

On reducing the order of arm-passes bandit streaming algorithms under memory bottleneck

Santanu Rathod¹

¹IIT-Bombay

santanusrathod@gmail.com

Abstract

In this work we explore multi-arm bandit streaming model, especially in cases where the model faces resource bottleneck. We build over existing algorithms conditioned by limited arm memory at any instance of time. Specifically, we improve the amount of streaming passes it takes for a bandit algorithm to incur a $O(\sqrt{T \log(T)})$ regret by a logarithmic factor, and also provide 2-pass algorithms with some initial conditions to incur a similar order of regret.

1 Introduction

In this paper we improve upon previous work done on regret minimization in Multi-Armed Bandit (MAB) [1] in constrained memory setting. In the multi-armed bandit setting, in general, the arms can be thought as handles in slot machines, with each handle giving a reward each time we pull it; the rewards for a particular arm follow an instance dependent but fixed distribution. The objective is to maximize the rewards accumulated in the end, or to put it in another way learn a policy to pull arms in a particular (regret minimizing or reward maximizing) manner.

With multi-arm bandits having abundant applications in healthcare, finance, dynamic-pricing models, recommender systems, etc. it makes sense to think about budgetary and resource constraints, especially in settings where the number of levers are large. In [6], the authors proposed a bandit algorithm for making online portfolio choices via exploiting correlations among multiple arms, now given that high frequency trading firms operate on large volumes of data *frequently* they can surely benefit from near optimal performance in resource constraint setting. Even in social good problems like drug-testing [8], and other practically relevant tasks like crowd-sourcing [9] that are modelled using multi armed bandits it makes economic sense to explore options that operate near optimally without storing entire statistics of the system

To that end, in [2] the authors have established an instance independent $O(\sqrt{T \log(T)})$ bound with $O(\log(T))$ streaming passes and recently [10] have established a lower

bound of $O(T^{2/3})$ for any single pass algorithm. Our objective thus was to explore the domain between $\log(T)$ and one pass, and try to minimise the regret accumulated while also reducing the number of passes to the extent that we can.

Our contributions:

- We propose a variation of Algorithm-1 in [2] which accrues $O(\sqrt{T \log(T)})$ regret with $\log(\log(T))$ passes instead of $\log(T)$ passes.
- Using the analysis devised for above result we then propose a 2-pass-algorithm, with some instance dependent initial conditions, with $O(\sqrt{T \log(T)})$ regret.
- Simulations to corroborate the results.

We also include an instance dependent two-pass-hybrid algorithm with some prior information about the system which incurs a $\log(T)$ regret. In the following sections we first start by describing the streaming model, RAM model, that we use, describe the algorithms and the key intuitions behind proving the result, and then doing same with 2-pass algorithms.

2 Related work

Right from the seminal work of [13] the predominant body of literature in stochastic multi-armed bandit is dedicated to the regret minimisation task on finite and infinite bandit instances. Later, a number of salient algorithms like UCB1 ([4]), Thompson Sampling ([14]; [15]), have been shown to achieve the order optimal cumulative regret on the finite instances. The study on multi arm bandit algorithms under constraint resources however is still limited despite its myriad practical applications today. [16] where they provide an instance dependant optimal regret with $O(1)$ storage of arms and [2] providing instance independent $\sqrt{T \log(T)}$ regret with $\log(T)$ passes. These results can however be further improved as shown in the following sections.

Since early 60s and 50s finite memory hypothesis testing has been looked at by researchers ([13]; [17]). In multi armed bandit setting [18] first presented a finite memory algorithm for two-armed Bernoulli instance, achieving an average reward which converges to optimality, with high probability. The approach consisted of a collection of

interleaved test and trial blocks, where each test block is divided into several sub-blocks and the switching among these sub-blocks is governed by a finite state machine. However, he considered only two-armed Bernoulli instances, and the approach guarantees only an asymptotic convergence of the empirical average reward. Hence, this setup is not very interesting, as our objective is to present a finite-time analysis of regret for general bandit instances.

To that end, we now present our bandit streaming setup and corresponding algorithms and analysis.

3 Preliminaries

3.1 RAM-Model

It should be noted that given any bandit instance $B = (A, D)$, as we are not considering any special structure in A, D , putting a restriction on an algorithm to use a bounded number of words of space, either restricts the horizon of pulls, or restricts the algorithm to store statistics of only bounded number of arms simultaneously. In this paper, we consider the latter and assume M to be that number. We adopt the word RAM model ([11], [12]), that considers a word as the unit of space. This model facilitates to consider that each of the input values and variables can be stored in $O(1)$ word space. For finite bandit instances ($|A| < \infty$), we consider a word to be consisted of $O(\log(T))$ bits. Therefore, our algorithm needs space-complexity of $O(M \log(T) + \log(|A|))$ bits. We call this set of arm indices whose statistics are stored as arm memory and its cardinality as arm memory size. Hence, an algorithm with arm memory size M can store the statistics of at most M arms. Also, it should be noted that an algorithm is allowed to pull an arm only if it is stored in the memory. Hence, before pulling a new arm (which is not currently in the arm memory), the algorithm should replace an arm in its arm memory with this new arm. It is interesting to note that the algorithms that work with $M = 1$, can only keep the stat of the arm it is currently pulling. Therefore, switching to a new arm costs such an algorithm to lose all the experience gained by sampling the previous arm. However, for a finite bandit instance, as the algorithms are allowed to remember all the arm indices, such an algorithm can store the gained experience by storing a bounded number of arm indices for possible further special treatment.

3.2 Simple Regret

It is one of the popular problems in *pure exploration* bandit setting which focuses on the design of strategies making the best possible use of available numerical resources (e.g., as cpu time) in order to optimize the performance of some decision-making task. If $b_t \in A$ is the arm recommended by the algorithm after t -th pull, then the simple regret at t is defined as,

$$E[r_t^*] = \mu^* - E[\mu_{b_t}] \quad (1)$$

In particular [3] studies the relationship between simple regret and cumulative regret, with results showing that

upper bounds on cumulative regret should also lead to upper bounds on simple regret.

Figure 1. from [3] shows the framework of a pure-exploration problem. Where the forecaster can be interpreted as the algorithm in consideration for the bandits, e.g. UCB1 or Thompson sampling. The forecaster then, at the end of its duration, recommends a particular arm based on some criteria, which can for example be a. Most played arm, b. Empirically best arm, or c. Arm with highest upper confidence and so forth. In this study we're concerned with most played arm(MPA) as our recommendation strategy.

Theorem 3.1 (Distribution-free upper bound on Simple-Regret of UCB-MPA by [3]). *Given a K -sized set of arms K as input, if UCB-MPA runs for a horizon of T pulls such that $T \geq K(K+2)$, then for some constant $C > 0$, it achieves*

$$\text{the expected simple regret } E[r_T^*] \leq C \sqrt{\frac{K \log(T)}{T}}.$$

Parameters: K probability distributions for the rewards of the arms, ν_1, \dots, ν_K .

For each round $t = 1, 2, \dots$,

- (1) the forecaster chooses $I_t \in \{1, \dots, K\}$;
- (2) the environment draws the reward Y_t for that action (also denoted by $X_{I_t, T_{I_t}(t)}$ with the notation introduced in the text);
- (3) the forecaster outputs a recommendation $J_t \in \{1, \dots, K\}$;
- (4) if the environment sends a stopping signal, then the game takes an end; otherwise, the next round starts.

Figure 1: The pure exploration problem for multi-armed bandits (with a finite number of arms)

From hereon \ln : logarithm with base e and \log : logarithm with base 2.

4 Multi pass algorithm

Algorithm 1, which we call UCB-LAM (UCB-limited arm memory), is presented in this section. We establish the upper-bound on cumulative regret and also the improvement in the order of passes over UCB-M proposed in [2]. The allocation strategy in UCB-LAM is UCB1 [4] for now, although it can in principle be MOSS, Thompson Sampling, etc. After exploring this multi pass algorithm we then move on to studying constant pass algorithms.

4.1 Intuition behind UCB-LAM

We know that main driving idea behind any multi-arm bandit algorithm focused on minimising regret is to sample rewards from the best arm with time. Although when faced with resource bottlenecks it's not even guaranteed that the best arm will be present in the arm memory, let alone consistently pulling it. Thus, intuitively, any algorithm that's minimizing regret under limited arm memory ought to take care that, a. The probability of the best arm being present in arm memory increases with time, b. Given (a) the best arm is sampled often. To that end we work towards UCB-LAM and also provide probabilistic analysis solidifying our intuitions, (a) and (b).

4.2 Description of UCB-LAM

In algorithm 1, UCB-LAM, we're given A : a set of K arms, $K > M \geq 2$: arm memory size. Now when $M \geq K$ we can simply run UCB1 [4] on the arms A since the arm memory can accommodate all the arms. But when $M < K$ we sample arms in phases. There are a total of x_0 phases and each phase w is divided into $h_0 = \lceil \frac{K-1}{M-1} \rceil$ sub-phases to accommodate appropriate M -sized subset of arms $S^{w,j}$ that're allowed in the memory for phase w , and sub-phase j . Once the Allocation Strategy is applied on $S^{w,j}$ the sub-phase then recommends $\hat{a} \in S^{w,j}$ to the next sub-phase $S^{w,j+1}$ or $S^{w+1,1}$.

Each of the h_0 sub-phases present in phase w are allotted a time-duration of b_w , with $b_1 = M(M+2)$ {change? Why?} and with the time allotted to a sub-phase increasing as the phase increases with $b_w = b_{w-1}^2$. The increasing time allocation with phase indicates the algorithms growing confidence in the best arm being present in arm memory and it being sampled often.

4.3 Preliminary arguments for regret analysis

We note that with K arms, M memory size, and horizon T , the number of sub-phases is $h_0 = \lceil \frac{K-1}{M-1} \rceil$ and the total number of phases or arm-passes is upper bounded by $x_0 = 1 + \lceil \log(\log_{M(M+2)}(\frac{T}{h_0})) \rceil$ (Lemma A.1 in Appendix A).

As we'll further see in our regret bifurcation and analysis further ahead, we need to upper-bound the mean of the recommended arms between two consecutive sub-phases. Let $a_*^{y,j} \in S^{y,j}$ be the arm recommended by sub-phase j to $j+1$, then we need to bound $r^{y,j} = \mu_*^{y,j-1} - \mu_*^{y,j}$. Since we know that, $\mu_*^{y,j-1} \leq \max_{a \in S^{y,j}} \mu_a$, we can write

$$r^{y,j} = \mu_*^{y,j-1} - \mu_*^{y,j} = \max_{a \in S^{y,j}} \mu_a - \mu_*^{y,j} - \epsilon$$

We thus get:

Corollary 4.0.1. *Using Theorem 3.1 in phase y , at the end of each sub-phase j , the approximate simple regret with respect*

to $\mu_{a_^{y,j}}$ is upper bounded as $E[r^{y,j}] \leq C \sqrt{\frac{K \log(b_y)}{b_y}}$.*

We also need to be cognizant of the fact that the arm recommended by a sub-phase might not be the optimal arm, because in the worst case scenario the allocation strategy will only be sampling sub-optimal arms and thus incurring a huge regret. It is thus imperative to bound the approximate simple regret and to take care that optimal arm is included the arm-memory, and once included, it is also the arm that's recommended to the next sub-phase with high probability. We'll come across these arguments naturally while doing the regret analysis.

Lemma 4.1. *Consider events,*

1. A : Best arm(universal) is in the current (w, j) instance.
2. B : Best arm(universal) is recommended to next instance $((w, j+1)$ or $(w+1, 1))$.

Let $s = (w, j)$ be the current instance with phase w and sub-phase j , and let r be the arm recommended to the next instance $(w, j+1)$ or $(w+1, 1)$ and let $\mu_*^{w,j}$ be the mean of that arm. Then if

- $T_1 = P(A_s = 1) \times \{b_w(E[\mu^* - \mu_*^{w,j}])P(B_s = 0|A_s = 1)\}$
- $T_2 = P(A_s = 0) \times \{b_w(E[\mu^* - \mu_*^{w,j}])\}$

We have,

$$\frac{T_1 + T_2}{b_w} \leq 2C \frac{(M-1) \cdot h_0}{(b_{w-1}/M-1)^2} (h_0 + 1) \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}}$$

The proof is presented in the Appendix A. Next we bifurcate the regret appropriately and use the above results to find the upper bound the cumulative regret (R_T^*)

Bifurcation of R_T^* . For any given phase w , and a sub-phase j , let $\mu_*^{w,j}$ = mean of the most played arm a for $a \in S^{w,j}$, and $R_{w,j}$ be the incurred regret. Then,

$$\begin{aligned} R_{w,j} &= b_w \mu^* - \sum_{t=1}^{b_w} E[\mu_{a_t}] \\ &= b_w E[\mu^* - \mu_*^{w,j}] + \sum_{t=1}^{b_w} (E[\mu_*^{w,j}] - E[\mu_{a_t}]) \end{aligned}$$

Now let $R_{w,j}^{(1)} = b_w(\mu^* - \mu_*^{w,j})$, and $R_{w,j}^{(2)} = \sum_{t=1}^{b_w} (E[\mu_*^{w,j}] - E[\mu_{a_t}])$, we write,

$$R_T^* = \sum_{w=1}^{x_0} \sum_{j=1}^{h_0} R_{w,j} = \sum_{w=1}^{x_0} \sum_{j=1}^{h_0} (R_{w,j}^{(1)} + R_{w,j}^{(2)}) \quad (2)$$

Having bifurcated cumulative regret R_T^* in terms of R^1, R^2 we can see that minimizing R^1 would essentially mean that with high probability, as we go further in phases, the best arm in memory is actually the optimal arm. As can be seen from **Lemma A.2** for phase w , the $P(\text{best arm is recommended}) = P(J_{b_w} = i^*) \geq 1 - \frac{K-1}{\alpha-1} (b_w/K-1)^{2(\alpha-1)}$, for $\alpha = 2$, increases with phase, which is also the key argument that we use while proving **Lemma 4.2** below.

Lemma 4.2. *For $2 \leq M < K$, and for $T \geq KM^2(M+2)$, let $R^{(1)} = \sum_{w=1}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(1)}$, then,*

$$R^1 \leq C_0 + C_2 \times \log(\log_{b_1}(\frac{T}{h_0}))$$

where C_2 is a constant depending on K, M .

For the proof, refer Appendix A. For calculating $\sum_{w=1}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(2)}$, can be upper-bounded using the problem independent upper-bound on the cumulative regret of UCB1 [4], which we restate below.

And R^2 can be interpreted as a local cumulative regret, or the regret accrued with respect to the best arm present in a particular memory instance. In the worst case scenario we expect the arms to be present in increasing order of means, and thus the local regret then would thus increase linearly with instances and will be within a certain factor of the local cumulative of the instance containing best arm. This what **Lemma 4.4** puts mathematically.

Lemma 4.3. (Distribution-Free Upper Bound on Cumulative Regret of UCB1 [4]). Given a set of K -arms as the input, for any horizon T , the cumulative regret incurred by UCB1 $R_T^* \leq 12\sqrt{TK \log(T)} + 6K$. Further, if $T \geq K/2$, then $R_T^* \leq 18\sqrt{TK \log(T)}$.

Next using Lemma 4.3 we upper bound $\sum_{w=1}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(2)}$, proof is given in Appendix A.

Lemma 4.4. For $2 \leq M < K$, and for $T \geq KM^2(M+2)$, let, $R^{(2)} = \sum_{w=1}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(2)}$, then,

$$R^{(2)} \leq C_0 + C_4 \sqrt{(\log_{b_1}(\frac{T}{h_0})) \cdot (\frac{T}{h_0})}$$

Where C_0, C_4 are constants depending on K and M .

Theorem 4.5 (Main result). For $2 \leq M < K$, and for $T \geq KM^2(M+2)$, let $R^{(1)} = \sum_{w=1}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(1)}$ and $R^{(2)} = \sum_{w=1}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(2)}$, so for cumulative regret $R_T^* = R^{(1)} + R^{(2)}$ we have,

$$R_T^* \leq 2C_0 + C_2 \times (\log(\log_{b_1}(\frac{T}{h_0}))) + C_4 \sqrt{(\log_{b_1}(\frac{T}{h_0})) \cdot \frac{T}{h_0}}$$

where C_0, C_1, C_4 are functions of K, M .

Proof. One can easily see how combining bounds from **Lemma 4.2** and **Lemma 4.4** will give us the required upper-bound on cumulative regret of **Algorithm 1**. \square

5 Constant pass algorithms

Now that we've introduced **UCB-LAM** which gives us $O(\sqrt{T \log(T)})$ cumulative regret in $O(\log(\log(T)))$ passes, we further explore constant pass algorithms in similar spirit. Decreasing the number of passes is practically very desirable since it frees up the memory reservoir required to store the dormant arms not being considered by the allocation strategy at any instance{come up with better motivation}, and in extension having *constant* number of passes will aid in determining when a section of memory space will be freed (after c passes) and thereby enabling us to better plan the use of the memory as opposed to the case where the number of passes is dependant on the time horizon thereby restricting the scope of any prior planning for the memory in consideration.

Algorithm 1 UCB-LAM(limited arm memory)

Input: A : the set of K arms indexed by $[K]$, $M(\geq 2)$: Arm memory size

```

1: if  $M \geq K$  then
2:   Run UCB1 on  $A$  until horizon
3: else
4:    $b_1 = M(M+2)$ . {Initial horizon per sub-phase}
5:    $\hat{a} = 1$ . {Initial arm recommendation}
6:    $w = 1$ . {Counts the number of phases}
7:    $h_0 = \lceil \frac{K-1}{M-1} \rceil$ . {The number of sub-phases in a phase}
8:   No random shuffling for now
9:   while the horizon is not finished do
10:     $l = 0$ 
11:    for  $j = 1, \dots, h_0$ ; if the horizon is not finished do
12:       $S^{w,j} = \{l+1, \dots, \min\{l+1+(M-1), K\}\}$ 
13:       $l =$  The highest arm index in  $S^{w,j}$ .
14:      if  $\hat{a} \notin S^{w,j}$  then
15:         $S^{w,j} = \{\hat{a}\} \cup S^{w,j} \setminus \{l\}$ .
16:         $l = l - 1$ 
17:      end if
18:      {ALLOCATION STRATEGY}
19:      Run UCB1 on  $S^{w,j}$  for horizon of  $b_w$  pulls or the remaining horizon; whichever is smaller.
20:      {RECOMMENDATION STRATEGY}
21:       $\hat{a} =$  The most played arm in  $S^{w,j}$ 
22:    end for
23:     $w = w + 1$ . {Increment phase count}
24:     $b_w = (b_{w-1})^2$ . {Increment horizon per sub-phase}
25:  end while
26: end if

```

5.1 2-pass UCB-LAM

2-pass UCB-LAM is an extension of UCB-LAM when restricted to two passes. The with the only difference being for us to know the time horizon beforehand – which is obvious. Since when we say that a certain algorithm takes 2-passes, we ought to know *when the first pass ends* and *when the second pass begins*.

From **Lemma B.1** we know for sure that the total number of passes required by **Algorithm-2** is two. Having said that we now put across the regret analysis and argument for **Algorithm-2**.

Bifurcation of regret

We know that the regret obtained from **Algorithm 2** R_T^* can be bifurcated as, following steps as we did the regret bifurcation of **Algorithm-1**-

For any given phase w , and a sub-phase j , let $\mu_*^{w,j}$ = mean of the most played arm a for $a \in S^{w,j}$, and $R_{w,j}$ be the incurred regret, Then,

Algorithm 2 2-pass UCB-LAM

Input: A : the set of K arms indexed by $[K]$, $M(\geq 2)$: Arm memory size, T : Total time horizon, large enough such that $T \geq K * (1 + \frac{4\alpha \ln(T)}{M\Delta_i^2})$

```
1: if  $M \geq K$  then
2:   Run UCB1 on  $A$  until horizon
3: else
4:    $\hat{a} = 1$ . { Initial arm recommendation }
5:    $w = 1$ . { Counts the number of phases }
6:    $h_0 = \lceil \frac{K-1}{M-1} \rceil$ . { The number of sub-phases in a phase }
7:    $b_1 = \frac{\sqrt{1 + 4\frac{T}{h_0}} - 1}{2}$ 
8:    $b_2 = \frac{2\frac{T}{h_0} + 1 - \sqrt{1 + 4\frac{T}{h_0}}}{2} = b_1^2$ 
9:   while the horizon is not finished do
10:     $l = 0$ 
11:    for  $j = 1, \dots, h_0$ ; if the horizon is not finished do
12:       $S^{w,j} = \{l+1, \dots, \min\{l+1+(M-1), K\}\}$ 
13:       $l =$  The highest arm index in  $S^{w,j}$ .
14:      if  $\hat{a} \notin S^{w,j}$  then
15:         $S^{w,j} = \{\hat{a}\} \cup S^{w,j} \setminus \{l\}$ .
16:         $l = l - 1$ 
17:      end if
18:      { ALLOCATION STRATEGY }
19:      Run UCB1 on  $S_{w,j}$  for horizon of  $b_w$  pulls or the remaining horizon; whichever is smaller.
20:      { RECOMMENDATION STRATEGY }
21:       $\hat{a} =$  The most played arm in  $S^{w,j}$ 
22:    end for
23:     $w = w + 1$ . { Increment phase count }
24:  end while
25: end if
```

$$\begin{aligned} R_{w,j} &= b_w \mu^* - \sum_{t=1}^{b_w} E[\mu_{a_t}] \\ &= b_w E[\mu^* - \mu_*^{w,j}] + \sum_{t=1}^{b_w} (E[\mu_*^{w,j}] - E[\mu_{a_t}]). \end{aligned}$$

Now let $R_{w,j}^{(1)} = b_w(\mu^* - \mu_*^{w,j})$, and $R_{w,j}^{(2)} = \sum_{t=1}^{b_w} (E[\mu_*^{w,j}] - E[\mu_{a_t}])$, we write,

$$R_T^* = \sum_{w=1}^2 \sum_{j=1}^{h_0} R_{w,j} = \sum_{w=1}^2 \sum_{j=1}^{h_0} (R_{w,j}^{(1)} + R_{w,j}^{(2)})$$

We now deal with R_1 and R_2 separately as we did earlier.

Lemma 5.1. For $2 \leq M < K$, and total time-horizon T , we get that, for Algorithm-2,

$$R_2 \leq C_2 + C_0 \sqrt{T + 0.25h_0} + C_1 \sqrt{T \log(T/h_0)}$$

Where C_0, C_1, C_2 are constants depending on M, K .

As we can see the order of $R_2 - \text{regret}$ accrued by **Algorithm-2** almost same as the order of $R_2 - \text{regret}$ accrued by **Algorithm-1**. Essentially R_2 here can be interpreted as the summation of all the *local* regrets, where local means being limited to one's memory instance without being aware of all the arms. We now move onto R_1 .

Lemma 5.2. For $2 \leq M < K$, and total time-horizon T , we get that, for Algorithm-2,

$$R_1 = \sum_{w=1}^2 \sum_{j=1}^{h_0} (R_{w,j}^{(1)}) \leq O(\sqrt{T})$$

{refer Appendix for the proof}. R^1 here can be physically interpreted as the penalty we accrue as the result of recommending sub-optimal arm instead of the best arm. While there can be several reasons for this, like a. Best arm is not present in the current memory instance, b. Best arm is present but isn't recommended for some reason, and all of these factors are analysed for in the proofs. {comment on how R_1 here differs from R_1 for algorithm-1}

Theorem 5.3. Given a set of K arms A with $K \leq 3$, an arm memory of size M , and the total time horizon T Algorithm-2 will incur a cumulative regret $R_T^* = O(A_1 \sqrt{T} + A_2 \sqrt{T \log(T)})$, where A_1, A_2 are constants depending on K, M .

Proof. For phase w , and sub-phase j , we know that, $R_{w,j}^{(1)} = b_w(\mu^* - \mu_*^{w,j})$, and $R_{w,j}^{(2)} = \sum_{t=1}^{b_w} (E[\mu_*^{w,j}] - E[\mu_{a_t}])$, and we write,

$$R_T^* = \sum_{w=1}^2 \sum_{j=1}^{h_0} R_{w,j}^* = \sum_{w=1}^2 \sum_{j=1}^{h_0} (R_{w,j}^{(1)} + R_{w,j}^{(2)})$$

And from **Lemma 5.1** we know that

$$R_2 \leq C_2 + C_0 \sqrt{T + 0.25h_0} + C_1 \sqrt{T \log(T/h_0)}$$

,and from **Lemma 5.2** we know that $R_1 \leq O(\sqrt{T})$, thus we get,

$$R_T^* = R_1 + R_2 \leq O(A_1 \sqrt{T} + A_2 \sqrt{T \log(T)}) \quad (3)$$

where A_1, A_2 are constants depending on K, M . \square

5.2 Constant pass algorithm from a kind-of explore-exploit perspective

As we've noted in the sections above, to accrue lesser regret in resource bottleneck settings intuitively what any algorithm tries to do is maximize the probability of the best arm being in the arm memory and being recommended often. In this subsection we explore a relatively simple algorithm that performs somewhat better than **2-pass UCB-LAM** theoretically speaking, building up on exactly the points we mentioned.

We'll first start with a **pseudo algorithm** and then based on our analysis build up **Algorithm-4**.

Algorithm 3 (pseudo) 2-pass-hybrid

Input: A : the set of K arms indexed by $[K]$, $M(\geq 2)$: Arm memory size, T : Total time horizon

- 1: **1st Pass:**
 - 2: Uniformly play all the arms (details later)
 - 3: Recommend the arm with highest $\bar{\mu}$
 - 4: **2nd Pass:**
 - 5: Run UCB [4] for the current sub-phase for time b_2
 - 6: Recommend the most played arm to the next sub-phase
 - 7: Keep doing it for the remaining time
-

Description of (pseudo) 2-pass-hybrid

For the set of K arms A , arm memory $M \geq 2$, and total time horizon T , a. In the first pass we sample all arms for an equal time-horizon, the main idea is to get an estimate of empirically best arm, b. Recommend the empirically best arm to the second pass, and in the second pass run the similar allocation strategy that we've used in the previous algorithms, i.e. bifurcate the memory appropriately and recommend the most played arm to the next memory instance. The underlying idea being that once the best arm is being captured, it'll be further recommended with high probability. And since the the probability of making a mistake on the best arm is very low the expected regret will have reasonable bound order.

We find the necessary conditions for our **(pseudo) 2-pass-hybrid** to perform optimally and build **Algorithm-4** out of it.

Lemma 5.4. For given set A of K arms, M arm memory, $\Delta_{\min} = \min_{i \in A} \{\mu^* - \mu^i\}$, total time horizon T , then

- The optimal total time duration($h_0 b_0$) spent in 1st pass by **Algorithm-3** is,

$$h_0 b_1 = \frac{h_0}{\Delta_{\min}^2} \log(1 + \frac{\Delta_{\min}^2}{K} T^2)$$

$$\text{where } h_0 = \lceil \frac{K-1}{M-1} \rceil.$$

- And for the above optimal time duration spent in the first pass, the cumulative regret accrued (R_T^*)

$$R_T^* = \frac{K}{\Delta_{\min}^2} \log(f(T)) \times (1 - \frac{1}{f(T)}) + \frac{T}{f(T)}$$

$$\text{where } f(T) = 1 + \frac{\Delta_{\min}^2}{K} T^2$$

Proof. Let X_i^k be the reward drawn from arm i at time k , therefore $\bar{\mu}_i = \frac{\sum_{k=1}^{b_0} X_i^k}{b_1}$. We thus want a bound on $P_{i \neq i^*}(\bar{\mu}_i \geq \bar{\mu}_{i^*})$ or that sub-optimal arm is recommended. Let E_i be the event such that $\{\bar{\mu}_i \geq \bar{\mu}_{i^*}\}$

$$\begin{aligned} P(E_i) &= P(\bar{\mu}_{i^*} - \bar{\mu}_i - (\Delta_i) \leq -\Delta_i) \\ &= P(\frac{\sum X_{i^*}^j - \sum X_i^j}{b_1} - \Delta_i \leq -\Delta_i) \\ &= P(\sum X_{i^*}^j - \sum X_i^j - b_0 \Delta_i \leq -b_0 \Delta_i) \end{aligned}$$

Let $S_{2n} = \sum X_{i^*}^j - \sum X_i^j$, $t = -b_0 \Delta_i$, so the above inequality can be rewritten as $P(S_{2n} - E[S_{2n}] \leq -t)$, and thus using Hoeffding's inequality we get:

$$\begin{aligned} P(S_{2n} - E[S_{2n}] \leq -t) &\leq \exp(\frac{-2t^2}{2b_1}) \\ P(S_{2n} - E[S_{2n}] \leq -t) &\leq \exp(\frac{-2(b_0^2)(\Delta_i^2)}{2b_1}) \\ \implies P(S_{2n} - E[S_{2n}] \leq -t) &\leq \exp(-\Delta_i^2 b_1) \\ \implies P(E_i) &\leq \exp(-\Delta_i^2 b_1) \end{aligned}$$

From above we now know that $P(E_i) \leq \exp(-\Delta_i^2 b_1)$. Now let \overline{BA} be the event such that the recommended arm is **not** the best arm or $P(\overline{BA}) = 1 - P(BA) = 1 - P(\cap_{i \in [K]} \{\overline{E_i}\})$, where E_i is the event that i th arm is the recommended arm. Thus we get,

$$1 - P(\cap_{i \in [K]} \{\overline{E_i}\}) = 1 - \prod_{i=1}^K (1 - P(E_i)) \quad (4)$$

Since we know that $P(E_i) \leq \exp(-\Delta_i^2 b_1)$

$$\begin{aligned} P(\overline{BA}) &\leq 1 - \prod_{i=1}^K (1 - \exp(-b_1 \Delta_i)) \\ &\leq 1 - \prod_{i=1}^K (1 - \exp(-b_1 \Delta_{\min})) \\ &\leq 1 - (1 - K \times \exp(-b_1 \Delta_{\min})) \\ &\leq K \times \exp(-b_1 \Delta_{\min}) \end{aligned}$$

Thus, we get that the $P(\text{The best arm isn't recommended}) = P(\overline{BA}) \leq K \times \exp(-b_1 \Delta_{\min})$ or that the $P(\text{best arm is recommended}) = P(BA) \geq 1 - K \times \exp(-b_1 \Delta_{\min})$

Calculating and optimizing the regret: We'll calculate the total regret by bifurcating it in two passes. We know that for the first pass the regret accrues linearly for the worst case, or:

$$R_1 \leq K b_1 \quad (5)$$

For the second pass however, it's a bit involved. Let BA be the event that best arm is recommended by first pass at the beginning of the second pass, then we have:

$$R_2 \leq (1 - P(BA)) \times (h_0 b_2) + P(BA) \times O(h_0 \sqrt{\frac{\log(b_2)}{b_2}}) \quad (6)$$

Which can be easily derived from using Lemma 4.4 in [2] and arguing that in the worst case regret will be accrued linearly. For the sake of simplicity we ignore the second term in equation (3), and using the expression for $P(\overline{BA}) \leq K \times \exp(-b_1 \Delta_{\min})$ which we derived in the earlier section

of the analysis, we convert our argument into a simple optimization problem where we want to find:

$$b_{1,\text{opt}}, b_{2,\text{opt}} = \underset{b_1, b_2}{\operatorname{argmin}} (K \times b_1 + e^{-b_1 \Delta_{\min}^2} \times (h_0 b_1)) \quad (7)$$

Such that, $T = K b_1 + h_0 b_2$.

Let $R = K \cdot b_1 + e^{-b_1 \Delta_{\min}^2} \cdot (h_0 b_2)$, substituting b_2 in terms of b_1 we get,

$$R = K b_1 + e^{-b_1 \Delta_{\min}^2} (T - K b_1) \quad (8)$$

Differentiating wrt b_1 we get:

$$\frac{dR}{db_1} = 0 \implies K + e^{-b_1 \Delta_{\min}^2} (-K) + \quad (9)$$

$$e^{-b_1 \Delta_{\min}^2} (T - K b_1) (-2 b_1 \Delta_{\min}^2) = 0 \quad (10)$$

$$\implies K = e^{-b_1 \Delta_{\min}^2} (K + b_1 \Delta_{\min}^2 (T - K b_1)) \quad (11)$$

$$\implies e^{b_1 \Delta_{\min}^2} = \frac{K}{K + b_1 \Delta_{\min}^2 (T - K b_1)} \quad (12)$$

$$\implies e^{b_1 \Delta_{\min}^2} \geq \frac{K}{K + b_1 T \Delta_{\min}^2} \quad (13)$$

$$\implies e^{b_1 \Delta_{\min}^2} \geq \frac{K}{K + \Delta_{\min}^2 T^2} \quad \{b_1 < T\} \quad (14)$$

$$\implies -b_1 \Delta_{\min}^2 \geq \log\left(\frac{K}{K + \Delta_{\min}^2 T^2}\right) \quad (15)$$

$$\implies b_1 \leq \frac{1}{\Delta_{\min}^2} \log\left(1 + \frac{\Delta_{\min}^2}{K} T^2\right) \quad (16)$$

Thus we get (a).

$$b_1 = \frac{1}{\Delta_{\min}^2} \log\left(1 + \frac{\Delta_{\min}^2}{K} T^2\right)$$

as the optimal time spent in the first pass.

Substituting the expression for $b_1 = \frac{1}{\Delta_{\min}^2} \log\left(1 + \frac{\Delta_{\min}^2}{K} T^2\right)$ in equation (6) and letting $f(T) = 1 + \frac{\Delta_{\min}^2}{K} T^2$ we get, (b).

$$R_T^* = \frac{K}{\Delta_{\min}^2} \log(f(T)) \times \left(1 - \frac{1}{f(T)}\right) + \frac{T}{f(T)}$$

□

Using **Lemma 5.4** we now propose a 2-pass **Algorithm-4**.

6 Simulations

We performed simulations to compare the performances of UCB-M from [2], UCB-LAM above, and standard UCB1. We set parameters to be $K=30$, $M=4$, and the bandit arms had means varying as $\mu_i = 0.99 - 0.1 * i$; $i \in [0, K]$ with each arm i following a Bernoulli distribution with mean μ_i . The order

Algorithm 4 2-pass-hybrid

Input: A : the set of K arms indexed by $[K]$, $M(\geq 2)$: Arm memory size, T : Total time horizon, $\Delta_{\min} = \min_i \{\mu^* - \mu_i\}$

```

1: if  $M \geq K$  then
2:   Run UCB1 on  $A$  until horizon
3: else
4:    $\hat{a} = 1$ . { Initial arm recommendation}
5:    $w = 1$ . {Counts the number of phases}
6:    $h_0 = \lceil \frac{K-1}{M-1} \rceil$ . { The number of sub-phases in a phase}
7:    $b_1 = \frac{1}{\Delta_{\min}^2} \log\left(1 + \frac{\Delta_{\min}^2}{K} T^2\right)$ 
8:    $b_2 = T - b_1$ 
9:   while the horizon is not finished do
10:     $l = 0$ 
11:    for  $j = 1, \dots, h_0$ ; if the horizon is not finished do
12:       $S^{w,j} = \{l+1, \dots, \min\{l+1+(M-1), K\}\}$ 
13:       $l =$  The highest arm index in  $S^{w,j}$ .
14:      if  $\hat{a} \notin S^{w,j}$  then
15:         $S^{w,j} = \{\hat{a}\} \cup S^{w,j} \setminus \{l\}$ .
16:         $l = l - 1$ 
17:      end if
18:      if  $w=1$  then
19:        {ALLOCATION STRATEGY}
20:        Play each arm in  $S^{w,j}$   $b_1$  times.
21:        {RECOMMENDATION STRATEGY}
22:         $\hat{a} =$  empirically best arm. Or  $\hat{a} = \max_i \mu_i$ 
23:      else
24:        {ALLOCATION STRATEGY}
25:        Run UCB1 on  $S^{w,j}$  for horizon of  $b_w$  pulls or the remaining horizon; whichever is smaller.
26:        {RECOMMENDATION STRATEGY}
27:         $\hat{a} =$  The most played arm in  $S^{w,j}$ 
28:      end if
29:    end for
30:     $w = w + 1$ . {Increment phase count}
31:  end while
32: end if

```

of arrival was randomised and simulation shows the average over 10 simulations.

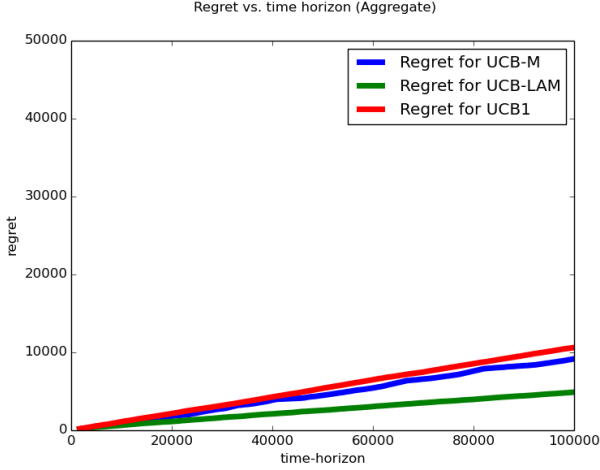


Figure 2: Aggregate regret vs. Time horizon compared between UCB-M, UCB-LAM, UCB1

We thus observe that the order of regret is comparable across the algorithms, with UCB-LAM having $O(\log(\log(T)))$ number of passes over $O(\log(T))$ number of passes of UCB-M.

7 Conclusion

We’ve thus explored the gap we set out to explore, which is the regret bound behaviour between one-pass [10] and $\log(T)$ passes. We’ve shown that in instance independent setting with no prior information about the system, $O(\sqrt{T \log(T)})$ regret can be achieved with simply $\log(\log(T))$ passes instead of $\log(T)$ passes. We’ve now also know that under large enough time horizons it’s possible to achieve $O(\sqrt{T \log(T)})$ regret with simply two passes. It makes sense intuitively because with the number of passes limited to two we want to learn the best arm behaviour as accurately as possible. However the behaviour of regret in smaller time horizons with constant number of passes needs to be explored further and we defer that to future studies.

8 Acknowledgement

This work wouldn’t have been possible without timely feedback and discussions from Vishakha Patil (PhD student IISc Bangalore) and Dr. Arindam Khan (Assistant Professor, IISc Bangalore), especially with regards to setting the over-arching goal of exploring the domain between constant passes and $O(\log(T))$ passes.

9 Appendix

A Proofs for Section 4

Lemma A.1. For a given K -sized set of arms A , and an arm memory size $M < K$, the number of phases **Algorithm 1** executes is upper bounded by $x_0 = 1 + \lceil \log(\log_{M(M+2)}(\frac{T}{h_0})) \rceil$.

Proof. Firstly we’d like to make it clear that there exists $\{K, M\}$ time horizon: T such that the time horizon is complete while the algorithm is in between a particular phase. We have $b_w = (b_{w-1})^2$ or the length of each sub-phase in phase w varies as $\{b_1^{2^{1-1}}, b_1^{2^{2-1}}, b_1^{2^{3-1}}, \dots, b_1^{2^{x_0-1}}\}$, where x_0 is the total number of phases or total number of passes. Now,

$$T = \sum_{w=1}^{x_0} \sum_{j=1}^{h_0} b_{w,j} \quad (17)$$

$$= \sum_{w=1}^{x_0} h_0 b_w \dots \dots \{b_w = b_{w,1} = \dots = b_{w,h_0}\} \quad (18)$$

$$\Rightarrow T \geq h_0 b_{x_0} \quad (19)$$

$$\Rightarrow T \geq h_0 (b_1^{2^{x_0-1}}) \quad (20)$$

$$\Rightarrow \log_{b_1}(\frac{T}{h_0}) \geq 2^{x_0-1} \quad (21)$$

$$\Rightarrow \log \log_{b_1}(\frac{T}{h_0}) \geq x_0 - 1 \quad (22)$$

Since $b_1 = M(M+2)$

$$\Rightarrow x_0 \leq \lceil \log \log_{M(M+2)}(\frac{T}{h_0}) \rceil + 1 \quad (23)$$

Hence proved. \square

Lemma A.2. Consider events,

1. A : Best arm(universal) is in the current (w, j) instance.
2. B : Best arm(universal) is recommended to next instance $((w, j+1)$ or $(w+1, 1))$.

And let J_{b_w} be the arm recommended by the recommendation strategy for **UCB-LAM**. Then for a particular instance $s = (w, j)$, M : arm-memory and b_w : time-horizon, $P(B_s = 1 | A_s = 1) = P(J_{b_w} = i^*) \geq 1 - \frac{M-1}{(b_w/M-1)^2}$

Proof. Using a side-result from **Lemma 1** in [3] we get that whenever the most played arm J_{b_w} is different from the optimal arm i^* then at least one of the suboptimal arms i is such that $T_i(b_w) \geq a_i b_w$, where a_1, \dots, a_K are real numbers such that $a_1 + a_2 + \dots + a_K = 1$ and $a_i \geq 0, \forall i$.

And that $P\{T_i(b_w) \geq a_i b_w\} \leq \frac{1}{\alpha-1} (a_i b_w - 1)^{2(1-\alpha)}$, where T_i : number of pulls for arm i and α refers to $UCB(\alpha)$ algorithm used during allocation strategy.

Now, we know that:

$$P(J_{b_w} = i^*) + (\sum_{i \in A \setminus \{i^*\}} P(J_{b_w} = i)) = 1$$

$$\Rightarrow P(J_{b_w} = i^*) = 1 - \sum_{i \in A \setminus \{i^*\}} P(J_{b_w} = i)$$

When i is the most played arm or ($J_{b_w} = i$) we know that $T_i \geq b_w/K$ and from **Lemma A.5** which is derived from from [3] we know that

$$\begin{aligned}
P_{i \neq i^*}(J_{b_w} = i) &\leq \frac{1}{\alpha - 1} (b_w/K - 1)^{2(1-\alpha)} \\
\Rightarrow 1 - \sum_{i \in A \setminus \{i^*\}} P(J_{b_w} = i) &\geq \\
(1 - \sum_{i \in A \setminus \{i^*\}} \frac{1}{\alpha - 1} (b_w/K - 1)^{2(\alpha-1)}) & \\
\Rightarrow P(J_{b_w} = i^*) &\geq 1 - \frac{K-1}{\alpha-1} (b_w/K - 1)^{2(\alpha-1)}
\end{aligned}$$

Substituting arm memory: $M, \alpha = 2$ for our case, we get $P(J_{b_w} = i^*) \geq 1 - \frac{M-1}{(b_w/M-1)^2}$

□

Corollary A.2.1. Consider events,

1. A : Best arm(universal) is in the current (w, j) instance.
2. B : Best arm(universal) is recommended to next instance $((w, j+1)$ or $(w+1, 1))$.

For an instance $s = (w, j)$ the probability of the optimal arm not being present in the current arm memory, $P(A_s = 0) \leq \frac{(M-1) \cdot h_0}{(b_{w-1}/M-1)^2}$

Proof. From **Lemma A.2** above we know that (probability of recommending best-arm)

$$P(B_s = 1 | A_s = 1) \geq 1 - \frac{M-1}{(b_w/M-1)^2}.$$

Let $t_w = b_w/M$, and $m = (w-1, k')$ be the memory instance in the previous phase $w-1$ containing the universal best arm a^* .

We know that $P(A_s = 1) = \Pi_{m=(w-1, k')}^{(w, j-1)} P(B_m = 1 | A_m = 1)$, because the probability that the current instance will have the best arm depends on the fact that previous instance had the best arm which it passed on – all the way to the instance $(w-1, k)$ which actually had the best arm. Thus we get,

$$\begin{aligned}
\Pi_{m=(w-1, k')}^{(w, j-1)} P(B_m = 1 | A_m = 1) &\geq \\
\Pi_{m=(w-1, k')}^{(w, j-1)} (1 - \frac{M-1}{(t_m-1)^2}) &
\end{aligned}$$

We're considering the worst case scenario here when the best arm lies in the previous phase $w-1$. As going from $m = (w-1, k')$ to $(w, j-1)$ has a phase change, $w-1$ to

w , we bifurcate the expression in two terms, a. considering phase $w-1 : k_1 = h_0 + 1 - k'$ and considering phase $w : k_2 = j-1$.

$$\begin{aligned}
\Rightarrow \Pi_{m=(w-1, k')}^{(w, j-1)} P(B_m = 1 | A_m = 1) &\geq \\
(1 - \frac{M-1}{(t_{w-1}-1)^2})^{k_1} \times (1 - \frac{M-1}{(t_w-1)^2})^{k_2} &
\end{aligned}$$

Where: $k_1 + k_2 \leq h_0$. Thus,

$$\begin{aligned}
1 - \Pi_{m=(w-1, k')}^{(w, j-1)} P(B_m = 1 | A_m = 1) &\leq \\
1 - (1 - \frac{M-1}{(t_{w-1}-1)^2})^{k_1} \times (1 - \frac{M-1}{(t_w-1)^2})^{k_2} & \\
\Rightarrow P(A_s = 0) &\leq \\
1 - (1 - \frac{M-1}{(t_{w-1}-1)^2})^{k_1} \times (1 - \frac{M-1}{(t_w-1)^2})^{k_2} &
\end{aligned}$$

$$\Rightarrow P(A_s = 0) \leq 1 - (1 - \frac{M-1}{(t_{w-1}-1)^2})^{h_0}$$

The above expression is valid even in the case where the best arm lies in phase w in instance (w, k') , instead of $w-1$, with $j > k'$. Primarily because $b_{w-1} < b_w$.

Using binomial expansion and $t_w = b_w/M$ thus we get,

$$P(A_s = 0) \leq \frac{(M-1) \cdot h_0}{(b_{w-1}/M-1)^2}$$

□

The significance of above lemmas is that they show, theoretically, that as we move ahead in time the probability that the best arm will be contained in the memory increases drastically which is what we guess intuitively.

Lemma A.3. Consider events,

1. A : Best arm(universal) is in the current (w, j) instance.
2. B : Best arm(universal) is recommended to next instance $((w, j+1)$ or $(w+1, 1))$.

Let $s = (w, j)$ be the current instance with phase w and sub-phase j , and let r be the arm recommended to the next instance $(w, j+1)$ or $(w+1, j)$ and let $\mu_*^{w, j}$ be the mean of that arm. Then $E[\mu^* - \mu_*^{w, j}]$ can be expressed in terms of T_1, T_2 or $b_w E[\mu^* - \mu_*^{w, j}] = T_1 + T_2$, where:

- $T_1 = P(A_s = 1) \times \{b_w(E[\mu^* - \mu_*^{w, j}])P(B_s = 0 | A_s = 1)\}$
- $T_2 = P(A_s = 0) \times \{b_w(E[\mu^* - \mu_*^{w, j}])\}$

Proof. Consider given events:

1. A : Best arm(universal) is in the current (w, j) instance.

2. B: Best arm(universal) is recommended to next instance $((w, j+1)$ or $(w+1, 1)$).

We know that:

- $P(B=1|A=0)=0$
- $P(B=0|A=0)=1$
- $P(B=1|A=1)=x(\text{unknown})$
- $P(B=0|A=1)=1-x$

Now we know that $R_{w,j}^1 = b_w E[\mu^* - \mu_*^{w,j}]$.

Let \mathbf{s} : current memory instance (w, j) . We know that $E[E[X]] = E[X]$, thus we write,

$$\begin{aligned} R_{w,j}^1 &= P(A_s = 1) \cdot \{b_w(E[\mu^* - \mu_*^{w,j}])P(B_s = 1|A_s = 1)\} \\ &\quad + P(A_s = 1) \cdot \{b_w(E[\mu^* - \mu_*^{w,j}])P(B_s = 0|A_s = 1)\} \\ &\quad + P(A_s = 0) \cdot \{b_w(E[\mu^* - \mu_*^{w,j}])P(B_s = 1|A_s = 0)\} \\ &\quad + P(A_s = 0) \cdot \{b_w(E[\mu^* - \mu_*^{w,j}])P(B_s = 0|A_s = 0)\} \end{aligned}$$

Using $P(B|A)$ values above, we get,

$$\begin{aligned} R_{w,j}^1 &= P(A_s = 1) \cdot \{b_w(E[\mu^* - \mu_*^{w,j}])P(B_s = 0|A_s = 1)\} \\ &\quad + P(A_s = 0) \cdot \{b_w(E[\mu^* - \mu_*^{w,j}])\} \end{aligned}$$

Now from above, let:

$$\begin{aligned} T_1 &= P(A_s = 1) \cdot \{b_w(E[\mu^* - \mu_*^{w,j}])P(B_s = 0|A_s = 1)\} \\ T_2 &= P(A_s = 0) \cdot \{b_w(E[\mu^* - \mu_*^{w,j}])\} \end{aligned}$$

$$\text{Hence, } R_{w,j}^1 = b_w E[\mu^* - \mu_*^{w,j}] = T_1 + T_2 \quad \square$$

Lemma 4.1 Consider events,

1. A: Best arm(universal) is in the current (w, j) instance.
2. B: Best arm(universal) is recommended to next instance $((w, j+1)$ or $(w+1, 1)$).

Let $\mathbf{s} = (w, j)$ be the current instance with phase w and sub-phase j , and let r be the arm recommended to the next instance $(w, j+1)$ or $(w+1, 1)$ and let $\mu_*^{w,j}$ be the mean of that arm. Then if

- $T_1 = P(A_s = 1) \cdot \{b_w(E[\mu^* - \mu_*^{w,j}])P(B_s = 0|A_s = 1)\}$
- $T_2 = P(A_s = 0) \cdot \{b_w(E[\mu^* - \mu_*^{w,j}])\}$

We have,

$$\frac{T_1 + T_2}{b_w} \leq 2C \frac{(M-1) \cdot h_0}{(b_{w-1}/M-1)^2} (h_0 + 1) \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}}$$

Proof. We prove it in two steps, wherein first step we establish an upper-bound on T_1 and in the second step we establish an upper-bound on T_2 .

Step 1. We know that $P(A_s = 1) \leq 1$, and from **Lemma A.2**, $P(B_s = 1|A_s = 1) = P(J_n = i^*) \geq$

$$1 - \frac{M-1}{(b_w/M-1)^2} \implies P(B_s = 0|A_s = 1) = 1 - P(J_n = i^*) \leq \frac{M-1}{(b_w/M-1)^2}$$

Thus using the probabilistic bounds from above and **Theorem 3.1** we get that,

$$T_1 \leq 1 \times \frac{M-1}{(b_w/M-1)^2} \times b_w C \sqrt{\frac{M \log(b_w)}{b_w}}$$

Step 2. Establishing bounds for T_2 isn't so straightforward and will thus be slightly more involved, since the best arm isn't present in the current instance $\mathbf{s} = (w, j)$. Let k be the minimum sub-phase in phase w that has the best-arm a^* or let $k = \min\{i \in [h_0]; a^* \in S^{w,i}\}$, we then analyse cases where a. $j \geq k$ and b. $j < k$ separately. let $\mu_*^{w,j}$ be the mean of that arm recommended by instance (w, j) or $\mu_*^{w,j}$

Step 2.a (for $j \geq k$)

We can write $E[\mu^* - \mu_*^{w,j}] = E[\mu^* - \mu_*^{w,k}] + \sum_{i=k}^{j-1} E[\mu_*^{w,i} - \mu_*^{w,i+1}]$

From **Corollary 4.0.1** we know that approximate simple regret $E[r^y] = E[\mu_*^{w,i} - \mu_*^{w,i+1}]$ can be bounded by $C \sqrt{\frac{M \log(b_w)}{b_w}}$. And using **Theorem 3.1** we can bound $E[\mu^* - \mu_*^{w,k}]$. We thus get,

$$E[\mu^* - \mu_*^{w,j}] \leq C h_0 \sqrt{\frac{M \log(b_w)}{b_w}}$$

Step 2.b(for step $j < k$)

Let $l = \min\{i \in [h_0]; a^* \in S^{w-1,i}\}$, or the sub-phase in the previous phase containing best arm a^* . Thus we can then write,

$$\begin{aligned} E[\mu^* - \mu_*^{w,j}] &\leq E[\mu^* - \mu_*^{w-1,l}] \\ &\quad + \sum_{i=l}^{h_0-1} E[\mu_*^{w-1,i} - \mu_*^{w-1,i+1}] \\ &\quad + E[\mu_*^{w-1,h_0} - \mu_*^{w,1}] \\ &\quad + \sum_{i=1}^{j-1} E[\mu_*^{w,i} - \mu_*^{w,i+1}] \end{aligned}$$

We'll now bound each of the above four terms, a. we know that $a^* \in S^{w-1,l}$ thus by **Theorem 3.1** we get that,

$$E[\mu^* - \mu_*^{w-1,l}] \leq C \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}}, \text{ b. using Corol-}$$

lary 4.0.1 we can bound $\sum_{i=l}^{h_0-1} E[\mu_*^{w-1,i} - \mu_*^{w-1,i+1}] \leq$

$$C(h_0 - l) \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}}$$

as it's nothing but the summation of consecutive recommended arms, c. for the last two terms similar to point (b) above, only now the time horizon for a

sub-phase is b_w , we can again invoke **Corollary 4.0.1** and get the following,

$$E[\mu^* - \mu_*^{w,j}] \leq C(h_0 - l + 1) \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}} + Cj \sqrt{\frac{M \log(b_w)}{b_w}}$$

Since, $b_{w-1} < b_w$
 $\Rightarrow \sqrt{\frac{M \log(b_w)}{b_w}} < \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}}$. Thus we get that,

$$E[\mu^* - \mu_*^{w,j}] \leq C(h_0 + j - l + 1) \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}}$$

Because there has to be atleast one instance containing best-arm a^* in h_0 consecutive instances, $j \leq l$ and thus we get,

$$E[\mu^* - \mu_*^{w,j}] \leq C(h_0 + 1) \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}}$$

Using results from **Step 2.a** and **Step 2.b** we get that
 $E[\mu^* - \mu_*^{w,j}] \leq C(h_0 + 1) \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}}$. Finally using **Corollary A.2.1** we get that,

$$T_2 \leq \frac{(M-1) \cdot h_0}{(b_{w-1}/M-1)^2} b_w C(h_0 + 1) \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}}$$

Step 3 Combining results from **Step 1**, **Step 2**, and using the fact that $b_{w-1} < b_w$ along with

$$\sqrt{\frac{M \log(b_w)}{b_w}} < \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}}$$

we finally get that,

$$\frac{T_1 + T_2}{b_w} \leq 2C \frac{(M-1) \cdot h_0}{(b_{w-1}/M-1)^2} (h_0 + 1) \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}}$$

□

Lemma A.4. If a sequence S is such that $S_w = (b)^{m^w/2} m^{w/2}$, then if $\sum(S) = \sum_{w=1}^{\log_m \log_b(T)} (b)^{m^w/2} m^{w/2}$, we have,

$$\sum(S) \leq \sqrt{\frac{m}{m-1} (\log_b(T)) \cdot (T + \frac{T^{1/m-1}}{b^{m/m-1}})}$$

Proof. We use **Cauchy-Schwartz Inequality** which states that $\sum_{i=1}^k a_i b_i \leq \sqrt{\sum_{i=1}^k a_i^2} \cdot \sqrt{\sum_{i=1}^k b_i^2}$, where $a_i, b_i \in C, \forall i \in [1, k]$.

Let $K = \log_m \log_b(T)$, $A = \sqrt{\sum_{w=1}^K b^{m^w}}$, $B = \sqrt{\sum_{w=1}^K m^w}$, applying **Cauchy-Schwartz Inequality** on $\sum(S)$ we get that,

$$\sum(S) \leq A \cdot B \quad (24)$$

$$\begin{aligned} B &= \sqrt{\sum_{w=1}^K m^w} = \sqrt{\frac{m(m^K - 1)}{m - 1}} \\ &= \sqrt{\frac{m}{m - 1} (\log_b(T) - 1)} \\ \Rightarrow B &\leq (\sqrt{\frac{m}{m - 1} (\log_b(T))}) \end{aligned} \quad (25)$$

Now, to bound A consider,

$$\begin{aligned} A^2 &= \sum_{w=1}^K b^{m^w} \\ A^2 &= \sum_{w=1}^{\log_m \log_b(T)-1} b^{m^w} + b^{m^{\log_m \log_b(T)}} \\ A^2 &= \sum_{w=1}^{\log_m \log_b(T)-1} b^{m^w} + T \end{aligned}$$

We know that, $\sum_{w=1}^{K-1} m^w = (m^K - m)/(m - 1)$

Since $\sum a_i \leq \prod a_i, \forall a_i \geq 2$

Therefore, $\sum_{w=1}^{K-1} b^{m^w} \leq b \sum_{w=1}^{K-1} m^w$

$$\begin{aligned} \Rightarrow \sum_{w=1}^{K-1} b^{m^w} &\leq b^{(m^K - m)/(m-1)} \\ \Rightarrow \sum_{w=1}^{K-1} b^{m^w} &\leq \frac{T^{1/m-1}}{b^{m/m-1}} \\ \Rightarrow A^2 &\leq T + \frac{T^{1/m-1}}{b^{m/m-1}} \end{aligned}$$

$$\Rightarrow A \leq (\sqrt{T + \frac{T^{1/m-1}}{b^{m/m-1}}}) \quad (26)$$

Thus from (7),(8),(9) we get that:

$$\sum(S) \leq \sqrt{\frac{m}{m-1} (\log_b(T)) * (T + \frac{T^{1/m-1}}{b^{m/m-1}})}.$$

□

Lemma 4.2 For $2 \leq M < K$, and for $T \geq KM^2(M+2)$, let $R^{(1)} = \sum_{w=1}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(1)}$, then,

$$R^1 \leq C_0 + C_2 \times \log(\log_{b_1}(\frac{T}{h_0}))$$

where C_2 is a constant depending on K, M .

Proof. We know that,

$$\begin{aligned} R^{(1)} &= \sum_{w=1}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(1)} \\ &= \sum_{j=1}^{h_0} R_{1,j}^{(1)} + \sum_{w=2}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(1)} \\ &\leq h_0 b_1 + \sum_{w=2}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(1)} \\ &\leq h_0 b_1 + \sum_{w=2}^{x_0} \sum_{j=1}^{h_0} b_w E[\mu^* - \mu_*^{w,j}] \end{aligned}$$

From **Lemma A.3** we know that $E[\mu^* - \mu_*^{w,j}]$ can be expressed in terms of T_1, T_2 and thus invoking **Lemma 4.1** we get,

$$\begin{aligned} R^1 &\leq h_0 b_1 \\ &\quad + \sum_{w=2}^{x_0} h_0 b_w (2C \frac{(M-1) \cdot h_0(h_0+1)}{(b_{w-1}/M-1)^2} \sqrt{\frac{M \log(b_{w-1})}{b_{w-1}}}) \end{aligned}$$

Let $C_1 = 2C \frac{(M-1) \cdot h_0^2(h_0+1)}{1} \sqrt{M}$, then

$$R^1 \leq h_0 b_1 + C_1 \sum_{w=2}^{x_0} \frac{b_w}{(b_{w-1}/M-1)^2} \sqrt{\frac{\log(b_{w-1})}{b_{w-1}}}$$

Using the fact that $\frac{\log(x)}{x} \leq 1/2, \forall x > 1$ we get,

$$R^1 \leq h_0 b_1 + \frac{C_1}{\sqrt{2}} \sum_{w=2}^{x_0} \frac{b_w}{(b_{w-1}/M-1)^2}$$

Since $b_w = b_{w-1}^2$ we can write,

$$\begin{aligned} R^1 &\leq h_0 b_1 + \frac{C_1 M^2}{\sqrt{2}} \sum_{w=2}^{x_0} \frac{1}{(1 - \frac{M}{b_{w-1}})^2} \\ &\leq h_0 b_1 + \frac{C_1 M^2}{\sqrt{2}} \sum_{w=2}^{x_0} \frac{1}{(1 - \frac{M}{b_1})^2} \\ &\leq h_0 b_1 + \frac{C_1 M^2}{\sqrt{2}(1 - \frac{M}{b_1})^2} \times (\log(\log_{b_1}(\frac{T}{h_0}))) \end{aligned}$$

Thus we get $R^1 \leq C_0 + C_2 \times \log(\log_{b_1}(\frac{T}{h_0}))$, where $C_2 = \frac{C_1 M^2}{\sqrt{2}(1 - \frac{M}{b_1})^2}$ is a constant depending on K, M . \square

Lemma 4.4 For $2 \leq M < K$, and for $T \geq KM^2(M+2)$, let, $R^{(2)} = \sum_{w=1}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(2)}$, then,

$$R^{(2)} \leq C_0 + C_4 \sqrt{(\log_{b_1}(\frac{T}{h_0})) * (\frac{T}{h_0})}$$

Where C_0, C_4 are constants depending on K and M .

Proof. For any sub-phase j of any phase $w \geq 2$, due to **Lemma 3**, we know that there exists a constant C such that $R_{w,j}^{(2)} \leq C \sqrt{b_w M \log(b_w)}$. Therefore,

$$\begin{aligned} R^{(2)} &= \sum_{w=1}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(2)} \\ &= \sum_{j=1}^{h_0} R_{1,j}^{(2)} + \sum_{w=2}^{x_0} \sum_{j=1}^{h_0} R_{w,j}^{(2)} \end{aligned}$$

Now $R_{w,j}^{(2)} = \sum_{t=1}^{b_w} (E[\mu_*^{w,j}] - E[\mu_{a_t}])$, and we know that $\mu_*^{w,j} \leq \max_{a \in S^{w,j}} \mu_a$. Thus using **Lemma 4.3** we get that,

$$\begin{aligned} R^{(2)} &\leq h_0 b_1 + C \sum_{w=2}^{x_0} \sum_{j=1}^{h_0} \sqrt{b_w M \log(b_w)} \\ &\leq h_0 b_1 + C h_0 \sum_{w=2}^{x_0} \sqrt{(b_1)^{2^{w-1}} 2^{w-1} (M \log(b_1))} \\ &\leq h_0 b_1 + C h_0 (M \log(b_1)) \sum_{w=1}^{x_0-1} \sqrt{(b_1)^{2^w} 2^w} \\ &\leq h_0 b_1 + C h_0 (M \log(b_1)) \sum_{w=1}^{\log \log_{b_1}(T/h_0)} \sqrt{(b_1)^{2^w} 2^w} \end{aligned}$$

Let $C_3 = C(\frac{K-1}{M-1})M \log(M(M+2))$, and $C_0 = \frac{(K-1)(M)(M+2)}{(M-1)}$.

$$\leq h_0 b_1 + C_3 \sum_{w=1}^{\log \log_{b_1}(T/h_0)} \sqrt{(b_1)^{2^w} 2^w}$$

Invoking **Lemma A.4** and using $m = 2$ we get

$$\leq h_0 b_1 + C_3 \sqrt{(\log_{b_1}(T/h_0)) \cdot (T/h_0 + \frac{(T/h_0)^{1/(1)}}{b_1^{2/(2-1)}})}$$

Thus we get,

$$R^{(2)} \leq C_0 + C_4 \sqrt{(\log_{b_1}(\frac{T}{h_0})) * (\frac{T}{h_0})}$$

$$\text{where } C_4 = C(\frac{K-1}{M-1})M \log(M(M+2)) \sqrt{1 + \frac{1}{b_1}}$$

$$\text{and } C_0 = \frac{(K-1)(M)(M+2)}{(M-1)}$$

□

Lemma A.5 (From Lemma 1 in [3]). *Let a_1, \dots, a_K be real numbers such that $a_1 + \dots + a_K = 1$ and $a_i \geq 0$ for all i , with the additional property that for all suboptimal arms i and all optimal arms i^* , one has $a_i \leq a_{i^*}$. Let n : total time horizon, and J_n : max played arm recommended. Then for $\alpha > 1$, and sufficiently large time horizons $a_i n \geq 1 + \frac{4\alpha \ln(n)}{\Delta_i^2}$, the allocation strategy given by UCB(α) associated with the recommendation given by the most played arm ensures that the probability of the suboptimal arm being the max played arm:*

$$P_{i \neq i^*}(J_n = i) \leq \frac{1}{\alpha - 1} (n/K - 1)^{2(1-\alpha)}$$

Proof. We first prove that whenever the most played arm J_n is different from an optimal arm i^* , then at least one of the suboptimal arms i is such that $T_i(n) \geq a_i n$. To do so, we use a contrapositive method and assume that $T_i(n) < a_i n$ for all suboptimal arms. Then,

$$(\sum_{i=1}^K a_i) n = n = \sum_{i=1}^K T_i(n) < \sum_{i_*} T_{i_*}(n) + \sum_i a_i n$$

where, in the inequality, the first summation is over the optimal arms, the second one, over the suboptimal ones. Therefore, we get

$$\sum_{i^*} a_{i^*} n < \sum_{i^*} T_{i^*}(n)$$

and there exists at least one optimal arm i^* such that $T_{i^*} > a_{i^*} n$. Since by definition of the vector (a_1, \dots, a_K) , one has $a_i \leq a_{i^*}$ for all suboptimal arms, it comes that

$T_i < a_i n \leq a_{i^*} n < T_{i^*}(n)$ for all suboptimal arms, and the most played arm J_n is thus an optimal arm.

A side-result extracted from [[7], proof of Theorem 7], see also [[4], proof of Theorem 1], states that for all suboptimal arms i and rounds $t \geq K + 1$,

$$P\{I_t = i, T_i(t-1) \geq l\} \leq 2t^{1-2\alpha} \quad (l \geq \frac{4\alpha \ln(n)}{\Delta_i^2})$$

We denote by $\lceil x \rceil$ the upper integer part of a real number x . For a suboptimal arm i and since by the assumptions on n and a_i , the choice $l = \lceil a_i n \rceil - 1$ satisfies $l \geq K + 1$ and $l \geq (4\alpha \ln(n))/\Delta_i^2$,

$$\begin{aligned} P\{T_i(n) \geq a_i n\} &= P\{T_i(n) \geq \lceil a_i n \rceil\} \\ &\leq \sum_{t=\lceil a_i n \rceil}^n P\{T_i(t-1) = \lceil a_i n \rceil - 1, I_t = i\} \\ &\leq \sum_{t=\lceil a_i n \rceil}^n 2t^{1-2\alpha} \leq 2 \int_{\lceil a_i n \rceil - 1}^{\infty} v^{1-2\alpha} dv \\ &\leq \frac{1}{\alpha - 1} (a_i n - 1)^{2(1-\alpha)}, \end{aligned}$$

With the uniform choice of $a_i = 1/K$ we finally get,

$$P_{i \neq i^*}(J_n = i) \leq P(T_i(n) \geq n/K) \leq \frac{1}{\alpha - 1} (n/K - 1)^{2(1-\alpha)}$$

□

B Proofs for Section 5

Lemma B.1. *Total number of passes in 2-pass UCB-LAM is two.*

Proof. From Algorithm-2 above we know that $b_1 = \frac{\sqrt{1 + 4\frac{T}{h_0}} - 1}{2}$ and $b_2 = \frac{2\frac{T}{h_0} + 1 - \sqrt{1 + 4\frac{T}{h_0}}}{2}$. So the total time taken for the first two passes ($T_{\text{two-pass}}$) is:

$$\begin{aligned} T_{\text{two-pass}} &= h_0 \times b_1 + h_0 \times b_2 \\ &= \frac{h_0 \sqrt{1 + 4\frac{T}{h_0}} - 1}{2} + \frac{2T + h_0 - h_0 \sqrt{1 + 4\frac{T}{h_0}}}{2} \\ &\implies T_{\text{two-pass}} = T \end{aligned}$$

Thus we see that the total time horizon is depleted by the end of second pass, and thus the total number of passes in Algorithm-2 in two. □

Lemma 5.1 *For $2 \leq M < K$, and total time-horizon T , we get that, for Algorithm-2,*

$$R_2 \leq C_2 + C_0 \sqrt{T + 0.25h_0} + C_1 \sqrt{T \log(T/h_0)}$$

Where C_0, C_1, C_2 are constants depending on M, K .

Proof. We know that R_2 can be written as,

$$\begin{aligned} R_2 &= \sum_{w=1}^2 \sum_{j=1}^{h_0} (R_{w,j}^{(2)}) \\ &= \sum_{w=1}^2 \sum_{j=1}^{h_0} \left(\sum_{t=1}^{b_w} (E[\mu_*^{w,j}] - E[\mu_{a_t}]) \right) \\ &\leq h_0 b_1 + \sum_{j=1}^{h_0} \left(\sum_{t=1}^{b_2} (E[\mu_*^{2,j}] - E[\mu_{a_t}]) \right) \end{aligned}$$

From **Lemma 4.3** we get that given a set of K arms as the input for any horizon T , the cumulative regret incurred by UCB1 when $T \geq K/2$, is upper-bounded by $18\sqrt{TK \log(T)}$ or $R_T^* \leq 18\sqrt{TK \log(T)}$. And we know that $\mu_*^{w,j} \leq \max_{a \in S^{w,j}} \mu_a \implies E[\mu_*^{w,j} - \mu_{a_t}] \leq E[\max_{a \in S^{w,j}} \mu_a - \mu_{a_t}]$. Thus,

$$R_2 \leq h_0 b_1 + h_0 C \sqrt{b_2 M \log(b_2)}$$

$$\begin{aligned} \text{Substituting } b_1 &= \frac{\sqrt{1 + 4\frac{T}{h_0}} - 1}{2} \text{ and } b_2 = \\ &= \frac{2\frac{T}{h_0} + 1 - \sqrt{1 + \frac{4T}{h_0}}}{2}, \text{ we get,} \end{aligned}$$

$$R_2 \leq 0.5(\sqrt{h_0^2 + 4h_0 T} - h_0) + C_1 \sqrt{T \log(T/h_0)}$$

where $C_1 = f(M, K)$, $C_0 = g(M, K)$, $C_2 = h(M, K)$
Thus we get that

$$R_2 \leq C_2 + C_0 \sqrt{T + 0.25h_0} + C_1 \sqrt{T \log(T/h_0)}.$$

□

Lemma 5.2 For $2 \leq M < K$, and total time-horizon T , we get that, for Algorithm-2,

$$R_1 = \sum_{w=1}^2 \sum_{j=1}^{h_0} (R_{w,j}^{(1)}) \leq O(\sqrt{T})$$

Proof. We know that R_1 can be written as,

$$\begin{aligned} R_1 &= \sum_{w=1}^2 \sum_{j=1}^{h_0} \sum_{t=1}^{b_w} (E[\mu^* - \mu_*^{w,j}]) \\ \implies R_1 &= \sum_{w=1}^2 \sum_{j=1}^{h_0} b_w (E[\mu^* - \mu_*^{w,j}]) \\ R_1 &\leq h_0 b_1 + \sum_{j=1}^{h_0} b_2 (E[\mu^* - \mu_*^{2,j}]) \end{aligned}$$

From **Lemma A.3** we know that $b_2 E[\mu^* - \mu_*^{2,j}]$ can be written in terms of T_1 and T_2 , or $b_2 E[\mu^* - \mu_*^{2,j}] = T_1 + T_2$, where,

$$\begin{aligned} T_1 &= P(A_s = 1) \times \{b_w(E[\mu^* - \mu_*^{w,j}])P(B_s = 0|A_s = 1) \\ T_2 &= P(A_s = 0) \times \{b_w(E[\mu^* - \mu_*^{w,j}])\} \end{aligned}$$

Where $s = (w, j)$ the current instance and $\mu_*^{w,j}$ is the mean of the arm recommended to the next instance. And invoking **Lemma 4.1** will give us that,

$$T_1 + T_2 \leq 2Cb_2 \frac{(M-1) \cdot h_0}{(b_1/M-1)^2} (h_0+1) \sqrt{\frac{M \log(b_1)}{b_1}}$$

Thus we get that,

$$R_1 \leq h_0 b_1 + 2Cb_2 \frac{(M-1) \cdot h_0^2}{(b_1/M-1)^2} (h_0+1) \sqrt{\frac{M \log(b_1)}{b_1}}$$

Substituting values for b_1, b_2 for **Algorithm-2**

$$\begin{aligned} b_1 &= \frac{\sqrt{1 + 4\frac{T}{h_0}} - 1}{2} \\ b_2 &= \frac{2\frac{T}{h_0} + 1 - \sqrt{1 + \frac{4T}{h_0}}}{2} = b_1^2 \end{aligned}$$

in the above inequality we finally get that $R_1 \leq O(\sqrt{T})$. □

References

- [1] Berry, D. and Fristedt, B. Bandit Problems: Sequential Allocation of Experiments. Chapman and Hall, 1985.
- [2] Regret Minimisation in Multi-Armed Bandits Using Bounded Arm Memory. Arghyrai Chaudhari and Shivaram Kalyankrishnan <https://arxiv.org/abs/1901.08387>
- [3] Bubeck, S., Munos, R., and Stoltz, G. Pure exploration in multi-armed bandits problems. In Algorithmic Learning Theory, pp. 23–37. Springer Berlin Heidelberg, 2009.
- [4] Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. Machine Learning, 47(2-3):235–256, 2002.
- [5] Audrey Durand, Charis Achilleos, Demetris Iacovides, Katerina Strati, Georgios D Mitsis, and Joelle Pineau. Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In Machine Learning for Healthcare Conference, pages 67–82, 2018
- [6] Weiwei Shen, Jun Wang, Yu-Gang Jiang, and Hongyuan Zha. Portfolio choices with orthogonal bandit learning. In Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.
- [7] J.-Y. Audibert, R. Munos, and C. Szepesvari. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. Theoretical Computer Science, 410:1876–1902, 2009
- [8] Armitage, P. Sequential Medical Trials. Blackwell Scientific Publications, 1960

- [9] Tran-Thanh, L., Stein, S., Rogers, A., and Jennings, N. R. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artif. Intl.*, 214:89 – 111, 2014.
- [10] Arnab Maiti, Vishakha Patil, Arindam Khan. Streaming Algorithms for Stochastic Multi-armed Bandits *NeurIPS* 2021
- [11] Aho, A. V., Hopcroft, J. E., and Ullman, J. D. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
- [12] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to Algorithms*, Third Edition. The MIT Press, 2009.
- [13] Robbins, H. Some aspects of the sequential design of experiments. *Bulletin of the AMS*, 58(5):527–535, 1952.
- [14] Chapelle, O. and Li, L. An empirical evaluation of Thompson sampling. In *Advances in neural information processing systems*, pp. 2249–2257, 2011.
- [15] Agrawal, S. and Goyal, N. Analysis of Thompson sampling for the multi-armed bandit problem. In *Proc. of the 25th Annual Conf. on Learning Theory*, volume 23, pp. 39.1– 39.26, Edinburgh, Scotland, 2012. PMLR.
- [16] Liao, D., Price, E., Song, Z., and Yang, G. Stochastic multi-armed bandits in constant space. In *Proc. AIS-TATS 2018*, volume 84, pp. 386–394. PMLR, 2018.
- [17] Cover, T. M., Freedman, M. A., and Hellman, M. E. Optimal finite memory learning algorithms for the finite sample problem. *Information and Control*, 30(1):49 – 85, 1976.
- [18] Cover, T. M. A note on the two-armed bandit problem with finite memory. *Information and Control*, 12(5):371 – 377, 1968.