Abstract

We consider the problem of identifying any k out of the best m arms in an n-armed stochastic multi-armed bandit. Framed in the PAC setting, this particular problem generalises both the problem of "best subset selection" (**?**) and that of selecting "one out of the best m" arms (**?**). In applications such as crowd-sourcing and drug-designing, identifying a single good solution is often not sufficient. Moreover, finding the best subset might be hard due to the presence of many indistinguishably close solutions. Our generalisation of identifying exactly k arms out of the best m, where $1 \leq k \leq m$, serves as a more effective alternative. We present a lower bound on the worst-case sample complexity for general k, and a fully sequential PAC algorithm, LUCB-k-m, which is more sample-efficient on easy instances. Also, extending our analysis to infinite-armed bandits, we present a PAC algorithm that is independent of n, which identifies an arm from the best ρ fraction of arms using at most an additive poly-log number of samples than compared to the lower bound, thereby improving over **?** and **?**. The problem of identifying $k > 1$ distinct arms from the best ρ fraction is not always well-defined; for a special class of this problem, we present lower and upper bounds. Finally, through a reduction, we establish a relation between upper bounds for the "one out of the best ρ" problem for infinite instances and the "one out of the best m" problem for finite instances. We conjecture that it is more efficient to solve "small" finite instances using the latter formulation, rather than going through the former.

1. Introduction

The stochastic multi-armed bandit (**??**) is a well-studied abstraction of decision making under uncertainty. Each *arm* of a bandit represents a decision. A *pull* of an arm represents taking the associated decision, which produces a real-valued reward. The reward is drawn i.i.d. from a distribution corresponding to the arm, independent of the pulls of other arms. At each round, the experimenter may consult the preceding history of pulls and rewards to decide which arm to pull.

The traditional objective of the experimenter is to maximise the expected cumulative reward over a horizon of pulls, or equivalently, to minimise the *regret* with respect to always pulling an optimal arm. Achieving this objective requires a careful balance between *exploring* (to reduce uncertainty about the arms' expected rewards) and *exploiting* (accruing high rewards). Regret-minimisation algorithms have been used in a variety of applications, including clinical trials (**?**), adaptive routing (**?**), and recommender systems (**?**).

Of separate interest is the problem of *identifying* an arm with the highest mean reward (**???**), under what is called the "pure exploration" regime. For applications such as product testing (**?**) and strategy selection (**?**), there is a dedicated phase in the experiment in which the rewards obtained are inconsequential. Rather, the objective is to identify the best arm either (1) in the minimum number of trials, for a given confidence threshold (**??**), or alternatively, (2) with minimum error, after a given number of trials (**??**). Our investigation falls into the first category, which is termed the "fixed confidence" setting. Conceived by Bechhofer (**?**), best-arm-identification in the fixed confidence setting has received a significant amount of attention over the years (**????**). The problem has also been generalised to identify the best subset of arms (**??**).

More recently, **?** have introduced the probem of identifying a single arm from among the best m in an n-armed bandit. This formulation is particularly useful when the number of arms is large, and in fact is a viable alternative even when the number of arms is *infinite*. In many practical scenarios, however, it is required to identify more than a single good arm. For example, imagine that a company needs to complete a task that is too large to be accomplished by a single worker, but which can be broken into 5 subtasks, each capable of being completed by one worker. Suppose there are a total of 1000 workers, and an indepdendent pilot survey has revealed that at least 15% of them have the skills to complete the subtask. To address the company's need, surely it would *suffice* to identify the best 5 workers for the subtask. However, if workers are to be identified based on a skill test that has stochastic outcomes, it would be unnecessarily expensive to indeed identify the "best subset". Rather, it would be enough to merely identify any 5 workers from among the best 150. This is precisely the problem we consider in our paper: identifying any k out of the best m arms in an n-armed bandit. In addition to distributed crowdsourcing (**?**), applications of this problem include the management of large sensor networks (**?**), wherein multiple reliable sensors must be identified using minimal testing, and in drug design (**?**, Chapter 43), to identify a promising set of candidate biomarkers.

The problem assumes equal significance from a theoretical standpoint, since it generalises both the "best subset selection" problem (**?**) (taking $k = m$) and that of selecting a "single arm from the best subset" (**?**) (taking $k = 1$). Unlike best subset selection, the problem remains feasible to solve even when n is large or infinite, as long as m/n is some constant $\rho > 0$. Traditionally, infinite-armed bandits have been tackled by resorting to side information such as distances between arms (**??**) or the structure of their distribution of rewards (**?**). This approach introduces additional parameters, which might not be easy to tune in practice. Alternatively, good arms can be reached merely by selecting arms *at random* and testing them by pulling. This latter approach has been applied successfully both in the regret-minimisation setting (**?**) and in the fixed-confidence setting (**??**). Our formulation paves the way for identifying "many" (k) "good" (in the top m among n) arms in this manner.

The interested reader may refer to Table 1 right away for a summary of our theoretical results, which are explained in detail after formally specifying the (k, m, n) and (k, ρ) problems in Section 2. In Section 3 we present our algorithms and analysis for the finite setting, and in Section 4 for the infinite setting. We present experimental results in Section 5, and conclude with a discussion in Section 6.

2. Problem Definition and Contributions

Let $\mathcal{A}$ be the set of arms in our given bandit instance. With each arm $a \in \mathcal{A}$, there is an associated reward distribution supported on a subset of $[0, 1]$, with mean μ_a. When pulled, arm $a \in \mathcal{A}$ produces a reward drawn i.i.d. from the corresponding distribution, and independent of the pulls of other arms. At each round, based on the preceding sequence of pulls and rewards, an algorithm either decides which arm to pull, or stops and returns a set of arms.

For a finite bandit instance with n arms, we take $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$, and assume, without loss of generality, that for arms $a_i, a_j \in \mathcal{A}$, $\mu_{a_i} \geq \mu_{a_j}$ whenever $i \leq j$. Given a tolerance $\epsilon \in [0, 1]$ and $m \in \{1, 2, \ldots, n\}$, we call an arm $a \in \mathcal{A}$ (ϵ, m)-optimal if $\mu_a \geq \mu_{a_m} - \epsilon$. We denote the set of all the (ϵ, m)-optimal arms as $\mathcal{TOP}_m(\epsilon) \overset{\text{def}}{=} \{a : \mu_a \geq \mu_{a_m} - \epsilon\}$. For brevity we denote $\mathcal{TOP}_m(0)$ as $\mathcal{TOP}_m$.

Definition (k, m, n) Problem. An instance of the (k, m, n) problem is of the form $(\mathcal{A}, n, m, k, \epsilon, \delta)$, where $\mathcal{A}$ is a set of arms with $|\mathcal{A}| = n \geq 2$; $m \in \{1, 2, \ldots, n - 1\}$; $k \in \{1, \ldots, m\}$; tolerance $\epsilon \in (0, 1]$; and mistake probability $\delta \in (0, 1]$. An algorithm $\mathcal{L}$ is said to solve (k, m, n) if for every instance of (k, m, n), it terminates with probability 1, and returns k *distinct* (ϵ, m)-optimal arms with probability at least $1 - \delta$.

The (k, m, n) problem is interesting from a theoretical standpoint because it covers an entire range of problems, with single-arm identification ($m = 1$) at one extreme and subset selection ($k = m$) at the other. Thus, any bounds on the sample complexity of (k, m, n) also apply to Q-F (**?**) and to SUBSET (**?**). In this paper, we show that any algorithm that solves (k, m, n) must incur $\Omega\left(\frac{n}{(m-k+1)\epsilon^2} \log\left(\frac{\binom{m}{k-1}}{\delta}\right)\right)$ pulls for some instance of the problem. We are unaware of bounds in the fixed-confidence setting that involve such a combinatorial term inside the logarithm.

Table 1 places our bounds in the context of previous results. The bounds shown in the table consider the worst-case across problem instances; in practice one can hope to do better on easier problem instances by adopting a fully sequential sampling strategy. Indeed we adapt the LUCB1 algorithm (**?**) to solve (k, m, n), denoting the new algorithm LUCB-k-m. Our analysis shows that for $k = 1$, and $k = m$, the upper bound on the sample complexity of this algorithm matches with those of $\mathcal{F}_2$ (**?**) and LUCB1 (**?**), respectively, up to a multiplicative constant. Empirically, LUCB-k-m with $k = 1$ appears to be more efficient than $\mathcal{F}_2$ for solving Q-F.

Along the same lines that **?** (**?**) define the Q-P problem for infinite instances, we define a generalisation of Q-P for selecting many good arms, which we denote (k, ρ). Given a set of arms $\mathcal{A}$, a sampling distribution $P_{\mathcal{A}}$, $\epsilon \in (0, 1]$, and $\rho \in [0, 1]$, an arm $a \in \mathcal{A}$ is called $[\epsilon, \rho]$-optimal if $P_{a' \sim P_{\mathcal{A}}}\{\mu_a \geq \mu_{a'} - \epsilon\} \geq 1 - \rho$. For $\rho, \epsilon \in [0, 1]$, we define the set of all $[\epsilon, \rho]$-optimal arms as $\mathcal{TOP}_\rho(\epsilon)$. As before, we denote $\mathcal{TOP}_\rho(0)$ as $\mathcal{TOP}_\rho$. A straightforward generalisation of Q-P is as follows.

Definition (k, ρ) Problem. An instance of the (k, ρ) problem is of the form $(\mathcal{A}, P_{\mathcal{A}}, k, \rho, \epsilon, \delta)$, where $\mathcal{A}$ is a set of arms; $P_{\mathcal{A}}$ is a probability distribution over $\mathcal{A}$; quantile fraction $\rho \in (0, 1]$; tolerance $\epsilon \in (0, 1]$; and mistake probability $\delta \in (0, 1]$. Such an instance is *valid* if $|\mathcal{TOP}_\rho| \geq k$, and *invalid* otherwise. An algorithm $\mathcal{L}$ is said to solve (k, ρ), if for every *valid* instance of (k, ρ), $\mathcal{L}$ terminates with probability 1, and returns k *distinct* $[\epsilon, \rho]$-optimal arms with probability at least $1 - \delta$.

At most k-equiprobable instances. Observe that (k, ρ) is well-defined only if the given instance has at least k distinct arms in $\mathcal{TOP}_\rho$; we term such an instance *valid*. It is worth noting that even valid instances can require an arbitrary amount of computation to solve. For example, consider an instance with $k > 1$ arms in $\mathcal{TOP}_\rho$, one among which has a probability γ of being picked by $P_{\mathcal{A}}$, and the rest each a probability of $(\rho - \gamma)/(k - 1)$. Since the arms have to be identified by sampling from $P_{\mathcal{A}}$, the probability of identifying the latter $k - 1$ arms diminishes to 0 as $\gamma \to \rho$, calling for an infinite number of guesses. To avoid this scenario, we restrict our analysis to a special class of valid instances in which $P_{\mathcal{A}}$ allocates no more than ρ/k probability to any arm in $\mathcal{TOP}_\rho$. We refer to such instances as "at most k-equiprobable" instances. Formally, a (k, ρ) problem instance given by $(\mathcal{A}, P_{\mathcal{A}}, k, \rho, \epsilon, \delta)$ is called "at most k-equiprobable" if $\forall a \in \mathcal{TOP}_\rho$, $\Pr_{\mathbf{a}' \sim P_{\mathcal{A}}}\{\mathbf{a}' = a\} \leq \frac{\rho}{k}$.[1]

[1] In a recent paper, **?** claim to solve the (k, ρ) problem. However, they do not notice that the problem can be ill-posed. Also, even with an at most k-equiprobable instance as input, their algo-

Table 1. Lower and upper bounds on the expected sample complexity (worst case over problem instances). The bounds for (k, ρ), $k > 1$ are for the special class of "at most k-equiprobable" instances.

Problem	Lower Bound	Previous Upper Bound	Current Upper Bound
$(1, 1, n)$ Best-Arm	$\Omega\left(\frac{n}{\epsilon^2}\log\frac{1}{\delta}\right)$ **(?)**	$O\left(\frac{n}{\epsilon^2}\log\frac{1}{\delta}\right)$ **(?)**	Same as previous
(m, m, n) SUBSET	$\Omega\left(\frac{n}{\epsilon^2}\log\frac{m}{\delta}\right)$ **(?)**	$O\left(\frac{n}{\epsilon^2}\log\frac{m}{\delta}\right)$ **(?)**	Same as previous
$(1, m, n)$ Q-F	$\Omega\left(\frac{n}{m\epsilon^2}\log\frac{1}{\delta}\right)$ **(?)**	$O\left(\frac{n}{m\epsilon^2}\log^2\frac{1}{\delta}\right)$	$O\left(\frac{1}{\epsilon^2}\left(\frac{n}{m}\log\frac{1}{\delta} + \log^2\frac{1}{\delta}\right)\right)$ **This paper**
(k, m, n) Q-F$_k$	$\Omega\left(\frac{n}{(m-k+1)\epsilon^2}\log\frac{\binom{m}{k-1}}{\delta}\right)$ **This paper**	-	$O\left(\frac{k}{\epsilon^2}\left(\frac{n\log k}{m}\log\frac{k}{\delta} + \log^2\frac{k}{\delta}\right)\right)^*$ **This paper** (*for $k \geq 2$)
$(1, \rho)$ $(\lvert\mathcal{A}\rvert = \infty)$ Q-P	$\Omega\left(\frac{1}{\rho\epsilon^2}\log\frac{1}{\delta}\right)$ **(?)**	$O\left(\frac{1}{\rho\epsilon^2}\log^2\frac{1}{\delta}\right)$	$O\left(\frac{1}{\epsilon^2}\left(\frac{1}{\rho}\log\frac{1}{\delta} + \log^2\frac{1}{\delta}\right)\right)$ **This paper**
(k, ρ) $(\lvert\mathcal{A}\rvert = \infty)$ Q-P$_k$	$\Omega\left(\frac{k}{\rho\epsilon^2}\log\frac{k}{\delta}\right)$ **This paper**	-	$O\left(\frac{k}{\epsilon^2}\left(\frac{\log k}{\rho}\log\frac{k}{\delta} + \log^2\frac{k}{\delta}\right)\right)^*$ **This paper** (*for a special class with $k \geq 2$)

o understand the notion of (k, ρ) let us take a concrete example. We clarify the notion of (k, ρ) via the following example.

Example 1. *Given a set of arms $\mathcal{A} = [0, 1]$, such that $\mu_a = a$. Now for some $\gamma \in [0, 0.5]$,bwe define a sampling distribution $P_{\mathcal{A}}^{(\gamma)}$ on $\mathcal{A}$ such that $\Pr_{a\sim P_{\mathcal{A}}^{(\gamma)}}\{\mu_a \in [0, 0.5]\} = 0.4$, $\Pr_{a\sim P_{\mathcal{A}}^{(\gamma)}}\{\mu_a \in (0.5, 0.95] \cup (0.98, 1)\} = 0$, $\Pr_{a\sim P_{\mathcal{A}}^{(\gamma)}}\{\mu_a(0.95, 0.98]\} = \gamma$, $\Pr_{a\sim P_{\mathcal{A}}^{(\gamma)}}\{\mu_a = 1\} = 0.6 - \gamma$.*

$$\Pr_{a\sim P_{\mathcal{A}}^{(\gamma)}}\{0 \leq \mu_a \leq 0.5\} = 0.4, \Pr_{a\sim P_{\mathcal{A}}^{(\gamma)}}\{0.5 < \mu_a \leq 0.95\} = 0,$$
$$\Pr_{a\sim P_{\mathcal{A}}^{(\gamma)}}\{0.95 < \mu_a < 1\} = \gamma, \Pr_{a\sim P_{\mathcal{A}}^{(\gamma)}}\{\mu_a = 1\} = 0.6 - \gamma.$$

For $\rho = 0.5$, $k = 2$, $\epsilon = 0.1$, and $\delta \in (0, 1)$, we define an instance of (k, ρ) as $\mathcal{I}^{(\gamma)} \overset{\text{def}}{=} (\mathcal{A}, P_{\mathcal{A}}^{(\gamma)}, k, \rho, \epsilon, \delta)$. Now according to the definition of (k, ρ), the instance $\mathcal{I}^{(0)}$ is invalid; for all $\nu \in (0, 0.1)$ an instance $\mathcal{I}^{(\nu)}$ is invalid, but there exist k distinct $[\epsilon, \rho]$-optimal arms; however, its solution will cease to exist for $\epsilon < 0.02$. Therefore, such an instance is not interesting, as it may lack existence of arbitrarily close approximate solution. Again for any $\gamma > 0.1$, $\mathcal{I}^{(\gamma)}$ is a valid instance; however, as $\gamma \downarrow 0.1$, it gets arbitrarily harder to solve $\mathcal{I}^{(\gamma)}$.

Note that any instance of the $(1, \rho)$ or Q-P (**?**) problem is necessarily valid and at most 1-equiprobable. Interestingly, we improve upon the existing upper bound for this problem, so it matches the lower bound up to an *additive* $O\left(\frac{1}{\epsilon^2}\log^2\frac{1}{\delta}\right)$ term. Below we summarise our contributions.

1. We generalise two previous problems—Q-F and SUBSET (**?**)—via (k, m, n). In Section 3 we derive a lower bound on the worst case sample complexity to solve (k, m, n), which generalises existing lower bounds for Q-F and SUBSET.

2. In Section 3.2 we extend LUCB1 (**?**) to present a fully-sequential algorithm—*<u>LUCB</u> for <u>k</u> out of <u>m</u>* or LUCB-k-m—to solve (k, m, n). We shows that for $k = 1$, and $k = m$ the upper bound on its expected sample complexity matches with those of $\mathcal{F}_2$, and LUCB1, respectively, up to a constant factor.

3. In Section 4 we present algorithm $\mathcal{P}_3$ to solve Q-P with a sample complexity that is an additive $O((1/\epsilon^2)\log^2(1/\delta))$ term away from the lower bound. We extend it to an algorithm KQP-1 for solving at most k-equiprobable (k, ρ) instances. Also, $\mathcal{P}_3$ and KQP-1 can solve Q-F and (k, m, n) respectively, and their sample complexities are the tightest instance-independent upper bounds as yet.

4. In Section 4.3 we present a general relation between the upper bound on the sample complexities for solving Q-F and Q-P. This helps in effectively transferring any improvement in the upper bound on the former to the latter. Also, we conjecture the existence of a class of Q-F instances that can be solved more efficiently than their "corresponding" Q-P instances.

5. In Section 5 we experimentally show that LUCB-k-m is significantly more efficient than $\mathcal{F}_2$ for solving Q-F.

3. Algorithms for Finite Instances

We begin our technical presentation by furnishing a lower bound on the sample complexity of algorithms for (k, m, n).

rithm fails to escape the $(1/\rho)\log^2(1/\delta)$ dependence.

3.1. Lower Bound on the Sample-Complexity

Theorem 3.1. *[Lower Bound for (k, m, n)] Let $\mathcal{L}$ be an algorithm that solves (k, m, n). Then, there exists an instance $(\mathcal{A}, n, m, k, \epsilon, \delta)$, with $0 < \epsilon \leq \frac{1}{\sqrt{32}}$, $0 < \delta \leq \frac{e^{-1}}{4}$, and $n \geq 2m$, $1 \leq k \leq m$, on which the expected number of pulls performed by $\mathcal{L}$ is at least $\frac{1}{18375} \cdot \frac{1}{\epsilon^2} \cdot \frac{n}{m-k+1} \ln \frac{\binom{m}{k-1}}{4\delta}$.*

The detailed proof of the theorem is given in Appendix 7. The proof generalises lower bound proofs for both (m, m, n) (**?**, see Theorem 8) and $(1, m, n)$ (**?**, see Theorem 3.3). The core idea in these proofs is to consider two sets of bandit instances, $\mathcal{I}$ and $\mathcal{I}'$, such that over "short" trajectories, an instance from $\mathcal{I}$ will yield the same reward sequences as a corresponding instance from $\mathcal{I}'$, with high probability. Thus, any algorithm will return the same set of arms for both instances, with high probability. However, by construction, no set of arms can be simultaneously correct for both instances—implying that a correct algorithm must encounter sufficiently "long" trajectories. Our main contribution is in the design of $\mathcal{I}$ and $\mathcal{I}'$ when $k \in \{1, 2, \ldots, m\}$ (rather than exactly 1 or m) arms have to be returned.

Our algorithms to achieve improved *upper* bounds for Q-F and (k, m, n) (across bandit instances) follow directly from methods we design for the infinite-armed setting in Section 4 (see Corollary 4.2 and Corollary 4.5). In the remainder of this section, we present a fully-sequential algorithm for (k, m, n) whose expected sample complexity varies with the "hardness" of the input instance.

3.2. An Adaptive Algorithm for Solving (k, m, n)

Algorithm 1 describes LUCB-k-m, a fully sequential algorithm, which for $k = 1$ has the same guarantee on sample-complexity as $\mathcal{F}_2$, but empirically appears to be more economical. The algorithm generalises LUCB1 (**?**), which solves (m, m, n).

Algorithm 1 LUCB-k-m: Algorithm to select k (ϵ, m)-optimal arms

```
Input: $\mathcal{A}$ (such that $|\mathcal{A}| = n$), $k, m, \epsilon, \delta$.
Output: $k$ distinct $(\epsilon, m)$-optimal arms from $\mathcal{A}$.
  Pull each arm $a \in \mathcal{A}$ once. Set $t = n$.
  while $ucb(l^t_*, t+1) - lcb(h^t_*, t+1) > \epsilon$. do
    $t = t + 1$.
    $A^t_1 \overset{\text{def}}{=}$ Set of $k$ arms with the highest empirical means.
    $A^t_3 \overset{\text{def}}{=}$ Set of $n - m$ arms with the lowest empirical means.
    $A^t_2 \overset{\text{def}}{=} \mathcal{A} \setminus (A^t_1 \cup A^t_3)$.
    $h^t_* = \arg\max_{\{a \in A^t_1\}} lcb(a, t)$.
    $m^t_* = \arg\min_{\{a \in A^t_2\}} u^t_a$.
    $l^t_* = \arg\max_{\{a \in A^t_3\}} ucb(a, t)$.
    pull $h^t_*, m^t_*, l^t_*$.
  end while
  Return $A^t_1$.
```

At each round t, we partition $\mathcal{A}$ into three subsets. We keep the k arms with the highest empirical averages in A^t_1, the $n - m$ arms with the lowest empirical averages in A^t_3, and the rest in A^t_2; ties are broken arbitrarily (uniformly at random in our experiments). At each round we choose a *contentious* arm from each of these three sets: from A^t_1 we choose h^t_*, the arm with the lowest lower confidence bound (LCB); from A^t_2 the arm which is least pulled is chosen, and called m^t_*; from A^t_3 we choose l^t_*, the arm with the highest upper confidence bound (UCB). The algorithm stops as soon as the difference between the lower confidence bound of h^t_*, and the upper confidence bound of l^t_* becomes no larger than the tolerance ϵ.

Let B_1, B_2, B_3 be corresponding sets based on the true means: that is, subsets of $\mathcal{A}$ such that $B_1 \overset{\text{def}}{=} \{1, 2, \cdots, k\}$, $B_2 = \{k+1, k+2, \cdots, m\}$ and $B_3 = \{m+1, m+2, \cdots, n\}$. For any two arms $a, b \in \mathcal{A}$ we define $\Delta_{ab} \overset{\text{def}}{=} \mu_a - \mu_b$. For the sake of convenience we slightly overload this notation as

$$\Delta_a = \begin{cases} \mu_a - \mu_{m+1} & \text{if } a \in B_1 \\ \mu_k - \mu_{m+1} & \text{if } a \in B_2 \\ \mu_m - \mu_a & \text{if } a \in B_3. \end{cases} \tag{1}$$

We note that $\Delta_k = \Delta_{k+1} = \cdots = \Delta_m = \Delta_{m+1}$. Let $u^*(a, t) \overset{\text{def}}{=} \left\lceil \frac{32}{\max\{\Delta_a, \frac{\epsilon}{2}\}^2} \ln \frac{k_1 n t^4}{\delta} \right\rceil$ for all $a \in \mathcal{A}$, where $k_1 = 5/4$. Now, we define the hardness term as $H_\epsilon = \sum_{a \in \mathcal{A}} \frac{1}{\max\{\Delta_a, \epsilon/2\}^2}$.

Theorem 3.2. *[Expected Sample Complexity of* LUCB*-k-m]* LUCB*-k-m solves (k, m, n) using an expected sample complexity upper bounded by $O\left(H_\epsilon \log \frac{H_\epsilon}{\delta}\right)$.*

Appendix-A describes the proof in detail. The core argument is similar to that for Algorithm $\mathcal{F}_2$ by **?**. However, it subtly differs due to the different strategy for choosing arms since the output set is not necessarily singleton. In practice, one can use tighter confidence bound calculations (we use KL-divergence based confidence bounds in our experiments) to get even better sample complexity.

Next, we are going to consider infinite-armed bandit instances, and present the algorithms to solve them.

4. Algorithm for Infinite Instances

Before proceeding to the identification of k $[\epsilon, \rho]$-optimal arms in infinite-armed bandits, we revisit the case of $k = 1$. To find a single $[\epsilon, \rho]$-optimal arm, the sample complexity of all the existing algorithms (**??**) scales as $(1/\rho\epsilon^2)\log^2(1/\delta)$, for the given mistake probability δ. In this section we present an algorithm $\mathcal{P}_3$ whose sample complexity is only an *additive* poly-log factor away from the lower bound of $\Omega((1/\rho\epsilon^2)\log 1/\delta)$ (**?**, Corollary 3.4).

4.1. Solving Q-P Instances

$\mathcal{P}_3$ is a two-phase algorithm. In the first phase, it runs a sufficiently large number of independent copies of $\mathcal{P}_2$ and chooses a large subset of arms (say of size u), in which every arm is $[\epsilon, \rho]$-optimal with probability at least $1 - \delta'$, where δ' is some small *constant*. The value u is chosen in a manner such that at least one of the chosen arms is $[\epsilon/2, \rho]$-optimal with probability at least $\delta/2$. The second phase solves the best arm identification problem $(1, 1, u)$ by applying MEDIAN ELIMINATION.

Algorithm 2 describes $\mathcal{P}_3$. It uses $\mathcal{P}_2$ (**?**) with MEDIAN ELIMINATION as a subroutine, to select an $[\epsilon, \rho]$-optimal arm with confidence $1 - \delta'$. We have assumed $\delta' = 1/4$, in practice the one can choose any sufficiently small value for it, which will merely affect the multiplicative constant in the upper bound.

Algorithm 2 $\mathcal{P}_3$

```
Input: $\mathcal{A}, \epsilon, \delta$.
Output: One $[\epsilon, \rho]$-optimal arm.
  Set $\delta' = 1/4$, $u = \left\lceil \frac{1}{\delta'} \log \frac{2}{\delta} \right\rceil = \left\lceil 4 \log \frac{2}{\delta} \right\rceil$.
  Run $u$ copies of $\mathcal{P}_2(\mathcal{A}, \rho, \epsilon/2, \delta')$ and form set $S$ with the output arms.
  Identify an $(\epsilon/2, 1)$-optimal arm in $S$ using MEDIAN ELIMINATION with confidence at least $1 - \delta/2$.
```

Theorem 4.1. *[Correctness and Sample Complexity of $\mathcal{P}_3$] $\mathcal{P}_3$ solves* Q-P*, with sample complexity* $O(\epsilon^{-2}(\rho^{-1} \log(1/\delta) + \log^2(1/\delta)))$.

Proof. First we prove the correctness and then upper-bound the sample complexity.

Correctness. First we notice that each copy of $\mathcal{P}_2$ outputs an $[\epsilon/2, \rho]$-optimal arm with probability at least $1 - \delta'$. Now, $S \cap \mathcal{TOP}_\rho = \emptyset$ can only happen if all the u copies of $\mathcal{P}_2$ output sub-optimal arms. Therefore, $\Pr\{S \cap \mathcal{TOP}_\rho = \emptyset\} = (1 - \delta')^u \leq \delta/2$. On the other hand, the mistake probability of MEDIAN ELIMINATION is upper bounded by $\delta/2$. Therefore, by taking union bound, we get the mistake probability is upper bounded by δ. Also, the mean of the output arm is not less than $\frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon$ from the $(1 - \rho)$-th quantile.

Sample complexity. First we note that, for some appropriate constant C, the sample complexity (SC) of each of the u copies of $\mathcal{P}_2$ is $\frac{C}{\rho(\epsilon/2)^2} \left(\ln \frac{2}{\delta'}\right)^2 \in O\left(\frac{1}{\rho\epsilon^2}\right)$. Hence, SC of all the u copies $\mathcal{P}_2$ together is upper bounded by $\frac{C_1 \cdot u}{\rho\epsilon^2}$, for some constant C_1. Also, for some constant C_2, the sample complexity of MEDIAN ELIMINATION is upper bounded by $\frac{C_2 \cdot u}{(\epsilon/2)^2} \ln \frac{2}{\delta} \leq \frac{C_3}{\epsilon^2} \ln^2 \frac{2}{\delta}$. Adding the sample complexities and substituting for u yields the bound. □

Corollary 4.2. *$\mathcal{P}_3$ can solve any instance of* Q-F $(\mathcal{A}, n, m, \epsilon, \delta)$ *with sample complexity* $O\left(\frac{1}{\epsilon^2}\left(\frac{n}{m} \log \frac{1}{\delta} + \log^2 \frac{1}{\delta}\right)\right)$.

Proof. Let, $(\mathcal{A}, n, m, \epsilon, \delta)$ be the given instance of Q-F. We partition the set $\mathcal{A}^\infty = [0, 1]$ in to n equal segments and associate each with a unique arm in $\mathcal{A}$, and such that no two different subsets get associated with the same arm. Now, defining $P_{A^\infty} = Uniform[0, 1]$, and $\rho' = m/n$, we realise that solving the Q-P instance $(\mathcal{A}^\infty, P_{\mathcal{A}^\infty}, \rho', \epsilon, \delta)$ solves the original Q-F instance, thereby proving the corollary. □

At this point it is of natural interest to find an efficient algorithm to solve (k, ρ). Next, we discuss the extension of Q-P to (k, ρ), and present lower and upper bound on the sample complexity needed to solve it.

4.2. Solving "At Most k-equiprobable" (k, ρ) Instances

Now, let us focus on identifying k $[\epsilon, \rho]$-optimal arms. In Theorem 4.3 we derive the lower bound on the sample complexity to solve an instance (k, ρ) by reducing it to solving a SUBSET problem as follows.

Theorem 4.3. *[Lower Bound on the Sample Complexity for Solving (k, ρ)] For every $\epsilon \in (0, \frac{1}{\sqrt{32}}]$, $\delta \in (0, \frac{1}{\sqrt{32}}]$, and $\rho \in (0, \frac{1}{2}]$, there exists an instance of (k, ρ) given by $(\mathcal{A}, P_\mathcal{A}, \rho, \epsilon, \delta)$, such that any algorithm that solves (k, ρ) incurs at least $C \cdot \frac{k}{\rho\epsilon^2} \ln \frac{k}{8\delta}$ samples, where $C = \frac{1}{18375}$.*

Proof. We shall prove the theorem by contradiction. Let us assume that the statement is incorrect. Therefore, there exists an algorithm ALG that ALG can solve any instance of (k, ρ) using no more than $C \cdot \frac{k}{\rho\epsilon^2} \ln \frac{k}{8\delta}$ samples, for $C = \frac{1}{18375}$. Now, let $(n, \mathcal{A}, m, \epsilon, \delta)$ be an instance of SUBSET (**?**) with $n \geq 2m$. Letting $P_\mathcal{A} = Uniform\{1, 2, \ldots, n\}$, $k = m$, and $\rho = m/n$, we create an instance of (k, ρ) as $(\mathcal{A}, P_\mathcal{A}, \rho, k, \epsilon, \delta)$. Therefore, solving this (k, ρ) instance will solve the original SUBSET instance. According our claim, ALG solves the original SUBSET instance using at most $C \cdot \frac{k}{(k/n)\epsilon^2} \ln \frac{k}{8\delta} = C \cdot \frac{m}{(m/n)\epsilon^2} \ln \frac{m}{8\delta} = C \cdot \frac{n}{\epsilon^2} \ln \frac{m}{8\delta}$ samples. This observation contradicts the lower bound on the sample complexity for solving SUBSET (**?**, Theorem 8); thereby proving the theorem. □

Algorithm for solving at most k-equiprobable (k, ρ) instances. Let, for any $\mathcal{S} \subseteq \mathcal{A}$, $\nu(\mathcal{S}) \stackrel{\text{def}}{=} \Pr_{a \sim P_\mathcal{A}}\{a \in \mathcal{S}\}$. Therefore, $\nu(\mathcal{A}) = 1$. Now, we present an algorithm KQP-1 that can solve any at most k-equiprobable instance of (k, ρ). Algorithm 3 describes KQP-1. At each phase y, it solves an instance of Q-P to output an arm, say $a^{(y)}$, from $\mathcal{TOP}_\rho(\epsilon)$. In the next phase, it updates the bandit instance $\mathcal{A}^{y+1} = \mathcal{A}^y \setminus \{a^{(y)}\}$, the sampling distribution $P_{\mathcal{A}^{y+1}} = \frac{1}{1 - \nu(\mathcal{A} \setminus \mathcal{A}^{y+1})} P_{\mathcal{A}^y}$, and the target quantile

$\rho^{y+1} = \rho^y - \nu(a^{(y)})$. However, as we are not given the explicit form of $P_{\mathcal{A}}$, we realise $P_{\mathcal{A}^{y+1}}$ by rejection-sampling—if $a' \in \mathcal{A} \setminus \mathcal{A}^{y+1}$ is chosen by $P_{\mathcal{A}}$, we simply discard a', and continue to sample $P_{\mathcal{A}}$ one more time. Because $\nu(\{a^y\})$ is not known explicitly, we rely on the fact that $\nu(\{a^y\}) \leq \rho/k$: it is for this reason we require the instance to be at most k-equiprobable. Therefore, in each phase $y \geq 1$, $\rho^y - \rho/k \leq \rho^{y+1} \leq \rho^y - \nu\{a^y\}$, and hence, KQP-1 solves an instance of Q-P given by $(\mathcal{A}^y, P_{\mathcal{A}^y}, \rho - (y-1)\rho/k, \epsilon, \delta)$.

Algorithm 3 KQP-1: Algorithm to solve a at most k-equiprobable (k, ρ) instances

```
Input: $\mathcal{A}, P_{\mathcal{A}}, k, \rho, \epsilon, \delta$.
Output: Set of $k$ distinct arms from $\mathcal{TOP}_\rho(\epsilon)$.
  $\mathcal{A}^1 = \mathcal{A}, \rho^1 = \rho$.
  for $y = 1, 2, 3, \cdots, k$ do
    Run $\mathcal{P}_3$ to solve the Q-P instance given by
    $(\mathcal{A}^y, P_{\mathcal{A}^y}, \rho^y, \epsilon, \frac{\delta}{k})$, and let $a^{(y)}$ be the output.
    $\mathcal{A}^{y+1} = \mathcal{A}^y \setminus \{a^{(y)}\}$.
    $\rho^{y+1} = \rho^y - ((y-1)\rho)/k$.
  end for
```

In Theorem 4.4 we present an upper bound on the expected sample complexity of KQP-1.

Theorem 4.4. *Given any at most k-equiprobable instance of (k, ρ) with $k > 1$,* KQP-*1 solves the instance with expected sample-complexity upper bounded by* $O\left(\frac{k}{\epsilon^2}\left(\frac{\log k}{\rho}\log\frac{k}{\delta} + \log^2\frac{k}{\delta}\right)\right)$.

Proof. We break the proof in two parts: upper-bounding the sample complexity, and proving correctness.

Sample complexity: In phase y, the sample complexity of $\mathcal{P}_3$ is upper-bounded as $\text{SC}(y) \leq \frac{C}{\rho^y\epsilon^2}\log\frac{k}{\delta}$, for some constant C. Therefore, the sample complexity of KQP-1 is upper bounded as

$$\begin{aligned}
\sum_{y=1}^{k}\text{SC}(y) &\leq \sum_{y=1}^{k}\frac{C}{\epsilon^2}\left(\frac{1}{\rho^y}\log\frac{k}{\delta} + \log^2\frac{k}{\delta}\right), \\
&\leq \frac{C}{\epsilon^2}\left(\log\frac{k}{\delta}\sum_{y=1}^{k}\frac{1}{\rho - (y-1)\frac{\rho}{k}} + k\log^2\frac{k}{\delta}\right), \\
&= \frac{Ck}{\epsilon^2}\left(\frac{1}{\rho}\log\frac{k}{\delta}\sum_{y=1}^{k}\frac{1}{k-y+1} + \log^2\frac{k}{\delta}\right), \\
&\leq \frac{C'k}{\epsilon^2}\left(\frac{\log k}{\rho}\log\frac{k}{\delta} + \log^2\frac{k}{\delta}\right),
\end{aligned}$$

for $k > 1$, and some constant C'.

Correctness: Letting E_y be the event that $a^{(y)} \notin \mathcal{TOP}_\rho(\epsilon)$, the probability of mistake by KQP-1 can be upper bounded as $\Pr\{\text{Error}\} = \Pr\{\exists y \in \{1, \cdots, k\}\ E_y\} \leq \sum_{y=1}^{k}\Pr\{E_y\} \leq \sum_{y=1}^{k}\frac{\delta}{k} = \delta$. □

Corollary 4.5. KQP-*1 can solve any instance of (k, m, n) given by $(\mathcal{A}, n, m, k, \epsilon, \delta)$ with $k \geq 2$, using* $O\left(\frac{k}{\epsilon^2}\left(\frac{n\log k}{m}\log\frac{k}{\delta} + \log^2\frac{k}{\delta}\right)\right)$ *samples.*

We note that though the sample complexity of KQP-1 is independent of size of the bandit instance $\mathcal{A}$, and every instance of (k, m, n) given by $(\mathcal{A}, n, m, m, \epsilon, \delta)$, can be solved by KQP-1 by posing it as an instance of (k, ρ) given by $(\mathcal{A}, Uniform\{\mathcal{A}\}, m/n, m, \epsilon, \delta)$. However, for $k = m$, the sample complexity of KQP-1 reduces to $O\left(\frac{1}{\epsilon^2}\left(n\log m \cdot \log\frac{m}{\delta} + \log^2\frac{m}{\delta}\right)\right)$, which is higher than the sample complexity of Halving (**?**), that needs only $O\left(\frac{n}{\epsilon^2}\log\frac{m}{\delta}\right)$ samples. Hence, for the best subset selection problem in finite instances Halving is preferable to KQP-1. However, in the very large instances, where the probability of picking any given arm from $\mathcal{TOP}_\rho$ is close to zero, (k, ρ) is the ideal problem to solve, and KQP-1 is the first solution that we propose.

Corollary 4.6. *Every instance of (k, ρ) given by $(\mathcal{A}, P_{\mathcal{A}}, k, \rho, \epsilon, \delta)$, such that $|\mathcal{A}| = \infty$, and for all finite subset $S \subset \mathcal{A}$, $\Pr_{a\sim P_{\mathcal{A}}}\{a \in S\} = 0$; can be solved within a sample-complexity $O\left(k\epsilon^{-2}\left(\rho^{-1}\log(k/\delta) + \log^2(k/\delta)\right)\right)$, by independently solving k different* Q-P *instances, each given by $(\mathcal{A}, P_{\mathcal{A}}, k, \rho, \epsilon, \delta/k)$.*

The correctness of Corollary 4.6 gets proved by noticing the fact that all the k outputs are unique with probability 1, and then taking union bound over mistake probabilities. Before going to the experiments, we present an important result on the hardness of solving Q-P.

4.3. On the Hardness of Solving Q-P

Theorem 4.7 presents a general relation between the upper bound on sample complexities for solving Q-F and Q-P.

Theorem 4.7. *Let $\gamma : \mathbb{Z}^+ \times \mathbb{Z}^+ \times [0,1] \times [0,1] \mapsto \mathbb{R}^+$. If every instance of* Q-F *given by $(\mathcal{A}, n, m, \epsilon, \delta)$, can be solved within the sample-complexity $O\left(\frac{n}{m\epsilon^2}\log\frac{1}{\delta} + \gamma(n, m, \epsilon, \delta)\right)$, then, every instance of* Q-P *given by $(\mathcal{A}, P_{\mathcal{A}}, \rho, \epsilon, \delta)$ can be solved within the sample-complexity* $O\left((1/\rho\epsilon^2)\log(1/\delta) + \gamma\left(\lceil 8\log(2/\delta)\rceil, \lfloor 4\log(2/\delta)\rfloor, \epsilon/2, \delta/2\right)\right)$.

We assume that there exists an algorithm OptQF that solves every instance of Q-F given by $(\mathcal{A}, n, m, \epsilon, \delta)$, using $O\left(\frac{n}{m\epsilon^2}\log\frac{1}{\delta} + \gamma(n, m, \epsilon, \delta)\right)$ samples. We establish the upper bound on sample complexity for solving Q-P by constructing an algorithm OptQP that follows an approach similar to $\mathcal{P}_3$. Specifically, OptQP reduces the input Q-P instance to an instance of Q-F using $O\left(\frac{1}{\rho\epsilon^2}\log\frac{1}{\delta}\right)$ samples. Then, it solves that Q-F using OptQF as the subroutine. The detailed proof is given in Appendix-C.

Corollary 4.8. *Corollary 4.2 shows that every* Q-F *is solvable in $O\left(\frac{1}{\epsilon^2}\left(\frac{n}{m}\log\frac{1}{\delta} + \log^2\frac{1}{\delta}\right)\right)$ samples. Hence,*

$\gamma(n, m, \epsilon, \delta) \in O\left(\frac{1}{\epsilon^2} \log^2 \frac{1}{\delta}\right)$, *and therefore, every* Q-P *is solvable in* $O\left(\frac{1}{\epsilon^2}\left(\frac{1}{\rho} \log \frac{1}{\delta} + \log^2 \frac{1}{\delta}\right)\right)$ *samples.*

On the other hand, if the lower bound for solving Q-F *provided by* **?** *matches the upper bound up to a constant factor, then* $\gamma(n, m, \epsilon, \delta) \in \Theta\left(\frac{n}{m\epsilon^2} \log \frac{1}{\delta}\right)$. *In that case,* Q-P *is solvable using* $\Theta\left(\frac{1}{\rho\epsilon^2} \log \frac{1}{\delta}\right)$ *samples.*

It is interesting to find a $\gamma(\cdot)$ such that the upper bound presented in Theorem 4.7 matches the lower bound up to a constant factor. We notice, Theorem 4.7 guarantees that there exists a constant C, such that for any given ϵ, δ, and $m \leq n/2$, $\gamma(n, m, \epsilon, \delta) \leq C \cdot \gamma\left(\lceil 8 \log(2/\delta) \rceil, \lfloor 4 \log(2/\delta) \rfloor, \frac{\epsilon}{2}, \frac{\delta}{2}\right)$. However, for $n < \lceil 8 \log(2/\delta) \rceil$ we believe Q-F can be solved more efficiently than posing it as Q-P. We present it as a conjecture.

Definition For $g : \mathbb{Z}^+ \times \mathbb{Z}^+ \times [0,1] \times [0,1] \mapsto \mathbb{R}^+$ we say Q-F is solvable in $\Theta(g(\cdot))$, if there exists an algorithm that solves every instance of Q-F given by $(\mathcal{A}, n, m, \epsilon, \delta)$ in $O(g(n, m, \epsilon, \delta))$ samples, and there exists an instance of Q-F given by $(\bar{\mathcal{A}}, \bar{n}, \bar{m}, \bar{\epsilon}, \bar{\delta})$ such that every algorithm needs at least $\Omega(g(\bar{n}, \bar{m}, \bar{\epsilon}, \bar{\delta}))$ samples to solve it.

Conjecture 4.1. *There exists a constant* $C > 0$, *and functions* $g : \mathbb{Z}^+ \times \mathbb{Z}^+ \times [0,1] \times [0,1] \mapsto \mathbb{R}^+$, *and* $h : \mathbb{Z}^+ \times \mathbb{Z}^+ \times [0,1] \times [0,1] \mapsto \mathbb{R}^+$, *such that for every* $\delta \in (0,1]$, *there exists an integer* $n_0 < C \log \frac{2}{\delta}$, *such that for every* $n \leq n_0$, Q-F *is solvable in* $\Theta(g(n, m, \epsilon, \delta))$ *samples, and its equivalent* Q-P *(obtained by posing the the instance of* Q-F *as an instance of* Q-P, *as done in proving Corollary 4.2) needs at least* $\Omega(h(n, m, \epsilon, \delta))$ *samples, then* $\lim_{\delta \downarrow 0} \frac{g(n,m,\epsilon,\delta)}{h(n,m,\epsilon,\delta)} \to 0$.

Next, we empirically compare LUCB-k-m for $k = 1$ with $\mathcal{F}_2$ on different instances, and also we study empirical performance of LUCB-k-m by varying k.

5. Experiments and Results

We begin our experimental evaluation by comparing $\mathcal{F}_2$ (**?**) and LUCB-k-m based on the number of samples drawn on different instances of Q-F or $(1, m, n)$. $\mathcal{F}_2$ is a fully-sequential algorithm that resembles LUCB-k-m, but subtle differences in the way the algorithms partition $\mathcal{A}$ and select arms to pull lead to different results. At each time step t, $\mathcal{F}_2$ creates three partitions of $\mathcal{A}$—$\bar{A}_1(t)$, $\bar{A}_2(t)$, and $\bar{A}_3(t)$. It puts the arm with the highest LCB in $\bar{A}_1(t)$; among the rest, it puts $m - 1$ arms with the highest UCBs in $\bar{A}_2(t)$; and the rest $n - m$ arms in $\bar{A}_3(t)$; ties are broken at random. At each time step t, it samples three arms—the arm in $\bar{A}_1(t)$, the least sampled arm in $\bar{A}_2(t)$, and the arm with the highest UCB in $\bar{A}_3(t)$.

We take five Bernoulli instance of sizes $n = 10, 20, 50, 100,$ and 200, with the means linearly spaced between 0.999 and 0.001 (both inclusive), and sorted in descending order. We name the bandit instance of size n as $\mathcal{I}_n$. Now, setting $\epsilon = 0.05, \delta = 0.001$, and $m = 0.1 \times n$, we run the experiments and compare the number of samples drawn by $\mathcal{F}_2$ and LUCB-k-m to solve these five instances for $k = 1$. In our implementation we have used KL-divergence based confidence bounds (**??**) for both $\mathcal{F}_2$ and LUCB-k-m. As depicted by Figure 1, as the number of arms (n) increases, the sample complexity of both the algorithms increases due to increase in hardness H_ϵ. However, the sample complexity of $\mathcal{F}_2$ increases much faster than LUCB-k-m.

As shown by **?** the efficiency of LUCB1 comes from the quick identification of the most optimal arm due to a large separation from the $m + 1$-th arm. Intuitively, the possible reason for $\mathcal{F}_2$ to incur more samples is the delay in prioritising the optimal arm to pull more frequently. This should result in a smaller fraction of total samples taken from the best arm. Figure 2 affirms this intuition. It represents a comparison between $\mathcal{F}_2$ and LUCB-k-m on the number of samples obtained by three "ground-truth" groups—B_1, B_2, and B_3 on $\mathcal{I}_{10}$, keeping $k = 1$ and varying m from 1 to 5. We note that the lesser the difference between k and m, the higher the hardness (H_ϵ), and both $\mathcal{F}_2$ and LUCB-k-m find it hard to identify a correct arm. Hence, for $k = m = 1$, both of them spend almost the same fraction of pulls to the best arm. However, as m becomes larger, keeping $k = 1$, the hardness of the problem reduces, but $\mathcal{F}_2$ still struggles to identify the best arm and results in spending a significantly a lesser fraction of the total pulls to it, compared to LUCB-k-m.

In this paper, we have developed algorithms specifically for the (k, m, n) problem; previously one might have solved (k, m, n) either by solving (k, k, n) or (m, m, n): that is choosing the *best* k- or m-sized subset. In Figure 3 we present a comparison of the sample complexities for solving (k, m, n) and the best subset-selection problems. Fixing $\mathcal{A} = \mathcal{I}_{20}$, $n = 20$, $m = 10$, (k, m, n) instances are given by and varying $k \in \{1, 3, 5, 8, 10\}$, whereas, for the best subset-selection problem we set $m = k$. As expected, the number of samples incurred is significantly lesser for solving the problem instances with $k < m$, thereby validating the use of LUCB-k-m.

6. Conclusion

Identifying one arm out of the best m, in an n-armed stochastic bandit is an interesting problem identified by **?**. They have mentioned the scenarios where identifying the best subset is practically infeasible. However, there are numerous examples in practice that demand efficient identification of multiple good solutions instead of only one; for example, assigning a distributed crowd-sourcing task, identification of

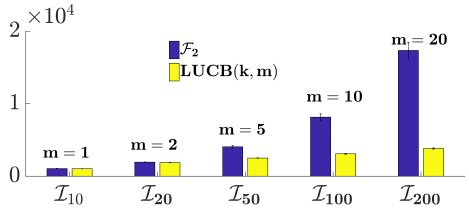

Figure 1. Comparison of incurred sample complexities by $\mathcal{F}_2$ and LUCB-k-m to solve Q-F with $m = 0.1 \times n$, on the five instances detailed in Section 5. y-axis represents the number of samples averaged over 100 runs, with standard error bars.

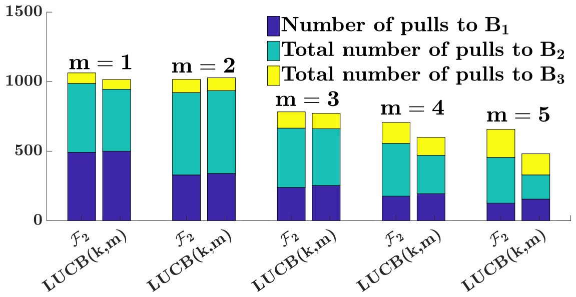

Figure 2. Comparison between $\mathcal{F}_2$ and LUCB-k-m on the number of pulls received by the camps B_1, B_2 and B_3, for solving different instances of Q-F on $\mathcal{I}_{10}$, by varying m from 1 to 5. Recall that B_1 is the singleton set, with the best arm being the only member. y-axis represents the number of samples averaged over 100 runs.

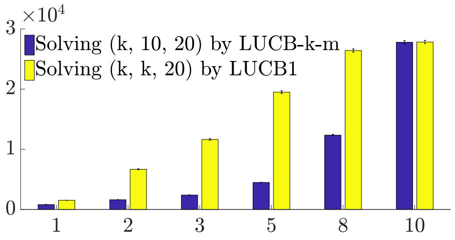

Figure 3. Comparison of number of samples incurred for solving different instances of (k, m, n) defined on $\mathcal{I}_{20}$, by setting $m = 10$, and varying $k \in \{1, 2, 3, 5, 8, 10\}$. x-axis represents k, and y-axis represents the number of samples averaged over 100 runs, with standard error bars.

good molecular combinations in drug designing, etc. In this paper, we present (k, m, n)—a generalised problem of identifying k out of the best m arms. Setting $k = 1$, (k, m, n) gets reduced to selection of one out of the best m arms, while setting $k = m$, makes it identical with the "subset-selection" (**?**). We have presented a lower bound on the sample complexity to solve (k, m, n). We have also presented a fully sequential adaptive PAC algorithm, LUCB-k-m, that solves (k, m, n), with expected sample complexity matching up to a constant factor that of $\mathcal{F}_2$ (**?**) and LUCB1 (**?**) for $k = 1$ and $k = m$, respectively. We have empirically compared LUCB-k-m to $\mathcal{F}_2$ on different problem instances, and shown that LUCB-k-m outperforms $\mathcal{F}_2$ by a large margin in terms of the number of samples as n grows.

For the problem of identification of a single $[\epsilon, \rho]$-optimal (**?**) arm in infinite bandit instances, the existing upper bound on the sample complexity differs from the lower bound by a multiplicative $\log \frac{1}{\delta}$ factor. It was not clear whether the lower bound was loose, or the upper can be improved, and left as an interesting problem to solve (**?**). In this paper we reduce the gap by furnishing an upper bound which is optimal up to an *additive* poly-log term. Further, we show that the problem of identification k distinct $[\epsilon, \rho]$-optimal arms is not well-posed in general, but when it is, we derive a lower bound on the sample complexity. Also, we identify a class of well-posed instances for which we present an efficient algorithm. In the end we show that how improving the upper bound on the sample-complexity for solving Q-F instances can be translated in improving upper bound on the sample-complexity for solving Q-P. However, we conjecture that there exists a set of Q-F instances and a corresponding set of Q-P instances, such that every instance of Q-F requires lesser number of samples to solve than the corresponding Q-P instance in the other set. Showing correctness of the conjecture and improving the lower and the upper bound on the sample complexities are some interesting directions we leave for future work.

7. Lower Bound on the Worst Case Sample Complexity to Solve (k, m, n)

Theorem 3.1. *[Lower Bound for (k, m, n)] Let $\mathcal{L}$ be an algorithm that solves (k, m, n). Then, there exists an instance $(\mathcal{A}, n, m, k, \epsilon, \delta)$, with $0 < \epsilon \leq \frac{1}{\sqrt{32}}$, $0 < \delta \leq \frac{e^{-1}}{4}$, and $n \geq 2m$, $1 \leq k \leq m$, on which the expected number of pulls performed by $\mathcal{L}$ is at least $\frac{1}{18375} \cdot \frac{1}{\epsilon^2} \cdot \frac{n}{m-k+1} \ln \frac{\binom{m}{k-1}}{4\delta}$.*

The proof technique for Theorem 3.1 follows a path similar to that of (**?**, Theorem 8), but differs in the fact that any k of the m (ϵ, m)-optimal arms needs to be returned as opposed to all the m.

7.1. Bandit Instances:

Assume we are given a set of n arms $\mathcal{A} = \{0, 1, 2, \cdots, n-1\}$. Let $I_0 \overset{\text{def}}{=} \{0, 1, 2, \cdots, m-k\}$ and $\mathcal{I}_l \overset{\text{def}}{=} \{I : I \subseteq \{\mathcal{A} \setminus I_0\} \wedge |I| = l\}$. Also for $I \subseteq \{m-k+1, m-k+2, \cdots, n-1\}$, we define

$$\bar{I} \overset{\text{def}}{=} \{m-k+1, m-k+2, \cdots, n-1\} \setminus I.$$

With each $I \in \mathcal{I}_{k-1} \cup \mathcal{I}_m$ we associate an n-armed bandit instance $\mathcal{B}^I$, in which each arm a produces a reward from a Bernoulli distribution with mean μ_a defined as:

$$\mu_a = \begin{cases} \frac{1}{2} & \text{if } a \in I_0 \\ \frac{1}{2} + 2\epsilon & \text{if } a \in I \\ \frac{1}{2} - 2\epsilon & \text{if } a \in \bar{I}. \end{cases} \tag{2}$$

Notice that all the instances in $\mathcal{I}_{k-1} \cup \mathcal{I}_m$ have exactly m (ϵ, m)-optimal arms. For $I \in \mathcal{I}_{k-1}$, all the arms in I_0 are (ϵ, m)-optimal, but for $I \in \mathcal{I}_m$ they are not. With slight overloading of notation we write $\mu(S)$ to denote the multi-set consisting of means of the arms in $S \subseteq \mathcal{A}$.

The key idea of the proof is that without sufficient sampling of each arm, it is not possible to correctly identify k of the (ϵ, m)-optimal arms with high probability.

7.2. Bounding the Error Probability:

We shall prove the theorem by first making the following assumption, which we shall demonstrate leads to a contradiction.

Assumption 1. *Assume, that there exists an algorithm $\mathcal{L}$, that solves each problem instance in (k, m, n) defined on bandit instance $\mathcal{B}^I$, $I \in \mathcal{I}_{k-1}$, and incurs a sample complexity SC_I. Then for all $I \in \mathcal{I}_{k-1}$, $\mathbb{E}[\mathrm{SC}_I] < \frac{1}{18375} \cdot \frac{1}{\epsilon^2} \cdot \frac{n}{m-k+1} \ln \left(\frac{\binom{m}{m-k+1}}{4\delta} \right)$, for $0 < \epsilon \leq \frac{1}{\sqrt{32}}$, $0 < \delta \leq \frac{e^{-1}}{4}$, and $n \geq 2m$, where $C = \frac{1}{18375}$.*

For convenience, we denote by $\Pr_I$ the probability distribution induced by the bandit instance $\mathcal{B}^I$ and the possible randomisation introduced by the algorithm $\mathcal{L}$. Also, let $S_\mathcal{L}$ be the set of arms returned (as output) by $\mathcal{L}$, and T_S be the total number of times the arms in $S \subseteq \mathcal{A}$ get sampled until $\mathcal{L}$ stops.

Then, as $\mathcal{L}$ solves (k, m, n), for all $I \in \mathcal{I}_{k-1}$

$$\Pr_I\{S_\mathcal{L} \subseteq I_0 \cup I\} \geq 1 - \delta. \tag{3}$$

Therefore, for all $I \in \mathcal{I}_{k-1}$

$$\mathbb{E}_I[T_\mathcal{A}] \leq C \frac{n}{m-k+1} \ln \left(\frac{\binom{m}{m-k+1}}{4\delta} \right). \tag{4}$$

7.2.1. Changing $\Pr_I$ to $\Pr_{I \cup Q}$ where $Q \in \bar{I}$ s.t. $|Q| = m-k+1$:

Consider an arbitrary but fixed $I \in \mathcal{I}_{k-1}$. Consider a fixed partitioning of $\mathcal{A}$, into $\left\lfloor \frac{n}{m-k+1} \right\rfloor$ subsets of size $(m-k+1)$ each. If Assumption (1) is correct, then for the instance $\mathcal{B}^I$, there are at most $\left\lfloor \frac{n}{4(m-k+1)}) \right\rfloor - 1$ partitions $B \subset \bar{I}$, such that $\mathbb{E}_I[T_B] \geq \frac{4C}{\epsilon^2} \ln \left(\frac{1}{4\delta} \right)$. Now, as $\left\lfloor \frac{n-m}{m-k+1} \right\rfloor - \left(\left\lfloor \frac{n}{4(m-k+1)} \right\rfloor - 1 \right) \geq \left\lfloor \frac{n}{4(m-k+1)} \right\rfloor + 1 > 0$; therefore, there exists at least one subset $Q \in \bar{I}$ such that $|Q| = m-k+1$, and $\mathbb{E}_I[T_Q] < \frac{4C}{\epsilon^2} \ln \left(\frac{\binom{m}{m-k+1}}{4\delta} \right)$. Define $T^* = \frac{16C}{\epsilon^2} \ln \left(\frac{\binom{m}{m-k+1}}{4\delta} \right)$. Then using Markov's inequality we get:

$$\Pr_I\{T_Q \geq T^*\} < \frac{1}{4}. \tag{5}$$

Let $\Delta = 2\epsilon T^* + \sqrt{T^*}$ and also let K_Q be the total rewards obtained from Q.

Lemma 7.1. *If $I \in \mathcal{I}_{k-1}$ and $Q \in \bar{I}$ s.t. $|Q| = m-k+1$, then*

$$\Pr_I \left\{ T_Q \leq T^* \wedge K_Q \leq \frac{T_Q}{2} - \Delta \right\} \leq \frac{1}{4}.$$

Proof. Let $K_Q(t)$ be the total sum obtained from Q at the end of the trial t. As for $\mathcal{B}^{I_0}$, $\forall j \in Q$ $\mu_j = 1/2 - 2\epsilon$, hence selecting and pulling one arm at each trial from Q following any rule (deterministic or probabilistic) is equivalent to selection of a single arm from Q for once and subsequently perform pulls on it. Hence whatever be the strategy of pulling one arm at each trial from Q, the expected reward for each pull will be $1/2 - 2\epsilon$. Let r_i be the i.i.d. reward obtained from the i^{th} trial. Then $K_Q(t) = \sum_{i=1}^{t} r_i$ and $Var[r_i] = \left(\frac{1}{2} - 2\epsilon\right)\left(\frac{1}{2} + 2\epsilon\right) = \left(\frac{1}{4} - 4\epsilon^2\right) < \frac{1}{4}$. As $\forall i : 1 \leq i \leq t$, r_i are i.i.d., we get $Var[K_Q(t)] = \sum_{i=1}^{t} Var(r_i) < \frac{t}{4}$.

Now we can write the following:

$$\begin{aligned}
& \Pr_I\left\{\min_{1\le t\le T^*}\left(K_Q(t)-t\left(\frac{1}{2}-2\epsilon\right)\right)\le -\sqrt{T^*}\right\} \\
\le\ & \Pr_I\left\{\max_{1\le t\le T^*}\left|K_Q(t)-t\left(\frac{1}{2}-2\epsilon\right)\right|\ge \sqrt{T^*}\right\} \\
\le\ & \frac{Var[K_Q(T^*)]}{T^*}<\frac{1}{4}, \qquad (6)
\end{aligned}$$

wherein we have used Kolmogorov's inequality. □

Lemma 7.2. *Let $I \in \mathcal{I}_{k-1}$ and $Q \in \mathcal{I}_{m-k+1}$ such that $Q \subseteq \bar{I}$, and let W be some fixed sequence of rewards obtained by a single run of algorithm $\mathcal{L}$ on $\mathcal{B}^I$ such that $T_Q \le T^*$ and $K_Q \ge \frac{T_Q}{2} - \Delta$, then:*

$$\Pr_{I\cup Q}\{W\} > \Pr_I\{W\}\cdot\exp(-32\epsilon\Delta). \qquad (7)$$

Proof. Recall the fact that all the arms in Q have the same mean. Hence, if chosen one at each trial (following any strategy), the expected reward at each trial remains the same. Hence the probability of getting a given reward sequence generated from Q is independent of the sampling strategy. Again as the arms in Q have higher mean in $\mathcal{B}^Q$, the probability of getting the sequence (of rewards) decreases monotonically as the 1-rewards for $\mathcal{B}^{I_0}$ become fewer. So we get

$$\begin{aligned}
&\Pr_{I\cup Q}\{W\} = \Pr_I\{W\}\frac{\left(\frac{1}{2}+2\epsilon\right)^{K_Q}\left(\frac{1}{2}-2\epsilon\right)^{T_Q-K_Q}}{\left(\frac{1}{2}-2\epsilon\right)^{K_Q}\left(\frac{1}{2}+2\epsilon\right)^{T_Q-K_Q}} \\
&\ge \Pr_I\{W\}\frac{\left(\frac{1}{2}+2\epsilon\right)^{\left(\frac{T_Q}{2}-\Delta\right)}\left(\frac{1}{2}-2\epsilon\right)^{\left(\frac{T_Q}{2}+\Delta\right)}}{\left(\frac{1}{2}-2\epsilon\right)^{\left(\frac{T_Q}{2}-\Delta\right)}\left(\frac{1}{2}+2\epsilon\right)^{\left(\frac{T_Q}{2}+\Delta\right)}} \\
&= \Pr_I\{W\}\cdot\left(\frac{\frac{1}{2}-2\epsilon}{\frac{1}{2}+2\epsilon}\right)^{2\Delta} \\
&> \Pr_I\{W\}\cdot\exp(-32\epsilon\Delta)\left[\text{for } 0<\epsilon\le\frac{1}{\sqrt{32}}\right].
\end{aligned}$$

□

Lemma 7.3. *If (5) holds for an $I \in \mathcal{I}_{k-1}$ and $Q \in \mathcal{I}_{m-k+1}$ such that $Q \subseteq \bar{I}$, and if $\mathcal{W}$ is the set of all possible reward sequences W, obtained by algorithm $\mathcal{L}$ on $\mathcal{B}^I$, then $\Pr_{I\cup Q}\{\mathcal{W}\} > \left(\Pr_I\{\mathcal{W}\}-\frac{1}{2}\right)\cdot 4\delta$. In particular,*

$$\Pr_{I\cup Q}\{S_{\mathcal{L}} \subseteq I_0 \cup I\} > \frac{\delta}{\binom{m}{m-k+1}}. \qquad (8)$$

Proof. Let for some fixed sequence (of rewards) W, T_Q^W and K_Q^W respectively denote the total number of samples received by the arms in Q and the total number of 1-rewards obtained before the algorithm $\mathcal{L}$ stopped. Then:

$$\begin{aligned}
&\Pr_{I\cup Q}\{W\} = \Pr_{I\cup Q}(W : W\in\mathcal{W}) \\
&\ge \Pr_{I\cup Q}\left\{W : W\in\mathcal{W}\bigwedge T_Q^W \le T^*\bigwedge K_Q^W\ge\frac{T_Q^W}{2}-\Delta\right\} \\
&> \Pr_I\left\{W : W\in\mathcal{W}\bigwedge T_Q^W \le T^*\bigwedge K_Q^W\ge\frac{T_Q^W}{2}-\Delta\right\}\cdot\exp(-32\epsilon\Delta) \\
&\ge \left(\Pr_I\left\{W : W\in\mathcal{W}\bigwedge T_Q^W\le T^*\right\}-\frac{1}{4}\right)\cdot\exp(-32\epsilon\Delta) \\
&\ge \left(\Pr_I\{\mathcal{W}\}-\frac{1}{2}\right)\cdot\frac{4\delta}{\binom{m}{m-k+1}}\ \text{ for } C=\frac{1}{18375},\ \delta<\frac{e^{-1}}{4}.
\end{aligned}$$

In the above, the 3rd, 4th and the last step are obtained using Lemma 7.2, Lemma 7.1 and Equation (5) respectively. The inequality (8) is obtained by using inequality (3), as $\Pr_I\{S_{\mathcal{L}} \in I_0\} > 1-\delta \ge 1-\frac{e^{-1}}{4} > \frac{3}{4}$. □

7.2.2. SUMMING OVER $\mathcal{I}_{k-1}$ AND $\mathcal{I}_m$

Now, we sum up the probability of errors across all the instances in $\mathcal{I}_{k-1}$ and $\mathcal{I}_m$. If the Assumption 1 is true, using the pigeon-hole principle we show that there exists some instance for which the mistake probability is greater than δ.

$$
\begin{aligned}
&\sum_{J \in \mathcal{I}_m} \Pr_J\{S_{\mathcal{L}} \not\subseteq J\} \\
&\geq \sum_{J \in \mathcal{I}_m} \sum_{\substack{J' \subset J \\ :|J'|=m-k+1}} \Pr_J\{S_{\mathcal{L}} \subseteq \{J \setminus J'\} \cup I_0\} \\
&\geq \sum_{J \in \mathcal{I}_m} \sum_{\substack{J' \subset J \\ :|J'|=m-k+1}} \Pr_J\{\exists a \in I_0 : S_{\mathcal{L}} = \{J \setminus J'\} \cup \{a\}\} \\
&= \sum_{J \in \mathcal{I}_m} \sum_{\substack{J' \subset J \\ :|J'|=m-k+1}} \sum_{I \in \mathcal{I}_{k-1}} \mathbb{1}[I \cup J' = J] \cdot \Pr_J\{S_{\mathcal{L}} \subseteq I \cup I_0\} \\
&= \sum_{J \in \mathcal{I}_m} \sum_{\substack{J' \subset \mathcal{A} \setminus I_0 \\ :|J'|=m-k+1}} \sum_{I \in \mathcal{I}_{k-1}} \mathbb{1}[I \cup J' = J] \cdot \Pr_J\{S_{\mathcal{L}} \subseteq I \cup I_0\} \\
&= \sum_{J \in \mathcal{I}_m} \sum_{I \in \mathcal{I}_{k-1}} \sum_{\substack{J' \subset \mathcal{A} \setminus I_0 \\ :|J'|=m-k+1}} \mathbb{1}[I \cup J' = J] \cdot \Pr_J\{S_{\mathcal{L}} \subseteq I \cup I_0\} \\
&= \sum_{I \in \mathcal{I}_{k-1}} \sum_{J \in \mathcal{I}_m} \sum_{\substack{J' \subset \bar{I} \\ :|J'|=m-k+1}} \mathbb{1}[I \cup J' = J] \cdot \Pr_J\{S_{\mathcal{L}} \subseteq I \cup I_0\} \\
&= \sum_{I \in \mathcal{I}_{k-1}} \sum_{\substack{J' \subset \bar{I} \\ :|J'|=m-k+1}} \sum_{J \in \mathcal{I}_m} \mathbb{1}[I \cup J' = J] \cdot \Pr_J\{S_{\mathcal{L}} \subseteq I \cup I_0\} \\
&= \sum_{I \in \mathcal{I}_{k-1}} \sum_{\substack{J' \subset \bar{I} \\ :|J'|=m-k+1}} \Pr_{I \cup J'}\{S_{\mathcal{L}} \subseteq I \cup I_0\}
\end{aligned}
$$

Recall that $\forall I \in \mathcal{I}_{k-1}$ there exists a set $Q \subset \mathcal{A} \setminus \{I \cup I_0\}$: $|Q| = (m - k + 1)$, such that $T_Q < T^*$. Therefore,

$$
\begin{aligned}
&\sum_{J \in \mathcal{I}_m} \Pr_J\{S_{\mathcal{L}} \not\subseteq J\} \\
&\geq \sum_{I \in \mathcal{I}_{k-1}} \sum_{\substack{J' \subset \bar{I} \\ :|J'|=m-k+1}} \Pr_{I \cup J'}\{S_{\mathcal{L}} \subseteq I \cup I_0\} \\
&> \sum_{I \in \mathcal{I}_{k-1}} \sum_{\substack{J' \subset \bar{I} \\ :|J'|=m-k+1}} \frac{\delta}{\binom{m}{m-k+1}} \\
&\geq \sum_{I \in \mathcal{I}_{k-1}} \binom{n-m}{m-k+1} \cdot \frac{\delta}{\binom{m}{m-k+1}} \\
&\geq \binom{n-(m-k+1)}{k-1} \cdot \binom{n-m}{m-k+1} \cdot \frac{\delta}{\binom{m}{m-k+1}} \\
&= \binom{n-(m+k-1)}{m} \delta \\
&= |\mathcal{I}_m| \delta.
\end{aligned}
$$

Hence, we get a contradiction to Assumption 1, thereby proving the theorem.

8. Analysis of LUCB-k-m

Let at time t, $\hat{p}_a^t$ be the empirical mean of the arm $a \in \mathcal{A}$, and u_a^t be the number of times the arm a has been pulled until (and excluding) time t. For a given $\delta \in (0,1]$, we define $\beta(u_a^t, t, \delta) = \sqrt{\frac{1}{2u_a^t} \ln \frac{k_1 n t^4}{\delta}}$, where $k_1 = 5/4$. We define upper and lower confidence bound on the estimate of the true mean of arm $a \in \mathcal{A}$ as $ucb(a,t) = \hat{p}_a + \beta(u_a^t, t, \delta)$, and $lcb(a,t) = \hat{p}_a - \beta(u_a^t, t, \delta)$ respectively.

To analyse the sample complexity, first we define some events, at least one of which must occur if the algorithm does not stop at the round t.

Definition (PROBABLE EVENTS) Let $a, b \in \mathcal{A}$, such that $\mu_a > \mu_b$. During the run of the algorithm, any of the following five events may occur:
i) The empirical mean of an arm may falls outside the upper or the lower confidence bound. We define it as:

$$CROSS_a^t \overset{\text{def}}{=} \{ucb(a,t) < \mu_a \vee lcb(a,t) > \mu_a\}.$$

ii) The empirical mean of arm a may be lesser than that of arm b; we definite as:

$$ErrA(a,b,t) \overset{\text{def}}{=} \{\hat{p}_a^t < \hat{p}_b^t\}.$$

iii) The lower and upper confidence bounds of arm a may fall below those of arm b; we define them as:

$$ErrL(a,b,t) \overset{\text{def}}{=} \{lcb(a,t) < lcb(b,t)\},$$
$$ErrU(a,b,t) \overset{\text{def}}{=} \{ucb(a,t) < ucb(b,t)\}.$$

iv) If an arm's confidence bounds are above a certain radius (say d), we define that event as

$$NEEDY_a^t(d) \overset{\text{def}}{=} \{\{lcb(a,t) < \mu_a - d\} \vee \{ucb(a,t) > \mu_a + d\}\}.$$

We show that any arm a, if sampled sufficiently, that is $u_a^t \geq u^*(a,t)$, then occurrence of any of the PROBABLE EVENTS imply occurrence of $CROSS_a^t$. First we show that if $CROSS_a^t$ does not occur for any $a \in \mathcal{A}$, then occurrence of any one of the PROBABLE EVENTS implies the occurrence of $NEEDY_a^t(\cdot)$ or $NEEDY_b^t(\cdot)$.

Lemma 8.1. *[Expressing* PROBABLE EVENTS *in terms of $NEEDY_a^t$ and $CROSS_a^t$] To prove that* $\{\neg CROSS_a^t \wedge \neg CROSS_b^t\} \wedge \{ErrA(a,b,t) \vee ErrU(a,b,t) \vee ErrL(a,b,t)\} \implies \{NEEDY_a^t\left(\frac{\Delta_{ab}}{2}\right) \vee NEEDY_b^t\left(\frac{\Delta_{ab}}{2}\right)\}$.

Proof. $\mathbf{ErrA(a,b,t)}$: To prove that $\neg\{CROSS_a^t \vee CROSS_b^t\} \wedge ErrA(a,b,t) \implies NEEDY_a^t\left(\frac{\Delta_{ab}}{2}\right) \vee NEEDY_b^t\left(\frac{\Delta_{ab}}{2}\right)$.

$$\begin{aligned}
&ErrA(a,b,t) \implies \hat{p}_a^t < \hat{p}_b^t \\
&\implies \hat{p}_a^t - (p_a - \beta(u_a^t,t,\delta)) < \hat{p}_b^t - (p_b + \beta(u_b^t,t,\delta) + \\
&(\beta(u_a^t,t,\delta) + \beta(u_b^t,t,\delta)) - \Delta_{ab}/2) \\
&\implies NEEDY_a^t\left(\frac{\Delta_{ab}}{2}\right) \vee NEEDY_b^t\left(\frac{\Delta_{ab}}{2}\right).
\end{aligned}$$

$\mathbf{ErrU(a,b,t)}$: To prove that $\neg\{CROSS_a^t \vee CROSS_b^t\} \wedge ErrU(a,b,t) \implies NEEDY_b^t\left(\frac{\Delta_{ab}}{2}\right)$.
Assuming $\neg CROSS_a^t \wedge \neg CROSS_b^t$ we get

$$\begin{aligned}
&ErrU(a,b,t) \implies \{ucb(b,t) > ucb(a,t)\} \\
&\implies \{\hat{p}_b^t + \beta(u_b^t,t,\delta) > \hat{p}_a^t + \beta(u_a^t,t,\delta)\} \\
&\implies \{\hat{p}_b^t > \mu_b + \beta(u_b^t,t,\delta)\} \vee \{\hat{p}_a^t < \mu_a - \beta(u_a^t,t,\delta)\} \vee \\
&\quad \{2\beta(u_b^t,t,\delta) > \Delta_{ab}\} \\
&\implies NEEDY_b^t\left(\frac{\Delta_{ab}}{2}\right).
\end{aligned}$$

$\mathbf{ErrL(a,b,t)}$: To prove that $\neg\{CROSS_a^t \vee CROSS_b^t\} \wedge ErrL(a,b,t) \implies NEEDY_a^t\left(\frac{\Delta_{ab}}{2}\right)$.
Assuming $\neg CROSS_a^t \wedge \neg CROSS_b^t$ we get

$$\begin{aligned}
&ErrL(a,b,t) \implies \{lcb(b,t) > lcb(a,t)\}\\
&\implies \{\hat{p}_b^t - \beta(u_b^t,t,\delta) > \hat{p}_a^t - \beta(u_a^t,t,\delta)\}\\
&\implies \{\hat{p}_b^t > \mu_b + \beta(u_b^t,t,\delta)\} \vee \{\hat{p}_a^t < \mu_a - \beta(u_a^t,t,\delta)\} \vee\\
&\qquad \{2\beta(u_a^t,t,\delta) > \Delta_{ab}\}\\
&\implies NEEDY_a^t\left(\frac{\Delta_{ab}}{2}\right).
\end{aligned}$$

□

We show that given a threshold d, if an arm a is sufficiently sampled, such that $\beta(u_a^t,t,\delta) \leq \frac{d}{2}$, then $NEEDY_a^t$ infers $CROSS_a^t$.

Lemma 8.2. *For any $a \in \mathcal{A}$, $\{NEEDY_a^t(d) | \beta(u_a^t,t,\delta) < d/2\} \implies CROSS_a^t$.*

Proof. First, we show that $\{lcb(a,t) < \mu_a - d | \beta(u_a^t,t,\delta) < d/2\} \implies CROSS_a^t$,

$$\begin{aligned}
&\{lcb(a,t) < \mu_a - d | \beta(u_a^t,t,\delta) < d/2\}\\
&\implies \{\hat{p}_a^t - \beta(u_a^t,t,\delta) < \mu_a - d | \beta(u_a^t,t,\delta) < d/2\}\\
&\implies \{\hat{p}_a^t < \mu_a - d + \beta(u_a^t,t,\delta) | \beta(u_a^t,t,\delta) < d/2\}\\
&\implies \{\hat{p}_a^t < \mu_a - d/2 | \beta(u_a^t,t,\delta) < d/2\}\\
&\implies CROSS_a^t. \qquad (9)
\end{aligned}$$

The other part follows the similar way. □

By the very definition of confidence bound, at any round t, the probability that the empirical mean of an arm will lie outside it, is very low. In other words, the probability of occurrence $CROSS_a^t$ is very low for all t and $a \in \mathcal{A}$.

Lemma 8.3. *[Upper bounding the probability of $CROSS_a^t$] $\forall a \in \mathcal{A}$ and $\forall t \geq 0$, $\Pr\{CROSS_a^t\} \leq \frac{\delta}{knt^4}$. Hence, $P\left[\exists t \geq 0 \wedge \exists a \in \mathcal{A} : CROSS_a^t | u_a^t \geq 0\right] \leq \frac{\delta}{k_1 t^3}$.*

Proof. $\Pr\{CROSS_a^t\}$ is upper bounded by using Hoeffding's inequality, and the next statement gets proved by taking union bound over all arms and t. □

Now, recalling the definition of h_*^t, and l_*^t from Algorithm 1, we present the key logic underlying the analysis of LUCB-k-m. The idea is to show that if the algorithm has not stopped, then one of those PROBABLE EVENTS must have occurred. Then using Lemma 8.1, and Lemma 8.2, Lemma 8.3 we show that beyond a certain number of rounds, the probability that LUCB-k-m will continue is sufficiently small. Lastly, using the argument based on pigeon-hole principle, similar to Lemma 5 of **?**, we establish the upper bound on the sample complexity. Below we present the core logic that shows, until the algorithm stops one of the PROBABLE EVENTS must occur.

Case 1 $h_*^t \in B_1 \wedge l_*^t \in B_1$

```
[H]
  if $\exists b_3 \in A_1^t \cap B_3$ then
    Then $ErrL(h_*^t, b_3, t)$ has occurred.
  else
    $\exists b_3 \in A_2^t \cap B_3$
    Then $ErrA(h_*^t, b_3, t)$ has occurred.
  end if
```

Case 2 $h^t_* \in B_1 \wedge l^t_* \in B_2$

```
[H]
  if $\exists b_3 \in A^t_1 \cap B_3$ then
    Then $ErrL(h^t_*, b^t_3, t)$ has occurred.
  else
    $\exists b_3 \in A^t_2 \cap B_3$.
    if $\Delta_{h^t_* l^t_*} \geq \frac{\Delta_{h^t_*}}{2}$ then
      Then $NEEDY^t_{h^t_*}(\Delta_{h^t_*}/4) \vee NEEDY^t_{l^t_*}(\Delta_{h^t_*}/4)$ has occurred.
    else
      Then $ErrL(l^t_*, b^t_3, t)$ has occurred.
    end if
  end if
```

Case 3 $h^t_* \in B_1 \wedge l^t_* \in B_3$

```
[H]
  Then $NEEDY^t_{h^t_*}(\Delta_{h^t_*}/4) \vee NEEDY^t_{l^t_*}(\Delta_{l^t_*}/4)$ has occurred.
```

Lemma 8.4 (H)**.** *If $T = CH_\epsilon \ln\left(\frac{H_\epsilon}{\delta}\right)$, then for $C \geq 2732$, the following holds:*

$$T > 2 + 2\sum_{a \in \mathcal{A}} u^*(a, T).$$

Proof. This proof is taken from Appendix B.3 of **?**.

$$\begin{aligned}
&2 + 2\sum_{a \in \mathcal{A}} u^*(a, T) = 2 + 64\sum_{a \in \mathcal{A}} \left\lceil \frac{1}{\max(\Delta_a, (\epsilon/2))^2} \ln \frac{knt^4}{\delta} \right\rceil \\
&\leq 2 + 64n + 64H_\epsilon \ln \frac{knT^4}{\delta} \\
&= 2 + 64n + 64H_\epsilon \ln k + 64H_\epsilon \ln \frac{n}{\delta} + 256H_\epsilon \ln T \\
&< (66 + 64\ln k)H_\epsilon + 64H_\epsilon \ln \frac{n}{\delta} + 256H_\epsilon \left[\ln C + \ln H_\epsilon + \ln\ln \frac{H_\epsilon}{\delta}\right] \\
&< (66 + 64\ln k)H_\epsilon + 64H_\epsilon \ln \frac{n}{\delta} + 256H_\epsilon \left[\ln C + \ln H_\epsilon + \ln\ln \frac{H_\epsilon}{\delta}\right] \\
&< 130H_\epsilon + 64H_\epsilon \ln \frac{n}{\delta} + 256H_\epsilon \left[\ln C + \ln H_\epsilon + \ln \frac{H_\epsilon}{\delta}\right]
\end{aligned}$$

Case 4 $h^t_* \in B_2 \wedge l^t_* \in B_1$

```
[H]
  if $\Delta_{h^t_* l^t_*} \geq \frac{\Delta_{h^t_*}}{2}$ then
    Then $ErrA(l^t_*, h^t_*, t)$ has occurred.
  else
    if $\exists b_3 \in A^t_1 \cap B_3$ then
      Then $ErrL(h^t_*, b^t_3, t)$ has occurred.
    else
      $\exists b_3 \in A^t_2 \cap B_3$
      $\therefore ErrA(l^t_*, b_3, t)$ has occurred.
    end if
  end if
```

Case 5 $h^t_* \in B_2 \wedge l^t_* \in B_2$ and $\Delta_{h^t_* l^t_*} > 0$

[H]

```
Here, $\exists b_1 \in (A^t_2 \cup A^t_3) \cap B_1$ and $\exists b_3 \in (A^t_1 \cup A^t_2) \cap B_3$
if $|\Delta_{h^t_* l^t_*}| < \Delta_{h^t_*}/2$ then
    if $\Delta_{b_1 h^t_*} > \Delta_{b_1}/4$ then
        if $b_1 \in A^t_2) \cap B_1$ then
            $ErrA(b_1, h^t_*, t)$
        else
            $b_1 \in A^t_3 \cap B_1$
            $ErrU(b_1, l^t_*, t)$ has occurred.
        end if
    else
        $\Delta_{b_1 h^t_*} \leq \Delta_{b_1}/4$ and hence $\Delta_{l^t_* b_3} \geq \Delta_{l^t_*}/4$
        if $b_3 \in A^t_2 \cap B_3$ then
            $ErrA(l^t_*, b_3, t)$ has occurred.
        else
            $b_3 \in A^t_1 \cap B_3$
            $ErrL(h^t_*, b_3, t)$ has occurred.
        end if
    end if
else
    $|\Delta_{h^t_* l^t_*}| > \Delta_{h^t_*}/2$
    $NEEDY^t_{h^t_*}(\Delta_{h^t_*}/4) \vee NEEDY^t_{l^t_*}(\Delta_{h^t_*}/4)$ has occurred.
end if
```

$$
\begin{aligned}
&< 130H_\epsilon + 64H_\epsilon \ln \frac{H_\epsilon}{\delta} + 256H_\epsilon \left[\ln C + 2\ln \frac{H_\epsilon}{\delta}\right] \\
&< (706 + 256 \ln C)H_\epsilon \ln \frac{H_\epsilon}{\delta} < CH_\epsilon \ln \frac{H_\epsilon}{\delta} \quad [\text{For } C \geq 2732]\,.
\end{aligned}
$$

□

Lemma 8.5. *Let $T^* = \left\lceil 2732H_\epsilon \ln\left(\frac{H_\epsilon}{\delta}\right)\right\rceil$. For every $T > T^*_1$, the probability that the Algorithm 1 has not terminated after T rounds of sampling is at most $\frac{8\delta}{T^2}$.*

Proof. Letting $\bar{T} = \frac{T}{2}$ we define two events for $\bar{T} \leq t \leq T-1$: $E^{(1)} \overset{\text{def}}{=} \exists a \in \mathcal{A} : CROSS^t_a$ and $E^{(2)} \overset{\text{def}}{=} \exists NEEDY^t_a\left(\frac{\Delta_a}{4}\right)$. If the algorithm stops for $t < \bar{T}$, then there is nothing to prove. On the contrary, let the algorithm has not stopped after $t > \bar{T}$ and neither $E^{(1)}$ nor $E^{(2)}$ has occurred. Letting N_{rounds} be the the required number of rounds beyond $\bar{T}$, we can

Case 5 (continued) $h^t_* \in B_2 \wedge l^t_* \in B_2$ and $\Delta_{h^t_* l^t_*} \leq 0$

```
[H]
  Here, $\exists b_1 \in (A^t_2 \cup A^t_3) \cap B_1$ and $\exists b_3 \in (A^t_1 \cup A^t_2) \cap B_3$
  if $|\Delta_{h^t_* l^t_*}| < \Delta_{h^t_*}/2$ then
    if $\Delta_{b_1 l^t_*} > \Delta_{b_1}/4$ then
      if $b_1 \in A^t_2 \cap B_1$ then
        $ErrA(b_1, h^t_*, t)$ has occurred.
      else
        $b_1 \in A^t_3 \cap B_1$
        $ErrU(b_1, l^t_*, t)$ has occurred.
      end if
    else
      $\Delta_{b_1 l^t_*} \leq \Delta_{b_1}/4$ and hence $\Delta_{h^t_* b_3} \geq \Delta_{h^t_*}/4$
      if $b_3 \in A^t_2 \cap B_3$ then
        $ErrA(l^t_*, b_3, t)$ has occurred.
      else
        $b_3 \in A^t_1 \cap B_3$
        $ErrL(h^t_*, b_3, t)$ has occurred.
      end if
    end if
  else
    $|\Delta_{h^t_* l^t_*}| > \Delta_{h^t_*}/2$
    $NEEDY^t_{h^t_*}(\Delta_{h^t_*}/4) \vee NEEDY^t_{l^t_*}(\Delta_{h^t_*}/4)$ has occurred.
  end if
```

upper bound it as:

$$\begin{aligned}
N_{rounds} &= \sum_{t=\bar{T}} \left\{ \mathbb{1}\left[NEEDY^t_{h^t_*}\left(\frac{\Delta_{h^t_*}}{4}\right) \vee NEEDY^t_{m^t_*}\left(\frac{\Delta_{m^t_*}}{4}\right) \vee NEEDY^t_{l^t_*}\left(\frac{\Delta_{l^t_*}}{4}\right)\right]\right\} \\
&\leq \sum_{\bar{T}}^{T-1} \sum_{a \in \mathcal{A}} \mathbb{1}\left[a \in \{h^t_*, m^t_*, l^t_*\} \wedge NEEDY^t_a\left(\frac{\Delta_a}{4}\right)\right] \\
&= \sum_{\bar{T}}^{T-1} \sum_{a \in \mathcal{A}} \mathbb{1}[a \in \{h^t_*, m^t_*, l^t_*\} \wedge (u^t_a < u^*(a,t))] \\
&\leq \sum_{\bar{T}}^{T-1} \sum_{a \in \mathcal{A}} \mathbb{1}[a \in \{h^t_*, m^t_*, l^t_*\} \wedge (u^t_a < u^*(a,t))] \\
&\leq \sum_{a \in \mathcal{A}} \sum_{\bar{T}}^{T-1} \mathbb{1}[(a \in \{h^t_*, m^t_*, l^t_*\}) \wedge (u^t_a < u^*(a,t))] \\
&\leq \sum_{a \in \mathcal{A}} u^*(a,t).
\end{aligned}$$

Using Lemma 8.4, $T \geq T^* \Rightarrow T > 2 + 2\sum_{a \in \mathcal{A}} u^*(a,t)$. Hence, if neither $E^{(1)}$ nor $E^{(2)}$ occurs then the algorithm runs for at most $\bar{T} + N_{rounds} \leq \lceil T/2 \rceil + \sum_{a \in \mathcal{A}} 16u^*(a,t) < T$ number of rounds.

The probability that the algorithm does not stop within T rounds, is upper-bounded by $P[E^{(1)} \vee E^{(2)}]$. Applying Lemma 8.2 and Lemma 8.3,

$$P[E^{(1)} \vee E^{(2)}] \leq \sum_{t=\bar{T}}^{T-1} \left(\frac{\delta}{k_1 t^3} + \frac{\delta}{k t^4}\right) \leq \sum_{t=\bar{T}}^{T-1} \frac{\delta}{k_1 t^3}\left(1 + \frac{2}{t}\right) \leq \left(\frac{T}{2}\right) \frac{8\delta}{k_1 T^3}\left(1 + \frac{4}{T}\right) < \frac{8\delta}{T^2}.$$

□

Case 6 $h_*^t \in B_2 \wedge l_*^t \in B_3$

```
[H]
  if $\Delta_{h_*^t l_*^t} \geq \frac{\Delta_{l_*^t}}{2}$ then
    Then $NEEDY^t_{h_*^t}(\Delta/4) \vee NEEDY^t_{l_*^t}(\Delta_{l_*^t}/4)$ has occurred.
  else
    $\Delta_{h_*^t l_*^t} < \frac{\Delta_{l_*^t}}{2}$
    $\therefore \forall b_1 \in \{A_2^t \cup A_3^t\} \cap B_1, \Delta_{b_1 h_*^t} > \frac{\Delta_{b_1}}{2}$.
    if $\exists b_1 \in A_2^t \cap B_1$ then
      $ErrA(b_1, h_*^t, t)$ has occurred.
    else
      $\exists b_1 \in A_3^t \cap B_1$.
      Then $ErrU(b_1^t, l_*^t, t)$ has occurred.
    end if
  end if
```

Case 7 $h_*^t \in B_3 \wedge l_*^t \in B_1$

```
[H]
  $\therefore ErrA(l_*^t, h_*^t, t)$ has occurred.
```

Theorem 3.2. *[Expected Sample Complexity of* LUCB-*k-m]* LUCB-*k-m solves* (k, m, n) *using an expected sample complexity upper bounded by* $O\left(H_\epsilon \log \frac{H_\epsilon}{\delta}\right)$.

Using Lemma 8.4, and Lemma 8.5 the expected sample complexity of the Algorithm 1 can be upper bounded as

$$E[SC] \leq 2\left(T_1^* + \sum_{t=T_1^*}^{\infty} \frac{8\delta}{T^2}\right) \leq 5464 \cdot \left(H_\epsilon \ln\left(\frac{H_\epsilon}{\delta}\right)\right) + 32. \tag{10}$$

Case 8 $h_*^t \in B_3 \wedge l_*^t \in B_2$

```
[H]
  if $\Delta_{h_*^t l_*^t} \geq \frac{\Delta_{h_*^t}}{2}$ then
    $ErrA(l_*^t, h_*^t, t)$ has occurred.
  else
    $\Delta_{h_*^t l_*^t} < \frac{\Delta_{h_*^t}}{2}$
    $\therefore \forall b_1 \in \{A_2^t \cup A_3^t\} \cap B_1, \Delta_{b_1 l_*^t} > \frac{\Delta_{b_1}}{2}$.
    if $\exists b_1 \in A_2^t \cap B_1$ then
      $ErrA(b_1, h_*^t, t)$ has occurred.
    else
      $\exists b_1 \in A_3^t \cap B_1$.
      $\therefore ErrU(b_1, l_*^t, t)$ has occurred.
    end if
  end if
```

```
Case 9 $h_*^t \in B_3 \wedge l_*^t \in B_3$
[H]
  $\exists b_1 \in \{A_2^t \cup A_3^t\} \cap B_1$
  if $\exists b_1 \in A_2^t \cap B_1$ then
    $ErrA(b_1, h_*^t, t)$ has occurred.
  else
    $\exists b_1 \in A_3^t \cap B_1$
    $\therefore ErrA(b_1, l_*^t, t)$ has occurred.
  end if
```

9. Proof of Theorem 4.7

Algorithm 4 describes OPTQP. It uses $\mathcal{P}_2$ (**?**) with MEDIAN ELIMINATION as the subroutine (inside $\mathcal{P}_2$), to select an $[\epsilon, \rho]$-optimal arm with confidence $1 - \delta'$. We have assumed $\delta' = 1/4$, in practice the one can choose any sufficiently small value for it, which will merely affect the multiplicative constant in the upper bound.

```
Algorithm 4 OPTQP
Input: $\mathcal{A}, \epsilon, \delta$, and OPTQF.
Output: A single $[\epsilon, \rho]$-optimal arm
  Set $\delta' = 1/4$, $u = \left\lceil \frac{1}{2(0.5-\delta')^2} \cdot \log \frac{2}{\delta} \right\rceil = \left\lceil 8 \log \frac{2}{\delta} \right\rceil$.
  Run $u$ copies of $\mathcal{P}_2(\mathcal{A}, \rho, \epsilon/2, \delta')$ and form set $S$ with the output arms.
  Return the output from OPTQF $\left(S, u, \lfloor \frac{u}{2} \rfloor, 1, \frac{\epsilon}{2}, \frac{\delta}{2}\right)$.
```

Theorem 9.1. *[Correctness and Sample Complexity of* OPTQP*] If* OPTQF *exists, then* OPTQP *solves* Q-P*, within the sample complexity* $\Theta\left(\frac{1}{\rho\epsilon^2} \log \frac{1}{\delta} + \gamma(\cdot)\right)$.

Proof. First we prove the correctness and then upper bound the sample complexity.

Correctness. First we notice that each copy of $\mathcal{P}_2$ outputs an $[\epsilon/2, \rho]$-optimal arm with probability at least $1 - \delta'$. Also, OPTQF outputs an $[\epsilon/2, \rho]$-optimal arm with probability $1 - \delta$. Let, $\hat{X}$ be the fraction of sub-optimal arms in S. Then $\Pr\{\hat{X} \geq \frac{1}{2}\} = \Pr\{\hat{X} - \delta' \geq \frac{1}{4}\} \leq \exp(-2 \cdot (\frac{1}{4})^2 \cdot u) = \exp(-2 \cdot \frac{1}{16} \cdot 8 \log \frac{2}{\delta}) < \frac{\delta}{2}$. On the other hand, the mistake probability of OPTQF is upper bounded by $\delta/2$. Therefore, by taking union bound, we get the mistake probability is upper bounded by δ. Also, the mean of the output arm is not less than $\frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon$ from the $(1 - \rho)$-th quantile.

Sample complexity. First we note that, for some appropriate constant C, the sample complexity (SC) of each of the u copies of $\mathcal{P}_2$ is $\frac{C}{\rho(\epsilon/2)^2} \left(\log \frac{2}{\delta'}\right)^2 \in O\left(\frac{1}{\rho\epsilon^2}\right)$. Hence, SC of all the u copies $\mathcal{P}_2$ together is upper bounded by $\frac{C_1 \cdot u}{\rho\epsilon^2}$, for some constant C_1. Also, for some constant C_2, the sample complexity of OPTQF is upper bounded by $C_2\left(\frac{u}{(u/2)(\epsilon/2)^2} \log \frac{2}{\delta} + \gamma(\cdot)\right) = C_2\left(\frac{8}{\epsilon^2} \log \frac{2}{\delta} + \gamma(\cdot)\right)$. Now, adding the sample complexities, and substituting for u we prove the bound. □