On the Asymptotic Optimality of Confidence Interval Based Algorithms for Fixed Confidence MABs

Kushal Kejriwal, Nikhil Karamchandani, Jayakrishnan Nair

Indian Institute of Technology Bombay, Mumbai, India
kushalk333@gmail.com, {nikhilk, jayakrishnan.nair}@ee.iitb.ac.in

Abstract

In this work, we address the challenge of identifying the optimal arm in a stochastic multi-armed bandit scenario with the minimum number of arm pulls, given a predefined error probability in a fixed confidence setting. Our focus is on examining the asymptotic behavior of sample complexity and the distribution of arm weights upon termination, as the error threshold is scaled to zero, under confidence-interval based algorithms. Specifically, we analyze the asymptotic sample complexity and termination weight fractions for the well-known LUCB algorithm, and introduce a new variant, the LUCB Greedy algorithm. We demonstrate that the upper bounds on the sample complexities for both algorithms are asymptotically within a constant factor of the established lower bounds.

1 Introduction

The best arm identification (BAI) problem is a classical construct in multi-armed bandits (MABs), where the goal of the learner is to identify the best among a finite basket of arms. Each arm is associated with an apriori unknown reward distribution, and the learner can obtain a sample from the reward distribution of any arm by 'pulling' it. BAI is relevant in a variety of applications, including clinical trials (Thananjeyan et al. 2021), crowdsourcing (Didwania, Nair, and Hemachandra 2022), recommendation systems (Yao et al. 2022), and and Monte Carlo tree search (Kaufmann and Koolen 2017).

In the *fixed confidence* BAI problem, the learner must identify the best arm, defined as the one with the highest mean reward, such that the probability of mis-identification (a.k.a., error) is at most a prescribed threshold δ. A learning algorithm that satisfies this property is said to be sound, and one seeks to design sound algorithms that make the fewest number of pulls (on average). There are broadly, two types of algorithms for fixed confidence BAI:
(i) confidence interval based algorithms, and
(ii) algorithms based on generalized likelihood ratio (GLR) tests.

The former category includes algorithms such as Action Elimination, (Even-Dar, Mannor, and Mansour 2006), UGapE (Gabillon, Ghavamzadeh, and Lazaric 2012), UCB (Audibert, Bubeck, and Munos 2010), and LUCB (Kalyanakrishnan et al. 2012), (Kaufmann and Kalyanakrishnan 2013); see (Jamieson and Nowak 2014) for a survey. These algorithms maintain confidence intervals on the mean reward of each arm (which are constructed using suitable concentration inequalities); these confidence intervals are used to determine which arm to sample next, and/or when to terminate. Upper bounds on the sample complexity (i.e., the number of pulls made until termination) of these algorithms tend to be loose (and typically contain logarithmic factors that do not arise in any known lower bounds). As a result, sample complexity upper bounds for confidence interval based algorithms typically do not admit a direct comparison with information theoretic lower bounds.

On the other hand, GLR based algorithms, including the celebrated Track & Stop algorithm (Garivier and Kaufmann 2016), the Transport Cost Balancing algorithm (Mukherjee and Tajer 2023a; Jourdan et al. 2022), and top-two algorithms (Mukherjee and Tajer 2023b; Bandyopadhyay, Juneja, and Agrawal 2024) use likelihood ratios to define stopping rules. These algorithms, which sample arms in a manner that seeks to track the optimal pull fractions suggested by the information theoretic lower bound, are analysed for sample complexity in the asymptotic regime $\delta \downarrow 0$. This asymptotic regime typically admits an exact characterization of the scaling of the sample complexity (including multiplicative constants). Moreover, such a scaling enables a direct (asymptotic) comparison between sample complexity upper bounds and information theoretic lower bounds.

In this paper, we perform an asymptotic (as $\delta \downarrow 0$) analysis of confidence interval based algorithms. Specifically, we consider the classical LUCB (Kalyanakrishnan et al. 2012), which is the state of the art algorithm in the space of confidence interval based algorithms, and analyse its sample complexity in the asymptotic regime as $\delta \downarrow 0$. We show that the sample complexity as well as the pull fractions (i.e., the fraction of pulls each arm has received) under LUCB admit, asymptotically, a clean and interpretable characterization. Moreover, our asymptotic sample complexity bound for LUCB admits a direct comparison with the available (algorithm-agnostic) lower bounds. This shows that the performance of LUCB is (asymptotically) within a *known* constant factor of the lower bound; such a strong optimality assertion is not available in the literature for confidence inter-

val based algorithms, to the best of our knowledge.

Motivated by the above, we also propose and analyse a variant of LUCB, called LUCB Greedy. While the LUCB algorithm samples, in each round, the (empirically) top arm and a (suitably defined) challenger arm, LUCB Greedy samples *one* of these arms, seeking to greedily maximize the separation between their confidence intervals. Interestingly, we find that LUCB Greedy also admits an asymptotic analysis analogous to LUCB. Moreover, neither algorithm dominates the other; which algorithm has a smaller (asymptotic) sample complexity depends on the specific MAB instance.

The remainder of this paper is organized as follows. We provide describe the MAB formulation and state some prelimiries in Section 2. The asymptotic analysis of the LUCB algorithm is performed in Section 3, and the corresponding analysis of LUCB Greedy is performed in Section 4. We present numerical case studies to complement our theoretical results in Section 5.

2 Problem Formulation

Consider a stochastic multi-armed bandit setting with K arms. Each arm i has a reward characterized by a 1-subGaussian distribution ν_i, and let the mean reward be denoted by μ_i. Let $\mu_1 > \mu_2 > \ldots > \mu_K$[1]. Let the suboptimality gaps be denoted by $\Delta_i = \mu_1 - \mu_i$ for $i \in \{2, 3, \ldots, K\}$, and let $\Delta_1 = \Delta_2$. At each time instant, the learner can pull an arm i and observe reward $R_{i,t} \sim \nu_i$. Rewards from the same arm i are independent and identically distributed (i.i.d.) samples from the distribution ν_i; also, reward samples across distinct arms are assumed independent.

The broad goal of the learner is to identify the 'best' arm, i.e., the one with the highest mean reward, by sequentially selecting arms and sampling from their associated reward distributions. An online learning scheme for this setup consists of the following components: (1) a *sampling rule* which considers the history of the reward observations seen thus far and decides which arm to pull next; (2) a *stopping rule* which prescribes when the learner will not sample arms any further; and (3) a *recommendation rule* which outputs an estimate for the best arm once the algorithm terminates.

In particular, we consider the *fixed confidence* best arm identification problem where the objective is to devise *sound* online learning schemes which can identify the best arm with some pre-defined target error probability. For $\delta \in [0, 1]$, we will say that a scheme $\mathcal{A}$ is δ-probably correct (δ-PC) if, for any underlying problem instance $\mathcal{I}$, the probability that the algorithm output is incorrect is at most δ. The performance of a δ-PC scheme $\mathcal{A}$ is measured via its random stopping time $t_\delta^{\mathcal{I}}(\mathcal{A})$ over an instance $\mathcal{I}$; and our results will be expressed in terms of expectation and high probability bounds. For ease of notation, going forward we will suppress the instance $\mathcal{I}$ and the algorithm $\mathcal{A}$ in the notation for stopping time since these will generally be clear from context; we will refer to the stopping time simply as $t(\delta)$.

[1]For the sake of simplicity, we assume the means of the non-optimal arms to be unique; see also Section 1.5 in the supplementary material.

In this work, we will focus on the LUCB sampling rule (Kalyanakrishnan et al. 2012) and introduce its variant LUCB Greedy. Let $N_i(t)$ denote the number of pulls of arm i till time instant t, and let $w_i(t)$ denote the corresponding pull fraction at t, i.e, $w_i(t) = N_i(t)/t$. In particular, we study the asymptotic stopping times and pull fractions for LUCB and LUCB Greedy, as the target error probability δ becomes small.

2.1 Lower Bound

We first discuss a well-known lower bound on the expected stopping time for any δ-PC scheme; see for example (Garivier and Kaufmann 2016; Lattimore and Szepesvári 2020). Consider the special case of a 1-Gaussian bandit instance $\boldsymbol{\mu}$, i.e., the mean rewards are given by $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_K)$ and the reward distribution of each arm is a unit variance Gaussian. . For a bandit instance $\boldsymbol{\mu}$, let Alt($\boldsymbol{\mu}$) be the set of bandit instances where the best arm is not the same as in $\boldsymbol{\mu}$. Finally, let Σ_K denote the set of probability distributions on $\{1, 2, \ldots, K\}$. Then we have the following instance-dependent lower bound (Garivier and Kaufmann 2016) on the sample complexity of any δ-PC scheme. [2]

Theorem 1. *For any δ-PC strategy and any bandit model $\boldsymbol{\mu}$, let $t(\delta)$ denote the stopping time of the algorithm. Then*

$$\liminf_{\delta \to 0} \frac{\mathbb{E}(t(\delta))}{\log(1/\delta)} \geq T^*(\boldsymbol{\mu}) \tag{1}$$

where

$$T^*(\boldsymbol{\mu})^{-1} = \sup_{\omega \in \sum_K} \inf_{\lambda \in Alt(\boldsymbol{\mu})} \left(\sum_{a=1}^{K} \omega_a d(\mu_a, \lambda_a) \right). \tag{2}$$

The quantity $T^*(\boldsymbol{\mu})$ can be bounded in terms of the suboptimality gap $\left(\sum_{i=1}^{K}(1/\Delta_i^2)\right)$ as follows:

$$2\left(\sum_{i=1}^{K} \frac{1}{\Delta_i^2}\right) \leq T^*(\boldsymbol{\mu}) \leq 4\left(\sum_{i=1}^{K} \frac{1}{\Delta_i^2}\right). \tag{3}$$

3 An Asymptotic View of LUCB

In this section, we perform an asymptotic analysis of the celebrated LUCB algorithm. Specifically, we show that LUCB's performance admits a clean and interpretable characterization in the asymptotic regime as $\delta \downarrow 0$. This characterization provides useful insights into the working of LUCB (which do not follow from the available order-sense upper bounds on sample complexity), and also enables a more direct comparison with the available information-theoretic lower bounds for best arm identification (see Section 2.1).

3.1 The LUCB Algorithm

We begin by stating the LUCB algorithm formally. For analytical convenience, we analyse a slight modification of the originally proposed LUCB algorithm (Kalyanakrishnan et al. 2012). Specifically, the algorithm we analyse differs from the classical one in the following ways:

[2]The lower bound holds more broadly for exponential families of bandit models.

- **Forced exploration (FE)**: We enforce a certain minimum (sublinear) exploration of all arms.
- **Non-forced-exploration (NFE) clock**: Denoted by t, this pull counter excludes those pulls that were performed due to forced exploration. The confidence widths are parameterized by this counter t.

Note that since the number of FEs scales sublinearly with t, one would expect that the overall sample complexity, denoted by $\hat{t}$, is dominated by t asymptotically; we prove that this is indeed the case. Thus, it suffices to analyse the NFE counter t at the termination of the algorithm.

Algorithm 1: LUCB Algorithm

1: **Inputs**: δ, α
2: **Initialize** $t, \hat{t} = K$, term = False
3: **for** $i \in [K]$ **do**
4: Sample arm i, obtain R_i
5: $X_i \leftarrow R_i$
6: $N_i, \hat{N}_i \leftarrow 1$
7: $\hat{\mu}_i \leftarrow X_i / \hat{N}_i$
8: $UCB_i \leftarrow \hat{\mu}_i + \sqrt{2\log(2Kc(N_i^\alpha)/\delta)/N_i}$
9: $LCB_i \leftarrow \hat{\mu}_i - \sqrt{2\log(2Kc(N_i^\alpha)/\delta)/N_i}$
10: **end for**
11: $a_{top} \leftarrow \arg\max \hat{\mu}_i$
12: $a_{chl} \leftarrow \arg\max_{i \neq a_{top}} UCB_i$
13: $\mathcal{U} \leftarrow \{a : \hat{N}_a < \sqrt{\hat{t}} - K/2\}$
14: **while** term = False **do**
15: **if** $\mathcal{U} \neq \varnothing$ **then**
16: $j \leftarrow \arg\min_{a \in \mathcal{U}} \hat{N}_a$
17: Sample arm j, obtain R_j
18: $X_j \leftarrow X_j + R_j$
19: $\hat{N}_j \leftarrow \hat{N}_j + 1$
20: $\hat{\mu}_j \leftarrow X_j / \hat{N}_j$
21: $\hat{t} \leftarrow \hat{t} + 1$
22: $\mathcal{U} \leftarrow \{a : \hat{N}_a < \sqrt{\hat{t}} - K/2\}$
23: **else if** $\mathcal{U} = \varnothing$ **then**
24: **for** $i \in \{a_{top}, a_{chl}\}$ **do**
25: Sample arm i, obtain R_i
26: $X_i \leftarrow X_i + R_i$
27: $\hat{N}_i \leftarrow \hat{N}_i + 1$
28: $N_i \leftarrow N_i + 1$
29: $\hat{\mu}_i \leftarrow X_i / \hat{N}_i$
30: $UCB_i \leftarrow \hat{\mu}_i + \sqrt{2\log(2Kc(N_i^\alpha)/\delta)/N_i}$
31: $LCB_i \leftarrow \hat{\mu}_i - \sqrt{2\log(2Kc(N_i^\alpha)/\delta)/N_i}$
32: $t \leftarrow t + 1$
33: $\hat{t} \leftarrow \hat{t} + 1$
34: **if** $LCB_{a_{top}} \geq UCB_{a_{chl}}$ **then**
35: term = True
36: **end if**
37: **end for**
38: **end if**
39: $a_{top} \leftarrow \arg\max \hat{\mu}_i$
40: $a_{chl} \leftarrow \arg\max_{i \neq a_{top}} UCB_i$
41: **end while**
42: **return** a_{top}

The algorithm is stated formally as Algorithm 1. Note the use of two pull counters $\hat{N}_i, N_i$ for each arm, and two aggregate pull counters $\hat{t}$ and t. As noted above, $\hat{t}$ and t denote, respectively, the (total) number of arms pulls and the (total) number of NFE pulls made by the algorithm. Similarly, $\hat{N}_i$ and N_i denote the number of pulls of arm i and the number of NFE pulls of arm i, respectively. Crucially, while the empirical mean estimate $\hat{\mu}_i$ for each arm i is based on *all* pulls of arm i (see Lines 20 and 29), the confidence width is defined using the NFE counter alone. Specifically, the confidence width for arm i as defined as $\sqrt{2\log(2Kc(N_i^\alpha)/\delta)/N_i}$ (see, for example, Lines 30 and 31). Here, the hyperparameter $\alpha \in (1, e/2]$ and the constant $c := \sum_{n=1}^{\infty} \frac{1}{n^\alpha}$.

As noted above, our modification of LUCB introduces forced exploration of all arms. The set of under-explored arms (which are then explored 'forcibly') is defined in Line 13; an arm is considered under-explored if the total number of pulls it has received is less than $\sqrt{\hat{t}} - K/2$. Lines 14 to 22 describe the case where the set of under-explored arms is not empty. In this case, we pull the arm that has been sampled the least number of times. On the other hand, Lines 23 to 38 describe the (main) case where the set of under-explored arms is empty. In this case, we follow the classical LUCB algorithm and pull two arms: (i) the *top arm* a_{top}, defined as the arm with the highest mean estimate (see Line 39), and (ii) the *challenger arm* a_{chl}, defined as the arm with the highest UCB (excluding a_{top}; see Line 40).

Finally, Line 35 describes the termination condition. The algorithm terminates when the LCB of the top arm exceeds the UCB of the challenger arm; the top arm is then reported as the best arm.

It is easy to show that the LUCB algorithm as described above is sound, i.e., δ-PC. Since the arguments for establishing this are standard (see (Kalyanakrishnan et al. 2012)), we omit the proof.

3.2 Heuristic Calculations

Before presenting our formal results, we present heuristic calculations of the asymptotic scaling of the stopping time and pull fractions under LUCB. These calculations provide useful intuition for the formal results that follow later in this section.

We begin with the following observations.

- Since the number of FE pulls scales sub-linearly with time, it suffices to analyse the scaling of $t(\delta)$ (total NFE pulls until termination) and $N_i(t(\delta))$ (NFE pulls of arm i until termination) as $\delta \downarrow 0$.
- Thanks to forced exploration, $\hat{N}_i(t(\delta)) \uparrow \infty$ as $\delta \downarrow 0$. By the law of large numbers, we expect $\hat{\mu}_i \approx \mu_i \; \forall i$ for large enough t. This also implies that once t is large enough, arm 1 will remain, with high probability (w.h.p.), the top arm, i.e., $a_{top} = 1$.
- We naturally expect that $t(\delta)$ scales proportionally

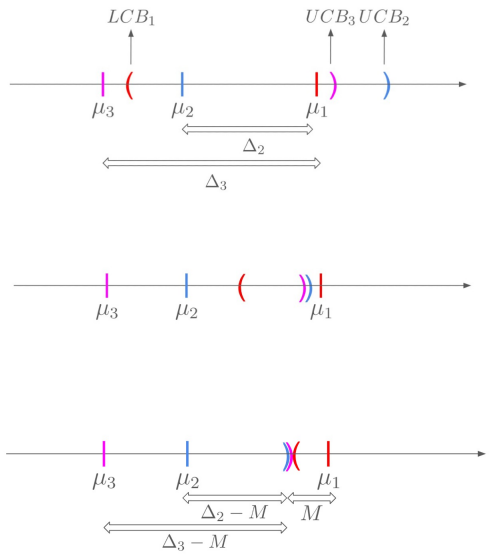

Figure 1: Asymptotic evolution of confidence intervals under LUCB

to $\log(1/\delta)$ as $\delta \downarrow 0$. Suppose that[3]

$$t(\delta) \sim \frac{4}{M^2} \log(1/\delta). \tag{4}$$

Since for large enough t, arm 1 is the top arm w.h.p. and hence gets sampled once every two pulls, we expect

$$N_1(t(\delta)) \sim \frac{t(\delta)}{2} \sim \frac{2}{M^2} \log(1/\delta). \tag{5}$$

Based on these observations, consider the asymptotic evolution of the confidence intervals. Figure 1 provides a visualisation of how these intervals evolve for the case of 3 arms with means $\mu_1 > \mu_2 > \mu_3$. Assuming $\hat{\mu}_i \approx \mu_i$, we center these intervals around the true arm means. The top panel of Figure 1 illustrates an intermediate scenario. In this case, arm 2 is the challenger arm; LUCB sampling would result in the UCB of this challenger arm decreasing, and the LCB of arm 1 (the top arm) increasing. As the sampling continues, we see two things happening (see the second panel of Figure 1):

- The UCBs of the non-optimal arms aligning,
- The separation gap the between the LCB of arm 1 and UCBs of the non-optimal arms decreasing.

Finally, once the separation between the LCB of arm 1 and the UCBs of the non-optimal arms shrinks to zero, the algorithm terminates (see the bottom panel of Figure 1).

Consequently, at termination, we expect:

$$LCB_1(t(\delta)) \approx UCB_i(t(\delta)) \quad \forall\, i \neq 1$$

$$\implies \Delta_i \approx \sqrt{\frac{2\log(1/\delta)}{N_1(t(\delta))}} + \sqrt{\frac{2\log(1/\delta)}{N_i(t(\delta))}} \quad \forall\, i \neq 1$$

[3]Here, $f(\delta) \sim g(\delta)$ means $\lim_{\delta \downarrow 0} \frac{f(\delta)}{g(\delta)} = 1$.

Here, we have ignored the lower order term $\log(2KcN_i^\alpha)$ from the expression of the confidence width. Thus, using (4) and (5), we have:

$$\frac{\Delta_i}{M} \approx 1 + \sqrt{\frac{t(\delta)}{2N_i(t(\delta))}} \quad \forall\, i \neq 1$$

$$\implies w_i(t(\delta)) := \frac{N_i(t(\delta))}{t(\delta)} \sim \frac{1}{2}\left(\frac{M}{\Delta_i - M}\right)^2 \quad \forall\, i \neq 1$$

Finally, since $w_1(t(\delta)) := \frac{N_1(t(\delta))}{t(\delta)} \sim \frac{1}{2}$, we conclude, using $\sum_{i=1}^K w_i(t(\delta)) = 1$, that

$$\frac{1}{M^2} = \sum_{i=2}^{K} \left(\frac{1}{\Delta_i - M}\right)^2. \tag{6}$$

In conclusion, our heuristic calculation suggests that the asymptotic scaling of the termination time is given by (4), where M is the unique positive solution of (6). Indeed, the instance-dependent constant M has another interpretation: it is the distance between the (asymptotic) LCB-UCB separation point and the mean of arm 1; see the bottom panel of Figure 1. Moreover, the limiting pull fraction of arm 1 equals $1/2$, whereas the limiting pull fraction of a non-optimal arm $i \neq 1$ is inversely proportional to $(\Delta_i - M)^2$.

3.3 Formal Asymptotic Results for LUCB

We now formalize the intuitions developed in Section 3.2. We begin by characterizing the almost sure scaling of the stopping time and the asymptotic weight fractions at termination for the LUCB algorithm.[4]

Theorem 2. *Under the LUCB algorithm stated as Algorithm 1, the stopping time $\hat{t}(\delta)$ satisfies*

$$\limsup_{\delta \to 0} \frac{\hat{t}(\delta)}{\log(1/\delta)} \leq \frac{4}{M^2}$$

almost surely, where M is the unique positive solution of (6).

Additionally, the limiting pull fractions at termination are given by:

$$\lim_{\delta \to 0} \frac{N_j(\hat{t}(\delta))}{\hat{t}(\delta)} = \begin{cases} \frac{1}{2} & j = 1 \\ \frac{1}{2}\left(\frac{M}{\Delta_j - M}\right)^2 & j \neq 1 \end{cases}$$

The statement of Theorem 2 is in line with the intuition developed in Section 3.2, except that (i) we now explicitly account for the forced explorations, and (ii) we only establish an asymptotic upper bound on the stopping time (i.e., we do not establish the matching lower bound suggested by (4)).

[4]A technicality here: To analyse an *almost sure* scaling, we must describe the operation of LUCB for each $\delta \in (0,1)$ on a common probability space (so that $\hat{t}(\delta)$ is well defined for all δ on each sample path). This is achieved by defining, on each sample path, an infinite sequence of samples for each arm. This allows LUCB to be 'run in parallel' with difference choices of δ on the same sample path.

Remark 1. *An analogous asymptotic scaling has been established for the Track & Stop (T&S) algorithm (see (Garivier and Kaufmann 2016, Proposition 13)). Relative to this result, the key technical challenge we face is that unlike in the case of T&S, the sampling rule of LUCB is intimately tied to the confidence threshold δ (via the confidence intervals). This means that as δ is scaled, the sampling choices under LUCB change over the entire run of the algorithm.*

Next, we establish the (technically harder) scaling of the expected stopping time under LUCB. The result is structurally similar to the almost-sure case (Theorem 2), barring an additional factor of $\alpha(>1)$ factor that arises due to the application of certain bounds on Lambert functions (these bounds are not used in our almost sure results; see supplementary material. Recall that α is the hyperparameter that is used to define the confidence widths).

Theorem 3. *Under the LUCB algorithm stated as Algorithm 1, the stopping time $\hat{t}(\delta)$ satisfies*

$$\limsup_{\delta \to 0} \frac{\mathbb{E}(\hat{t}(\delta))}{\log(1/\delta)} \leq \frac{4\alpha}{M^2}, \tag{7}$$

where the constant M is as defined in Theorem 2.

Naturally, one might be tempted to just take α arbitrarily close to 1 to optimize the bound above. While doing so does improve the asymptotic scaling of the mean stopping time, it can have an adverse effect on the mean stopping time in the pre-limit (i.e., for moderate choices of δ), due to the larger value of the parameter c in the confidence width.

We now provide a (highly simplified) sketch of the main idea used to prove Theorem 3; the formal proof can be found in the supplementary material. Based on the ideas described in Section 3.2, we define constants N_i for each arm i, such that w.h.p., for any arm $j \neq 1$, N_j pulls suffice in order to separate its confidence interval from that of arm 1. We then argue that $t_{up} = \sum N_j$ is an upper bound on the stopping time w.h.p. This is because by time t_{up}, at least one of the arms $j \neq 1$ must have received N_j pulls, implying that $UCB_j < LCB_1$. But since LUCB pulls the (non-top) arm with the highest UCB, this also implies $UCB_\ell < LCB_1$ for $\ell \neq 1$. The proof is completed by showing that the asymptotic scaling of t_{up} agrees with the statement of the theorem.

Lastly, we state the following corollary, which relates the upper bound of Theorem 3 to known information-theoretic lower bounds on sample complexity, for the special case of a Gaussian bandit instance. Specifically, the corollary shows that the expected stopping time under LUCB is asymptotically at most 6α times the information-theoretic lower bound. To the best of our knowledge, this is the first exact (not 'order sense') comparison between an (instance dependent) upper bound for LUCB and an (instance dependent) information-theoretic lower bound.

Corollary 3.1. *Consider the special case of a 1-Gaussian bandit instance $\boldsymbol{\mu}$ (i.e., the reward distribution of each arm is a unit variance Gaussian). Then*

$$\limsup_{\delta \to 0} \frac{\mathbb{E}(\hat{t}(\delta))}{\log(1/\delta)} \leq 8\alpha \left(\sum_{i=1}^{K} \frac{1}{\Delta_i^2} \right) + 4\alpha \left(\sum_{i=3}^{K} \frac{1}{\Delta_i^2} \right)$$

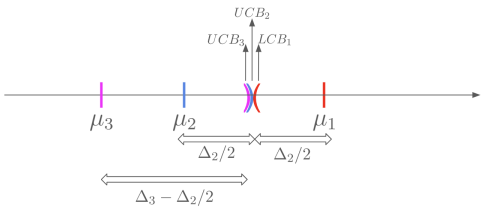

Figure 2: Asymptotic view of the confidence intervals upon termination under LUCB Greedy for 3-armed instance

$$\leq 12\alpha \left(\sum_{i=1}^{K} \frac{1}{\Delta_i^2} \right) \leq 6\alpha T^*(\boldsymbol{\mu}).$$

The above bounding is not tight, owing to difference in structure between the upper bound of Theorem 3 and the best lower bound for BAI (see Theorem 1). Empirically, we find that the asymptotic sample complexity under LUCB is at most 4α times the lower bound; see Section 5.

4 A Greedy Variant of LUCB

We now describe and analyse a variant of the LUCB algorithm, which we call LUCB Greedy. This algorithm seeks to be more opportunistic in its sampling strategy, but is amenable to an asymptotic analysis similar to that performed for LUCB in Section 3. Interestingly, we find that neither algorithm dominates the other with respect to asymptotic scaling of the stopping time.

LUCB Greedy differs from LUCB in its sampling rule—instead of sampling *both* the top arm a_{top} and the challenger arm a_{chl} (see Line 24), LUCB Greedy samples *one* of these two arms; specifically the one that has received the fewer number of pulls thus far. Intuitively, this choice (greedily) results in the most reduction in the separation between the LCB of a_{top} and the UCB of a_{chl} (assuming the confidence interval centers don't change). The remaining details of LUCB Greedy are identical to LUCB; a formal pseudocode of the scheme is provided in supplementary material.

Similar to the LUCB algorithm, it is easy to show that LUCB Greedy is sound, i.e, δ-PC. In the remainder of this section, we present an asymptotic analysis of LUCB Greedy.

4.1 Heuristic Calculations

As before, we begin by presenting heuristic calculations of the asymptotic scaling of the stopping time and pull fractions under LUCB Greedy. Since the arguments are similar to those made earlier for LUCB, we only highlight the differences here.

- Upon termination, we expect the UCBs of the non-optimal arms to be aligned; see Figure 2. It also follows then that arm 2 has the highest number of pulls among the non-optimal arms.
- Additionally, we expect $N_1(t(\delta)) \sim N_2(t(\delta))$; intuitively, this is because the sampling pulls the top arm only when it has had fewer pulls that the (current) challenger

arm. Consequently arm 1's pulls get 'tied' with the most-pulled non-optimal arm, i.e., arm 2.

Thus, $LCB_1(t(\delta)) \approx UCB_2(t(\delta))$, implying

$$\Delta_2 \approx \sqrt{\frac{2\log(1/\delta)}{N_1(t(\delta))}} + \sqrt{\frac{2\log(1/\delta)}{N_2(t(\delta))}} \approx 2\sqrt{\frac{2\log(1/\delta)}{N_1(t(\delta))}}$$

$$\implies N_1(t(\delta)) \sim N_2(t(\delta)) \sim \frac{8\log(1/\delta)}{\Delta_2^2}$$

Similarly, for $i > 2$, $LCB_1(t(\delta)) \approx UCB_i(t(\delta))$, implying

$$\Delta_i \approx \sqrt{\frac{2\log(1/\delta)}{N_1(t(\delta))}} + \sqrt{\frac{2\log(1/\delta)}{N_i(t(\delta))}} \implies$$

$$\Delta_i - \Delta_2/2 \approx \sqrt{\frac{2\log(1/\delta)}{N_i(t(\delta))}},\ N_i(t(\delta)) \sim \frac{2\log(1/\delta)}{(\Delta_i - \Delta_2/2)^2}$$

Interestingly, we see that under LUCB Greedy, the (asymptotic) separation point between the LCB of arm 1 and the UCBs of the non-optimal arms is halfway between μ_1 and μ_2 (see Figure 2). Moreover, the limiting pull fractions admit an explicit closed form; the limiting pull fraction of arm i is inversely proportional to $(\Delta_i - \Delta_2/2)^2$ (i.e., the square of the gap between μ_i and the separation point).

4.2 Formal Asymptotic Results for LUCB Greedy

We now formalize the intuitions developed in Section 4.1. We begin by characterizing the almost sure scaling of the stopping time and the asymptotic weight fractions at termination for LUCB Greedy. Define

$$M_g := \left(\frac{8}{\Delta_2^2} + \sum_{i=3}^{K} \frac{1}{\left(\Delta_i - \frac{\Delta_2}{2}\right)^2}\right). \tag{8}$$

Theorem 4. *Under the LUCB Greedy algorithm, the stopping time $\hat{t}(\delta)$ satisfies $\limsup_{\delta\to 0} \frac{\hat{t}(\delta)}{\log(1/\delta)} \leq 2M_g$ almost surely, where M_g is as defined in* (8)*. Additionally, the limiting pull fractions at termination are given by:*

$$\lim_{\delta\to 0} \frac{N_i(\hat{t}(\delta))}{\hat{t}(\delta)} = \left(\frac{M_g}{\Delta_i - \Delta_2/2}\right)^2.$$

Next, we establish the scaling of the expected stopping time under LUCB Greedy. As before, this result is structurally similar to the almost-sure case, barring the additional factor of $\alpha(>1)$ factor that arises due to Lambert bounding.

Theorem 5. *Under the LUCB Greedy algorithm, the stopping time $\hat{t}(\delta)$ satisfies $\limsup_{\delta\to 0} \frac{\mathbb{E}(\hat{t}(\delta))}{\log(1/\delta)} \leq 2\alpha M_g$, where M_g is as defined in* (8)*.*

Finally, similar to Corollary 3.1, we relate the upper bound of Theorem 5 to known information-theoretic lower bounds on sample complexity, for the special case of a Gaussian bandit instance. Specifically, we show that the expected stopping time under LUCB Greedy is asymptotically at most 4α times the information-theoretic lower bound.

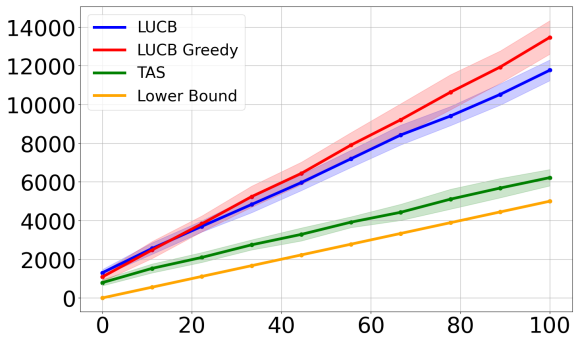

Figure 3: Sample complexity (with 1 standard deviation confidence band) vs $\log(1/\delta)$ for instance $\boldsymbol{\mu_1}$ for LUCB and LUCB Greedy, and the information theoretic lower bound

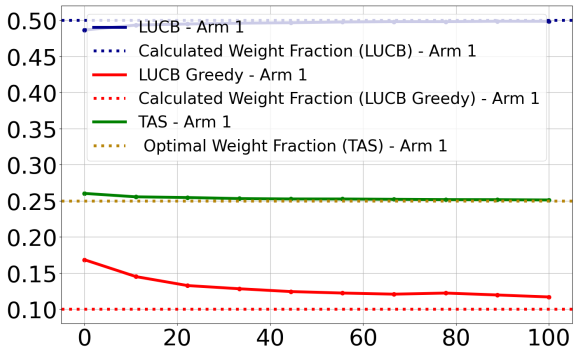

Figure 4: Pull fractions at termination of the top-arm vs $\log(1/\delta)$ for instance $\boldsymbol{\mu_1}$ under LUCB and the LUCB Greedy, along with the optimal pull fraction

Corollary 5.1. *Consider the special case of a 1-Gaussian bandit instance $\boldsymbol{\mu}$. Then*

$$\limsup_{\delta\to 0} \frac{\mathbb{E}(\hat{t}(\delta))}{\log(1/\delta)} \leq 8\alpha \left(\sum_{i=1}^{K} \frac{1}{\Delta_i^2}\right) \leq 4\alpha T^*(\boldsymbol{\mu}). \tag{9}$$

Interestingly, the upper bound (9) can be shown to be tight (for example, it is matched for any 2-armed instance).

5 Simulations and Results

In the preceding sections, we have derived theoretical bounds on the asymptotic sample complexity of the LUCB and LUCB Greedy algorithms. In particular, for the case of 1-Gaussian instances, Corollaries 3.1 and 5.1 show that as the target error probability $\delta \downarrow 0$, the multiplicative gaps between the expected sample complexity of LUCB and LUCB Greedy, and the information-theoretic lower bound are at most 6α and 4α respectively. Note that these are only upper bounds, and we find that empirically, the gap is smaller. Furthermore, there are bandit instances where LUCB outperforms its greedy counterpart and vice-versa.

Below, we compare the empirical performance of LUCB and LUCB Greedy via numerical simulations, with the value of hyper-parameter α chosen to be 1.2. We assume the arm reward distributions to be 1-Gaussian. We present sample

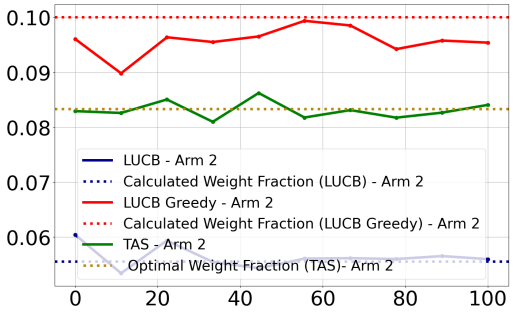

Figure 5: Pull fraction at termination of arm 2 vs $\log(1/\delta)$ for instance $\boldsymbol{\mu_1}$ under LUCB and the LUCB Greedy, along with the optimal pull fraction

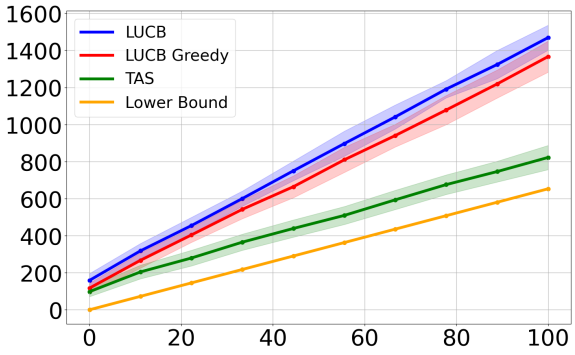

Figure 6: Sample complexity (with 1 standard deviation confidence band) vs $\log(1/\delta)$ for instance $\boldsymbol{\mu_2}$ for LUCB and LUCB Greedy, and the information theoretic lower bound

complexity results which are averaged over 50 independent runs of the corresponding algorithms.

In Figures 3, 4, and 5, we analyze a bandit instance with 10 arms where the true mean rewards are given by $\boldsymbol{\mu_1} = [2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2]$, i.e., the best arm has mean reward 2 and all the other arms have common mean reward 1.2. Figure 3 plots the average sample complexity against $\log(1/\delta)$, comparing the performance of the LUCB and LUCB Greedy algorithms against the information-theoretic lower bound (Equation 2). We observe that, for this specific bandit instance, the LUCB algorithm outperforms the LUCB Greedy algorithm. Figures 4 and 5 present the pull fractions of the top two arms for the LUCB Greedy and LUCB algorithms along with the optimal pull fractions (corresponding to the information-theoretic bound), showing that as $\delta \downarrow 0$ the pull fractions indeed converge to the asymptotic values indicated by our theoretical results (Theorems 2 and 4). From Figures 4 and 5, we also observe that the LUCB Greedy and the LUCB algorithms under-sample and over-sample the best arm relative to the optimal pull fraction respectively.

The above observation also provides an intuitive explanation for the superior performance of LUCB for this instance. Under LUCB, the separation point is closer to the true mean of the best arm, whereas in the LUCB Greedy algorithm, it is closer to the non-optimal arms. In this particular instance, all non-optimal arms have the same mean, necessitating more pulls for them to reach their respective separation points. Since LUCB tends to over-sample the best arm, it performs better here than the LUCB Greedy approach.

In Figures 6, 7, and 8, we analyze a bandit instance with 10 arms where the true mean rewards are $\boldsymbol{\mu_2} = [3, 1.4, 0.76, 0.65, 0.55, 0.47, 0.4, 0.33, 0.28, 0.22, 0.17]$. Figure 6 plots the average sample complexity vs $\log(1/\delta)$, comparing the performance of the LUCB and LUCB Greedy algorithms along with the lower bound . We observe that unlike the previous bandit instance, here the LUCB Greedy algorithm outperforms the LUCB algorithm. Intuitively, this is because for this instance, the non-optimal arms can reach the separation point in comparatively fewer pulls which better suits the LUCB Greedy scheme.

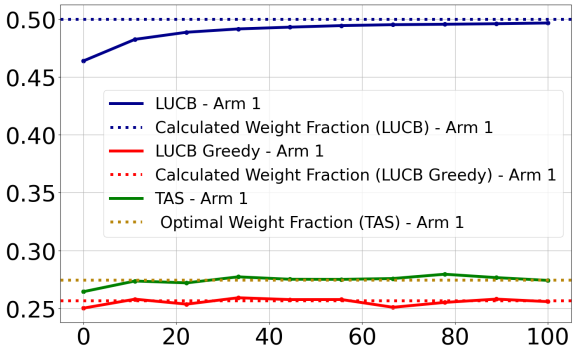

Figure 7: Pull fraction at termination of the top-arm vs $\log(1/\delta)$ for instance $\boldsymbol{\mu_2}$ for the LUCB and the LUCB Greedy algorithm along with the optimal pull fraction

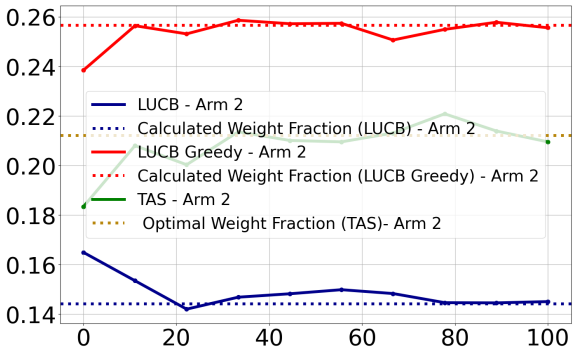

Figure 8: Pull fraction at termination for arm 2 vs $\log(1/\delta)$ for instance $\boldsymbol{\mu_2}$ for the LUCB and the LUCB Greedy algorithm along with the optimal pull fraction

Figures 7 and 8 present the pull fractions of the top two arms for the LUCB Greedy and LUCB algorithms along with the information-theoretically optimal values. As before, the pull fractions for LUCB and LUCB Greedy converge to the values suggested by our analytical results.

Acknowledgments

N.K. and J.N. acknowledge support from SERB via grant CRG/2021/002923 and two MATRICS grants.

References

Audibert, J.-Y.; Bubeck, S.; and Munos, R. 2010. Best Arm Identification in Multi-Armed Bandits. 41–53.

Bandyopadhyay, A.; Juneja, S.; and Agrawal, S. 2024. Optimal Top-Two Method for Best Arm Identification and Fluid Analysis. *arXiv preprint arXiv:2403.09123*.

Didwania, Y.; Nair, J.; and Hemachandra, N. 2022. Unsupervised Crowdsourcing with Accuracy and Cost Guarantees. In *20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks, WiOpt 2022, Torino, Italy, September 19-23, 2022*, 137–144. IEEE.

Even-Dar, E.; Mannor, S.; and Mansour, Y. 2006. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *Journal of Machine Learning Research*, 7(39): 1079–1105.

Gabillon, V.; Ghavamzadeh, M.; and Lazaric, A. 2012. Best Arm Identification: A Unified Approach to Fixed Budget and Fixed Confidence. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Garivier, A.; and Kaufmann, E. 2016. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, 998–1027. PMLR.

Jamieson, K.; and Nowak, R. 2014. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *2014 48th annual conference on information sciences and systems (CISS)*, 1–6. IEEE.

Jourdan, M.; Degenne, R.; Baudry, D.; de Heide, R.; and Kaufmann, E. 2022. Top two algorithms revisited. *Advances in Neural Information Processing Systems*, 35: 26791–26803.

Kalyanakrishnan, S.; Tewari, A.; Auer, P.; and Stone, P. 2012. PAC subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, 655–662.

Kaufmann, E.; and Kalyanakrishnan, S. 2013. Information complexity in bandit subset selection. *Journal of Machine Learning Research*, 30: 228–251.

Kaufmann, E.; and Koolen, W. M. 2017. Monte-Carlo tree search by best arm identification. *Advances in Neural Information Processing Systems*, 30.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.

Mukherjee, A.; and Tajer, A. 2023a. Best Arm Identification in Stochastic Bandits: Beyond $\beta-$optimality. arXiv:2301.03785.

Mukherjee, A.; and Tajer, A. 2023b. SPRT-based efficient best arm identification in stochastic bandits. *IEEE Journal on Selected Areas in Information Theory*, 4: 128–143.

Thananjeyan, B.; Kandasamy, K.; Stoica, I.; Jordan, M. I.; Goldberg, K.; and Gonzalez, J. E. 2021. PAC Best Arm Identification Under a Deadline. *CoRR*, abs/2106.03221.

Yao, F.; Li, C.; Nekipelov, D.; Wang, H.; and Xu, H. 2022. Learning the optimal recommendation from explorative users. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 9457–9465.