit the total regret to $O(T^{2/3})$ while satisfying the arm selection constraints in expectation. This outperforms the state-of-the-art regret bound of $O(T^{3/4})$ and expected constraint violation bound $o(1)$, which are derived in less challenging stochastic reward environments. Additionally, we provide a high-probability constraint violation bound of $O(\sqrt{T})$. To validate the effectiveness of the proposed B-FTRL algorithm, numerical results are presented to demonstrate its superiority in comparison to other existing methods.**

I. Introduction

Multi-armed bandits (MAB) problems are a fundamental class of sequential decision-making under uncertainty problems, which involve a crucial tradeoff between exploration and exploitation. In the simplest form, a learner is presented with a set of arms, each representing a different action, and in each round, the learner must choose an arm to play based on past observations. The objective is to maximize the cumulative reward obtained from the chosen arms, or equivalently, to minimize the cumulative regret, which measures the difference between the rewards obtained and the rewards that would have been obtained by always choosing the best arm. Depending on how rewards are generated, bandit problems can be classified as either stochastic bandits or adversarial bandits (also known as non-stochastic bandits). In stochastic bandits, rewards are drawn from fixed but unknown probability distributions associated with each arm. In contrast, adversarial bandits make no statistical assumptions about the reward generation process, and as a result, they are considered more challenging to solve. In both settings, the learner must balance exploration, trying out different arms to learn their potential rewards, and exploitation, choosing arms with the highest expected rewards based on current knowledge, in order to achieve good overall performance over time. The study of MAB has important applications in various fields, such as online advertising [1], recommendation systems [2], clinical trials [3] and resource allocation in network systems [4], where decisions must be made sequentially in the face of uncertainty.

To address sequential decision problems in more complex systems, researchers have proposed and studied various extensions of the basic MAB model. One well-explored variant is **combinatorial bandits** [5], where multiple arms can be played simultaneously in each decision round. This extension poses greater challenges for learning due to the significantly expanded decision space. Another important extension considers the inclusion of **switching costs** [6], which arise when changing actions between consecutive rounds incurs non-negligible overhead costs. In practical systems, such as network management or robotic control, these costs can be significant. Consequently, exploration of different actions must now consider the additional switching cost, making the task of maximizing the reward more complex. Recently, MAB problems under **arm selection constraints** [7] have gained significant attention, particularly in the domain of computer networking. These constraints model situations where fairness considerations require that each arm (representing a user or a specific task) should be played sufficiently often. Additionally, these constraints can be utilized to handle budget restrictions arising from the overhead associated with playing different arms. This paper is motivated by real-world systems that simultaneously feature combinatorial arms, switching costs, and arm selection constraints. Here, we present some applications that illustrate the relevance and significance of considering these combined features.

- *Cloud Resource Management*. In cloud computing environments, tasks can be assigned to different virtual machines (arms) with varying costs and processing capabilities. However, switching tasks between virtual machines incurs switching costs and may disrupt ongoing processes. Additionally, cloud providers may want to ensure fair utilization of resources by enforcing arm selection con-

Y. Huang and J. Xu's work is supported in part by NSF under grants 2006630, 2033681, 2029858, 2044991 and 2319780. The first two authors contributed equally to this work.

straints to maintain a minimum assignment ratio for each virtual machine. Adversarial Combinatorial Bandits with Switching Cost and Arm Selection Constraints (ACB-SCSC) can be used to optimize resource allocation in cloud systems while considering the cost of switching tasks and adhering to resource fairness requirements.

- *Distributed Sensing and Monitoring*. In distributed sensing systems, multiple sensors (arms) are deployed to monitor a large area. Switching sensors between monitoring tasks may introduce delays or data loss. Moreover, some sensors may have limited battery or communication resources, requiring the system to impose constraints on their utilization. ACB-SCSC can be applied to optimize sensor task allocation while minimizing switching costs and ensuring that sensors are used fairly and efficiently.
- *Vehicle Routing and Fleet Management*. In logistics and transportation, optimizing vehicle routes involves choosing between different delivery routes (arms) while considering switching costs associated with rerouting vehicles and adhering to constraints on vehicle utilization. ACB-SCSC can be employed to optimize vehicle routing decisions and maximize fleet efficiency.
- *Healthcare Resource Allocation*. In healthcare settings, patients may require different treatments (arms), and switching treatments may have associated costs or adverse effects on patient outcomes. Arm selection constraints may be applied to ensure a fair distribution of treatments among patients. ACB-SCSC can be used to optimize treatment allocation in healthcare to improve patient outcomes while considering switching costs and adhering to resource constraints.

Despite its wide applicability, there has been limited effort to address the challenging problem of combinatorial bandits considering both switching costs and arm selection constraints, especially in the adversarial setting. Such problems present significant challenges, as exploration to learn the quality of different arms leads to increased reward loss due to arm switching, with an unclear impact on constraint violation. In this paper, we propose a novel algorithm to tackle this complex sequential decision-making problem under uncertainty.

Specifically, we address a combinatorial bandits problem with semi-bandit feedback, where the learner observes the reward of each arm in the selected subset. The arm selection constraints are expressed as "anytime cumulative constraints", which impose stringent conditions that must hold cumulatively within each time slot, rather than considering long-term averages. Our proposed algorithm, named B-FTRL (Block-structured Follow-the-Regularized-Leader), draws inspiration from [8], which introduced a Follow-the-Regularized-Leader (FTRL) algorithm with a novel hybrid regularizer for general semi-bandit problems. This FTRL algorithm achieves an impressive $O(\log T)$ regret for stochastic environments and $O(\sqrt{T})$ regret for adversarial environments, where T denotes the number of decision rounds. The B-FTRL algorithm adopts a block structure that divides time into blocks, allowing arm switching only between blocks. This design significantly reduces the switching cost associated with arm selections. We provide theoretical analysis and show that B-FTRL achieves a regret bound of $O(T^{\frac{2a-b+1}{1+a}} + T^{\frac{b}{1+a}})$, where a and b are algorithm parameters. Additionally, we offer a high-probability constraint violation bound of $O(\sqrt{T})$. By carefully tuning the algorithm parameters, specifically setting $a = 1/2$ and $b = 1$, B-FTRL achieves an improved regret bound of $O(T^{2/3})$, outperforming the $O(T^{3/4})$ regret bound presented in [9]. Notably, this improvement is observed even in the more challenging setting where combinatorial arm selection is considered, unlike the setting in [9], which focused on the less complex stochastic environment without considering combinatorial arms.

II. Related Work

Adversarial Bandits Adversarial bandits problems deal with scenarios where the rewards for each arm follow an arbitrary process. In stochastic bandits, the minimax optimal regret is of order $O(\log T)$ [10], while in adversarial bandits, it increases to order $O(\sqrt{T})$ [11]. Recent research has attempted to bridge the gap by achieving a "best-of-both-worlds" outcome [12]–[16]: attaining $O(\log T)$ regret in stochastic environments and simultaneously achieving $O(\sqrt{T})$ regret even in adversarial environments.

Combinatorial Semi-Bandits Stochastic semi-bandit problems have seen several algorithms based on the optimistic principle, achieving a regret bound of $O(\log T)$ [5]. Algorithms with $O(\sqrt{T})$ regret for the adversarial semi-bandit setting have also been well-studied [17]–[21]. These algorithms typically rely on either Follow-the-Regularized-Leader (FTRL) or Follow-the-Perturbed-Leader (FTPL) techniques. In a particular work [8], a novel hybrid regularizer was introduced within the FTRL framework, leading to $O(\log T)$ regret in stochastic environments and $O(\sqrt{T})$ regret in adversarial environments. Building upon this idea, our B-FTRL algorithm adopts the same hybrid regularizer to address problems with switching costs and arm selection constraints.

Bandits with Switching Cost Dealing with bandits with switching costs often involves dividing time into blocks of increasing length. For stochastic bandits with switching costs, an asymptotically optimal $O(\log T)$ regret bound can be derived using a block-based Upper-Confidence-Bound (UCB) algorithm [6]. However, introducing a unit switching cost incurs a lower bound of $\tilde{\Omega}(T^{2/3})$ on the minimax regret [22]. Recent state-of-the-art block-based algorithms [23], [24] achieve $O(T^{2/3})$ regret in the more general adversarial bandits setting, matching the minimax lower bound, and $O(T^{1/3})$ regret in the stochastic bandits setting. Other approaches [25], such as those based on the Gittens index, have also been developed.

Bandits with Arm Selection Constraints Bandit problems with arm selection constraints have gained significant attention, starting with a work [7] that considered fairness constraints to ensure each arm is played a minimum fraction of time. A more comprehensive framework was later developed in another work [26]–[29]. These algorithms fall under the category of

pessimistic-optimistic algorithms, which consider long-term constraints and utilize the Lyapunov drift theorem [30] to ensure long-term queue stability. However, their results imply a constraint violation of $o(t)$ for the anytime cumulative version of these constraints. Follow-up works [31] have attempted to reduce the constraint violation by adding a "tightness" parameter to the virtual queues. Additionally, kernelized bandits with constraints were studied in [32], [33], achieving $O(\sqrt{T})$ regret and constraint violation.

Recent works [9], [34], [35] investigated constrained bandit problems with switching costs under the non-combinatorial setting. In the most recent work [9], a blocked-based pessimistic-optimistic algorithm was developed, achieving $O(T^{3/4})$ regret and $o(1)$ constraint violation in expectation. In comparison, our proposed algorithm achieves a $O(T^{2/3})$ regret and guarantees that constraints are satisfied in expectation. Moreover, we provide a high probability bound on the realized constraint violation.

III. Problem Formulation

A. Model

We investigate a sequential game involving a learner and an adversary that models the environment. The game consists of d fixed (base) arms. At each time $t = 1, 2, ...$, the learner must select no more than m out of these d arms to play, forming a set of arms called a super-arm denoted by $X_t \in \{0,1\}^d$, where $X_{t,i} = 1$ indicates that arm i is played, and $X_{t,i} = 0$ otherwise. Let $\mathcal{D} \subseteq \{0,1\}^d$ be the set of feasible super-arms that are composed of m base arms.

Reward: At each time t, the adversary selects a reward vector $\ell_t \in [-1,1]^d$ simultaneously with the learner's choice of the super-arm. The learner receives a reward $r_t = \langle X_t, \ell_t \rangle$ and observes the individual rewards of the selected base arms, denoted by $o_t = X_t \circ \ell_t \in [-1,1]^d$, where $\circ$ stands for element-wise product. It should be noted that for any unselected arm i, no reward is observed, but we let $o_{t,i} = 0$ for simplicity. Importantly, the adversary in this context is an oblivious one, meaning that the reward vectors $\ell_1, \ell_2, ...$ are pre-determined and independent of the learner's actions.

Switching Cost: Additionally, the problem accounts for costs incurred due to switching arms in consecutive time slots. A cost $h_t = H(X_t, X_{t-1})$ is incurred if the chosen super-arm X_t is different from the previous one, X_{t-1}. The switching cost function $H(\cdot,\cdot)$ can take various forms, for example, $H(X_t, X_{t-1}) = \|X_t - X_{t-1}\|_1$ or $H(X_t, X_{t-1}) = \mathbb{1}\{X_t \neq X_{t-1}\}$. Our model accommodates different switching cost functions, as long as the switching cost is upper-bounded by λ, defined as follows:

$$H(X_t, X_{t-1}) \leq \lambda \cdot \mathbb{1}\{X_t \neq X_{t-1}\}.$$

To complete the formulation, we set $X_0 = X_1$ to ensure that the first-slot switching cost $h_1 = H(X_1, X_0) = 0$.

Constraints: Furthermore, the problem involves K anytime cumulative constraints, each characterized by a d-dimensional coefficient $w^k \in [-1,1]^d$, where $w_i^k \in [-1,1]$ represents a penalty associated with constraint k of playing arm i at time t. The penalty of playing a super-arm $X_{t,i}$ at time t for constraint k is $c_t^k = \langle X_t, w_t^k \rangle$. The anytime cumulative constraints require that the cumulative penalty be no greater than 0 for each constraint k at any time t, i.e., $\sum_{\tau=1}^{t} c_\tau^k \leq 0, \forall t$.

The objective of the adversarial combinatorial bandits with switching cost and arm selection constraints problem is to devise an algorithm that selects the super-arm to play in each time slot, aiming to maximize the cumulative net reward (i.e., reward minus switching cost) subject to K anytime cumulative constraints. Formally, the problem can be expressed as follows:

$$\begin{aligned} \max_{X_1,...,X_T \in \mathcal{D}} \quad & \sum_{t=1}^{T} \left(\langle X_t, \ell_t \rangle - H(X_t, X_{t-1}) \right) \\ \text{subject to} \quad & \sum_{\tau=1}^{t} \langle X_\tau, w^k \rangle \leq 0, \quad \forall k = 1, ..., K, \forall t = 1, ..., T \end{aligned}$$

B. Regret

We evaluate the performance of our algorithm in terms of (expected) regret, which quantifies the difference in expected net reward between our algorithm and the best fixed randomized policy with complete information about the adversary's reward vectors. A fixed randomized policy is represented by a fixed vector $x \in [0,1]^d$ with the constraint that $\|x\|_1 \leq m$. In each time slot, the policy randomly selects an m-sized super-arm based on the probabilities defined by the vector x according to a sampling rule P (see Appendix).

The best fixed randomized policy can be found by solving the following constrained optimization problem:

$$\begin{aligned} \max_{x \in [0,1]^d, \|x\|_1 \leq m} \quad & \langle x, \sum_{t=1}^{T} \ell_t \rangle \\ \text{subject to} \quad & \langle x, w^k \rangle \leq 0, \quad \forall k = 1, ..., K \end{aligned}$$

Let x^* be the optimal solution of the randomized policy to the above problem, and let $\mathbf{OPT}(T)$ represent the optimal cumulative reward. It is important to note that $\mathbf{OPT}(T)$ does not account for the switching cost incurred when two realizations of X_t and X_{t-1}, generated according to the fixed policy x^*, are different. Therefore, it serves as an optimistic upper-bound on the actual cumulative net reward.

The regret of our algorithm is defined as:

$$\mathbf{REG}(T) = \mathbf{OPT}(T) - \mathbb{E}\left[\sum_{t=1}^{T} \langle X_t, \ell_t \rangle - H(X_t, X_{t-1}) \right].$$

In general, an algorithm, especially being a randomized policy, may not always produce actions that strictly adhere to the constraints in every time slot. In fact, prioritizing aggressive reward maximization can easily lead to constraint violations. Therefore, we also measure the performance of our algorithm in terms of constraint violation at every time t, which is defined as:

$$V_t^k = \sum_{\tau=1}^{t} \langle X_\tau, w^k \rangle.$$

In particular, $V_t^k \leq 0$ means constraint k in time slot t is satisfied while $V_t^k > 0$ means that constraint k is violated by an amount V_t^k.

IV. Algorithm

To tackle the constrained adversarial combinatorial bandits with switching cost problem, we introduce a novel algorithm called B-FTRL, where "B" stands for block. Our algorithm draws inspiration from the FTRL (Follow-The-Regularized-Leader) algorithm proposed in [8], which was originally proposed for combinatorial bandits problems under both stochastic and adversarial settings.

In the B-FTRL algorithm, we group the T time slots into several time blocks, indexed by $n = 1, 2, ...$, to minimize the impact of switching costs. Within each time block, the same super-arm is played in every time slot, and switching is only allowed in the first time slot of a block. Let $\mathcal{B}_n$ denote the set of time slots that belong to block n, and $|\mathcal{B}_n|$ be the length of that block. In the first time slot of block n, the algorithm computes the "regularized leader" using information from the past $n-1$ blocks. The regularized leader x_n is defined as follows:

$$x_n = \arg\max_{x \in \mathcal{X}} \langle x, \hat{L}_{n-1} \rangle - \eta_n^{-1} \Psi(x). \tag{1}$$

Here, $\mathcal{X}$ represents the feasible set with constraints: $x \in [0,1]^d$, $\|x\|_1 \leq m$, and $\langle x_i, w^k \rangle \leq 0$ for all k. The term η^n represents a time-decaying learning rate schedule. Let us now explain the remaining components of Equation (1), namely $\hat{L}_{n-1}$ and $\Psi(x)$.

Cumulative reward estimate: $\hat{L}_{n-1}$ is the cumulative reward estimate up to block n, i.e., $\hat{L}_{n-1} = \sum_{s=1}^{n-1} \hat{\ell}_s$, where $\hat{\ell}_s \in \mathbb{R}^d$ represents the total reward estimate in block s. Specifically, $\hat{\ell}_n$ is computed at the end of each time block n using the observed rewards $o_n = \sum_{t \in \mathcal{B}_n} X_t \circ \ell_t \in \mathbb{R}^d$, where X_t is the super-arm sampled using the sampling rule P (see Appendix) according to x_n, and it remains the same for all time slots in block n. The formula for $\hat{\ell}_n$ is as follows:

$$\hat{\ell}_{n,i} = \left(\frac{(o_{n,i}/|\mathcal{B}_n| + 1)\mathbb{1}_n(i)}{x_{n,i}} - 1 \right) |\mathcal{B}_n|, \quad \forall i = 1, \cdots, d$$

where $\mathbb{1}_n(i) \in \{0, 1\}$ indicates whether arm i is played in block n or not. It can be shown that $\hat{\ell}_{n,i}$ is an unbiased estimator of $\ell_{n,i}$ as follows:

$$\begin{aligned} \mathbb{E}[\hat{\ell}_{n,i}] =& x_{n,i} \left(\frac{\sum_{t \in \mathcal{B}_n} \ell_{t,i}/|\mathcal{B}_n| + 1}{x_{n,i}} - 1 \right) |\mathcal{B}_n| \\ &+ (1 - x_{n,i})(-1)|\mathcal{B}_n| \\ =& \sum_{t \in \mathcal{B}_n} \ell_{t,i} = \ell_{n,i}, \end{aligned}$$

where we slightly abuse notation to use $\ell_{n,i}$ to denote the total reward obtained by playing arm i in block n, while $\ell_{t,i}$ represents the reward by playing arm i in time slot t.

Algorithm 1 B-FTRL

Input: $\mathcal{X} = \{x | 0 \leq x \leq 1, \|x\|_1 \leq m, \sum_i x_i w_i^k \leq 0, \forall k\}$
Initialization: $\hat{L}_0 = (0, \ldots, 0), \eta_n = \beta/n, |\mathcal{B}_n| = \max\{\lceil \alpha\sqrt{n} \rceil, 1\}$
for block $n = 1, 2, ...,$ **do**
  Compute

$$x_n = \arg\max_{x \in \mathcal{X}} \left\{ \langle x, \hat{L}_{n-1} \rangle - \eta_n^{-1} \Psi(x) \right\},$$

  Sample $X_n \sim P(x_n)$ such that $\mathbb{E}_{X \sim P}[X_n] = x_n$
  Play X_n for all rounds $t \in \mathcal{B}_n$
  Observe $o_n = \sum_{t \in \mathcal{B}_n} X_t \circ \ell_t$
  Construct estimator $\hat{\ell}_n, \forall i$:

$$\hat{\ell}_{n,i} = \left(\frac{(o_{n,i}/|\mathcal{B}_n| + 1)\mathbb{1}_t(i)}{x_{n,i}} - 1 \right) \cdot |\mathcal{B}_n|$$

  Update $\hat{L}_n = \hat{L}_{n-1} + \hat{\ell}_n$
end for

Regularizer: The term $\Psi(x)$ is a regularization term with the following expression:

$$\Psi(x) = \sum_{i=1}^{d} \left(\sqrt{x_i} - (1 - x_i) \log(1 - x_i) \right).$$

In essence, $\Psi(x)$ combines the Tsallis entropy (with power $1/2$), denoted by $-\sum_{i=1}^{d} \sqrt{x_i}$, and the Shannon entropy on the complement of x, denoted by $\sum_{i=1}^{d} (1 - x_i) \log(1 - x_i)$.

After computing the "regularized leader" x_n, the algorithm samples $X_n \sim P(x_n)$ using the sampling rule P and plays the selected super-arm in every time slot within block n.

V. Regret Analysis

In this section, we analyze the regret of the proposed B-FTRL algorithm.

A. Main results

To facilitate our analysis, we divide regret into two parts:

$$\mathbf{REG}(T) = \mathbf{REG}^{\text{reward}}(T) + \mathbf{REG}^{\text{switching}}(T),$$

where $\mathbf{REG}^{\text{reward}}(T) = \mathbf{OPT} - \mathbb{E}\left[\sum_{t=1}^{T} \langle X_t, \ell_t \rangle\right]$ accounts for the reward difference between the optimal fixed randomized policy and our algorithm, and $\mathbf{REG}^{\text{switching}}(T) = \mathbb{E}\left[\sum_{t=1}^{T} H(X_t, X_{t-1})\right]$ accounts for the cost due to action switching. Theorem 1 presents the main results of the regret analysis.

Theorem 1. By setting the block length as $|\mathcal{B}_n| = \max\{\lceil \alpha\sqrt{n} \rceil, 1\}$ and the learning rate $\eta_n = \beta/n$, where $\alpha, \beta > 0$ are constants satisfying $\alpha\beta \leq \frac{\sqrt{2}-1}{2}$, B-FTRL ensures that the regret is bounded as $\mathbf{REG}(T) \leq O(T^{2/3})$. Moreover, the constraints are satisfied in expectation, i.e., $\mathbb{E}[V_T^k] \leq 0, \forall k$ and $V_T^k \leq O\left(\sqrt{T \log \frac{1}{\delta}}\right)$ with probability $1 - \delta$ for any $\delta \in (0, 1)$.

TABLE I
CUMULATIVE REWARD REGRET VS. SWITCHING REGRET

Parameter a	Parameter b	Reward regret	Switching regret
$1/2$	1	$O(T^{2/3})$	$O(T^{2/3})$
1	1	$O(T)$	$O(\sqrt{T})$
0	$1/2$	$O(\sqrt{T})$	$O(T)$

Now, let us discuss our regret and constraint violation bounds compared to those derived in [9], which studied a constrained stochastic bandits with switching costs problem. Our considered setting generalizes that in [9] by allowing adversarial rewards (instead of stochastic rewards) and combinatorial actions (instead of individual actions), making the comparison meaningful in this special case. **Regret**: In the stochastic reward setting, [9] proves a regret bound of $O(T^{3/4})$ for their algorithm. However, our B-FTRL algorithm can improve the regret bound to $O(T^{2/3})$ even in the more difficult adversarial reward setting. **Constraint Violation**: In [9], a $o(1)$ bound is provided on the expected constraint violation. In our algorithm, the expected constraint violation is guaranteed to be satisfied because the sampling vector x_t at every time slot t is chosen such that $\langle x_t, w^k\rangle \leq 0$. Moreover, we also provide a sublinear high-probability bound on the realized constraint violation.

In Theorem 2, we characterize the impact of algorithm parameters on the regret bound:

Theorem 2. By setting the block length as $|\mathcal{B}_n| = \max\{\lceil \alpha n^a \rceil, 1\}$ and the learning rate as $\eta_n = \beta n^{-b}$, where $\alpha\beta \leq \frac{\sqrt{2}-1}{2}$ and $b \geq a$, B-FTRL ensures the following bounds on the reward regret and the switching regret:

$$\mathbf{REG}^{\text{reward}}(T) \leq O\left(T^{\frac{2a-b+1}{1+a}} + T^{\frac{b}{1+a}}\right) \tag{2}$$

$$\mathbf{REG}^{\text{switching}}(T) \leq O\left(T^{\frac{1}{1+a}}\right) \tag{3}$$

Moreover, the constraints are satisfied in expectation, i.e., $\mathbb{E}[V_T^k] \leq 0, \forall k$ and $V_T^t \leq O\left(\sqrt{T\log\frac{1}{\delta}}\right)$ with probability $1-\delta$ for any $\delta \in (0,1)$.

Theorem 2 shows that B-FTRL is able to make a tradeoff between the reward regret and the switching regret. Specifically, fixing b and decreasing a decreases the reward regret at the expense of a larger switching regret. The values of a and b can be chosen depending on the relative importance of the reward and the switching cost. Table I lists the respective reward regret and switching cost for specific values of a and b. We remark that when $a = 0$ and $b = 1$, B-FTRL is equivalent to the algorithm in [8], in which the switching costs are not considered, and hence, our regret bound recovers their theoretical results.

B. Proofs

We start by introducing some preliminary definitions and results. Let $\Psi_n(x) = \eta_n^{-1}\Psi(x)$, and Φ_n be defined as

$$\Phi_n(C) = \max_{x\in\mathcal{X}}\{\langle x, C\rangle - \Psi_n(x)\} \tag{4}$$

Then we have $\nabla\Phi_n(C) = \arg\max_{x\in\mathcal{X}}\{\langle x, C\rangle - \Psi_n(x)\}$. Recall that the sampling vector x_n in our algorithm is chosen as $x_n = \arg\max_{x\in\mathcal{X}}\left\{\langle x, \hat{L}_{n-1}\rangle - \Psi_n(x)\right\}$ and hence $x_n = \nabla\Phi_n(\hat{L}_{n-1})$. Following the standard analysis for FTRL, we decompose the reward regret into a sum of stability and penalty terms:

$$\begin{aligned}
&\sum_{t=1}^{T}\langle x^*, \ell_t\rangle - \mathbb{E}[\sum_{t=1}^{T}\langle x_t, \ell_t\rangle] = \sum_{n=1}^{N}\langle x^*, \ell_n\rangle - \mathbb{E}[\sum_{n=1}^{N}\langle x_n, \ell_n\rangle] \\
&= \underbrace{\mathbb{E}\left[\sum_{n=1}^{N}\left(-\langle x_n, \ell_n\rangle + \Phi_n(\hat{L}_n) - \Phi_n(\hat{L}_{n-1})\right)\right]}_{\text{stability}} + \\
&\underbrace{\mathbb{E}\left[\sum_{n=1}^{N}\left(\Phi_n(\hat{L}_{n-1}) - \Phi_n(\hat{L}_n) + \langle x^*, \ell_n\rangle\right)\right]}_{\text{penalty}} \\
&= \mathbf{REG}^{\text{stability}} + \mathbf{REG}^{\text{penalty}}
\end{aligned}$$

where x^* is the optimal sampling vector in hindsight, N is the number of blocks and $\ell_n = \sum_{t\in\mathcal{B}_n}\ell_t$ with a slight abuse of notation. Lemmas 1 and 2 below develop bounds on the stability and the penalty parts of the regret, respectively. These results generalize those in [8], to handle block-based decision processes and randomized benchmarks in the regret definition. Their proofs can be found in the appendix.

Lemma 1. For any sequence of learning rate and block size sequences $\{\eta_n = \beta/n^b, |\mathcal{B}_n| = \max\{\lceil\alpha n^a\rceil, 1\}$ that satisfy $\alpha\beta \leq \frac{\sqrt{2}-1}{2}$ and $b \geq a$, the *stability* term in the regret is bounded as

$$\mathbf{REG}^{\text{stability}} \leq \sum_{n=1}^{N} 16\sqrt{2md}\eta_n|\mathcal{B}_n|^2. \tag{5}$$

Lemma 2. For any non-increasing sequence of learning rates $\{\eta_n\}_n$, the *penalty* term in the regret is bounded as

$$\mathbf{REG}^{\text{penalty}} \leq \frac{3}{2}\sqrt{md}\eta_N^{-1}. \tag{6}$$

We now prove Theorem 1 and Theorem 2 by combining the results in Lemma 1 and Lemma 2. In order to apply our results to blocks, we first derive an upper bound on the number of blocks N. Because $|\mathcal{B}_n|$ is chosen as $\max\{\lceil\alpha n^a\rceil, 1\}$, we have $|\mathcal{B}_n| \geq \alpha n^a$ and non-decreasing. Let $\Gamma = \left(\frac{a+1}{\alpha}\right)^{\frac{1}{a+1}} T^{\frac{1}{a+1}}$ and observe that:

$$\begin{aligned}
\sum_{n=1}^{\Gamma+1}|\mathcal{B}_n| &\geq \sum_{n=1}^{\Gamma+1}\alpha n^a \geq \int_0^{\Gamma+1}\alpha n^a dn \\
&\geq \int_0^{\Gamma}\alpha n^a\, dn = \frac{\alpha}{a+1}\Gamma^{a+1} \geq T.
\end{aligned}$$

Thus, we can upper bound N by $\Gamma + 1$.

To bound the regret incurred by our algorithm, we need to control the terms $\sum_{n=1}^{N}\eta_n|\mathcal{B}_n|^2$, η_N^{-1}, and the number of switches. Note that the number of switches is bounded by the

number of blocks, i.e., $\Gamma+1$. Thus, the cumulative switching regret satisfies

$$\mathbf{REG}^{\text{switching}}(T) \leq \lambda(\Gamma+1) \leq O(T^{\frac{1}{a+1}}).$$

The reward regret can be bounded by

$$\mathbf{REG}^{\text{reward}}(T) = \mathbf{REG}^{\text{stability}} + \mathbf{REG}^{\text{penalty}}$$
$$\leq \sum_{n=1}^{N} 16\sqrt{2md}\eta_n|\mathcal{B}_n|^2 + \frac{3}{2}\sqrt{md}\eta_N^{-1}.$$

By choosing $\eta_n = \beta/n^b$ and $|\mathcal{B}_n| = \max\{\lceil \alpha n^a \rceil, 1\}$,

$$\mathbf{REG}^{\text{reward}}(T) \leq O(\sum_{n=1}^{N} \alpha^2\beta n^{2a-b} + \sum_{n=1}^{N} \beta n^{-b} + N^b/\beta)$$
$$\leq O(T^{\frac{1+2a-b}{a+1}} + T^{\frac{1-b}{a+1}} + T^{\frac{b}{a+1}}) \leq O(T^{\frac{1+2a-b}{a+1}} + T^{\frac{b}{a+1}}),$$

where the second inequality uses $\sum_{n=1}^{N} n^{2a-b} \leq O(N^{1+2a-b})$, $\sum_{n=1}^{N} n^{-b} \leq O(N^{1-b})$ and the bound Γ on N. This proves the general regret bounds in Theorem 2. Now, by letting $a = 1/2$ and $b = 1$, the maximum order is the reward regret and the switching regret is minimized at $T^{2/3}$ and hence, the regret bound in Theorem 1 is proved.

Finally, we consider the constraint violation. Because in every slot t, the arms are chosen so that $\mathbb{E}[\langle X_t, w^k\rangle] = \langle x_t, w^k\rangle \leq 0$, the constraint is guaranteed to be satisfied in expectation in our algorithm, i.e., $\mathbb{E}[V_t^k] \leq 0$. Next, we bound the realized constraint violation using the Hoeffding inequality. Let $\sigma = \max_{k,i} |w_{k,i}|$. Then we have for all k,

$$\text{Prob}\left(V_T^k - \mathbb{E}[V_T^k] \geq \sqrt{T\log\frac{1}{\delta}}\right) \leq e^{-\frac{T\log\frac{1}{\delta}}{2\sigma^2 T}} = O(\delta).$$

Therefore,

$$V_t^k \leq \mathbb{E}[V_t^k] + O\left(\sqrt{T\log\frac{1}{\delta}}\right) \leq O\left(\sqrt{T\log\frac{1}{\delta}}\right), \text{w.p.} 1-\delta$$

VI. Numerical Results

In this section, we present the numerical results to assess the performance of our proposed B-FTRL algorithm and compare it against baseline approaches. While our algorithm is designed for the adversarial setting, we extend the evaluation to include both the adversarial and stochastic settings. The inclusion of the stochastic setting is valuable as it represents a special (and comparatively easier) case of the adversarial setting. Furthermore, evaluating our algorithm in the stochastic setting enables us to directly compare its performance with existing algorithms designed specifically for stochastic environments.

A. Setup

Rewards We conduct simulations with $d = 7$ individual arms. In the stochastic setting, the expected rewards for each arm are set as $\bar{\ell} = [0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]$. To introduce variability, a random noise ϵ uniformly sampled from $[-0.01, 0.01]$ is added to the realized reward at each time slot, i.e., $\ell_t = \bar{\ell} + \epsilon$.

In the adversarial setting, following the approach in [8], [16], we divide time into multiple phases as follows:

$$\underbrace{1, ..., t_1}_{T_1}, \underbrace{t_1+1, ..., t_2}_{T_2}, ..., \underbrace{t_{n-1}, ..., T}_{T_n}.$$

The length of phase s is defined as $T_s = \lceil 1.6^s \rceil$. Within each phase, the expected rewards are set as:

$$\bar{\ell}_s = \begin{cases} \bar{\ell} + 0.2, & \text{if } s \text{ is odd} \\ \bar{\ell} - 0.2, & \text{if } s \text{ is even} \end{cases}$$

Again, a random noise term ϵ uniformly sampled from $[-0.01, 0.01]$ is added to the realized reward $\ell_t = \bar{\ell}_s + \epsilon$ for time slots belonging to phase s.

Switching cost We incorporate the switching cost function as $H(X_t, X_{t-1}) = \|X_t - X_{t-1}\|_1$, which means that the switching cost is proportional to the L_1 norm (Manhattan distance) between the arm choices at two consecutive time slots.

Constraints We consider both $K = 1$ and $K = 3$ arm selection constraints. The constraint vectors w^k are uniformly and randomly sampled from the range $[-1, 1]^d$. These constraints provide additional criteria that need to be satisfied during the decision-making process.

B. Baseline algorithm

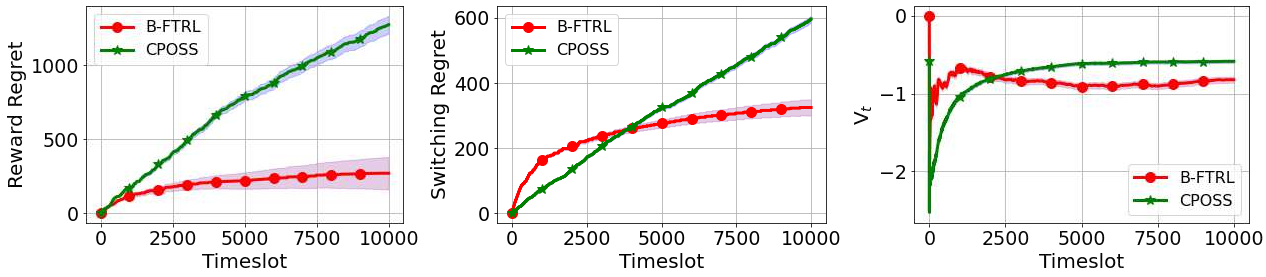
We extend the POSS algorithm proposed in [9] to the combinatorial setting, which we refer to as CPOSS, and utilize it as the baseline algorithm for performance comparison. In CPOSS, we adopt the same underlying concept as POSS but modify the arm selection strategy to choose the super-arm with the highest sum of the pessimistic-optimistic indices. It is worth noting that this modification does not impact the asymptotic bounds of the algorithm.

Regarding our B-FTRL algorithm, the default parameter values are set as follows: $\alpha = \frac{\sqrt{2}-1}{4}$, $\beta = 1$, $b = 1$, and $a = \frac{1}{2}$. We conduct a comprehensive evaluation by averaging the results over 30 independent random experiments to ensure robustness and reliability in our analysis.

C. Results in the stochastic setting

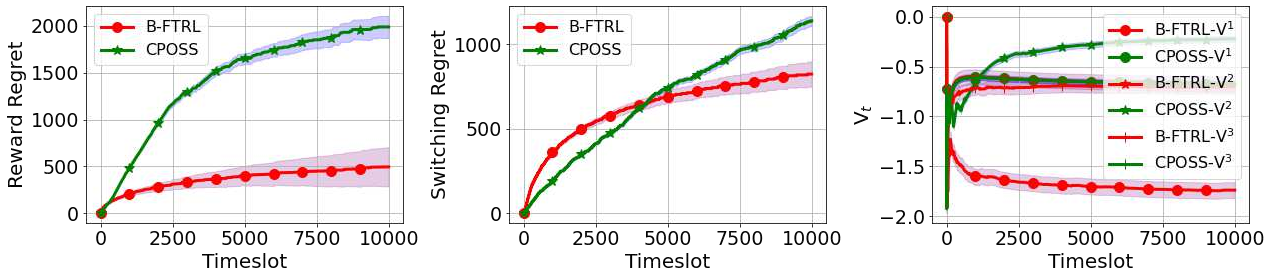
Figures 1 and 2 depict the reward regret, switching regret (equivalently, switching cost), and constraint violation achieved by B-FTRL and CPOSS in the stochastic setting. In the experiments corresponding to Figure 1, we set $m = 1$ and $K = 1$. Under this configuration, CPOSS reduces to POSS, enabling a direct comparison between B-FTRL and POSS. On the other hand, the experiments corresponding to Figure 2 were conducted with $m = 3$ and $K = 3$, allowing comparisons to be made in a more general setting.

1) Regret: Figures 1(a)(b) present a comparison of reward regret and switching regret achieved by B-FTRL and CPOSS, both with $m = 1$ and $K = 1$. Additionally, Figures 2(a)(b) showcase the results with $m = 3$ and $K = 3$. In both setups, B-FTRL and CPOSS exhibit sublinear growth in reward regret and switching regret over time, with B-FTRL consistently outperforming CPOSS (while we note that CPOSS's regret



(a) Reward regret (b) Switching regret (c) Time-average constraint value

Fig. 1. Performance Comparison in the Stochastic Setting ($m = 1$, $K = 1$).



(a) Reward regret (b) Switching regret (c) Time-average constraint value

Fig. 2. Performance Comparison in the Stochastic Setting ($m = 3$, $K = 3$).

is sublinear, though not as apparent). This trend aligns with the theoretical regret bound results: B-FTRL's regret follows $O(T^{2/3})$, while CPOSS's regret follows $O(T^{3/4})$. Notably, B-FTRL's bound of $O(T^{2/3})$ is derived in the more challenging adversarial setting, making its actual performance even better than the theoretical bound in the stochastic setting. Improving the theoretical bound for B-FTRL in the stochastic setting remains an interesting future research direction. Additionally, we observed that B-FTRL's regret exhibits higher variance, likely due to the arm sampling operation in the algorithm.

2) Constraint Violation: Figures 1(c) and 2(c) demonstrate that both CPOSS and B-FTRL satisfy the constraints in the experiments for both configurations. However, it is evident that B-FTRL has a distinct advantage in meeting the constraints. This is because B-FTRL ensures that the constraint is satisfied in expectation for each time slot, providing a more robust approach, whereas CPOSS only guarantees that the expected constraint violation is $o(1)$, which may not offer the same level of reliability in fulfilling the constraints.

D. Results in the adversarial setting

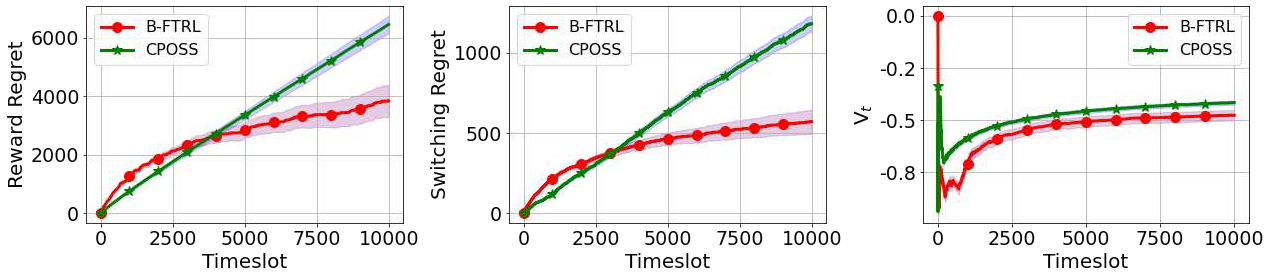
Figures 3 and 4 display the reward regret, switching regret (equivalently, switching cost), and constraint violation achieved by B-FTRL and CPOSS in the adversarial setting. Specifically, in the experiments corresponding to Figure 3, we set $m = 1$ and $K = 1$, and in the experiments corresponding to Figure 4, we set $m = 3$ and $K = 3$.

1) Regret: Figures 3(a)(b) provide a comparison of the reward regret and switching regret achieved by B-FTRL and CPOSS, with both algorithms configured with $m = 1$ and $K = 1$. Additionally, Figures 4(a)(b) showcase the results when $m = 3$ and $K = 3$. In line with the findings from the stochastic setting, B-FTRL continues to exhibit significantly superior performance compared to CPOSS. However, it is worth noting that both B-FTRL and CPOSS show relatively poorer performance in the adversarial setting compared to the stochastic setting. For B-FTRL, the regret still demonstrates sublinear growth over time, which aligns with our theoretical expectations. Nevertheless, the growth rate is faster in the adversarial setting than in the stochastic setting. As for CPOSS, since it is explicitly designed for the stochastic setting, its performance is notably worse when faced with adversarial conditions.

2) Constraint Violation: Figures 3(c) and 4(c) provide evidence that both CPOSS and B-FTRL successfully satisfy the constraints in the experiments for both configurations in the adversarial setting. However, it is notable that the constraint violation is larger (although still negative) for both algorithms in the adversarial environment, as they need to make more frequent adaptations to cope with the adversarial challenges. It is worth mentioning that CPOSS utilizes the virtual queue technique to handle constraints, which mitigates the impact of the adversarial reward setting on constraint violation compared to its effect on regret performance. Nevertheless, B-FTRL still demonstrates an advantage in effectively meeting the constraints when compared to CPOSS.

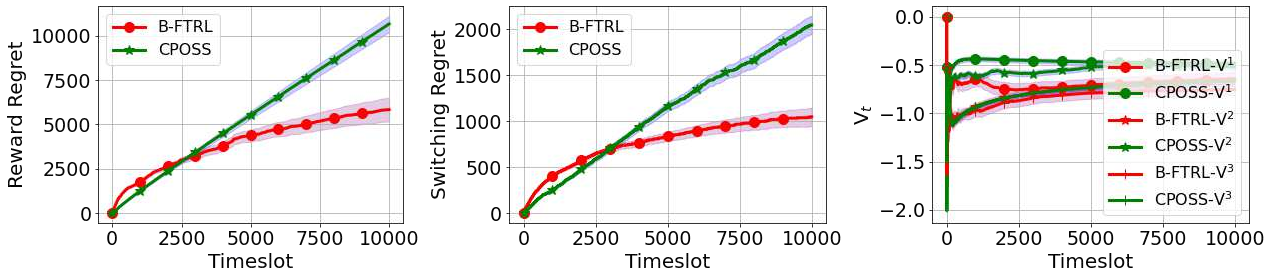
VII. Conclusion

This paper delves into a novel adversarial combinatorial bandits problem, encompassing switching costs and arm selection constraints, which hold immense practical relevance in cloud resource management, distributed sensing and monitoring, and



(a) Reward regret (b) Switching regret (c) Time-average constraint value

Fig. 3. Performance Comparison in the Adversarial Setting ($m = 1$, $K = 1$).



(a) Reward regret (b) Switching regret (c) Time-average constraint value

Fig. 4. Performance Comparison in the Adversarial Setting ($m = 3$, $K = 3$).

healthcare resource management, among other applications. We introduced a novel algorithm called B-FTRL, adept at effectively balancing exploration and exploitation while taking into account the switching costs and adhering to the arm selection constraints. Through both theoretical analysis and extensive numerical simulations, we demonstrated the significant superiority of B-FTRL over existing methods.

An interesting aspect of our work is that although the algorithm is designed for the adversarial setting, its techniques have the potential to tackle "best-of-both-worlds" problems, thus extending its applicability beyond the adversarial domain. As a future research direction, we aim to establish tighter regret bounds for B-FTRL in the stochastic bandits setting, further enhancing the algorithm's performance and versatility. Such advancements hold great promise in a wide range of real-world decision-making scenarios.

APPENDIX

PROOF OF LEMMA 1

In order to bound the stability term, we recall several properties of the potential function provided by [8], which we use in this proof:

$$\nabla\Psi_n(x) = \eta_n^{-1}\left(\frac{1}{2\sqrt{x_i}} + \log(1-x_i) + 1\right)_{i=1,\dots,d},$$

$$\nabla^2\Psi_n(x) = \eta_n^{-1}\text{diag}\left[\left(\frac{1}{4\sqrt{x_i^3}} + \frac{1}{1-x_i}\right)_{i=1,\dots,d}\right],$$

$$(\nabla^2\Psi_n(x))^{-1} \preceq \eta_n\text{diag}\left[\left(\min\left\{4\sqrt{x_i^3}, (1-x_i)\right\}\right)_{i=1,\dots d}\right].$$

We define the convex conjugate and the associated Bregman divergence of a convex function $f : \mathcal{C} \to \mathbb{R}$ as

$$f^*(\cdot) = \max_{x\in\mathcal{C}}\{\langle x, \cdot\rangle - f(x)\}$$

$$D_f(x, y) = f(x) - f(y) - \langle\nabla f(y), x - y\rangle$$

respectively. Then we have $\Psi_n^*(\cdot) = \max_{x\in\mathcal{X}}\{\langle x, \cdot\rangle - \Psi_n(x)\}$ and the following statements are true [36]

- **S1**: $\nabla\Psi_n^*(\cdot) = \arg\max_{x\in\mathcal{X}}\{\langle x, \cdot\rangle - \Psi_n(x)\}$, $\nabla\Psi_n = (\nabla\Psi_n^*)^{-1}$, $\nabla^2\Psi_n(x) = (\nabla^2\Psi_n^*(\nabla\Psi_n(x)))^{-1}$.
- **S2**: For any $x, y \in \mathbb{R}^d$, by Taylor's theorem, there exist a $z \in \text{Conv}(\{x, y\})$ such that $D_{\Psi_n^*}(x, y) = \frac{1}{2}\|x - y\|^2_{\nabla^2\Psi_n^*(z)}$.
- **S3**: For any L, let $\tilde{L} = \nabla\Psi_n(\nabla\Phi_n(L))$. Then it holds for any $\ell \in \mathbb{R}^d$, $D_{\Phi_n}(L + \ell, L) \le D_{\Psi_n^*}(\tilde{L} + \ell, \tilde{L})$.

Now, using these preliminary results, we have

$$\textbf{REG}^{\text{stability}} = \mathbb{E}\left[\sum_{n=1}^N D_{\Phi_n}(\hat{L}_n, \hat{L}_{n-1})\right]$$

$$\overset{(i)}{\le}\mathbb{E}\left[\sum_{n=1}^N D_{\Psi_n^*}(\nabla\Psi_n(x_n) + \hat{\ell}_n, \nabla\Psi_n(x_n))\right]$$

$$\overset{(ii)}{=}\mathbb{E}\left[\sum_{n=1}^N \frac{1}{2}||\hat{\ell}_n||^2_{\nabla^2\Psi_n^*(z_n)}\right] = \mathbb{E}\left[\sum_{n=1}^N \frac{1}{2}||\hat{\ell}_n||^2_{\nabla^2\Psi_n^*(\nabla\Psi_n(y_n))}\right]$$

$$\overset{(iii)}{=}\mathbb{E}\left[\sum_{n=1}^N \frac{1}{2}||\hat{\ell}_n||^2_{\nabla^2\Psi_n(y_n)^{-1}}\right],$$

where (i), (ii) and (iii) hold due to **S3**, **S2** and **S1**, respectively. Here, we let $\tilde{x}_n = \nabla\Psi_n^*(\nabla\Psi_n(x_n) + \hat{\ell}_n)$, and choose $z_n \in \text{Conv}\{\nabla\Psi_n(x_n), \nabla\Psi_n(\tilde{x}_n)\}$ and $y_n \in [x_n, \tilde{x}_n]$ such that $\nabla\Psi_n(y_n) = z_n$.

Next, we show that when $\alpha\beta \le \frac{\sqrt{2}-1}{2}$ and $b \ge a$ (which means $\eta_n|\mathcal{B}_n| \le \frac{\sqrt{2}-1}{2}$), the following always holds:

$$\tilde{x}_n = \nabla\Psi_n^*(\nabla\Psi_n(x_n) + \hat{\ell}_n) \le 2x_n \tag{7}$$

Because function $\nabla\Psi_n$ and $\nabla\Psi_n^*$ are symmetric and independent in each coordinate, it is sufficient to consider one dimension and drop the index i to prove (7). We consider only $x_n \le 1/2$, otherwise the statement is trivial since the range of $\Delta\Psi_n^*$ is $[0,1]^d$. Now suppose the opposite holds: $\tilde{x}_n > 2x_n$. Note that by the construction of $\tilde{\ell}_n$, we have $-|\mathcal{B}_n| \le \hat{\ell}_n \le \frac{2|\mathcal{B}_n|}{x_n}$. Since $\nabla\Psi_n(x_n)$ is strictly decreasing in $(0,1)$, we have

$$\begin{aligned}
-\hat{\ell}_n &= \nabla\Psi_n(x_n) - \nabla\Psi_n(x_n) - \hat{\ell}_n \\
&= \nabla\Psi_n(x_n) - \nabla\Psi_n(\nabla\Psi_n^*(\nabla\Psi_n(x_n) + \hat{\ell}_n)) \\
&= \nabla\Psi_n(x_n) - \nabla\Psi_n(\tilde{x}_n) > \nabla\Psi_n(x_n) - \nabla\Psi_n(2x_n) \\
&= \eta_n^{-1}\left(\frac{1}{2\sqrt{x_n}} + \log(1-x_n) - \frac{1}{2\sqrt{2x_n}} - \log(1-2x_n)\right) \\
&> \eta_n^{-1}\left(\frac{\sqrt{2}-1}{2\sqrt{2}}\right)\frac{1}{\sqrt{x_n}} > \eta_n^{-1}\left(\frac{\sqrt{2}-1}{2}\right),
\end{aligned}$$

where the second inequality uses the monotonicity of the log function and the last inequality uses $x_n \le 1/2$. Because $-\hat{\ell}_n \le |\mathcal{B}_n|$, we have $\eta_n|\mathcal{B}_n| \ge \frac{\sqrt{2}-1}{2}$, which leads to a contradiction when $\alpha\beta \le \frac{\sqrt{2}-1}{2}$ and $b \ge a$. Therefore (7) is proved. Thus, we have

$$\begin{aligned}
\nabla^2\Psi_n(y_n)^{-1} &\preceq \eta_n \cdot \mathrm{diag}\left[\left(4\sqrt{(\tilde{x}_{n,i})^3}\right)_{i=1,\dots,d}\right] \\
&\preceq \eta_n \cdot \mathrm{diag}\left[\left(4\sqrt{(2x_{n,i})^3}\right)_{i=1,\dots,d}\right].
\end{aligned}$$

Going back to the stability term, we thus have

$$\begin{aligned}
\mathbf{REG}^{\text{stability}} &\le \mathbb{E}\left[\sum_{n=1}^N \frac{1}{2}||\hat{\ell}_n||^2_{\nabla^2\Psi_n(y_n)^{-1}}\right] \\
&\le \sum_{n=1}^N \mathbb{E}\left[\frac{\eta_n}{2}\sum_{i=1}^d (\hat{\ell}_{n,i})^2 4\sqrt{(2x_{ni})^3}\right] \\
&\overset{(iv)}{\le} \sum_{n=1}^N \frac{\eta_n|\mathcal{B}_n|^2}{2}\sum_{i=1}^d \frac{4}{x_{ni}} 4\sqrt{(2x_{ni})^3} \\
&= \sum_{n=1}^N 16\sqrt{2}\eta_n|\mathcal{B}_n|^2\sum_{i=1}^d\sqrt{x_{n,i}} \overset{(v)}{\le} \sum_{n=1}^N 16\sqrt{2}\sqrt{md}\eta_n|\mathcal{B}_n|^2.
\end{aligned}$$

where (iv) is due to

$$\begin{aligned}
\mathbb{E}[(\hat{\ell}_{n,i})^2] &= |\mathcal{B}_n|^2\left(x_{n,i}\cdot\left(\frac{1+\frac{o_{n,i}}{|\mathcal{B}_n|}}{x_{n,i}} - 1\right)^2 + (1-x_{n,i})\cdot 1^2\right) \\
&\le |\mathcal{B}_n|^2\left(x_{n,i}\cdot\left(\frac{2}{x_{n,i}} - 1\right)^2 + (1-x_{n,i})\cdot 1^2\right) \\
&\le |\mathcal{B}_n|^2\left(\frac{4}{x_{n,i}} - 3\right) \le |\mathcal{B}_n|^2\frac{4}{x_{n,i}}.
\end{aligned}$$

and (v) applies the Cachy-Schwarz inequaility $\sum_{i=1}^d\sqrt{x_{n,i}} \le \sqrt{md}$ considering $\sum_{i=1}^d x_{n,i} \le m$.

PROOF OF LEMMA 2

We note that $-(1-x)\log(1-x) \le \frac{\sqrt{x}}{2}$ when $x \in [0,1]$. By the definition of $\Psi(x)$ and using the Cauchy-Schwarz inequality, we have

$$0 \le \Psi(x) \le \sum_{i=1}^d \frac{3}{2}\sqrt{x_i} \le \frac{3}{2}\sqrt{md}, \forall x \in \mathcal{X}.$$

Using the definitions of Φ_n, we have

$$\begin{aligned}
\mathbf{REG}^{\text{penalty}} = &\sum_{n=1}^N\Big(-\Phi_n(\hat{L}_n) + \Phi_n(\hat{L}_{n-1}) + \langle x^*, \hat{\ell}_n\rangle\Big) \\
=&\sum_{n=1}^N\Big(-\max_{x\in\mathcal{X}}\Big\{\langle x, \hat{L}_n\rangle - \eta_n^{-1}\Psi(x)\Big\} \\
&+\Big\{\langle x_n, \hat{L}_{n-1}\rangle - \eta_n^{-1}\Psi(x_n)\Big\}\Big) + \sum_{n=1}^N\langle x^*, \hat{\ell}_n\rangle \\
\le &-\langle x^*, \hat{L}_N\rangle + \eta_N^{-1}\Psi(x^*) \\
&-\sum_{n=1}^{N-1}\Big(\langle x_{n+1}, \hat{L}_n\rangle - \eta_n^{-1}\Psi(x_{n+1})\Big) \\
&+\sum_{n=1}^N\Big(\langle x_n, \hat{L}_{n-1}\rangle - \eta_n^{-1}\Psi(x_n)\Big) + \langle x^*, \hat{L}_N\rangle \\
=&\eta_N^{-1}\Psi(x^*) + \sum_{n=2}^N\eta_{n-1}^{-1}\Psi(x_n) - \sum_{n=1}^N\eta_n^{-1}\Psi(x_n) \\
\le&\eta_N^{-1}\Psi(x^*) + \sum_{n=2}^N(\eta_n^{-1} - \eta_{n-1}^{-1})\,(-\Psi(x_n)) \le \eta_N^{-1}\Psi(x^*)
\end{aligned}$$

Plugging the bound on $\Psi(x)$ yields the claimed result.

SAMPLING RULE $P(\cdot)$

For a given $x \in \mathrm{Conv}(\mathcal{X})$, one sampling rule P such that $\mathbb{E}_{X\sim P}[X] = x$ is the following:

We first define the following auxiliary vectors for $0 \le i \le m$, $0 \le j \le d-m$, and we define $\beta_{i,j} \in \mathrm{Conv}(\mathcal{X})$ as:

$$\beta_{i,j} = \left(\underbrace{1,\cdots,1}_{i}, \frac{m-i}{d-i-j}, \cdots, \frac{m-i}{d-i-j}, \underbrace{0,\cdots,0}_{j}\right)$$

It is trivial to sample with mean $\beta_{i,j}$ with the sampling rule:

$$P_{i,j} = \mathrm{Uniform}(\{x \in \mathcal{X} | x_{1,\cdots,i} = \mathbf{1} \wedge x_{d-j+1,\cdots,d} = \mathbf{0}\}).$$

Then we decompose $x = \sum_{s=0}^d p_{x,s}\beta_{i_s,j_s}$ such that $p_{x,s} \in [0,1]$,$\sum_{s=0}^d p_{x,s} = 1$,$(i_0,j_0) = (0,0)$ and $(i_{s+1}, j_{s+1}) - (i_s, j_s) \in \{(1,0),(0,1)\}$. In other words, either i or j increases by one from s to $s+1$. Finally the full sampling scheme is $\sum_{s=0}^d p_{x,s}\beta_{i_s,j_s}$.

When inputting an x, we first create sample s based on the sampling expression $\sum_{s=0}^d p_{x,s}\beta_{i_s,j_s}$, where i in β_{i_s} corresponds to the index of the arm that will definitely be sampled, j corresponds to the index of arms that will not be sampled, and the remaining $i-m$ arms are sampled based on a uniform distribution.

REFERENCES

[1] T. Geng, X. Lin, H. S. Nair, J. Hao, B. Xiang, and S. Fan, "Comparison lift: Bandit-based experimentation system for online advertising," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 17, 2021, pp. 15 117–15 126.

[2] 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*, 2010, pp. 661–670.

[3] S. S. Villar, J. Bowden, and J. Wason, "Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges," *Statistical science: a review journal of the Institute of Mathematical Statistics*, vol. 30, no. 2, p. 199, 2015.

[4] A. Ortiz, A. Asadi, M. Engelhardt, A. Klein, and M. Hollick, "Cbmos: Combinatorial bandit learning for mode selection and resource allocation in d2d systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2225–2238, 2019.

[5] R. Combes, S. Magureanu, and A. Proutiere, "Minimal exploration in structured stochastic bandits," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[6] R. Agrawal, M. Hedge, and D. Teneketzis, "Asymptotically efficient adaptive allocation rules for the multiarmed bandit problem with switching cost," *IEEE Transactions on Automatic Control*, vol. 33, no. 10, pp. 899–906, 1988.

[7] F. Li, J. Liu, and B. Ji, "Combinatorial sleeping bandits with fairness constraints," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1799–1813, 2019.

[8] J. Zimmert, H. Luo, and C.-Y. Wei, "Beating stochastic and adversarial semi-bandits optimally and simultaneously," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7683–7692.

[9] J. Steiger, L. Bin, J. Bo, and N. Lu, "Constrained bandit learning with switching costs for wireless networks," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023.

[10] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2, pp. 235–256, 2002.

[11] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.

[12] S. Bubeck and A. Slivkins, "The best of both worlds: Stochastic and adversarial bandits," in *Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, 2012, pp. 42–1.

[13] Y. Seldin and A. Slivkins, "One practical algorithm for both stochastic and adversarial bandits," in *International Conference on Machine Learning*. PMLR, 2014, pp. 1287–1295.

[14] P. Auer and C.-K. Chiang, "An algorithm with nearly optimal pseudo-regret for both stochastic and adversarial bandits," in *Conference on Learning Theory*. PMLR, 2016, pp. 116–120.

[15] Y. Seldin and G. Lugosi, "An improved parametrization and analysis of the exp3++ algorithm for stochastic and adversarial bandits," in *Conference on Learning Theory*. PMLR, 2017, pp. 1743–1759.

[16] J. Zimmert and Y. Seldin, "An optimal algorithm for stochastic and adversarial bandits," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 467–475.

[17] J.-Y. Audibert, S. Bubeck, and G. Lugosi, "Regret in online combinatorial optimization," *Mathematics of Operations Research*, vol. 39, no. 1, pp. 31–45, 2014.

[18] G. Neu, "First-order regret bounds for combinatorial semi-bandits," *Eprint Arxiv*, 2015.

[19] R. Combes, M. S. Talebi Mazraeh Shahi, A. Proutiere *et al.*, "Combinatorial bandits revisited," *Advances in neural information processing systems*, vol. 28, 2015.

[20] G. Neu and G. Bartók, "An efficient algorithm for learning with semi-bandit feedback," in *International Conference on Algorithmic Learning Theory*, 2013.

[21] C. Y. Wei and H. Luo, "More adaptive algorithms for adversarial bandits," 2018.

[22] O. Dekel, J. Ding, T. Koren, and Y. Peres, "Bandits with switching costs: T 2/3 regret," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, 2014, pp. 459–467.

[23] C. Rouyer, Y. Seldin, and N. Cesa-Bianchi, "An algorithm for stochastic and adversarial bandits with switching costs," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9127–9135.

[24] I. Amir, G. Azov, T. Koren, and R. Livni, "Better best of both worlds bounds for bandits with switching costs," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 800–15 810, 2022.

[25] M. Asawa and D. Teneketzis, "Multi-armed bandits with switching penalties," *IEEE transactions on automatic control*, vol. 41, no. 3, pp. 328–348, 1996.

[26] K. Guo and T. Q. S. Quek, "Dynamic computation offloading in multi-server mec systems: An online learning approach," in *GLOBECOM*, 2020.

[27] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, "An efficiency-boosting client selection scheme for federated learning with fairness guarantee," 2020.

[28] Q. Leng, S. Wang, X. Huang, Z. Shao, and Y. Yang, "Decentralized multi-agent bandit learning for intelligent internet of things systems," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2022, pp. 2118–2123.

[29] X. Gao, X. Huang, Y. Tang, Z. Shao, and Y. Yang, "History-aware online cache placement in fog-assisted iot systems: An integration of learning and control," *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14 683–14 704, 2021.

[30] M. Neely, *Stochastic network optimization with application to communication and queueing systems*. Springer Nature, 2022.

[31] X. Liu, B. Li, P. Shi, and L. Ying, "An efficient pessimistic-optimistic algorithm for stochastic linear bandits with general constraints," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 075–24 086, 2021.

[32] X. Zhou and B. Ji, "On kernelized multi-armed bandits with constraints," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14–26, 2022.

[33] Y. Deng, X. Zhou, A. Ghosh, A. Gupta, and N. B. Shroff, "Interference constrained beam alignment for time-varying channels via kernelized bandits," in *2022 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*. IEEE, 2022, pp. 25–32.

[34] S. Krishnasamy, P. Akhil, A. Arapostathis, R. Sundaresan, and S. Shakkottai, "Augmenting max-weight with explicit learning for wireless scheduling with switching costs," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2501–2514, 2018.

[35] S. Basu and S. Shakkottai, "Switching constrained max-weight scheduling for wireless networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2314–2322.

[36] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, "Convex analysis and optimization," *Pitman Advanced Pub. Program*, 2003.