Scaling Multi-Armed Bandit Algorithms

Edouard Fouché
Karlsruhe Institute of Technology
edouard.fouche@kit.edu

Junpei Komiyama
University of Tokyo
junpei@komiyama.info

Klemens Böhm
Karlsruhe Institute of Technology
klemens.boehm@kit.edu

ABSTRACT

The Multi-Armed Bandit (MAB) is a fundamental model capturing the dilemma between exploration and exploitation in sequential decision making. At every time step, the decision maker selects a set of arms and observes a reward from each of the chosen arms. In this paper, we present a variant of the problem, which we call the Scaling MAB (S-MAB): The goal of the decision maker is not only to maximize the cumulative rewards, i.e., choosing the arms with the highest expected reward, but also to decide how many arms to select so that, in expectation, the cost of selecting arms does not exceed the rewards. This problem is relevant to many real-world applications, e.g., online advertising, financial investments or data stream monitoring. We propose an extension of Thompson Sampling, which has strong theoretical guarantees and is reported to perform well in practice. Our extension dynamically controls the number of arms to draw. Furthermore, we combine the proposed method with ADWIN, a state-of-the-art change detector, to deal with non-static environments. We illustrate the benefits of our contribution via a real-world use case on predictive maintenance.

CCS CONCEPTS

• **Computing methodologies** → **Sequential decision making**; • **Theory of computation** → *Online learning algorithms*; • **Information systems** → Data streams; Data analytics.

KEYWORDS

Bandit Algorithms; Thompson Sampling; Adaptive Windowing; Data Stream Monitoring; Predictive Maintenance

ACM Reference Format:
Edouard Fouché, Junpei Komiyama, and Klemens Böhm. 2019. Scaling Multi-Armed Bandit Algorithms. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3292500.3330862

1 INTRODUCTION

1.1 Motivation

In the classical Multi-Armed Bandit (MAB) problem, a forecaster must choose one of K arms at each round, and playing it yields a reward. Her goal is to maximize the cumulative reward over time.

In a generalization of the problem, known as the Multiple-Play Multi-Armed Bandit (MP-MAB) [3, 20], the forecaster must choose L distinct arms per round, where L is an exogenous parameter.

KDD '19, August 4–8, 2019, Anchorage, AK, USA

While in some applications, such as web content optimization, L is given, there are many applications where an appropriate value of L is not obvious. We consider a new variant of the MP-MAB where the forecaster not only must choose the best arms, but also must "scale" L, i.e., change the number of plays, to maximize the reward and minimize the cost at any time. By doing so, the forecaster controls the efficiency of each observation. We name this setting the Scaling Multi-Armed Bandit (S-MAB) problem.

Think of a new casino game, which we call the "blind roulette": The player places bets on distinct numbers, and each number has an independent but unknown probability to be drawn. Bets are set to a fixed amount, e.g., one can only bet 1$ on a number, or nothing. In each round, the player must decide how many bets to place, and on which numbers. The casino then reveals to the player which ones of her bets were successful and pays the corresponding reward. To make the game more challenging, the casino may sometimes change the underlying probability of each number without notice.

While placing a few but confident bets may seem to be an economically efficient option, the absolute gain at the end of the day will not be large. On the other hand, placing many bets may not be a good strategy, as many numbers typically have a low chance to be drawn. To maximize her gain, the player must place as many bets as possible, as long as her expected gain is greater than the amount bet. Whenever the probabilities change, the player needs to adapt her behaviour, otherwise she may loose most of her bets and experience high regret w.r.t. an optimal (but unknown) strategy.

This game matches many real-world applications, e.g., the placement of online advertisements, investment in financial portfolios, or data stream monitoring. We elaborate on the latter using an example from predictive maintenance, our running example:

Example 1 (Correlation Monitoring). *Correlation often results from physical relationships between, say, the temperature and pressure of a fluid. When correlation changes, this means that the system is transitioning into another state, e.g., the fluid solidifies, or that equipment deteriorates or fails, e.g., there is a leak. When monitoring large factories, it is useful to maintain an overview of correlations to keep operation costs down. However, updating the complete correlation matrix continuously is impractical with current methods, since the data is typically high-dimensional and ever evolving. A more efficient solution consists in updating only a few elements of the matrix, based on a notion of utility, e.g., high correlation values. The system must minimize the cost of monitoring while maximizing the total utility, in a possibly non-static environment.*

Thus, the S-MAB problem introduces an additional trade-off: One wants to maximize the reward, but at the same time minimize the cost of each round/observation. The challenge here is threefold:

C1: Top-arms Identification. To maximize the reward from L plays, one needs to find the L arms with the highest expected reward. This is the traditional exploration-exploitation dilemma.

C2: Scale Decision. One should not play more arms than necessary: Playing many arms leads to high costs, but playing only a few arms leads to low rewards. One should set L accordingly to control the efficiency, i.e., the ratio of the rewards to the costs.

C3: Change Adaptation. The environment can either be static or non-static. In the second case, one needs to "forget" past knowledge whenever a change occurs. Forgetting that is too aggressive or too conservative leads to suboptimal results.

1.2 Contributions

This paper makes the following contributions:

We formalize a novel generalization of the MAB problem, the Scaling MAB (S-MAB). The novelty is that the decision maker not only decides which arms to play, but also how many, to maximize her cumulative reward under an efficiency constraint. To our knowledge, we are first to consider this setting.

We first propose S-TS, an algorithm to solve this problem in the static setting, i.e., when the distribution of rewards does not change over time. We leverage existing bandit algorithms, e.g., Thompson Sampling (TS), and show that the regret of our method (i.e., the difference from the outcome of a perfect oracle) only grows logarithmically with the number of time steps.

Then, we generalize our method for the non-static setting. To do so, we combine our algorithm with ADWIN [8], a state-of-the-art change detector, which is at the same time efficient and offers theoretical guarantees.

Finally, we validate our findings via experiments. The comparison with existing approaches shows that our method achieves state-of-the-art results. – We release our source code and data on GitHub[1], to ensure reproducibility.

2 RELATED WORK

Work on bandits traces back to 1933 [30], with the design of clinical trials. The theoretical guarantees of bandits remained largely unknown until recently [4, 5, 16, 18]. Our work builds on several facets of bandits, which have been studied separately, such as: anytime bandits [13, 19], multiple-play bandits (MP-MAB) [3, 20, 32] and bandits in non-static environments [6, 17, 27].

One can see the S-MAB problem as a direct extension of the MP-MAB, with the novelty that the player must control the number of plays over time. In particular, we build on the work from [20]. It shows that Multiple-Play Thompson Sampling (MP-TS) has optimal regret, while being computationally efficient. Nonetheless, we compare our results with other multiple-play models, such as variants of the celebrated UCB [4] and Exp3 [5] algorithms, namely CUCB [12], MP-KL-UCB [15, 20] and Exp3.M [32].

Our problem is different from the so-called profitable bandits [1], since they aim at maximizing a static notion of profit – as opposed to efficiency – which boils down to finding the individual arms for which the rewards exceed the costs in expectation. Moreover, our problem is more challenging than the MP-MAB and its extension called combinatorial MAB (CMAB) [12] in that we are interested in a set of arms where the model parameters μ_i satisfy an efficiency constraint (see Eq. (1)), and the algorithm needs to estimate them.

[1] https://github.com/edouardfouche/S-MAB

The S-MAB also is related to the budgeted multi-armed bandit model [31, 33], because it aims at maximizing a notion of efficiency, i.e., the ratio of the reward to the cost of playing arms. In our case, the total number of plays – the "budget" – is not an external constraint. Instead, the S-MAB decides how many arms to play based on its observations of the environment.

Bandits have readily been applied to a number of real-world applications, such as packet routing [7], online advertising [11], recommendation systems [23], robotic planning [28] and resource allocation [24]. Nonetheless, the application of bandits to data stream monitoring (see Example 1) has received much less attention.

For a broader overview of bandit algorithms, we refer the reader to recent surveys [9, 10, 22].

3 SCALING MULTI-ARMED BANDITS

In this section, we formally define the S-MAB problem and propose an algorithm, named Scaling Thompson Sampling (S-TS), to solve it in the static setting. Throughout our paper, we use the most common notations from the bandit literature, e.g., as in [9].

3.1 Problem Definition

Let there be K arms. Each arm $i \in [K] = \{1, \dots, K\}$ is associated with an unknown probability distribution v_i with mean μ_i.

At each round $t = 1, \dots, T$, the forecaster selects arms $I(t) \subset [K]$, then receives a corresponding reward vector $X(t)$. $L_t \leq K$ is the number of these arms. The rewards $X_i(t) \in X(t)$ of each arm i are i.i.d. samples from v_i. We make the classical assumption from bandit analysis that the rewards $X_i(t)$ are 0 or 1, i.e., the distribution of rewards from arm $i \in [K]$ follows a Bernoulli distribution with mean μ_i. The selection of each arm $i \in [K]$ is associated with a unit cost 1, where cost and reward do not need to have the same unit. Note that it is not very difficult to generalize our results to other reward distributions, as long as they are bounded.

Let $N_i(t)$ and $S_i(t)$ be the number of draws of arm i and the sum of the rewards obtained from it respectively before round t. Let $\hat{\mu}_i(t) = S_i(t)/N_i(t)$ be the empirical estimation of μ_i at time t. The forecaster is interested in maximizing the sum of the rewards over arms drawn, under the constraint that the sum of the rewards must be greater than the sum of the costs by an efficiency factor η^*. The parameter $\eta^* \in [0, 1]$ controls the trade-off between the cost of playing and the reward obtained, which is application-dependent.

Thinking of our "blind roulette" metaphor, assume that, whenever a bet is successful, the casino awards the double of the bet. Then, for a positive gain expectation, the player must set $\eta^* > 0.5$ and control η_t, the admitted cost per arm, to be greater than η^*.

In other words, at each step t, the forecaster is facing the following constrained optimization problem:

$$\max_{I(t)\subset[K]} \sum_{i\in I(t)} S_i(t) \quad s.t. \quad \eta_t = \frac{\sum_{i\in I(t)} \mu_i}{L_t} > \eta^* \tag{1}$$

The difficulty here is that the forecaster does not know μ_i, but only has access to an estimate $\hat{\mu}_i$ from previous observations.

$\sum_{i\in I(t)} S_i(t)$ is maximized when the forecaster chooses the arms with the highest expectation μ_i. For simplicity, we assume that all arms have distinct expectations (i.e., $\mu_i \neq \mu_j, \forall i \neq j$) and we assume without loss of generality that $\mu_1 > \mu_2 > \cdots > \mu_K$, and thus $[L_t]$ is

the top-L_t arms. Under the assumption that the forecaster always chooses $[L_t]$, the value of η_t is only determined by L_t, i.e., Eq. (1) is equivalent to finding the optimal number of plays L^*:

$$L^* = \max_{1 \le L \le K} L \quad s.t. \quad \frac{\sum_{i=1}^{L} \mu_i}{L} > \eta^* \tag{2}$$

Thus, the correct identification of the top-L_t arms (**C1**) is sine qua non to find the optimal number of plays L^* (**C2**). Next, in non-static environments, the expected rewards may change, i.e., $\mu_i : t \mapsto [0, 1]$ becomes a function of t, as does L^*. So the forecaster must adapt its estimation $\hat{\mu}_i$ (**C3**), in order to correctly select the arms with the highest reward, i.e., it needs to discard past observations. In this paper, we describe how we solve these challenges.

3.2 Scaling Thompson Sampling (S-TS)

Let us first assume that the environment is static. Our algorithm is the combination of two components:

(1) An MP-MAB algorithm to identify the top-L_t arms (**C1**).
(2) A so-called "scaling policy", to determine the value of L_{t+1} based on L_t and the observations at time t (**C2**).

For (1), we use an existing algorithm, MP-TS [20]. It is a Bayesian-inspired bandit algorithm, which maintains a Beta posterior with parameters α_i, β_i over each arm i. In each round, MP-TS samples an observation θ_i from each posterior and selects the top-L_t arms according to these observations. Then, the parameters of this posterior are adjusted based on the reward vector $X(t)$.

For (2), we propose to use a scaling policy, i.e., a strategy to control the number of plays, such that the empirical efficiency $\hat{\eta}_t$ remains larger than η^*. Whenever $\hat{\eta}_t \le \eta^*$, we "scale down", i.e., we set $L_{t+1} = L_t - 1$. Otherwise, we "scale up". When we are confident that adding one arm will lead to $\hat{\eta}_t \le \eta^*$, we stop scaling. To do so, we estimate B_t, an upper confidence bound for $\hat{\eta}_{t+1}$, assuming that $L_{t+1} = L_t + 1$. $\hat{B}_t$ is our estimator for B_t, based on the observations from the environment so far. The confidence is derived from the Kullback-Leibler divergence, as the so-called KL-UCB index [15, 25]. We name our policy Kullback-Leibler Scaling (KL-S):

$$L_{t+1} = \begin{cases} L_t - 1 & \text{if } \hat{\eta}_t \le \eta^* \\ L_t + 1 & \text{if } \hat{\eta}_t > \eta^* \text{ and } \hat{B}_t > \eta^* \\ L_t & \text{otherwise} \end{cases} \tag{3}$$

where $1 \le L_{t+1} \le K$ and

$$\hat{\eta}_t = \frac{1}{L_t} \sum_{i}^{I(t)} \hat{\mu}_i \qquad \hat{B}_t = \frac{L_t}{L_t + 1} \hat{\eta}_t + \frac{1}{L_t + 1} b_{\widehat{L_t+1}}(t) \tag{4}$$

$\hat{B}_t$ is the empirical estimator of

$$B_t = \frac{1}{L_t + 1} \sum_{i=1}^{L_t} \mu_i(t) + \frac{1}{L_t + 1} b_{L_t+1}(t).$$

where $b_i(t)$ is the KL-UCB index of arm i and $\widehat{L_t + 1}$ is the arm of the $(L_t + 1)$-th largest index. The KL-UCB index is as follows:

$$b_i(t) = \max_q \{N_i(t) d_{\mathrm{KL}}(\hat{\mu}_i(t), q) \le \log(t / N_i(t))\} \tag{5}$$

where d_{KL} is the Kullback-Leibler divergence.

In S-TS, the algorithms of (1) MP-TS and (2) KL-S are intertwined. See Algorithm 1. S-TS successively calls the two procedures MP-TS and KL-S, while maintaining the statistics N_i, S_i for each arm.

We initialize the scaling policy with $L_1 = K$. The rationale is that nothing is known about the reward distribution of the arms initially, so pulling a maximum number of arms is indeed informative.

Computational complexity of Algorithm 1: At each round, MP-TS draws a sample from a Beta distribution (Line 12), and KL-S computes the KL-UCB index for each arm (Line 25), which can be done efficiently via Newton's method. Given that these operations are done in constant time and that finding the top-L_t elements among K elements takes $O(K \log K)$ in the worst case ($L_t = K$), each round of the proposed algorithm takes $O(K \log K)$ time. The space complexity of the algorithm is $O(K)$ as it only keeps four statistics $(\alpha_i, \beta_i, N_i, S_i)$ per arm $i \in [K]$.

Algorithm 1 S-TS

Require: Set of arms $[K] = \{1, 2, \ldots, K\}$, target efficiency η^*
1: $\alpha_i(1) = 0, \beta_i(1) = 0 \quad \forall i \in [K]$
2: $N_i(1) = 0, S_i(1) = 0 \quad \forall i \in [K]$
3: $L_1 \leftarrow K$
4: **for** $t = 1, 2, \ldots, T$ **do**
5: $I(t), X(t) \leftarrow$ MP-TS(L_t) ▷ Play L_t arms (as in MP-TS)
6: **for** $i \in I(t)$ **do**
7: $N_i(t+1) = N_i(t) + 1$
8: $S_i(t+1) = S_i(t) + X_i(t)$
9: $L_{t+1} \leftarrow$ KL-S(L_t) ▷ Scale L_t for the next round

10: **procedure** MP-TS(L_t)
11: **for** $i = 1, \ldots, K$ **do**
12: $\theta_i(t) \sim \mathrm{Beta}(\alpha_i(t) + 1, \beta_i(t) + 1)$
13: Play arms $I(t) := \arg\max_{K' \subset [K], |K'| = L_t} \sum_i^{K'} \theta_i(t)$
14: Observe reward vector $X(t)$
15: **for** $i \in I(t)$ **do** ▷ Update parameters
16: $\alpha_i(t+1) = \alpha_i(t) + X_i(t)$
17: $\beta_i(t+1) = \beta_i(t) + (1 - X_i(t))$
18: **return** $I(t), X(t)$

19: **procedure** KL-S(L_t)
20: $S_i = S_i(t+1), N_i = N_i(t+1) \quad \forall i \in [K]$
21: $\hat{\mu}_i = S_i / N_i \quad \forall i \in [K]$ ▷ $\hat{\mu}_i = 1$, if $N_i = 0$
22: $\hat{\eta}_t = \sum_{i \in I(t)} \hat{\mu}_i$
23: **if** $\hat{\eta}_t \le \eta^*$ **then return** $\max(L_t - 1, 1)$ ▷ Scale down
24: **else**
25: $KL = \left\{ \max_q \{N_i \cdot d_{\mathrm{KL}}(\hat{\mu}_i, q) \le \log \frac{t+1}{N_i}\} : \forall i \in [K] \right\}$
26: $b_{\widehat{L_t+1}} = (L_t + 1)$-th largest element from KL
27: $\hat{B}_t = \frac{L_t}{L_t+1} \hat{\eta}_t + \frac{1}{L_t+1} b_{\widehat{L_t+1}}(t)$
28: **if** $\hat{B}_t > \eta^*$ **then return** $\min(L_t + 1, K)$ ▷ Scale up
29: **else return** L_t ▷ Do not scale

4 THEORETICAL ANALYSIS

In this section, we analyse the properties of scaling bandits. In particular, we measure the capability of an algorithm to control

the size of L_t by introducing a quantity called "pull regret". Our analysis is general: We show that not only S-TS (Algorithm 1) but that KL-S, combined with any MP-MAB algorithm of logarithmic regret, has logarithmic pull regret. We introduce our notation in Section 4.1 and proceed to our main theorem in Section 4.2. Details of the proofs are in Section 8, in the supplementary material.

4.1 Preliminaries

We assume there is a unique L^*, i.e., L^* is such that $\sum_{i=1}^{L^*} \mu_i / L^* > \eta^*$ and $\sum_{i=1}^{L^*+1} \mu_i/(L^*+1) < \eta^*$. Let $\Delta = \min(\Delta_a, \Delta_b)$ be the "gap", i.e., the absolute difference between η^* and the closest possible η_t, with $\Delta_a = (\sum_{i=1}^{L^*} \mu_i - \eta^*) > 0$ and $\Delta_b = (\eta^* - \sum_{i=1}^{L^*+1} \mu_i) > 0$.

Let us first generalize S-TS in Algorithm 2. A "base bandit" (MP-Base-Bandit, Line 4) is an abstract bandit algorithm that, given the reward information up to the last round and the current number of plays L_t, decides on $I(t)$, i.e., which arms to draw, and returns the reward vector $X(t)$ at each round t.

Algorithm 2 Scaling Bandit with General Base Bandit Algorithm

Require: Set of arms $[K]$, target efficiency η^*
1: $N_i(1) = 0$, $S_i(1) = 0$ $\quad \forall i \in [K]$
2: $L_1 \leftarrow K$
3: **for** $t = 1, 2, \ldots, T$ **do**
4: $\quad I(t), X(t) \leftarrow$ MP-Base-Bandit(L_t) ▷ Play L_t arms
5: $\quad$ **for** $i \in I(t)$ **do**
6: $\quad\quad N_i(t+1) = N_i(t) + 1$
7: $\quad\quad S_i(t+1) = S_i(t) + X_i(t)$
8: $\quad L_{t+1} \leftarrow$ KL-S(L_t) ▷ Scale L_t for the next round

If we set MP-Base-Bandit ≡ MP-TS (Algorithm 1, Line 10), then Algorithm 2 becomes Algorithm 1. As an alternative to MP-TS, one could consider for example MP-KL-UCB [15] (resp. CUCB [12]) that draws the top-L_t arms in terms of the KL-UCB indices (resp. UCB1 indices) or Exp3.M [32] that uses exponential weighting.

To evaluate whether our scaling strategy converges to the optimal number of pulls L^*, we define a new notion of "pull regret" as the absolute difference between the number of pulls L_t and L^*:

$$\mathrm{PReg}(T) = \sum_{t=1}^{T} \left| L^* - L_t \right| \tag{6}$$

The "standard" multiple-play regret, with varying L_t, measures how many suboptimal arms the algorithm draws. It is defined as:

$$\mathrm{Reg}(T) = \sum_{t=1}^{T} \left[\max_{I \subseteq [K], |I| = L_t} \sum_{i \in I} \mu_i - \sum_{i \in I(t)} \mu_i \right] \tag{7}$$

Notice that, when an algorithm uses $L_t = L^*$ in each round, the pull regret is 0, and the regret boils down to the existing MP-MAB. Achieving sublinear pull regret implies that the algorithm satisfies the efficiency constraint, while sublinear regret implies that it maximizes the total reward, so we need to minimize both regrets.

4.2 Regret Bound

For any event $\mathcal{X}$, let $\mathcal{X}^c$ be its complementary. For an event $\mathcal{X}$, $\mathbf{1}(\mathcal{X}) = 1$ if $\mathcal{X}$ holds or 0 otherwise.

Definition 4.1. (Top-L_t set) $I(t) : |I(t)| = L_t$ is a top-L_t set if it contains the L_t arms with highest expectation μ_i. Let $\mathcal{A}_t$ be the event that $I(t)$ is the top-L_t set.

Definition 4.2. (A logarithmic regret algorithm) A base bandit algorithm has logarithmic regret if there exists a distribution-dependent constant $C_{\mathrm{alg}} = C_{\mathrm{alg}}(\{\mu_i\})$ such that

$$\sum_{t=1}^{T} \Pr\left[\mathcal{A}_t^c\right] \leq C^{\mathrm{alg}} \log T$$

Remark 1. *Based on their existing analyses, one can prove that CUCB [12], MP-KL-UCB [15], and MP-TS [20] have logarithmic regret for varying L_t. We show that MP-TS has logarithmic regret in Section 8.1 in the supplementary material, using techniques from [2].*

The following theorem states that our policy has logarithmic pull regret, i.e., that the number of pulls converges to L^* when we combine it with a base bandit algorithm of logarithmic regret.

Theorem 4.3. (Logarithmic Pull Regret) *Let the general scaling bandit of Algorithm 2 with a base bandit algorithm of logarithmic regret be given. Then, there exist two distribution-dependent constants $C_*^{\mathrm{preg}}, C_*^{\mathrm{reg}} = C_*^{\mathrm{preg}}(\{\mu_i\}), C_*^{\mathrm{reg}}(\{\mu_i\})$ such that*

$$\mathbb{E}[PReg(T)] \leq C_*^{\mathrm{preg}} \log T, \tag{8}$$

Moreover, the standard regret of the proposed algorithm is bounded as

$$\mathbb{E}[Reg(T)] \leq C_*^{\mathrm{reg}} \log T. \tag{9}$$

Let us first define the events needed for the proof:

$$\mathcal{B}_t = \{L_t \leq L^* \cap \hat{\eta}_t > \eta^*\} \cup \{L_t > L^* \cap \hat{\eta}_t \leq \eta^*\} \tag{10}$$

$$\mathcal{C}_t = \{L_t \geq L^* \cup \hat{B}_t > \eta^*\} \tag{11}$$

$$\mathcal{D}_t = \{L_t < L^* \cup \hat{B}_t \leq \eta^*\} \tag{12}$$

The following lemmas are key to bound the pull regret:

Lemma 4.4. (Scaling) $L_t \in \{L^*, L^*+1\}$ *holds if*

$$\bigcap_{t'=t-K,\ldots,t-1} \mathcal{B}_{t'} \cap \mathcal{C}_{t'}.$$

Proof of Lemma 4.4. $\mathcal{B}_{t'} \cap \mathcal{C}_{t'}$ implies
- $L_{t'+1} = L_{t'} + 1$ if $L_{t'} < L^*$,
- $L_{t'+1} \in \{L^*, L^*+1\}$ if $L_{t'} = L^*$
- $L_{t'+1} = L_{t'} - 1$ if $L_{t'} \geq L^* + 1$

As $L^* - L_{t-K} < K$, there exists $t'_c \in \{t-K, t-K+1, \ldots, t-1\}$ such that $L_{t'_c} = L^*$ and after the round t'_c, $L_{t'} \in \{L^*, L^*+1\}$ holds. □

Lemma 4.5. (Sufficient condition of No-regret) $L_t = L^*$ *holds if*

$$\bigcap_{t'=t-K,\ldots,t-1} \{\mathcal{B}_{t'} \cap \mathcal{C}_{t'}\} \cap \mathcal{D}_{t-1}.$$

Proof of Lemma 4.5. Lemma 4.4 implies $L_{t-1} \in \{L^*, L^*+1\}$, which, combined with $\mathcal{B}_{t-1} \cap \mathcal{C}_{t-1} \cap \mathcal{D}_{t-1}$ implies that $L_t = L^*$. □

We can now proceed to the proof of Theorem 4.3:

Proof of Theorem 4.3. Lemma 4.5 implies that, if the direction of scaling is correct and the confidence bound is sufficiently small, then L_t goes to L^*. We decompose the pull regret using Lemma 4.5:

$$\begin{aligned}
&\mathrm{PReg}(T) \le K \sum_{t=1}^{T} \mathbf{1}[L_t \neq L^*] \\
&\text{(since PReg}(t) \text{ increases at most by } K \text{ at each round)} \\
&\le K \sum_{t=1}^{T} \mathbf{1}\left[\left(\bigcup_{t'=t-K}^{t-1} \left(\mathcal{A}_{t'}^c \cup \mathcal{B}_{t'}^c \cup C_{t'}^c\right) \cup \mathcal{D}_{t-1}^c\right) \cap L_t \neq L^*\right] \\
&\text{(by the contraposition of Lemma 4.5)} \\
&\le K + K \sum_{t=K+1}^{T} \left(\mathbf{1}\left[\bigcup_{t'=t-K}^{t-1} \left(\mathcal{A}_{t'}^c \cup \mathcal{B}_{t'}^c \cup C_{t'}^c\right)\right]\right. \\
&\qquad \left. + \mathbf{1}\left[\bigcap_{t'=t-K}^{t-1} \left(\mathcal{A}_{t'}^c \cap \mathcal{B}_{t'}^c \cap C_{t'}^c\right) \cap \mathcal{D}_{t-1}^c \cap L_t \neq L^*\right]\right) \\
&\le K + K^2 \sum_{t'=K+1}^{T} \left(\mathbf{1}[\mathcal{A}_{t'}^c] + \mathbf{1}[\mathcal{A}_{t'} \cap \mathcal{B}_{t'}^c] + \mathbf{1}[\mathcal{A}_{t'} \cap C_{t'}^c]\right) \\
&\quad + K \sum_{t=K+1}^{T} \mathbf{1}\left[\bigcap_{t'=t-K}^{t-1} \left(\mathcal{A}_{t'} \cap \mathcal{B}_{t'} \cap C_{t'}\right) \cap \mathcal{D}_{t-1}^c \cap L_t \neq L^*\right] \quad (13)
\end{aligned}$$

The following lemma, which is proven in the supplementary material, bounds each term in Eq. (13) in expectation.

Lemma 4.6. (Bounds on each term) *The following bounds hold:*

$$\underbrace{\sum_{t=1}^{T} \Pr[\mathcal{A}_{t'}^c] = O(\log T)}_{\text{(A)}} \quad ; \quad \underbrace{\sum_{t=1}^{T} \Pr[\mathcal{A}_t \cap \mathcal{B}_t^c] = O(1/\Delta^2)}_{\text{(B)}}$$

$$\underbrace{\sum_{t=1}^{T} \Pr[\mathcal{A}_t \cap C_t^c] = O(1/\Delta^2) + O(\log\log T)}_{\text{(C)}}$$

$$\underbrace{\sum_{t=K+1}^{T} \Pr\left[\bigcap_{t'=t-K}^{t-1} (\mathcal{A}_{t'} \cap \mathcal{B}_{t'} \cap C_{t'}) \cap \mathcal{D}_{t-1}^c, L_t \neq L^*)\right] = O(1/\Delta^2).}_{\text{(D)}}$$

Eq. (8) now follows from Lemma 4.6. Eq. (9) follows from the fact that the base bandit algorithm has logarithmic regret. □

5 NON-STATIC ADAPTATION (S-TS-ADWIN)

To handle the non-static setting (**C3**), we combine S-TS with Adaptive Windowing (ADWIN) [8], yielding S-TS-ADWIN.

ADWIN (Algorithm 3) monitors the expected value from a single (virtually infinite) stream of values $\{x_1, x_2, \dots\}$, where $x_i \in [0, 1]$. ADWIN maintains a window W of varying size $|W|$ so that the nature of the stream is consistent. ADWIN reduces the size of the window whenever two neighbouring subwindows have different mean, based on a statistical test with confidence δ. For each two subwindows of size $|W_1| + |W_2| = |W|$ with corresponding means $\hat{\mu}_{W_1}, \hat{\mu}_{W_2}$, ADWIN shrinks the windows to W_2 if

$$|\hat{\mu}_{W_1} - \hat{\mu}_{W_2}| \ge \epsilon_{\text{cut}}^{\delta} \quad \text{where} \quad \epsilon_{\text{cut}}^{\delta} = \sqrt{\frac{1}{2m} \log\left(\frac{4|W|}{\delta}\right)} \qquad (14)$$

and $m = 1/((1/|W_1|) + (1/|W_2|))$. The authors [8] showed that ADWIN efficiently adapts to both gradual and abrupt changes with theoretical guarantees (see Theorem 3.1 therein).

Algorithm 3 ADWIN

Require: Stream of values $\{x_1, x_2, \dots\}$, confidence level δ

$W \leftarrow \{\}$
for $t = 1, 2, \dots$ **do**
  $W \leftarrow W \cup \{x_t\}$
  Drop elements from the tail of W until $|\hat{\mu}_{W_1} - \hat{\mu}_{W_2}| < \epsilon_{\text{cut}}^{\delta}$ holds for every split $W = W_1 \cup W_2$.

The idea behind S-TS-ADWIN, showed in Algorithm 4, is to create an ADWIN instance A_i per arm i. At each step t, A_i obtains as input the reward from the corresponding arm $X_i(t)$ if $i \in I(t)$. Thus, each instance A_i maintains a time window W_i of variable size, which shrinks whenever ADWIN detects a change in μ_i.

However, for any bandit algorithm with logarithmic regret, the number of plays of suboptimal arms grows with $\log(T)$. That is, after some time, the A_j of any suboptimal arm j does not obtain any input, and thus no change can be detected for arm j.

Thus, we use $w_t = \min\{|W_i| : \forall i \in [K]\}$, i.e., the smallest window from each A_i, to estimate the statistics of any arm $i \in [K]$ at each step t. Here, we implicitly assume that the change points are "global", i.e., that they are shared across the $\mu_i, \forall i \in [K]$. In principle, changes may also be "local", e.g., a single μ_i changes. But we will show that despite this assumption, it works well in practice.

Algorithm 4 S-TS-ADWIN

Require: Set of arms $[K]$, target efficiency η^*, delta δ

$\alpha_i(1) = 0, \beta_i(1) = 0 \quad \forall i \in [K]$
$N_i(1) = 0, S_i(1) = 0 \quad \forall i \in [K]$
$A_i \leftarrow$ instantiate ADWIN with parameter $\delta, \forall i \in [K]$
$L_1 \leftarrow K$
for $t = 1, 2, \dots, T$ **do**
  $I(t), X(t) \leftarrow$ MP-TS(L_t) ▷ Play L_t arms (as in MP-TS)
  for $i \in I(t)$ **do**
    $N_i(t+1) = N_i(t) + 1$
    $S_i(t+1) = S_i(t) + X_i(t)$
    Add $X_i(t)$ into A_i
  $L_{t+1} \leftarrow$ KL-S(L_t) ▷ Scale L_t for the next round
  $w_t \leftarrow \min\{|W_i| : \forall i \in [K]\}$ ▷ Keep the smallest window
  $N_i(t+1) = \sum_{j=t-w_t}^{t} \mathbf{1}(i \in I(j))$
  $S_i(t+1) = \sum_{j=t-w_t}^{t} X_i(j) * \mathbf{1}(i \in I(j))$
  $\alpha_i(t+1) = S_i(t+1), \beta_i(t+1) = N_i(t+1) - S_i(t+1)$

By default, we set $\delta = 0.1$ for each instance, since [8] showed that it leads to a very low empirical false positive rate and good performance. We show in our experiments that this parameter does not have a significant impact on our results, and that S-TS-ADWIN performs very well against synthetic and real-world scenarios.

Computational complexity of Algorithm 4: We use the improved version of ADWIN, dubbed ADWIN2 [8]. For a window of size W, ADWIN2 takes $O(\log W)$ time per object. Since we have K instances of them, the time complexity of the ADWIN2 part is in $O(K \log W) = O(K \log T)$ per round. The space complexity of ADWIN2 is in $O(W)$, but the window typically shrinks rapidly in the case of a non-static environment. We show in our experiments that the scalability of S-TS-ADWIN is almost the same as S-TS.

6 EXPERIMENTS

This section evaluates the performance S-TS and S-TS-ADWIN. We compare against alternative "base bandits" and to the state-of-the-art non-static bandit algorithms. We also highlight the benefits of scaling by comparing against non-scaling bandits. We simulate scenarios with 10^5 steps, to evaluate our approach in static (Section 6.1) and non-static (Section 6.2) environments. Then, we present a study where we have monitored real-world data streams (Section 6.3). We will also verify the scalability of our approach.

We have implemented every approach in Scala and averaged our experimental results across 100 runs. Each algorithm was run single-threaded in a server with 64 cores at 3 GHz and 128 GB RAM.

6.1 Static Environment

In this section, our goal is to verify the capability of S-TS to find L^* and maximize the reward in a static environment. We compare S-TS in terms of pull regret and standard regret against alternative scaling bandits, i.e., by replacing the base bandit with MP-KL-UCB [15, 20], CUCB [12], and Exp3.M [32]. We adapt each algorithm so that they "scale". The prefix "S-" stands for the use of KL-S: For instance, S-KL-UCB is Algorithm 2 where the MP-Base-Bandit chooses the top-L_t arms based on the KL-UCB index.

We simulate a static scenario with K Bernoulli arms with known means $\mu_1, \dots \mu_K$ and $T = 10^5$, such that

$$\{\mu_i\}_{i=1}^{K} = \left\{ \frac{i}{K} - \frac{1}{3K} \right\}_{i=1}^{K} \tag{15}$$

Given this, the means of the K arms are distributed linearly between 0 and 1, such that, when $\eta^* = 0.9$, then $L^* = K/5$, and when $\eta^* = 0.8$, then $L^* = 2K/5$, and so on. We set $K = 100$. We measure the regret and the pull regret of each approach against a Static Oracle (SO), which always pulls the top-L^* arms in expectation.

In Figure 1, the first row shows the convergence to L^*. The second and last rows show the pull regret and standard regret respectively. We see that S-TS and S-KL-UCB perform best, since they obtain the lowest regret for both measures. When η^* is smaller, the number of pulls L_t converges faster to L^*, for two reasons: (i) The optimal number of pulls L^* is closer to the starting condition L_1, and (ii) a lower η^* allows more exploration and more plays per rounds. So the top-L^* arms are found in fewer rounds with higher confidence.

We also see that S-Exp3.M does not perform very well. S-Exp3.M targets at the adversarial bandit problem [32]. I.e., its assumptions regarding the rewards distribution are weaker. The policy, based on exponential weighting, forces Exp3.M to explore much, so that our scaling policy KL-S lets L_t quickly drop to 1. Nonetheless, we see that after a large number of steps, S-Exp3.M lets L_t increase again.

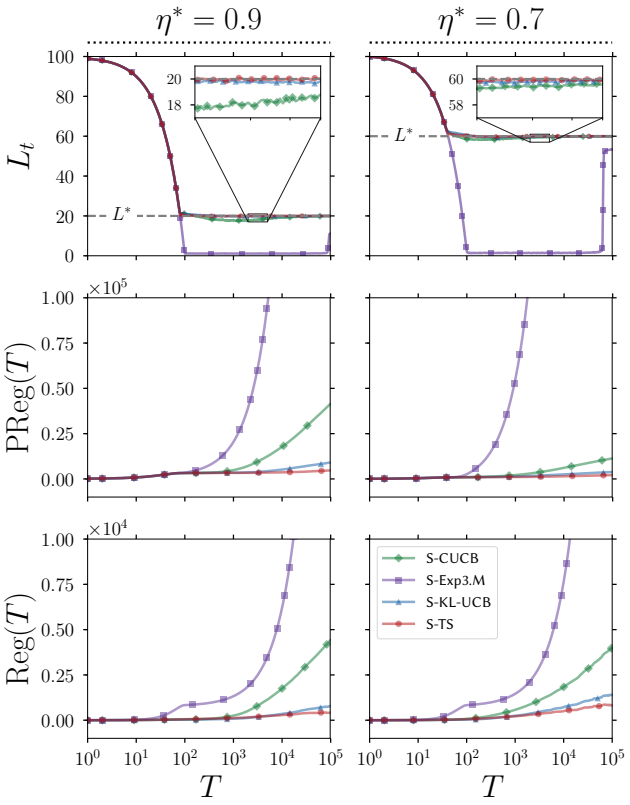

Figure 1: Static experiment. S-TS minimizes both regrets.

6.2 Non-Static Environment

In this section, we verify whether S-TS-ADWIN adapts to changes in the reward distribution. We compare our results against the following state-of-the-art non-static bandit algorithms: Discounted Thompson Sampling (dTS) [26] with parameter γ (discounting factor), Epsilon-Greedy (EG) [29] with parameter ϵ (probability of selecting the best arm greedy-wise) and Sliding Window UCB (SW-UCB) [16] with parameter w (size of the window).

We set $\eta^* = 0.6$ and use the previous static setup to generate our non-static scenarios. In line with the literature on concept drift [14], we simulate "gradual" and "abrupt" changes:

- **Gradual**: We place 60 equidistant change points over the time axis. For the first 30 change points, we set $\mu_h = 0$ for the arm $h \in K$ with the current highest expected reward. Then, we revert those changes in a "last in – first out" way. Thus, L^* evolves gradually from 80 to 20, and back.
- **Abrupt**: We place two change points, equidistant from the start and end. At the first one, we set $\mu_h = 0$ for the top-30 arms. We revert this change at the second change point. Thus, L^* abruptly changes from 80 to 20 and back.

Since the environment is non-static, μ_i and L^* now vary as a function of t. Thus, we measure regret against a piecewise static oracle, which "knows" $\mu_i(t)$ and $L^*(t)$.

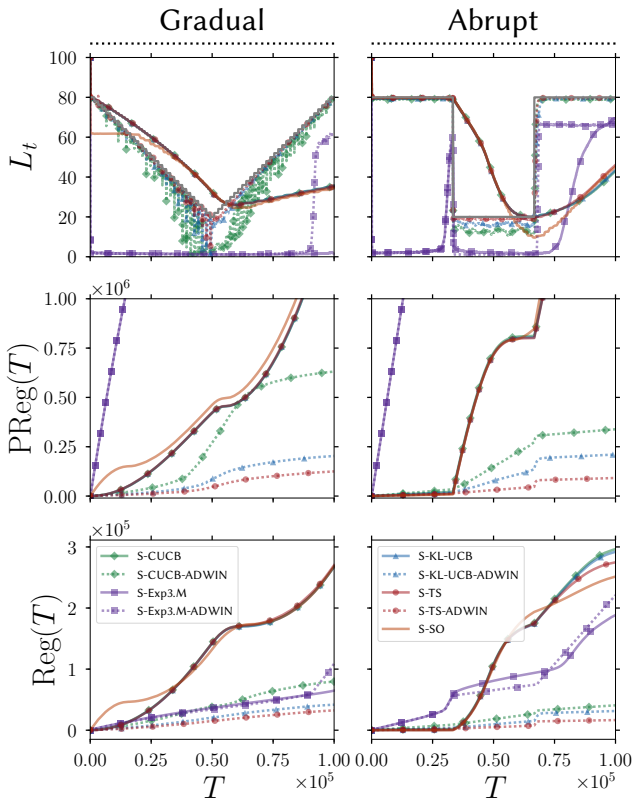

Figure 2: Non-static experiment: S-TS vs. S-TS-ADWIN.

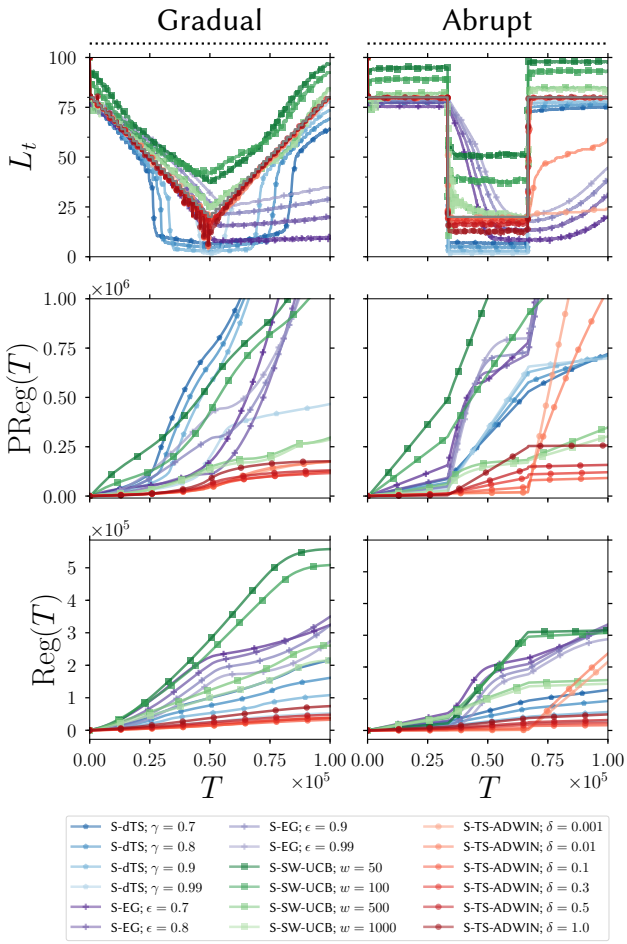

Figure 3: Non-static experiment: Non-static bandits.

A key result from this experiment is that S-TS, which assumes that arms do not change over time, fails to adapt to a changing environment, whereas our improvement, S-TS-ADWIN, does (Figure 2) and even outperforms all alternatives (Figure 3).

Figure 2 shows that S-CUCB(-ADWIN) and S-KL-UCB(-ADWIN) behave similarly to S-TS(-ADWIN), but have slightly higher regret and pull regret. S-Exp3.M has very high pull regret. Overall, we see that our adaptation based on ADWIN made it possible to handle both gradual and abrupt changes.

Figure 3 compares our approach to the existing non-static bandit alternatives. S-dTS tends to underestimate L^* in the case of a strong discounting factor, e.g., for $\gamma = 0.7$. On the contrary, S-SW-UCB overestimates L^*, in particular when the window size w is small. The behaviour of S-EG is similar to the one of static approaches: It does not adapt to change quickly.

We also see that S-TS-ADWIN is robust for a large range of δ, except for very small values, e.g., 0.01 and 0.001. The best results are obtained with $\delta = 0.1$, which is consistent with the results in [8]. Other approaches in turn are quite sensitive to their parameters. For example, we can see that a weak discounting factor of $\gamma = 0.99$ is beneficial for dTS in the case of a gradual change, but that more aggressive discounting is better with abrupt changes. The figure shows that our approach adapts to different kinds of change, as opposed to the other approaches, without tuning its parameter.

6.3 Real-World Example

In this section, we look at a real-world instance of Example 1. The data set corresponds to a week of measurements in a pyrolysis plant. It contains one measurement per second from a selection of 20 sensors, such as temperature, pressure, in various components.

We consider Mutual Information [21] as a measure of correlation, which we have computed pair-wise between all attributes over a sliding window of size 1000 (~ 15 minutes) with step size 100 (~ 1.5 minute). Our goal in this use case is to employ bandit algorithms as a "monitoring system" to keep an overview of large correlation values in the stream. Whenever the monitoring system detects a Mutual Information value higher than a threshold Γ, it obtains a reward of 1, otherwise 0. The challenge is to decide which coefficients to re-compute and how many of them at each step. This results in a S-MAB problem with 6048 steps and $(20 * 19)/2 = 190$ arms. Figure 4 is the reward matrix for $\Gamma = 2$. We see that there are fewer rewards at the beginning and end of time. This is because the week is bordered by periods of lower activity in the plant. In the weekends, we observe fewer correlations than during weekdays.

Since there is no ground truth $\{\mu_i(t)\}$, it is not possible to assess the pull regret nor the standard regret. Instead, we compare the

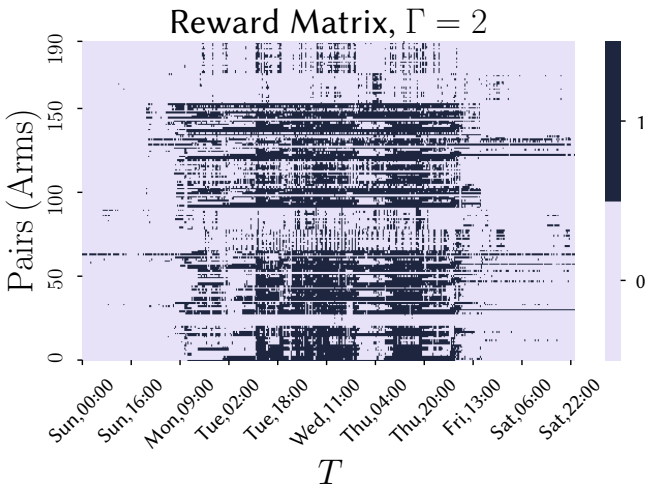

Figure 4: Real-world experiment: Distribution of rewards.

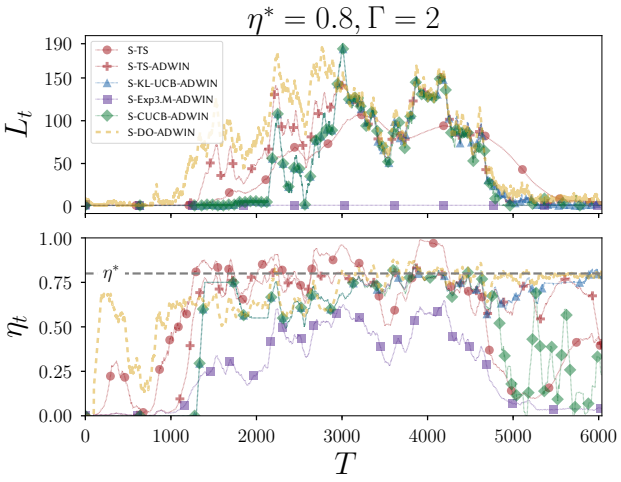

Figure 5: Real-world experiment: Variation of L_t and η_t.

rewards and costs across algorithms: Whenever an Algorithm A obtains more rewards than an Algorithm B for the same cost (i.e., number of plays), we conclude that A is superior to B.

We compare against oracles with different levels of knowledge: Random Oracle (RO), Static Oracle (SO) and Dynamic Oracle (DO) are shown as a black, green and gold dotted lines respectively.

In Figure 5, we set $\eta^* = 0.8$ and visualize the evolution of the number of plays over time for various approaches. S-TS-ADWIN is the closest match to S-DO-ADWIN, our strongest baseline. This indicates that S-TS-ADWIN adapts to changes of rewards to find a proper value for L_t, unlike static algorithms such as S-TS.

Figure 6 shows the relationship between the average reward and the average cost (in terms of number of plays) of each algorithm. S-TS-ADWIN consistently yields higher rewards than DO for the

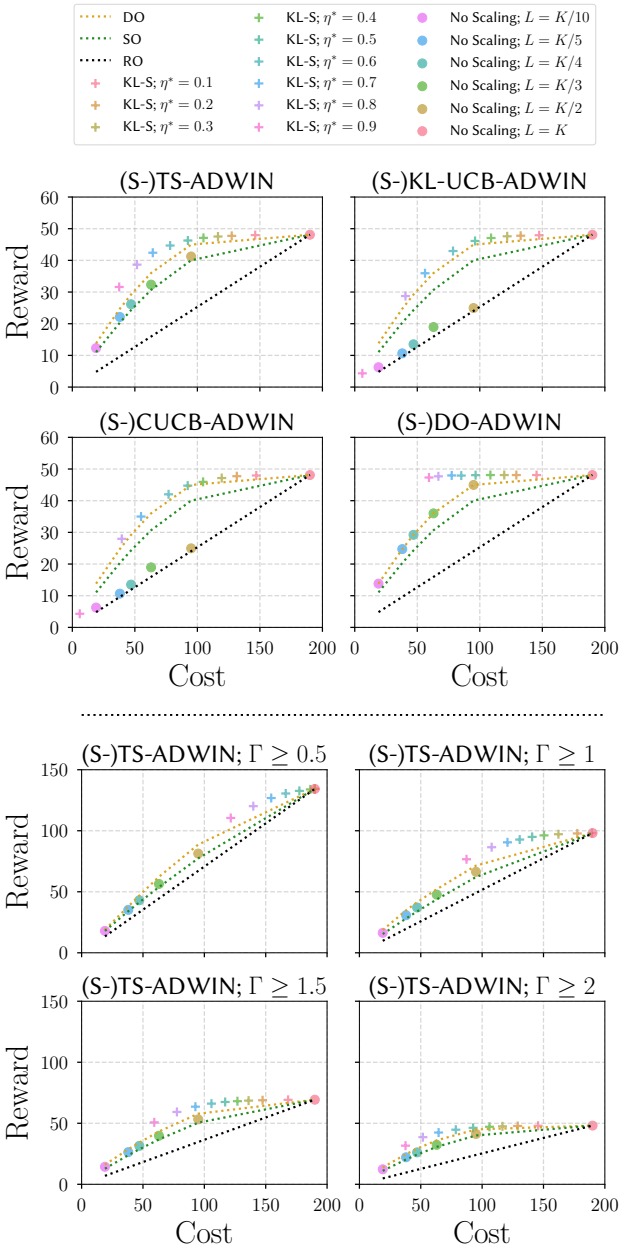

Figure 6: Real-world experiment: Scaling vs. Non-Scaling and variation of the utility criterion Γ with (S-)TS-ADWIN.

same costs. TS-ADWIN (without scaling) also is superior to SO. Here we see the full benefit of our scaling policy with S-DO-ADWIN: The scaling dynamic oracle consistently achieves nearly maximal reward, while pulling fewer arms than a non-scaling algorithm. In other words, it outperforms its non-scaling counterpart.

Surprisingly, the UCB-based approaches do not perform well without scaling; they are close to the random oracle (RO). We hypothesize that ADWIN keeps the size of the dynamic window small in the real-world setting; w_t remains small, affecting the sharpness

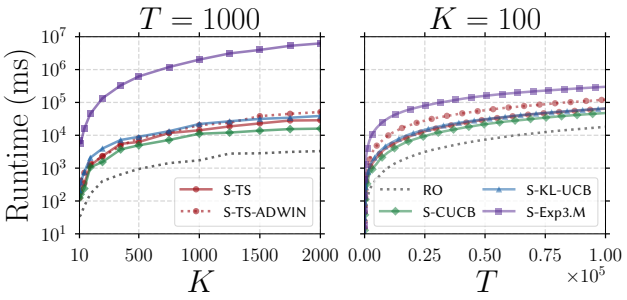

Figure 7: Scalability of bandit algorithms w.r.t. K and T.

of the confidence bound. However, when our scaling policy is used, both approaches perform slightly better than DO.

We also verify that S-TS-ADWIN can adapt to different environments by changing Γ, which influences the availability of rewards. We see here that the improvement against our baselines is consistent, i.e., our algorithm also adapts the number of plays per round.

Finally, we evaluate the scalability of our approach with growing K and T. To do so, we stick to our real world example and create versions of the problem of different size with 10 to 2000 arms and up to 10^5 steps, by resampling the arms and observations. Then, we run our real-world experiment with $\Gamma = 2$ and average the runtime with scaling parameter η^* from 0.1 to 0.9. Figure 7 shows the result. We see that each bandit approach scales linearly with the number of arms and the number of steps. S-CUCB is the fastest one, closely followed by S-TS and S-KL-UCB. S-Exp3.M is two orders of magnitude slower w.r.t. K. By comparing S-TS and S-TS-ADWIN, we see that the added computational burden from ADWIN is small and scales alike with an increasing number of arms and time steps. Each bandit approach, except Exp3.M, is at most one order of magnitude slower than choosing arms at random (RO). We see that our approach requires on average one millisecond to decide which arms to play when $K = 100$. This is typically less than the time required at each step to estimate Mutual Information on a single pair, using state-of-the-art estimators [21].

Altogether, our experiments verified that our algorithms, S-TS and S-TS-ADWIN, are both effective and efficient. S-TS outperforms state-of-the-art bandits in the static setting, while S-TS-ADWIN adapts better to different kinds of change than its competitors. In our real-world example, S-TS-ADWIN obtains almost all the rewards in the environment for a cost reduced by up to 50%, outperforming very competitive baselines, such as a non-scaling dynamic oracle.

7 CONCLUSION

We have formalized a novel bandit model, which captures the efficiency trade-off that is central to many real-world applications. We have proposed a new algorithm, S-TS, which combines Multiple-Play Thompson Sampling (MP-TS) with a new functionality to decide on the number of arms played per round, a so-called "scaling policy". Our analysis and experiments showed that it enjoys strong theoretical guarantees and very good empirical behaviour. We also proposed an extension of our algorithm for the non-static setting. We applied the proposed model to data stream monitoring and showed its utility. However, we expect the impact of our contribution to extend beyond this one application.

In the future, it will be interesting to look closer at the non-static setting. The static regret analysis already is quite involved, and extending it to non-static settings is a challenge. Our algorithm performs better than the existing approaches empirically. We hypothesize that this success is due a class of non-stationarity that our solution exploits – but which has not been formalized yet.

ACKNOWLEDGMENTS

This work was supported by the DFG Research Training Group 2153: "Energy Status Data – Informatics Methods for its Collection, Analysis and Exploitation" and the German Federal Ministry of Education and Research (BMBF) via Software Campus (01IS17042). We thank the pyrolysis team of the Bioliq®process for providing the data for our real-world use case (https://www.bioliq.de/english/).

REFERENCES

[1] Mastane Achab, Stéphan Clémençon, and Aurélien Garivier. 2018. Profitable Bandits. In *ACML (Proceedings of Machine Learning Research)*, Vol. 95. PMLR, 694–709. http://proceedings.mlr.press/v95/achab18a.html

[2] Shipra Agrawal and Navin Goyal. 2013. Further Optimal Regret Bounds for Thompson Sampling. In *AISTATS (JMLR Workshop and Conference Proceedings)*, Vol. 31. JMLR.org, 99–107. http://proceedings.mlr.press/v31/agrawal13a.html

[3] Venkatachalam Anantharam, Pravin Varaiya, and Jean Walrand. 1987. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-Part I: I.I.D. rewards. *IEEE Trans. Automat. Control* 32, 11 (1987), 968–976. https://doi.org/10.1109/TAC.1987.1104491

[4] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning* 47, 2-3 (2002), 235–256. https://doi.org/10.1023/A:1013689704352

[5] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. 1995. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*. 322–331. https://doi.org/10.1109/SFCS.1995.492488

[6] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. 2002. The Nonstochastic Multiarmed Bandit Problem. *SIAM J. Comput.* 32, 1 (2002), 48–77. https://doi.org/10.1137/S0097539701398375

[7] Baruch Awerbuch and Robert D. Kleinberg. 2004. Adaptive routing with end-to-end feedback: distributed learning and geometric approaches. In *STOC*. ACM, 45–53. https://doi.org/10.1145/1007352.1007367

[8] Albert Bifet and Ricard Gavaldà. 2007. Learning from Time-Changing Data with Adaptive Windowing. In *SDM*. SIAM, 443–448. https://doi.org/10.1137/1.9781611972771.42

[9] Sébastien Bubeck and Nicolò Cesa-Bianchi. 2012. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning* 5, 1 (2012), 1–122. https://doi.org/10.1561/2200000024

[10] Giuseppe Burtini, Jason L. Loeppky, and Ramon Lawrence. 2015. A Survey of Online Experiment Design with the Stochastic Multi-Armed Bandit. *CoRR* abs/1510.00757 (2015). http://arxiv.org/abs/1510.00757

[11] Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. 2008. Mortal Multi-Armed Bandits. In *NIPS*. 273–280.

[12] Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. 2016. Combinatorial Multi-Armed Bandit with General Reward Functions. In *NIPS*. 1651–1659.

[13] Rémy Degenne and Vianney Perchet. 2016. Anytime optimal algorithms in stochastic multi-armed bandits. In *ICML (JMLR Workshop and Conference Proceedings)*. JMLR.org, 1587–1595. http://proceedings.mlr.press/v48/degenne16.html

[14] João Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM Comput. Surv.* 46, 4 (2014), 44:1–44:37. https://doi.org/10.1145/2523813

[15] Aurélien Garivier and Olivier Cappé. 2011. The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond. In *COLT (JMLR Proceedings)*, Vol. 19. JMLR.org, 359–376. http://proceedings.mlr.press/v19/garivier11a.html

[16] Aurélien Garivier and Eric Moulines. 2008. On Upper-Confidence Bound Policies for Non-Stationary Bandit Problems. *CoRR* abs/0805.3415 (2008). https://arxiv.org/abs/0805.3415

[17] Aurélien Garivier and Eric Moulines. 2011. On Upper-Confidence Bound Policies for Switching Bandit Problems. In *ALT (Lecture Notes in Computer Science)*, Vol. 6925. Springer, 174–188. https://doi.org/10.1007/978-3-642-24412-4_16

[18] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. 2012. Thompson Sampling: An Asymptotically Optimal Finite-Time Analysis. In *ALT (Lecture Notes in Computer Science)*, Vol. 7568. Springer, 199–213. https://doi.org/10.1007/978-3-642-34106-9_18
[19] Robert D. Kleinberg. 2006. Anytime algorithms for multi-armed bandit problems. In *SODA*. ACM Press, 928–936. https://doi.org/10.1145/1109557.1109659
[20] Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa. 2015. Optimal Regret Analysis of Thompson Sampling in Stochastic Multi-armed Bandit Problem with Multiple Plays. In *ICML (JMLR Workshop and Conference Proceedings)*, Vol. 37. JMLR.org, 1152–1161. http://proceedings.mlr.press/v37/komiyama15.html
[21] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Phys. Rev. E* 69 (Jun 2004), 066138. Issue 6. https://doi.org/10.1103/PhysRevE.69.066138
[22] Tor Lattimore and Csaba Szepesvári. 2019. *Bandit Algorithms*. Cambridge University Press (preprint). https://banditalgs.com/
[23] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *WWW*. ACM, 661–670. https://doi.org/10.1145/1772690.1772758
[24] Tian Li, Jie Zhong, Ji Liu, Wentao Wu, and Ce Zhang. 2018. Ease.ml: Towards Multi-tenant Resource Sharing for Machine Learning Workloads. *PVLDB* 11, 5 (2018), 607–620. https://dl.acm.org/citation.cfm?id=3177737
[25] Odalric-Ambrym Maillard. 2017. Boundary Crossing for General Exponential Families. In *ALT (Proceedings of Machine Learning Research)*, Vol. 76. PMLR, 151–184. http://proceedings.mlr.press/v76/maillard17a.html
[26] Vishnu Raj and Sheetal Kalyani. 2017. Taming Non-stationary Bandits: A Bayesian Approach. *CoRR* abs/1707.09727 (2017). http://arxiv.org/abs/1707.09727
[27] Aleksandrs Slivkins and Eli Upfal. 2008. Adapting to a Changing Environment: the Brownian Restless Bandits. In *COLT*. Omnipress, 343–354.
[28] Vaibhav Srivastava, Paul Reverdy, and Naomi Ehrich Leonard. 2014. Surveillance in an abruptly changing world via multiarmed bandits. In *CDC*. IEEE, 692–697. https://doi.org/10.1109/CDC.2014.7039462
[29] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement learning : an introduction*. MIT Press.
[30] William R. Thompson. 1933. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika* 25, 3/4 (1933), 285–294. https://doi.org/10.2307/2332286
[31] Long Tran-Thanh, Archie C. Chapman, Enrique Munoz de Cote, Alex Rogers, and Nicholas R. Jennings. 2010. Epsilon-First Policies for Budget-Limited Multi-Armed Bandits. In *AAAI*. AAAI Press.
[32] Taishi Uchiya, Atsuyoshi Nakamura, and Mineichi Kudo. 2010. Algorithms for Adversarial Bandit Problems with Multiple Plays. In *ALT*, Vol. 6331. Springer, 375–389. https://doi.org/10.1007/978-3-642-16108-7_30
[33] Yingce Xia, Tao Qin, Weidong Ma, Nenghai Yu, and Tie-Yan Liu. 2016. Budgeted Multi-Armed Bandits with Multiple Plays. In *IJCAI*. IJCAI/AAAI Press, 2210–2216. http://www.ijcai.org/Abstract/16/315

8 PROOFS

8.1 Performance of MP-TS as a "base bandit"

We show that Remark 1 holds for MP-TS. Let the posterior sample of TS at round t be $\theta_i(t) \sim \mathrm{Beta}(\alpha_i(t)+1, \beta_i(t)+1)$. Note that $\mathcal{A}_t^c, L_t = l$ implies that there exists $i \le l, j > l$ such that $i \notin I_t$, $j \in I_t$. Let $d = \mu_i - \mu_j > 0$ and $x, y = \mu_j + d/3, \mu_j + 2d/3$. Let the events $\mathcal{E}_{i,\mu}(t) = \{\hat{\mu}_i(t) \le x\}$ and $\mathcal{E}_{i,\theta}(t) = \{\theta_i(t) \le y\}$. Let $\theta_{(l)}(t)$ be the l-th largest from $\{\theta_i(t)\}$ (ties broken arbitrarily). We have

$$\sum_{t=1}^{T} \Pr\left[\mathcal{A}_t^c\right] = \sum_{t=1}^{T} \sum_{l\in[K]} \Pr\left[\mathcal{A}_t^c \cap L_t = l\right] \le \sum_{t=1}^{T} \sum_{l\in[K]} \sum_{i\le l, j>l} \Pr\left[L_t = l \cap i \notin I_t \cap j \in I_t\right]. \quad (16)$$

Here,

$$\Pr\left[L_t = l \cap i \notin I_t \cap j \in I_t\right] \le \Pr\left[L_t = l \cap y \le \theta_j(t)\right] + \Pr\left[L_t = l \cap \theta_{(l)}(t) \le y \cap \theta_i(t) \le y\right]. \quad (17)$$

Let $p_{i,n} = \Pr[\theta_i(t) > y \cap N_i(t) = n]$. The following discussion is essentially equivalent to the Lemma 9 in [20]. Let $\theta_{(l)\setminus i}(t)$ be the value of the l-th largest among $\{\theta_j\}_{j\in[K]\setminus i}$. Due to place restrictions, we may write equivalently $\theta_{(l)\setminus i}(t) \equiv \theta^t_{(l)\setminus i}$. Thus, we have

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbf{1}\left[L_t = l \cap \theta_{(l)}(t) \le y \cap \theta_i(t) \le y\right] \\
&\le \sum_{n=1}^{T}\sum_{t=1}^{T} \mathbf{1}\left[L_t = l \cap \theta_{(l)}(t) \le y \cap \theta_i(t) \le y \cap N_i(t) = n\right] \\
&\le \sum_{n=1}^{T}\sum_{t=1}^{T} \mathbf{1}\left[L_t = l \cap \theta_{(l)\setminus i}(t) \le y \cap \theta_i(t) \le y \cap N_i(t) = n\right] \\
&\le \sum_{n=1}^{T}\sum_{m=1}^{T} \mathbf{1}\left[m \le \sum_{t=1}^{T} \mathbf{1}\left[L_t = l \cap \theta^t_{(l)\setminus i} \le y \cap \theta^t_i \le y \cap N^t_i = n\right]\right].
\end{aligned}$$

The event

$$m \le \sum_{t=1}^{T} \mathbf{1}\left[\theta_{(l)\setminus i}(t) \le y \cap \theta_i(t) \le y \cap N_i(t) = n\right] \quad (18)$$

implies that $\{\theta_{(l)\setminus i}(t) \le y \cap \theta_i(t) \le y \cap N_i(t) = n\}$ occurred at least m rounds, and $\{\theta_i(t) \le y\}$ occurred for the first m rounds such that Eq. (18) holds. By the statistical independence of $\theta_{(l)\setminus i}(t)$ and $\theta_i(t)$:

$$\Pr\left[m \le \mathbf{1}\left[L_t = l \cap \theta^t_{(l)\setminus i} \le y \cap \theta^t_i \le y \cap N^t_i = n\right]\right] \le (1-p_{i,n})^m.$$

and following the same steps as Lemma 9 in [20], we have

$$\sum_{n=1}^{T}\sum_{m=1}^{T}(1-p_{i,n})^m \le \sum_{n=1}^{T}\frac{1-p_{i,n}}{p_{i,n}} \le \frac{1}{(\mu_i - y)^2} \text{ (by Lemma 2 in [2])}. \quad (19)$$

Moreover, by Lemma 4 and Lemma 3 in [2]:

$$\begin{aligned}
\Pr\left[L_t = l \cap y \le \theta_j(t)\right] &\le \Pr\left[L_t = l \cap j \in I_t \cap y \le \theta_j(t) \cap x > \hat{\mu}_j\right] \\
&\quad + \Pr\left[L_t = l \cap j \in I_t \cap x \le \hat{\mu}_j\right] \\
&\le \left(\frac{\log T}{d_{\mathrm{KL}}(x,y)} + 1\right) + \left(\frac{1}{d_{\mathrm{KL}}(x,\mu_j)} + 1\right), \quad (20)
\end{aligned}$$

where $d_{\mathrm{KL}}(p,q) = p\log(p/q) + (1-p)\log((1-p)/(1-q))$ be the KL divergence between two Bernoulli distributions. From Eqs. (16), (17), (19), and (20), we have $\sum_{t=1}^{T}\Pr\left[\mathcal{A}_t^c\right] = O(\log T)$.

8.2 Proof of Lemma 4.6

In this section, we bound each of the terms (A)–(D) in Lemma 4.6.

Lemma 8.1. *Let*

$$\mathcal{G}_l(t) = \bigcap_{i\le l} \left\{|\hat{\mu}_i(t) - \mu_i| \le \Delta\right\}.$$

For $l \in [K]$, the following inequality holds:

$$\sum_{t=1}^{T} \Pr[\mathcal{A}_t \cap L_t = l \cap \mathcal{G}_l^c(t)] = O(1/\Delta^2). \quad (21)$$

Proof. Event $\mathcal{A}$ implies each of arm $i \le l$ is drawn, and thus

$$\sum_{t=1}^{T} \Pr[\mathcal{A}_t \cap L_t = l \cap \mathcal{G}_l^c(t)] \le 1 + \sum_{n=1}^{T} \Pr[|\hat{\mu}_{i,n} - \mu_i| \le \Delta] \le 1 + \frac{1}{2\Delta^2}\mathrm{e}^{-2n_c\Delta^2} \text{ (by Lemma 9.2)} = O\left(\frac{1}{\Delta^2}\right). \quad \square$$

Bounding Term (A): Term (A) is directly bounded by the fact that the base bandit algorithm has logarithmic regret.

Bounding Term (B): Note that $\mathcal{B}_t = \{L_t \le L^* \cap \hat{\eta}_t < \eta^*\} \cup \{L_t > L^* \cap \hat{\eta}_t \ge \eta^*\}$, and $\{\mathcal{A}_t \cap \mathcal{G}_l(t)\}$ implies $\mathcal{B}_t$. By Lemma 8.1,

$$\sum_{t=1}^{T} \Pr[\mathcal{A}_t \cap \mathcal{B}_l^c] \le \sum_{l\in[K]} \sum_{t=1}^{T} \Pr[\mathcal{A}_t \cap \mathcal{G}_l^c(t)] = O(1/\Delta^2). \quad (22)$$

Bounding Term (C): The event $\mathcal{A}_t \cap B_i^l(t) < \eta^*$ implies $\mathcal{G}_l^c(t) \cup b_{l+1}(t) \le \mu_{l+1} - \Delta$. By using this, we have

$$\begin{aligned}
&\sum_{t=1}^{T} \Pr[\mathcal{A}(t) \cap C^c(t)] \le \sum_{l=1}^{L^*-1} \sum_{t=1}^{T} \Pr[\mathcal{A}_t \cap L_t = l \cap B_l^l(t) \le \eta^*] \\
&\le \sum_{l=1}^{L^*-1} \sum_{t=1}^{T} \Pr\left[\mathcal{A}_t \cap L_t = l \cap \left(\mathcal{G}_l^c(t) \cup b_{l+1}(t) \le \mu_{l+1} - \Delta\right)\right] \\
&\le \sum_{l=1}^{L^*-1} \sum_{t=1}^{T} \left(\Pr[\mathcal{A}_t \cap L_t = l \cap \mathcal{G}_l^c(t))] + \Pr[b_{l+1}(t) \le \mu_{l+1} - \Delta]\right) \\
&\le \sum_{l=1}^{L^*-1} \sum_{t=1}^{T} \Pr[\mathcal{A}_t \cap L_t = l \cap \mathcal{G}_i^c(t))] + O(\log\log T) \\
&\quad \text{(By the union bound of Lemma 9.3 over } t \in [T]) \\
&= O(1/\Delta^2) + O(\log\log T) \quad \text{(by Lemma 8.1).} \quad (23)
\end{aligned}$$

Bounding Term (D): Note that $\bigcap_{t'=t-K}^{t-1} (\mathcal{A}_{t'} \cap \mathcal{B}_{t'} \cap C_{t'})$ implies $L_{t-1} \in \{L^*, L^*+1\}$. Thus, $L_t \ne L^*$ implies $B_i^{L_{t-1}}(t-1) > \eta^*$, and $B_i^{L_{t-1}}(t-1) > \eta^*$ implies $\mathcal{G}_{L_{t-1}}^c(t) \cup b_{L_{t-1}+1}(t) > \mu_{L_{t-1}+1} + \Delta$. Moreover, $\bigcap_{t'=t-K}^{t-1} (\mathcal{A}_{t'} \cap \mathcal{B}_{t'} \cap C_{t'}) \cap \mathcal{D}_{t-1}^c$ implies that arm L^*+1 is drawn in either round $t-1$ or round t, and thus the event

$$\bigcap_{t'=t-K}^{t-1} (\mathcal{A}_{t'} \cap \mathcal{B}_{t'} \cap C_{t'}) \cap \mathcal{D}_{t-1}^c \cap N_{L^*+1}(t) = n$$

occurs at most twice for each n. By using these we obtain

$$\begin{aligned}
&\sum_{t=K+1}^{T} \Pr\left[\bigcap_{t'=t-K}^{t-1} (\mathcal{A}_{t'} \cap \mathcal{B}_{t'} \cap C_{t'}) \cap \mathcal{D}_{t-1}^c \cap L_t \ne L^*)\right] \\
&\le K \sum_{l\in\{L^*,L^*+1\}} \sum_{t=1}^{T} \Pr[\mathcal{A}_t \cap \mathcal{G}_l^c(t)] + K + \frac{4\log T}{\Delta^2} \\
&\quad + \sum_{t=K+1}^{T} \Pr\left[b_{L^*+1}(t-1) \ge \mu_{L^*+1} + \Delta \cap N_i(t) \ge \frac{2\log T}{\Delta^2}\right] \\
&\le O(1/\Delta^2) + K + \frac{4\log T}{\Delta^2} \\
&\quad + \sum_{t=K+1}^{T} \Pr\left[b_{L^*+1}(t-1) \ge \mu_{L^*+1} + \Delta \cap N_{L^*+1}(t) \ge \frac{2\log T}{\Delta^2}\right] \\
&\quad \text{(by Lemma 8.1)} \\
&\le O(1/\Delta^2) + K + \frac{4\log T}{\Delta^2} \\
&\quad + 2 \sum_{n=\frac{2\log T}{\Delta^2}}^{T} \Pr\left[\bigcup_t \left(b_{L^*+1}^t \ge \mu_{L^*+1} + \Delta \cap N_{L^*+1}^t = n\right)\right], \quad (24)
\end{aligned}$$

and the last term is bounded as

$$\begin{aligned}
&\sum_{n=\frac{2\log T}{\Delta^2}}^{T} \Pr\left[\bigcup_t (b_{L^*+1}(t) \ge \mu_{L^*+1} + \Delta \cap N_{L^*+1}(t) = n)\right] \\
&\le \sum_{n=\frac{2\log T}{\Delta^2}}^{T} \Pr\left[n d_{\mathrm{KL}}(\hat{\mu}_{L^*+1,n}, \mu_{L^*+1} + \Delta) \le \log T\right] \\
&\le \sum_{n=\frac{2\log T}{\Delta^2}}^{T} \Pr\left[2n(\hat{\mu}_{L^*+1,n} - \mu_{L^*+1} - \Delta)^2 \le \log T\right] \\
&\quad \text{(by Pinsker's inequality)} \\
&= \sum_{n=\frac{2\log T}{\Delta^2}}^{T} \Pr\left[\hat{\mu}_{L^*+1,n} \le \mu + \Delta - \sqrt{\frac{\log T}{2n}}\right] \\
&\le \sum_{n=\frac{2\log T}{\Delta^2}}^{T} \Pr\left[\hat{\mu}_{L^*+1,n} \le \mu + \Delta/2\right] \le \sum_{n=\frac{2\log T}{\Delta^2}}^{T} \mathrm{e}^{-2n(\Delta/2)^2} = O(1/T) \\
&\quad \text{(by Hoeffding inequality).} \quad (25)
\end{aligned}$$

9 CONCENTRATION INEQUALITIES

The following inequalities are used to derive Lemma 4.6.

Lemma 9.1. (Hoeffding inequality) *Let $X_1, \ldots, X_n$ be independent random variables taking values in $[0,1]$ with mean $\mu = (1/n)\sum_i^n X_i$. Let $\hat{\mu} = (1/n)\sum_{i=1}^n X_i$. The following inequalities hold:*

$$\Pr[\hat{\mu} \ge \mu + \epsilon] \le \mathrm{e}^{-2n\epsilon^2} \text{ and } \Pr[\hat{\mu} \le \mu - \epsilon] \le \mathrm{e}^{-2n\epsilon^2}. \quad (26)$$

Lemma 9.2. (High-probability bound) *Let $\hat{\mu}_{i,n}$ be empirical estimate of μ_i at $N_i(t) = n$. For any $\epsilon > 0$ and $n_c > 0$, the following bound holds:*

$$\Pr\left[N_i(t) = n \cap \bigcup_{n=n_c}^{\infty} |\hat{\mu}_{i,n} - \mu| \ge \epsilon\right] \le \frac{1}{2\epsilon^2}\mathrm{e}^{-2n_c\epsilon^2}.$$

Lemma 9.2 is easily derived by using the Hoeffding inequality and the union bound over n.

Proof of Lemma 9.2.

$$\begin{aligned}
&\Pr\left[N_i(t) = n \cap \bigcup_{n=n_c}^{\infty} |\hat{\mu}_i(t) - \mu| \ge \epsilon\right] \\
&\le \sum_{n=n_c}^{\infty} \Pr\left[N_i(t) = n \cap |\hat{\mu}_i(t) - \mu| \ge \epsilon\right] \\
&\le \mathrm{e}^{-2n_c\epsilon^2} \sum_{n=0}^{\infty} \mathrm{e}^{-2n\epsilon^2} \text{ (by Hoeffding inequality)} \\
&= \mathrm{e}^{-2n_c\epsilon^2} \frac{\mathrm{e}^{2\epsilon^2}}{\mathrm{e}^{2\epsilon^2} - 1} \le \mathrm{e}^{-2n_c\epsilon^2} \frac{1}{2\epsilon^2}. \qquad \square
\end{aligned}$$

Lemma 9.3. (Underestimation of the KL-UCB index, Corollary 23 in [25]) *The following inequality holds: Let $\epsilon > 0$ be arbitrary. There exists constants $T_c, C_{KL} = C_{KL}(\{\mu_i\}, \epsilon)$ such that, for $t > T_c$*

$$\Pr[b_i(t) \le \mu_i - \epsilon] \le \frac{C_{KL}}{t\log t}. \quad (27)$$