Batched fixed-confidence pure exploration for bandits with switching constraints

Newton Mwai
Department of Computer Science and Engineering
Chalmers University of Technology
Göteborg, Sweden
mwai@chalmers.se

Milad Malekipirbazari
Department of Computer Science and Engineering
Chalmers University of Technology
Göteborg, Sweden
miladma@chalmers.se

Fredrik D. Johansson
Department of Computer Science and Engineering
Chalmers University of Technology
Göteborg, Sweden
fredrik.johansson@chalmers.se

Abstract

Many studies in multi-armed bandits focus on making exploration quick, often obliviously to constraints tied to exploration like the cost of switching between arms. Switching costs arise in many real-world settings like in healthcare when personalizing treatments, where successive assignment of the same treatment could be necessary for treatment to take effect; or in industrial applications where reconfiguring production is costly. Unfortunately, controlling for switching is significantly understudied outside of regret minimization. In this work, we present a bandit formulation with constraints on the arm switching frequency in fixed-confidence pure exploration and give a lower bound for this setting. We present a batched bandit algorithm called SPB C-Tracking inspired by track-and-stop algorithms, adapted to batch plays with a limited number of arm switches. Finally, we empirically demonstrate that our approach achieves quick stopping times even when constrained to a minimal switching limit.

1 Introduction

Sequential decision-making algorithms promise to improve outcomes in diverse applications, from healthcare to e-commerce, by systematically exploring alternative policies for action. Classically, an effective algorithm strikes a good balance between exploration and exploitation [29, 7, 26], converging as quickly as possible to a good or optimal policy. However, many applications come with costs tied to *switching* actions, and decision-making agents are incentivized to use the same action repeatedly. For example, in healthcare settings such as in clinical trials [4] and treatment personalisation for chronic diseases [22], switching treatments has costs for the patient: every time a treatment is changed, the patient has to get used to the new treatment and its potential side effects. In the personalization of web pages or apps, switching content or interface frequently may be inconveniencing or annoying to users,

38th Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2024).

and in industrial applications, switching actions could mean high costs of reconfiguring production setups. It is therefore desirable to limit the number of switches, even if they make exploration more efficient.

In the multi-armed bandit (MAB) problem, an agent sequentially samples actions from a set of unknown distributions, and it aims to sample (*explore*) them in a manner that helps it to learn about the underlying distributions; either quickly, or with high confidence given an exploration budget (*pure exploration*) [5, 14, 11], or in order to minimize the cumulative cost of choosing sub-optimal actions (*regret minimization*) [31, 12, 25, 27]. Switching in multi-armed bandits has been extensively studied in the regret minimization setting, [3, 9, 30, 2, 28] but it is less studied for pure exploration, possibly because satisfying fixed-confidence correctness is difficult while minimizing the total number of switches. Several works on regret minimization with switching costs use ideas around batching the action selection in time. Although there are works on batched bandits for pure exploration [18, 1, 23, 6], this area is still relatively under-explored.

In this work, we control arm switching in fixed-confidence pure exploration bandits based on a provided constraint on the switching rate. We batch exploration and track the optimal batched arm playing proportions from the setting's lower bound, where we use sparse batch arm playing proportions to attain lower switching. The intuition can be explained as follows: since in a batched bandit exploration is not allowed inside a batch, the arms selected to be played in a batch can be planned at the start of the batch in successive play segments if the number of arms in the sparse proportions are less than the available slots in the batch. The switching then only occurs when changing from one successive arm play segment to the next, or between batches.

Main contributions. 1) We propose a formulation to control the frequency of arm switches in fixed-confidence pure exploration in a batched bandit setting (Section 2). **2)** We provide a lower bound for the search time of any algorithm in this setting while controlling arm switches with a switching constraint (Section 4). **3)** We present a tracking-based algorithm, Sparse-Projected Batch C-Tracking (SPB C-Tracking) (Section 5). **4)** We perform an empirical evaluation and present results showing that our algorithm identifies the best arm quickly with a switching limitation compared to existing batch algorithms. It also compares favorably to classical optimal algorithms in the fixed-confidence setting (Section 6).

2 Problem Formulation

We consider the problem of fixed-confidence, pure exploration multi-armed bandits with a limit on the number of arm switches.

Let $\mathcal{A} = \{1, ..., K\}$ be a set of actions and $\mu_a \in \mathbb{R}$ the expected reward for playing arm $a \in \mathcal{A}$, with $\mu^* = \max_{a \in \mathcal{A}} \mu_a$. A bandit algorithm ϕ plays an arm a_t in successive rounds $t = 1, 2, ...$, before terminating according to a stopping criterion at time τ and recommending the arm $\hat{a}_\tau$. The total number of arm switches S_τ is the number of successive plays where the arms differ, $S_\tau = \sum_{t=2}^{\tau} [a_t \neq a_{t-1}]$. Our goal is to design a search strategy ϕ *to minimize the expected number of trials τ required to identify an optimal action with confidence at least* $1 - \delta$ *for a given* $\delta > 0$*, while limiting the expected rate of switching actions to* $\alpha \in [0, 1]$.

$$\begin{aligned} \underset{\phi}{\text{minimize}} \quad & \mathbb{E}_\phi[\tau] \\ \text{subject to} \quad & \mathbb{P}(\mu_{\hat{a}_\tau} < \mu^*) \leq \delta \\ & \mathbb{E}_\phi[S_\tau] \leq \alpha \mathbb{E}_\phi[\tau] \end{aligned} \tag{1}$$

To control the switching rate, we formulate a *batched* variant of the problem. In batched bandits [10, 15, 18, 6], arm plays are planned in a sequence at the start of a batch, and the rewards for all plays are given at the end of it for the bandit to update its model. Using batches of fixed size B allows us to ensure that all plays for a specific arm are made in sequence within the batches, and the number of switches in a batch is determined by the number of distinct arms in the plan. With this, the total number of switches in exploration will be attributed either to switching between arms *within* the batches when changing from one successive arm play segment to the next, or to changing arms *between* batches.

If all batches include at most s switches, we can bound the expected number of switches by $\mathbb{E}_\phi[S_\tau] \leq \mathbb{E}_\phi[\beta](s+1)-1$; where $\mathbb{E}_\phi[\beta]$ is the expected number of batches played by ϕ before terminating. The bound covers the number of switches within a batch (from one successive arm segment to the next) and the switches from one batch to the next. As a result, the second constraint in our objective above can be satisfied by keeping the switches in the batches low, requiring that the constraint in Eq. (1) holds for the right-hand side of the inequality above, $\forall b : \mathbb{E}_\phi[\beta](S^b+1)-1 \leq \alpha \mathbb{E}_\phi[\tau] = \alpha B\, \mathbb{E}_\phi[\beta]$. If this holds, with $\lfloor \cdot \rfloor$ the floor operator,

$$\forall b : S^b \leq s \coloneqq \lfloor \alpha B - 1 \rfloor \implies \mathbb{E}_\phi[S_\tau] \leq \alpha \mathbb{E}_\phi[\tau]\ . \tag{2}$$

We can now re-formulate our goal to be *to minimize the expected number of batches β required to identify an optimal action, with confidence at least $1-\delta$, while limiting the arm switches within the batch to be at most a pre-specified switching constraint $s \in \{0, ..., \min(K-1, B-1)\}$*,

$$\begin{aligned} \underset{\phi}{\text{minimize}} \quad & \mathbb{E}_\phi[\beta] \\ \text{subject to} \quad & \mathbb{P}\left(\mu_{\hat{a}_\beta} < \mu^*\right) \leq \delta \\ & S^b \leq s,\ \forall b \end{aligned} \tag{3}$$

In this work, all batches are planned deterministically. Randomized algorithms could achieve a non-integer *expected* number of switches but we do not explore that here.

3 Related Work

Multi-armed bandits with switching costs have been studied widely in the regret minimization setting [3, 9, 30, 2, 28], and analyses typically focus on regret bounds in both stochastic and adversarial settings. Recently, algorithms based on variations of the Tsallis-INF and EXP3 bandit algorithms have been presented. A key idea in these studies is the use of *blocks/batches* to control the frequency of arm switching [3, 30, 2] and we use this idea in the pure exploration setting, where studies of controlling switching in exploration are scarce.

Batched bandits are a setting where arms are planned in batches, and played as planned before rewards are given for the whole batch at once. The arms to be sampled in a batch are determined by the arms and rewards from previous batches. This setting has been studied widely in the regret minimization setting [10, 15, 16, 13, 19, 20] with the focus being how many batches are required to attain the optimal cumulative regret; using either static batch sizes or adaptive batch sizes. In pure exploration, a few studies in batched bandits exist [18, 1, 23, 6]. All of these works focus on arm elimination strategies, which are different in nature to our algorithm, which focuses on tracking optimal arm playing proportions in batches.

Tracking proportions is ubiquitous in the fixed-confidence pure exploration setting since the idea of Track-and-Stop algorithms was introduced by Garivier and Kaufmann [11], along with optimal instance-dependent asymptotic regime results. Our work focuses on how these results can be used in the batch setting with switching constraints. In this regard, Jourdan et al. [17] presented an interesting setting in pure exploration for combinatorial bandits with semi-bandit feedback which combines ideas from Garivier and Kaufmann [11], Degenne et al. [8] and the combinatorial bandits literature. In our work, since the batch size is static and the switching limit in a batch is provided, we can derive a related combinatorial bandit setting for the arm plays in batches. The difference in our setting relates to the successive play segments allowed to enable minimizing of switching, as well as our goal of identifying the best arm compared to a super arm.

4 Theoretical Discussion

For the fixed-confidence pure exploration setting, [11] presented a general lower bound for the expected stopping time $\mathbb{E}[\tau]$ of any algorithm returning the best arm with probability at least $1-\delta$, for some $\delta > 0$,

$$\mathbb{E}[\tau] \geq T^*(\mu)\ \mathrm{kl}(\delta, 1-\delta)\ . \tag{4}$$

$$\text{where } T^*(\mu)^{-1} := \sup_{w \in \Delta_1^K} \inf_{\lambda \in \mathrm{Alt}(\mu)} \left(\sum_{a=1}^{K} w_a d(\mu_a, \lambda_a) \right).$$

Here, $d(.)$ is the KL-divergence, and $\Delta_1^K := \{w \in_+^K : w_1 + \cdots + w_K = 1\}$ the simplex of possible arm playing proportions. This lower bound is derived by considering the optimal allocation of arm pulls w^* to minimize the worst-case instance-specific stopping time while ensuring that the probability of incorrectly identifying the best arm does not exceed a pre-specified confidence level δ. The term $T^*(\mu)$ represents the inverse of the optimal allocation of arm draws, characterized by the supremum over all possible allocation vectors w and the infimum over all alternative bandit models λ (that differ from μ in their optimal arm). This formulation captures the inherent difficulty of the best arm identification problem under the fixed-confidence setting, ensuring that any optimal strategy will asymptotically match this lower bound as δ approaches zero.

Garivier and Kaufmann [11] also introduced the idea of *track-and-stop* algorithms, which aim to *track* the optimal arm playing proportions, $w^*(\mu)$, matching this lower bound,

$$w^*(\mu) := \arg\max_{w \in \Delta_1^K} \inf_{\lambda \in \mathrm{Alt}(\mu)} \left(\sum_{a=1}^{K} w_a d(\mu_a, \lambda_a) \right) \tag{5}$$

An algorithm plays the arms following a *tracking rule* aiming for an overall arm proportion as close to the optimal proportions as possible. The *stop* in *track-and-stop* is because an algorithm also combines its tracking rule with a *stopping rule* which typically does a satistical test of whether the past observations allow assessing that, with a risk of at most δ, one arm is larger than the others. The Parallel Generalized Likelihood Ratio Test (GLRT) has become a standard stopping rule due to its ability to exploit the geometry of the distributions better, hence earlier stopping.

4.1 Lower Bound on $\mathbb{E}[\beta]$ with Batch Plays

To start with, we show that if we consider a batched horizon, we can obtain a lower bound on the expected number of batches β necessary for any δ-probably correct algorithm, following the reasoning in the lower bound from Garivier and Kaufmann [11] (Eq. (4)) as:

$$\mathbb{E}[\beta] \geq T_b^*(\mu)\ \mathrm{kl}(\delta, 1-\delta) \tag{6}$$

$$\text{where } T_b^*(\mu)^{-1} := 1/B \sup_{w \in \Delta_1^K} \inf_{\lambda \in \mathrm{Alt}(\mu)} \left(\sum_{a=1}^{K} \lceil w_a B \rceil \cdot d(\mu_a, \lambda_a) \right).$$

We get this by applying the transportation lemma of Kaufmann et al. [21] with

$$\sum_{a=1}^{K} \mathbb{E}_\mu[N_a(\beta)] \cdot d(\mu_a, \lambda_a) \geq \ \mathrm{kl}(\delta, 1-\delta)$$

for any alternative bandit instance $\lambda \in \mathrm{Alt}(\mu) = \{\lambda \in \mathbb{R}^K : \arg\max_i \lambda_i \neq \arg\max_j \mu_j\}$. Here, $N_a(\beta)$ is the total plays of arm a up to batch β. Combining the inequalities given by all alternatives λ_a yields a bound,

$$\begin{aligned}
\mathrm{kl}(\delta, 1-\delta) &\leq \inf_{\lambda \in \mathrm{Alt}(\mu)} \sum_{a=1}^{K} \mathbb{E}_\mu[N_a(\beta)] \cdot d(\mu_a, \lambda_a) \\
&= B \cdot \mathbb{E}_\mu[\beta] \inf_{\lambda \in \mathrm{Alt}(\mu)} \sum_{a=1}^{K} \frac{\mathbb{E}_\mu[N_a(\beta)]}{B \cdot \mathbb{E}_\mu[\beta]} \cdot d(\mu_a, \lambda_a) \\
&\leq 1/B\ \mathbb{E}_\mu[\beta] \sup_{w \in \Delta_1^K} \inf_{\lambda \in \mathrm{Alt}(\mu)} \sum_{a=1}^{K} \lceil w_a B \rceil \cdot d(\mu_a, \lambda_a)
\end{aligned}$$

The maximum rounding error from the ceiling operation above is an additional KB batches. Finally, replacing the strategy-dependent proportions of arm draws by their supremum yields the lower bound.

4.2 Minimizing Switching: Lower Bound with Sparsity Constraint in Batch Plays

In our setting, we aim to restrict switching costs associated with changing arms in a batched multi-armed bandit framework by limiting the number of switches in each batch to at most s. In a track-and-stop framework, this imposes a constraint on the number of non-zero arm proportions allowed in a batch. The batch switch limit inherently means that some arms will have a play count of zero in certain batches. For this, we cannot directly apply [11]'s analysis and algorithm per batch, since having arms with zero play proportion can not be the solution to the min-max problem. If an arm a is never played, $w_a = 0$, the adversary λ can exploit this and differ arbitrarily for that arm. This is also evident from Lemma 4 in Garivier and Kaufmann [11] which would be violated if proportions were 0.

By incorporating the arm sparsity constraint, we introduce a new dimension to the problem where planning the plays over successive batches becomes crucial. In other words, when all batches are done, we want to end up having played according to the minimizer of Eq. (5), but we can't play according to these proportions every batch. Arms with non-zero proportions must be scheduled in such a way that minimizes the overall switching cost across batches, thereby optimizing the exploration process while respecting the given constraints. To do this, we will study the optimal proportions of played *batch configurations* instead.

Given that the batch size is fixed and known, we can determine the sparsity-constrained integer playing configurations c of arm plays in the batch, ensuring that the total number of plays matches the desired sparsity. With a given batch switch limit s, we introduce the available integer playing configurations for plays corresponding to the desired sparsity, $\mathcal{C}^K_{B,s}$:

$$\mathcal{C}^K_{B,s} := \left\{ c \in \mathbb{N}^K : \sum_{a=1}^{K} c_a = B \text{ and } \|c\|_0 \leq s+1 \right\} \tag{7}$$

Here, $\|\cdot\|_0$ denotes the l_0-norm, which counts the number of nonzero elements in the vector. Each element c of $\mathcal{C}^K_{B,s}$ represents a planning of arms plays that can be executed in a batch of size B.

From the above definitions, considering $c_{a,b}$ as the number of plays of arm a in the feasible combination of plays in batch b, and considering $p_{c,b}$ as the probability of configuration c occurring in batch b, the total plays up to batch β can be expressed as:

$$N(\beta) = \sum_{a=1}^{K} \sum_{b=1}^{\beta} \sum_{c \in \mathcal{C}^K_{B,s}} p_{c,b} c_{a,b} \ ,$$

and the total plays of arm $a \in \mathcal{A}$ up to batch β can be given as:

$$N_a(\beta) = \sum_{b=1}^{\beta} \sum_{c \in \mathcal{C}^K_{B,s}} p_{c,b} c_{a,b} \ .$$

Based on the definition of batch configurations given in Eq. (7), we now derive a lower bound for batch plays with sparsity constraints:

Theorem 1. *Let $\Delta^{\mathcal{C}} := \Delta_1^{|\mathcal{C}^K_{B,s}|}$ be the simplex over sparse batch configurations. Given $\delta \in (0,1)$, for any strategy that is returns the best arm with probability at least $1-\delta$, and for any bandit model $\mu \in \mathcal{S}$, the following inequality holds:*

$$\mathbb{E}_\mu[\beta] \geq T^*_{bc}(\mu) \cdot \mathrm{kl}(\delta, 1-\delta), \tag{8}$$

where

$$T^*_{bc}(\mu)^{-1} := \sup_{p \in \Delta^{\mathcal{C}}} \inf_{\lambda \in \mathrm{Alt}(\mu)} \sum_{a=1}^{K} \sum_{c \in \mathcal{C}^K_{B,s}} p_c c_a d(\mu_a, \lambda_a). \tag{9}$$

In Eq. (9), the supremum is computed over the possible probabilities of choosing the sparse configurations, thereby incorporating the switch limit into the batch play optimization.

Although shared in form, our lower bound differs from the result in [11] in several key ways. First, we incorporate a switching constraint that limits the number of non-zero proportions within each batch, unlike Garivier and Kaufmann's method, which assumes no switching costs. Second, we use a batching strategy with sparsity constraints, meaning only a limited number of arms are played in each batch, whereas their method does not involve such batching or switch constraints. Finally, the playing proportions in our result are defined for entire batch configurations $\mathcal{C}_{B,s}^K$ of integer arm plays, a unique aspect not present in the fixed-confidence setting discussed in [11].

Our approach to combining batch plays with sparsity constraints is related to the combinatorial bandits discussed in Jourdan et al. [17]. In their work, they explore combinatorial bandits with semi-bandit feedback, which involves making decisions over combinations of arms. Similarly, our method involves managing combinations of arm plays within each batch, taking into account the sparsity constraint to limit switching between arms.

The combination set $\mathcal{C}_{B,s}^K$ introduced in our method is high-dimensional, making it challenging to deal with in practice. More particularly, for the case of at most s batch switches, we have

$$|\mathcal{C}_{B,s}^K| = \sum_{i=0}^{s} \binom{K}{i+1}\binom{B-1}{i}$$

and for the case of exactly s batch switches we have

$$|\mathcal{C}_{B,s}^K| = \binom{K}{s+1}\binom{B-1}{s}.$$

Addressing this high-dimensionality and optimizing the design of algorithms that efficiently handle these combinations remain areas for future work.

5 Algorithms

This section presents the core algorithm introduced in our study, Sparse-Projected Batch C-Tracking (SPB C-Tracking), which addresses the challenge of minimizing switching costs in a batched setting for pure exploration bandits. We include its pseudocode and a detailed explanation of its operation.

Algorithm 1 outlines the steps involved in this process. The SPB C-Tracking algorithm begins with initializing parameters, including the arm estimates, batch size, and batch switch limit. In each iteration, the algorithm computes the optimal arm proportions $w^*(\hat{\mu}_{b-1})$ using the current mean estimates with respect to the optimization problem given in Eq. (5). Then, following the original C-tracking algorithm [11], the algorithm computes $w^\epsilon(\hat{\mu}_{b-1})$ by projecting the optimal proportions onto the L^∞-bounded simplex $\Delta_{1,\epsilon}^K$, where

$$\Delta_{1,\epsilon}^K := \{(w_1, ..., w_K) \in [\epsilon, 1] : w_1 + ... + w_K = 1\}.$$

This step ensures that any arm receives a proportion of at least ϵ and that the proportions sum to one. Next, we project the C-tracking criterion $B\sum_{i=0}^{b-1} w^{\epsilon_i}(\hat{\mu}_i) - N$ onto a constrained set $\sum_s \cap \Delta_B^K$, ensuring that the number of non-zero proportions is limited by the batch switch limit s, where

$$\sum\nolimits_s = \left\{w \in \mathbb{R}^K : \|w\|_0 \le s+1\right\} \text{ and } \Delta_B^K = \left\{w \in \mathbb{R}_+^K : w_1 + \cdots + w_K = B\right\}.$$

This projection minimizes the L^2 norm difference between the projected proportions and the scaled sum of previous proportions and the current play counts. The projection method used is equivalent to the one proposed by Kyrillidis et al. [24], which employs a simple greedy algorithm for exact sparse projection. The resulting sparse-projected proportions $w^s(\hat{\mu}_{b-1})$ are then converted to integers $\bar{w}^s(\hat{\mu}_{b-1})$, which represent the number of times each arm will be played in the batch. Within each batch, the algorithm selects the arm a_t with the highest projected proportion and plays the selected arm ω_t times and observes the rewards r_t to $r_{t+\omega_t-1}$. The rewards are observed, and the arm estimates are updated accordingly. The process repeats until a criterion based on a confidence threshold is met (similar to Chernoff's stopping rule presented in [11]), ultimately returning the best arm identified.

6 Simulation Experiments

In our experiments, we're foremost interested in investigating if we can achieve a low stopping time $\mathbb{E}[\tau]$ while observing a minimal switching limit s during exploration. We also investigate the effects

Algorithm 1 Sparse-Projected Batch C-Tracking (SPB C-Tracking)

```
**Input** $K$ arms, $\delta \in (0,1), B$ : batch size, $s$ : batch switch limit
**Output** $\beta$, $\hat{a}_\beta$

$b \leftarrow 1$, $t \leftarrow 1$, $Z_1 \leftarrow 0$, $\hat{\mu}_0 = \mathbf{0}$, and $N_a \leftarrow 0$ for all $a \in \mathcal{A}$
while $Z_b \leq \log(\frac{\log(t)+1}{\delta})$ do
    Compute $w^*(\hat{\mu}_{b-1})$                                    ▷ See Eq. (5)
    Compute $w^{\epsilon_{b-1}}(\hat{\mu}_{b-1})$          ▷ $L^\infty$ proj. of $w^*(\hat{\mu}_{b-1})$ on to $\Delta^K_{1,\epsilon_{b-1}}$with $\epsilon_{b-1} = (K^2 + (b-1)B)^{-1/2}/2$
    Compute $w^s(\hat{\mu}_{b-1}) \in \arg\min_{w: w \in \sum_s \cap \Delta^K_B} \left\| w - B \sum_{i=0}^{b-1} w^{\epsilon_i}(\hat{\mu}_i) + N \right\|_2$
    Compute $\bar{w}^s(\hat{\mu}_{b-1}) = \text{integer}(w^s(\hat{\mu}_{b-1}))$
    while $t \leq bB$ do
        $a_t = \arg\max_{a \in \mathcal{A}} \; \bar{w}^s_a(\hat{\mu}_{b-1})$
        $\omega_t = \max_{a \in \mathcal{A}} \; \bar{w}^s_a(\hat{\mu}_{b-1})$
        Play $a_t$ $\omega_t$ times, and Observe $r_t$ to $r_{t+\omega_t-1}$
        Update $N_{a_t} \leftarrow N_{a_t} + \omega_t$ and $\hat{\mu}_{b,a_t} = \hat{\mu}_{b-1,a_t} + \frac{1}{N_{a_t}} \left( \sum_{j=t}^{t+\omega_t-1} r_j - \omega_t \hat{\mu}_{b-1,a_t} \right)$
        Update $\bar{w}^s_{a_t}(\hat{\mu}_{b-1}) \leftarrow 0$
        Update $t \leftarrow t + \omega_t$
    end while
    Update $b \leftarrow b + 1$
    Compute $Z_b$
end while

Return $\hat{a}_\beta$
```

of different combinations of s and the batch size B on $\mathbb{E}[\tau]$. As discussed in Section 2, s and B are inherently tied from $s := \lfloor \alpha B - 1 \rfloor$. For small batch sizes, the choice of s will have a large impact on which arm configurations can be played in a batch, and therefore on the nature of exploration. We aim to validate this empirically.

We compare our SPB C-Tracking algorithm to the BatchRacing algorithm [18]. Batch Racing is a Batched Racing algorithm that successively eliminates arms across the rounds, starting with the full set of available arms as a surviving set $S_1 = \mathcal{A}$. In each round of batches of size B, it uses a RoundRobin algorithm to determine plays in the batch uniformly. It is designed for top-k identification, to which end our setting lies in top-1. In each round, BatchRacing uses a UCB (resp LCB) to determine if there is any arm that is confidently top-k (resp not), and moves the arms to an accepted, A_t (resp rejected, R_t) set. The arms moved to the accepted or rejected sets are removed from the surviving sets and the process repeats until k arms are accepted, whereby it terminates and outputs the accepted set A_τ. In our setting $|A_\tau| = 1$.

6.1 Experimental setup

Our experimental simulation setting comprises a set of 8 arms from Gaussian distributions, with means $\mu = \{0.8, 0.65, 0.6, 0.55, 0.5, 0.45, 0.4, 0.35\}$ and standard deviation $\sigma = 1$. The algorithms (SPB C-Tracking, BatchRacing and Track and Stop C-Tracking [11]) are run with $\delta = 0.01$, and a pull limit of 20,000. We run experiments to investigate the effect of varying a switching limit s in SPB C-Tracking. In these, we *set* the number of switches $S^b = s$ for all batches b, rather than let them be bounded by s from above. We also compare the effect of the batch size in SPB C-Tracking and the BatchRacing baseline. We use the stopping time, $\mathbb{E}[\tau]$, as the evaluation metric and we additionally compare the stopping times of the batched algorithms to the stopping time of the un-batched Track and Stop C-Tracking. All experiments are done for 300 repetitions for each combination of batch size and switching limit.

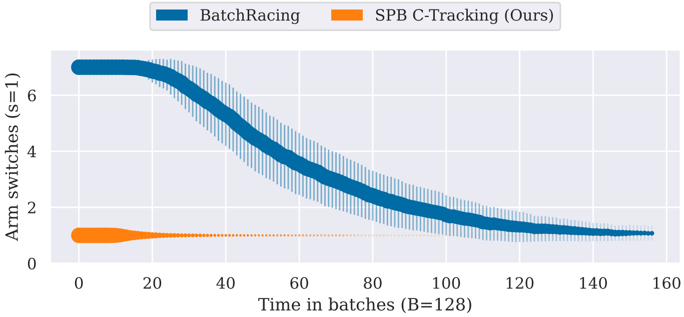

Figure 1: Comparison trace of switches along time in batches for SPB C-Tracking (Ours) vs BatchRacing (Baseline)

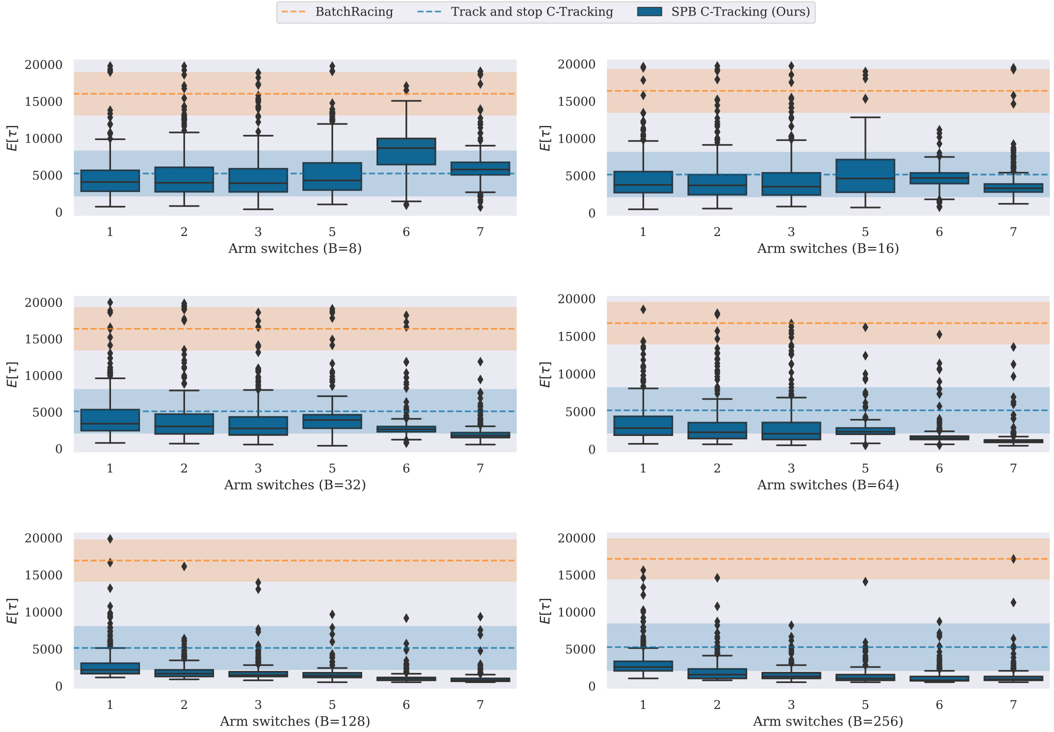

Figure 2: Comparison of stopping times $\mathbb{E}[\tau]$ over a range of switching limits s in SPB C-Tracking (Ours) with different batch sizes B. Track-and-stop C-tracking is not batched.

6.2 Results and discussion

In Figure 1, we see that our algorithm achieves faster stopping, as shown by the shorter batch horizons over the repeated runs, even when constrained to a minimal switching limit ($s = 1$) across the batch horizon. The switches behaviour of the BatchRacing algorithm shows that it starts with the maximum possible switches in early batches and eventually decreases to $s = 1$ in later batches. Successive elimination algorithms, including BatchRacing, which comprise the bulk of the limited work in batched bandits in pure exploration [18, 1, 23, 6] will always exhibit this switching behaviour. These algorithms are expected to have a high number of switches during exploration, as they always start with the whole set of arms as the feasible exploration set. This validates our approach of tracking the optimal arm-playing proportions, as it enables the computation of sparse proportions that yield a solution (SPB C-Tracking) for limiting switches during exploration.

The stopping times in Figure 2 show that as expected, SPB C-Tracking always outperforms the BatchRacing baseline. This can be explained by the algorithm's characteristic of tracking the optimal playing proportions from the lower bound. This is also visible from the stopping times of the similarly optimal-proportions-designed Track and Stop C-Tracking, to which SPB C-Tracking is fairly matched. A keener look reveals that SPB C-Tracking seems to perform slightly better than Track and Stop C-Tracking with higher batch sizes. This could be explained by our forced exploration (playing exactly $s+1$ arms) being better in larger batch sizes. For smaller batch sizes, this forced exploration appears to be costlier in exploration (e.g. see in $B=8, s=6$) likely due to playing arms that do not add value to the exploration utility. For the larger batch sizes ($B \geq 32$), it is also visible that the stopping times of SPB C-Tracking decrease as the switching limit is relaxed and more arms are explored in the batch—the algorithm is able to get information from more arms and hence learn more about their distributions. However, when batch sizes get too large, stopping times increase again (see Appendix Figure 3), most likely due to exploration being limited.

7 Conclusion

In this work, we presented a formulation to control arm switching frequency in fixed confidence pure exploration, and showed that it is possible to stop quicker even when constrained to a minimal switching limit, s. We present a batched algorithm that empirically demonstrates this, the SPB C-Tracking, that is designed to track the optimal playing proportions. Our algorithm achieves minimal switching via use of sparse arm playing proportions computed through a sparse simplex projection. A limitation of our approach is that the algorithm always plays exactly $s+1$ arms which is wasteful of exploration, evident in stopping times with lower batch sizes with $s \geq 5$. This will be revised in future work with s as an upper bound. Additionally, future work will focus on providing sample complexity results for SPB C-Tracking, as well as analyzing this problem in a combinatorial pure exploration bandit setting for the arm plays in batches analogous to Jourdan et al. [17]. This will be a step towards an algorithm that more explicitly aligns with the combinatorial sparse configurations lower bound in Eq. (8) as well as relieves the computation of the expensive optimization problem required for the optimal playing proportions $w^*(\hat{\mu}_{b-1})$.

References

[1] Arpit Agarwal, Shivani Agarwal, Sepehr Assadi, and Sanjeev Khanna. Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons. In *Conference on Learning Theory*, pages 39–75. PMLR, 2017.

[2] Idan Amir, Guy Azov, Tomer Koren, and Roi Livni. Better best of both worlds bounds for bandits with switching costs. *Advances in neural information processing systems*, 35:15800–15810, 2022.

[3] Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. *arXiv preprint arXiv:1206.6400*, 2012.

[4] Maryam Aziz, Emilie Kaufmann, and Marie-Karelle Riviere. On multi-armed bandit designs for dose-finding trials. *Journal of Machine Learning Research*, 22(14):1–38, 2021.

[5] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Algorithmic Learning Theory: 20th International Conference, ALT 2009, Porto, Portugal, October 3-5, 2009. Proceedings 20*, pages 23–37. Springer, 2009.

[6] Shengyu Cao, Simai He, Ruoqing Jiang, Jin Xu, and Hongsong Yuan. Best arm identification in batched multi-armed bandit problems. *arXiv preprint arXiv:2312.13875*, 2023.

[7] Herman Chernoff. Sequential design of experiments. *The Annals of Mathematical Statistics*, 30 (3):755–770, 1959. ISSN 00034851.

[8] Rémy Degenne, Wouter M Koolen, and Pierre Ménard. Non-asymptotic pure exploration by solving games. *Advances in Neural Information Processing Systems*, 32, 2019.

[9] Ofer Dekel, Jian Ding, Tomer Koren, and Yuval Peres. Bandits with switching costs: T 2/3 regret. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 459–467, 2014.

[10] Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. Batched multi-armed bandits problem. *Advances in Neural Information Processing Systems*, 32, 2019.

[11] Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 998–1027, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

[12] J. Gittens and Michael Dempster. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B: Methodological*, 41:148–177, 02 1979.

[13] Ben Hambly, Renyuan Xu, and Huining Yang. Recent advances in reinforcement learning in finance. *Mathematical Finance*, 33(3):437–503, 2023.

[14] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pages 423–439. PMLR, 2014.

[15] Tianyuan Jin, Jing Tang, Pan Xu, Keke Huang, Xiaokui Xiao, and Quanquan Gu. Almost optimal anytime algorithm for batched multi-armed bandits. In *International Conference on Machine Learning*, pages 5065–5073. PMLR, 2021.

[16] Tianyuan Jin, Pan Xu, Xiaokui Xiao, and Quanquan Gu. Double explore-then-commit: Asymptotic optimality and beyond. In *Conference on Learning Theory*, pages 2584–2633. PMLR, 2021.

[17] Marc Jourdan, Mojmír Mutnỳ, Johannes Kirschner, and Andreas Krause. Efficient pure exploration for combinatorial bandits with semi-bandit feedback. In *Algorithmic Learning Theory*, pages 805–849. PMLR, 2021.

[18] Kwang-Sung Jun, Kevin Jamieson, Robert Nowak, and Xiaojin Zhu. Top arm identification in multi-armed bandits with batch arm pulls. In *Artificial Intelligence and Statistics*, pages 139–148. PMLR, 2016.

[19] Cem Kalkanli and Ayfer Ozgur. Batched thompson sampling. *Advances in Neural Information Processing Systems*, 34:29984–29994, 2021.

[20] Cem Kalkanlı and Ayfer Özgür. Asymptotic performance of thompson sampling for batched multi-armed bandits. *IEEE Transactions on Information Theory*, 2023.

[21] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17:1–42, 2016.

[22] Newton Mwai Kinyanjui, Emil Carlsson, and Fredrik D Johansson. Fast treatment personalization with latent bandits in fixed-confidence pure exploration. *Transactions on Machine Learning Research*, 2023.

[23] Junpei Komiyama, Kaito Ariu, Masahiro Kato, and Chao Qin. Optimal simple regret in bayesian best arm identification. *arXiv preprint arXiv:2111.09885*, 2021.

[24] Anastasios Kyrillidis, Stephen Becker, Volkan Cevher, and Christoph Koch. Sparse projections onto the simplex. In *International Conference on Machine Learning*, pages 235–243. PMLR, 2013.

[25] T.L Lai and H Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

[26] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms.* Cambridge University Press, 2020.

[27] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[28] Yingying Li, James A Preiss, Na Li, Yiheng Lin, Adam Wierman, and Jeff S Shamma. Online switching control with stability and regret guarantees. In *Learning for Dynamics and Control Conference*, pages 1138–1151. PMLR, 2023.

[29] Herbert Robbins. Some aspects of the sequential design of experiments. 1952.

[30] Chloé Rouyer, Yevgeny Seldin, and Nicolo Cesa-Bianchi. An algorithm for stochastic and adversarial bandits with switching costs. In *International Conference on Machine Learning*, pages 9127–9135. PMLR, 2021.

[31] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. ISSN 00063444.

A Appendix: Proof of Theorem 1

Proof of Theorem 1: Consider $\delta \in (0, 1)$ and a bandit model μ in $\mathcal{S}$, along with a δ-PAC strategy. For each block $b \geq 1$, let $N_a(b)$ represent the number of times arm a is drawn up to the end of block b. According to Garivier and Kaufmann [11, Lemma 1], the expected number of draws for each arm and the Kullback-Leibler divergence between two bandit models with distinct optimal arms are related to the error probability δ:

$$\forall \lambda \in \mathcal{S} \ : \ a^*(\lambda) \neq a^*(\mu), \ \sum_{a=1}^{K} \mathbb{E}_\nu[N_a(\beta)] d(\mu_a, \lambda_a) \geq \mathrm{kl}(\delta, 1-\delta).$$

Rather than selecting a specific λ for each arm a to provide a lower bound on $\mathbb{E}_\mu[\beta]$, we integrate the inequalities from all alternative λs:

$$\mathrm{kl}(\delta, 1-\delta) \leq \inf_{\lambda \in \mathrm{Alt}(\mu)} \sum_{a=1}^{K} \mathbb{E}_\mu[N_a(\beta)] d(\mu_a, \lambda_a)$$

Let $\mathcal{C}^K_{B,s}$ be the available integer playing configurations for plays corresponding to the desired sparsity and $\Delta^{\mathcal{C}} := \Delta_1^{|\mathcal{C}^K_{B,s}|}$ be the simplex over sparse batch configurations. Then, we have

$$\begin{aligned}
\mathrm{kl}(\delta, 1-\delta) &\leq \inf_{\lambda \in \mathrm{Alt}(\mu)} \sum_{a=1}^{K} \mathbb{E}_\mu \left[\sum_{b=1}^{\beta} \sum_{c \in \mathcal{C}^K_{B,s}} c_{a,b} \right] d(\mu_a, \lambda_a) \\
&= \inf_{\lambda \in \mathrm{Alt}(\mu)} \sum_{a=1}^{K} \mathbb{E}_\mu[\beta] \mathbb{E}_\mu \left[\sum_{c \in \mathcal{C}^K_{B,s}} c_a \right] d(\mu_a, \lambda_a) \\
&\leq \mathbb{E}_\mu[\beta] \sup_{p \in \Delta^{\mathcal{C}}} \inf_{\lambda \in \mathrm{Alt}(\mu)} \sum_{a=1}^{K} \sum_{c \in \mathcal{C}^K_{B,s}} p_c c_a \, d(\mu_a, \lambda_a),
\end{aligned}$$

where the last inequality arises because the probabilities of arm draws specific to each batch are less than or equal to their maximum values. This substitution is made to derive a bound that applies to any δ-PAC algorithm. ■

B Appendix: Additional experimental results

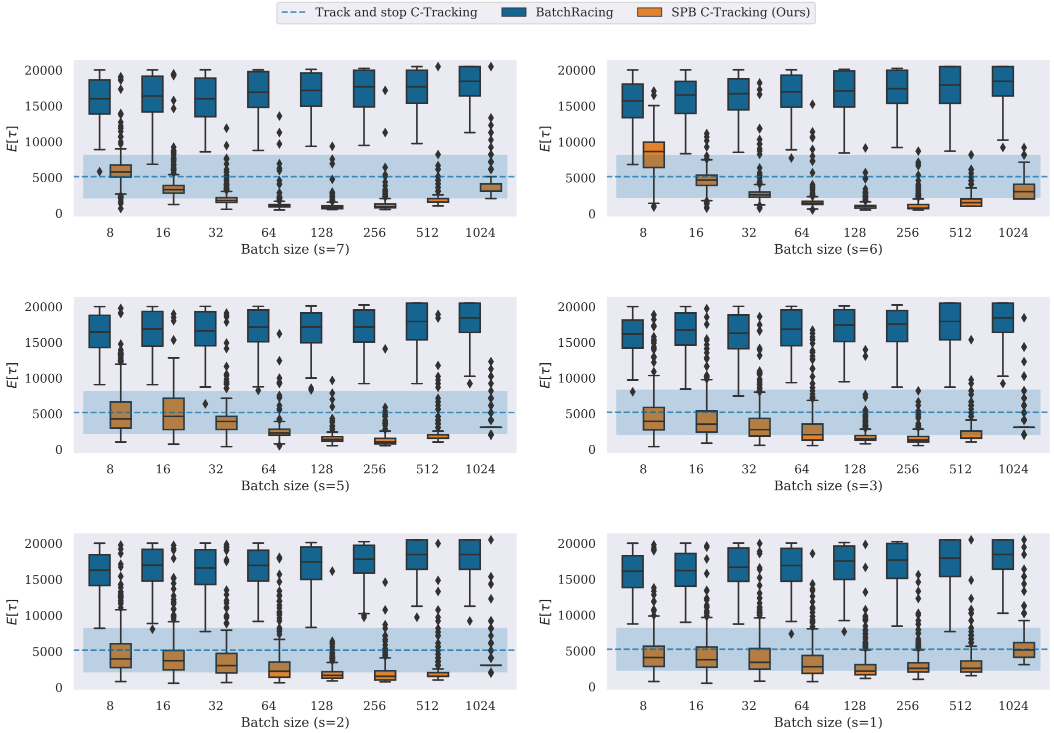

Figure 3: Effect of batch size on the stopping times for BatchRacing and SPB C-Tracking