Best Arm Identification: A Unified Approach to Fixed Budget and Fixed Confidence

Victor Gabillon, Mohammad Ghavamzadeh, Alessandro Lazaric

Best Arm Identification: A Unified Approach to Fixed Budget and Fixed Confidence

Victor Gabillon **Mohammad Ghavamzadeh** **Alessandro Lazaric**
INRIA Lille - Nord Europe, Team SequeL
{victor.gabillon,mohammad.ghavamzadeh,alessandro.lazaric}@inria.fr

Abstract

We study the problem of identifying the best arm(s) in the stochastic multi-armed bandit setting. This problem has been studied in the literature from two different perspectives: *fixed budget* and *fixed confidence*. We propose a unifying approach that leads to a meta-algorithm called unified gap-based exploration (UGapE), with a common structure and similar theoretical analysis for these two settings. We prove a performance bound for the two versions of the algorithm showing that the two problems are characterized by the same notion of complexity. We also show how the UGapE algorithm as well as its theoretical analysis can be extended to take into account the variance of the arms and to multiple bandits. Finally, we evaluate the performance of UGapE and compare it with a number of existing fixed budget and fixed confidence algorithms.

1 Introduction

The problem of *best arm(s) identification* [6, 3, 1] in the stochastic multi-armed bandit setting has recently received much attention. In this problem, a forecaster repeatedly selects an arm and observes a sample drawn from its reward distribution during an exploration phase, and then is asked to return the best arm(s). Unlike the standard multi-armed bandit problem, where the goal is to maximize the cumulative sum of rewards obtained by the forecaster (see e.g., [14, 2]), in this problem the forecaster is evaluated on the quality of the arm(s) returned at the end of the exploration phase. This abstract problem models a wide range of applications. For instance, let us consider a company that has K different variants of a product and needs to identify the best one(s) before actually placing it on the market. The company sets up a testing phase in which the products are tested by potential customers. Each customer tests one product at the time and gives it a score (a reward). The objective of the company is to return a product at the end of the test phase which is likely to be successful once placed on the market (i.e., the best arm identification), and it is not interested in the scores collected during the test phase (i.e., the cumulative reward).

The problem of best arm(s) identification has been studied in two distinct settings in the literature.

Fixed budget. In the *fixed budget* setting (see e.g., [3, 1]), the number of rounds of the exploration phase is fixed and is known by the forecaster, and the objective is to maximize the probability of returning the best arm(s). In the above example, the company fixes the length of the test phase before hand (e.g., enrolls a fixed number of customers) and defines a strategy to choose which products to show to the testers so that the final selected product is the best with the highest probability. Audibert et al. [1] proposed two different strategies to solve this problem. They defined a strategy based on upper confidence bounds, called UCB-E, whose optimal parameterization is strictly related to a measure of the complexity of the problem. They also introduced an elimination algorithm, called Successive Rejects, which divides the budget n in phases and discards one arm per phase. Both algorithms were shown to have nearly optimal probability of returning the best arm. Deng et al. [5] and Gabillon et al. [7] considered the extension of the best arm identification problem to the multi-

bandit setting, where the objective is to return the best arm for each bandit. Recently, Bubeck et al. [4] extended the previous results to the problem of m-best arm identification and introduced a new version of the Successive Rejects algorithm (with accept and reject) that is able to return the set of the m-best arms with high probability.

Fixed confidence. In the *fixed confidence* setting (see e.g., [11, 6]), the forecaster tries to minimize the number of rounds needed to achieve a fixed confidence about the quality of the returned arm(s). In the above example, the company keeps enrolling customers in the test until it is, e.g., 95% confident that the best product has been identified. Maron & Moore [11] considered a slightly different setting where besides a fixed confidence also the maximum number of rounds is fixed. They designed an elimination algorithm, called Hoeffding Races, based on progressively discarding the arms that are suboptimal with enough confidence. Mnih et al. [13] introduced an improved algorithm, built on the Bernstein concentration inequality, which takes into account the empirical variance of each arm. Even-Dar et al. [6] studied the *fixed confidence* setting without any budget constraint and designed an elimination algorithm able to return an arm with a required accuracy ϵ (i.e., whose performance is at least ϵ-close to the optimal arm). Kalyanakrishnan & Stone [9] further extended this approach to the case where the m-best arms must be returned with a given confidence. Finally, Kalyanakrishnan et al. [10] recently introduced an algorithm for the case of m-best arm identification along with a thorough theoretical analysis showing the number of rounds needed to achieve the desired confidence.

Although the *fixed budget* and *fixed confidence* problems have been studied separately, they display several similarities. In this paper, we propose a unified approach to these two settings in the general case of m-best arm identification with accuracy ϵ.[1] The main contributions of the paper can be summarized as follows:

Algorithm. In Section 3, we propose a novel meta-algorithm, called *unified gap-based exploration* (UGapE), which uses the same arm selection and (arm) return strategies for the two settings. This algorithm allows us to solve settings that have not been covered in the previous work (e.g., the case of $\epsilon \neq 0$ has not been studied in the fixed budget setting). Furthermore, we show in Appendix C that UGapE outperforms existing algorithms in some settings (e.g., it improves the performance of the algorithm by Mnih et al. [13] in the fixed confidence setting). We also provide a thorough empirical evaluation of UGapE and compare it with a number of existing fixed budget and fixed confidence algorithms in Appendix C.

Theoretical analysis. Similar to the algorithmic contribution, in Section 4, we show that a large portion of the theoretical analysis required to study the behavior of the two settings of the UGapE algorithm can be unified in a series of lemmas. The final theoretical guarantees are thus a direct consequence of these lemmas when used in the two specific settings.

Problem complexity. In Section 4.4, we show that the theoretical analysis indicates that the two problems share exactly the same definition of complexity. In particular, we show that the probability of success in the fixed budget setting as well as the sample complexity in the fixed confidence setting strictly depend on the inverse of the gaps of the arms and the desired accuracy ϵ.

Extensions. Finally, in Appendix B, we discuss how the proposed algorithm and analysis can be extended to improved definitions of confidence interval (e.g., Bernstein-based bounds) and to more complex settings, such as the multi-bandit best arm identification problem introduced in [7].

2 Problem Formulation

In this section, we introduce the notation used throughout the paper. Let $A = \{1, \dots, K\}$ be the set of arms such that each arm $k \in A$ is characterized by a distribution ν_k bounded in $[0, b]$ with mean μ_k and variance σ_k^2. We define the m-max and m-argmax operators as[2]

$$\mu_{(m)} = \max_{k \in A}^{m} \mu_k \qquad \text{and} \qquad (m) = \arg\max_{k \in A}^{m} \mu_k \ ,$$

where (m) denotes the index of the m-th best arm in A and $\mu_{(m)}$ is its corresponding mean so that $\mu_{(1)} \geq \mu_{(2)} \geq \ldots \geq \mu_{(K)}$. We denote by $S^m \subset A$ any subset of m arms (i.e., $|S^m| = m < K$) and by $S^{m,*}$ the subset of the m best arms (i.e., $k \in S^{m,*}$ iif $\mu_k \geq \mu_{(m)}$). Without loss of generality, we

[1] Note that when $\epsilon = 0$ and $m = 1$ this reduces to the standard best arm identification problem.

[2] Ties are broken in an arbitrary but consistent manner.

assume there exists a unique set $S^{m,*}$. In the following we drop the superscript m and use $S = S^m$ and $S^* = S^{m,*}$ whenever m is clear from the context. With a slight abuse of notation we further extend the m-max operator to an operator returning a set of arms, such that

$$\{\mu_{(1)}, \dots, \mu_{(m)}\} = \max_{k \in A}^{1..m} \mu_k \qquad \text{and} \qquad S^* = \arg\max_{k \in A}^{1..m} \mu_k \ .$$

For each arm $k \in A$, we define the gap Δ_k as

$$\Delta_k = \begin{cases} \mu_k - \mu_{(m+1)} & \text{if } k \in S^* \\ \mu_{(m)} - \mu_k & \text{if } k \notin S^* \end{cases} .$$

This definition of gap indicates that if $k \in S^*$, Δ_k represents the "advantage" of arm k over the suboptimal arms, and if $k \notin S^*$, Δ_k denotes how suboptimal arm k is. Note that we can also write the gap as $\Delta_k = |\max_{i \neq k}^{m} \mu_i - \mu_k|$. Given an accuracy ϵ and a number of arms m, we say that an arm k is (ϵ,m)-optimal if $\mu_k \geq \mu_{(m)} - \epsilon$. Thus, we define the ($\epsilon$,$m$)-best arm identification problem as the problem of finding a set S of m (ϵ,m)-optimal arms.

The (ϵ,m)-best arm identification problem can be formalized as a game between a stochastic bandit environment and a forecaster. The distributions $\{\nu_k\}$ are unknown to the forecaster. At each round t, the forecaster pulls an arm $I(t) \in A$ and observes an independent sample drawn from the distribution $\nu_{I(t)}$. The forecaster estimates the expected value of each arm by computing the average of the samples observed over time. Let $T_k(t)$ be the number of times that arm k has been pulled by the end of round t, then the mean of this arm is estimated as $\widehat{\mu}_k(t) = \frac{1}{T_k(t)} \sum_{s=1}^{T_k(t)} X_k(s)$, where $X_k(s)$ is the s-th sample observed from ν_k. For any arm $k \in A$, we define the notion of *arm simple regret* as

$$r_k = \mu_{(m)} - \mu_k, \tag{1}$$

and for any set $S \subset A$ of m arms, we define the *simple regret* as

$$r_S = \max_{k \in S} r_k = \mu_{(m)} - \min_{k \in S} \mu_k. \tag{2}$$

We denote by $\Omega(t) \subset A$ the set of m arms returned by the forecaster at the end of the exploration phase (when the alg. stops after t rounds), and by $r_{\Omega(t)}$ its corresponding simple regret. Returning m (ϵ,m)-optimal arms is then equivalent to having $r_{\Omega(t)}$ smaller than ϵ. Given an accuracy ϵ and a number of arms m to return, we now formalize the two settings of *fixed budget* and *fixed confidence*.

Fixed budget. The objective is to design a forecaster capable of returning a set of m (ϵ,m)-optimal arms with the largest possible confidence using a fixed budget of n rounds. More formally, given a budget n, the performance of the forecaster is measured by the probability $\widetilde{\delta}$ of not meeting the (ϵ,m) requirement, i.e., $\widetilde{\delta} = \mathbb{P}\big[r_{\Omega(n)} \geq \epsilon\big]$, the smaller $\widetilde{\delta}$, the better the algorithm.

Fixed confidence. The goal is to design a forecaster that stops as soon as possible and returns a set of m (ϵ,m)-optimal arms with a fixed confidence. We denote by $\widetilde{n}$ the time when the algorithm stops and by $\Omega(\widetilde{n})$ its set of returned arms. Given a confidence level δ, the forecaster has to guarantee that $\mathbb{P}\big[r_{\Omega(\widetilde{n})} \geq \epsilon\big] \leq \delta$. The performance of the forecaster is then measured by the number of rounds $\widetilde{n}$ either in expectation or high probability.

Although these settings have been considered as two distinct problems, in Section 3 we introduce a unified arm selection strategy that can be used in both cases by simply changing the stopping criteria. Moreover, we show in Section 4 that the bounds on the performance of the algorithm in the two settings share the same notion of complexity and can be derived using very similar arguments.

3 Unified Gap-based Exploration Algorithm

In this section, we describe the unified gap-based exploration (UGapE) meta-algorithm and show how it is implemented in the fixed-budget and fixed-confidence settings. As shown in Figure 1, both fixed-budget (UGapEb) and fixed-confidence (UGapEc) instances of UGapE use the same arm-selection strategy, SELECT-ARM (described in Figure 2), and upon stopping, return the m-best arms in the same manner (using Ω). The two algorithms only differ in their stopping criteria. More precisely, both algorithms receive as input the definition of the problem (ϵ, m), a constraint (the

budget n in UGapEb and the confidence level δ in UGapEc), and a parameter (a or c). While UGapEb runs for n rounds and then returns the set of arms $\Omega(n)$, UGapEc runs until it achieves the desired accuracy ϵ with the requested confidence level δ. This difference is due to the two different objectives targeted by the algorithms; while UGapEc aims at optimizing its budget for a given confidence level, UGapEb's goal is to optimize the quality of its recommendation for a fixed budget.

UGapEb (ϵ, m, n, a)

Parameters: accuracy ϵ, number of arms m, budget n, exploration parameter a

Initialize: Pull each arm k once, update $\widehat{\mu}_k(K)$ and set $T_k(K) = 1$

for $t = K+1, \dots, n$ **do**
  SELECT-ARM (t)
end for

Return $\Omega(n) = \arg\min_{J(t)} B_{J(t)}(t)$

UGapEc (ϵ, m, δ, c)

Parameters: accuracy ϵ, number of arms m, confidence level δ, exploration parameter c

Initialize: Pull each arm k once, update $\widehat{\mu}_k(K)$, set $T_k(K) = 1$ and $t \leftarrow K+1$

while $B_{J(t)}(t) \geq \epsilon$ **do**
  SELECT-ARM (t)
  $t \leftarrow t+1$
end while

Return $\Omega(t) = J(t)$

Figure 1: The pseudo-code for the UGapE algorithm in the fixed-budget (UGapEb) *(left)* and fixed-confidence (UGapEc) *(right)* settings.

Regardless of the final objective, how to select an arm at each round (arm-selection strategy) is the key component of any multi-arm bandit algorithm. One of the most important features of UGapE is having a unique arm-selection strategy for the fixed-budget and fixed-confidence settings. We now describe the UGapE's arm-selection strategy, whose pseudo-code has been reported in Figure 2. At each time step t, UGapE first uses the observations up to time $t-1$ and computes an index $B_k(t) = \max_{i \neq k}^{m} U_i(t) - L_k(t)$ for each arm $k \in A$, where

SELECT-ARM (t)

Compute $B_k(t)$ for each arm $k \in A$

Identify the set of m arms $J(t) \in \arg\min_{k \in A}^{1..m} B_k(t)$

Pull the arm $I(t) = \arg\max_{k \in \{l_t, u_t\}} \beta_k(t-1)$

Observe $X_{I(t)}\big(T_{I(t)}(t-1)+1\big) \sim \nu_{I(t)}$

Update $\widehat{\mu}_{I(t)}(t)$ and $T_{I(t)}(t)$

Figure 2: The pseudo-code for the UGapE's arm-selection strategy. This routine is used in both UGapEb and UGapEc instances of UGapE.

$$\forall t,\ \forall k \in A \qquad U_k(t) = \widehat{\mu}_k(t-1) + \beta_k(t-1) \qquad , \qquad L_k(t) = \widehat{\mu}_k(t-1) - \beta_k(t-1). \tag{3}$$

In Eq. 3, $\beta_k(t-1)$ is a confidence interval,[3] and $U_k(t)$ and $L_k(t)$ are high probability upper and lower bounds on the mean of arm k, μ_k, after $t-1$ rounds. Note that the parameters a and c are used in the definition of the confidence interval β_k, whose shape strictly depends on the concentration bound used by the algorithm. For example, we can derive β_k from the Chernoff-Hoeffding bound as

$$\text{UGapEb:}\ \ \beta_k(t-1) = b\sqrt{\frac{a}{T_k(t-1)}}\ , \qquad \text{UGapEc:}\ \ \beta_k(t-1) = b\sqrt{\frac{c \log \frac{4K(t-1)^3}{\delta}}{T_k(t-1)}}\ . \tag{4}$$

In Sec. 4, we discuss how the parameters a and c can be tuned and we show that while a should be tuned as a function of n and ϵ in UGapEb, $c = 1/2$ is always a good choice for UGapEc. Defining the confidence interval in a general form $\beta_k(t-1)$ allows us to easily extend the algorithm by taking into account different (higher) moments of the arms (see Appendix B for the case of variance, where $\beta_k(t-1)$ is obtained from the Bernstein inequality). From Eq. 3, we may see that the index $B_k(t)$ is an upper-bound on the simple regret r_k of the kth arm (see Eq. 1). We also define an index for a set S as $B_S(t) = \max_{i \in S} B_i(t)$. Similar to the arm index, B_S is also defined in order to upper-bound the simple regret r_S with high probability (see Lemma 1).

After computing the arm indices, UGapE finds a set of m arms $J(t)$ with minimum upper-bound on their simple regrets, i.e., $J(t) = \arg\min_{k \in A}^{1..m} B_k(t)$. From $J(t)$, it computes two arm indices $u_t =$

[3] To be more precise, $\beta_k(t-1)$ is the width of a confidence interval or a confidence radius.

$\arg\max_{j \notin J(t)} U_j(t)$ and $l_t = \arg\min_{i \in J(t)} L_i(t)$, where in both cases the tie is broken in favor of the arm with the largest uncertainty $\beta(t-1)$. Arms l_t and u_t are the worst possible arm among those in $J(t)$ and the best possible arm left outside $J(t)$, respectively, and together they represent how bad the choice of $J(t)$ could be. Intuitively, UGapE pulls the most uncertain between u_t or l_t allows Finally, the algorithm selects and pulls the arm $I(t)$ as the arm with the larger $\beta(t-1)$ among u_t and l_t, observes a sample $X_{I(t)}\big(T_{I(t)}(t-1)+1\big)$ from the distribution $\nu_{I(t)}$, and updates the empirical mean $\widehat{\mu}_{I(t)}(t)$ and the number of pulls $T_{I(t)}(t)$ of the selected arm $I(t)$.

There are two more points that need to be discussed about the UGapE algorithm. **1)** While UGapEc defines the set of returned arms as $\Omega(t) = J(t)$, UGapEb returns the set of arms $J(t)$ with the smallest index, i.e., $\Omega(n) = \arg\min_{J(t)} B_{J(t)}(t),\ t \in \{1, \dots, n\}$. **2)** UGapEc stops (we refer to the number of rounds before stopping as $\widetilde{n}$) when $B_{J(\widetilde{n}+1)}(\widetilde{n}+1)$ is less than the given accuracy ϵ, i.e., when even the mth worst upper-bound on the arm simple regret among all the arms in the selected set $J(\widetilde{n}+1)$ is smaller than ϵ. This guarantees that the simple regret (see Eq. 2) of the set returned by the algorithm, $\Omega(\widetilde{n}) = J(\widetilde{n}+1)$, to be smaller than ϵ with probability larger than $1-\delta$.

4 Theoretical Analysis

In this section, we provide high probability upper-bounds on the performance of the two instances of the UGapE algorithm, UGapEb and UGapEc, introduced in Section 3. An important feature of UGapE is that since its fixed-budget and fixed-confidence versions share the same arm-selection strategy, a large part of their theoretical analysis can be unified. We first report this unified part of the proof in Section 4.1, and then provide the final performance bound for each of the algorithms, UGapEb and UGapEc, separately, in Sections 4.2 and 4.3, respectively.

Before moving to the main results, we define additional notation used in the analysis. We first define event $\mathcal{E}$ as

$$\mathcal{E} = \big\{\forall k \in A,\ \forall t \in \{1, \dots, T\},\ \big|\widehat{\mu}_k(t) - \mu_k\big| < \beta_k(t)\big\}, \tag{5}$$

where the values of T and β_k are defined for each specific setting separately. Note that event $\mathcal{E}$ plays an important role in the sequel, since it allows us to first derive a series of results which are directly implied by the event $\mathcal{E}$ and to postpone the study of the stochastic nature of the problem (i.e., the probability of $\mathcal{E}$) in the two specific settings. In particular, when $\mathcal{E}$ holds, we have that for any arm $k \in A$ and at any time t, $L_k(t) \le \mu_k \le U_k(t)$. Finally, we define the complexity of the problem as

$$H_\epsilon = \sum_{i=1}^{K} \frac{b^2}{\max(\frac{\Delta_i + \epsilon}{2}, \epsilon)^2}\ . \tag{6}$$

Note that although the complexity has an explicit dependence on ϵ, it also depends on the number of arms m through the definition of the gaps Δ_i, thus making it a complexity measure of the (ϵ, m) best arm identification problem. In Section 4.4, we will discuss why the complexity of the two instances of the problem is measured by this quantity.

4.1 Analysis of the Arm-Selection Strategy

Here we report lower (Lemma 1) and upper (Lemma 2) bounds for indices B_S on the event $\mathcal{E}$, which show their connection with the regret and gaps. The technical lemmas used in the proofs (Lemmas 3 and 4 and Corollary 1) are reported in Appendix A. We first prove that for any set $S \neq S^*$ and any time $t \in \{1, \dots, T\}$, the index $B_S(t)$ is actually an upper-bound on the simple regret of this set r_S.

Lemma 1. *On event $\mathcal{E}$, for any set $S \neq S^*$ and any time $t \in \{1, \dots, T\}$, we have $B_S(t) \ge r_S$.*

Proof. On event $\mathcal{E}$, for any arm $i \notin S^*$ and each time $t \in \{1, \dots, T\}$, we may write

$$\begin{aligned} B_i(t) &= \max_{j \neq i}^{m} U_j(t) - L_i(t) = \max_{j \neq i}^{m} \big(\widehat{\mu}_j(t-1) + \beta_j(t-1)\big) - \big(\widehat{\mu}_i(t-1) - \beta_i(t-1)\big) \\ &\ge \max_{j \neq i}^{m} \mu_j - \mu_i = \mu_{(m)} - \mu_i = r_i\ . \end{aligned} \tag{7}$$

Using Eq. 7, we have

$$B_S(t) = \max_{i \in S} B_i(t) \ge \max_{i \in (S - S^*)} B_i(t) \ge \max_{i \in (S - S^*)} r_i = r_S,$$

where the last passage follows from the fact that $r_i \leq 0$ for any $i \in S^*$. □

Lemma 2. *On event $\mathcal{E}$, if arm $k \in \{l_t, u_t\}$ is pulled at time $t \in \{1, \dots, T\}$, we have*

$$B_{J(t)}(t) \leq \min\big(0, -\Delta_k + 2\beta_k(t-1)\big) + 2\beta_k(t-1). \tag{8}$$

Proof. We first prove the statement for $B(t) = U_{u_t}(t) - L_{l_t}(t)$, i.e.,

$$B(t) \leq \min\big(0, -\Delta_k + 2\beta_k(t-1)\big) + 2\beta_k(t-1). \tag{9}$$

We consider the following cases:
Case 1. $k = u_t$**:**
Case 1.1. $u_t \in S^*$**:** Since by definition $u_t \notin J(t)$, there exists an arm $j \notin S^*$ such that $j \in J(t)$. Now we may write

$$\mu_{(m+1)} \geq \mu_j \overset{\text{(a)}}{\geq} L_j(t) \overset{\text{(b)}}{\geq} L_{l_t}(t) \overset{\text{(c)}}{\geq} L_{u_t}(t) = \widehat{\mu}_k(t-1) - \beta_k(t-1) \overset{\text{(d)}}{\geq} \mu_k - 2\beta_k(t-1) \tag{10}$$

(a) and **(d)** hold because of event $\mathcal{E}$, **(b)** follows from the fact that $j \in J(t)$ and from the definition of l_t, and **(c)** is the result of Lemma 4. From Eq. 10, we may deduce that $-\Delta_k + 2\beta_k(t-1) \geq 0$, which together with Corollary 1 gives us the desired result (Eq. 9).

Case 1.2. $u_t \notin S^*$**:**
Case 1.2.1. $l_t \in S^*$**:** In this case, we may write

$$\begin{aligned} B(t) = U_{u_t}(t) - L_{l_t}(t) &\overset{\text{(a)}}{\leq} \mu_{u_t} + 2\beta_{u_t}(t-1) - \mu_{l_t} + 2\beta_{l_t}(t-1) \\ &\overset{\text{(b)}}{\leq} \mu_{u_t} + 2\beta_{u_t}(t-1) - \mu_{(m)} + 2\beta_{l_t}(t-1) \overset{\text{(c)}}{\leq} -\Delta_{u_t} + 4\beta_{u_t}(t-1) \end{aligned} \tag{11}$$

(a) holds because of event $\mathcal{E}$, **(b)** is from the fact that $l_t \in S^*$, and **(c)** is because u_t is pulled, and thus, $\beta_{u_t}(t-1) \geq \beta_{l_t}(t-1)$. The final result follows from Eq. 11 and Corollary 1.

Case 1.2.2. $l_t \notin S^*$**:** Since $l_t \notin S^*$ and the fact that by definition $l_t \in J(t)$, there exists an arm $j \in S^*$ such that $j \notin J(t)$. Now we may write

$$\mu_{u_t} + 2\beta_{u_t}(t-1) \overset{\text{(a)}}{\geq} U_{u_t}(t) \overset{\text{(b)}}{\geq} U_j(t) \overset{\text{(c)}}{\geq} \mu_j \overset{\text{(d)}}{\geq} \mu_{(m)} \tag{12}$$

(a) and **(c)** hold because of event $\mathcal{E}$, **(b)** is from the definition of u_t and the fact that $j \notin J(t)$, and **(d)** holds because $j \in S^*$. From Eq. 12, we may deduce that $-\Delta_{u_t} + 2\beta_{u_t}(t-1) \geq 0$, which together with Corollary 1 gives us the final result (Eq. 9).

With similar arguments and cases, we prove the result of Eq. 9 for $k = l_t$. The final statement of the lemma (Eq. 8) follows directly from $B_{J(t)}(t) \geq B(t)$ as shown in Lemma 3. □

Using Lemmas 1 and 2, we define an upper and a lower bounds on $B_{J(t)}$ in terms of quantities related to the regret of $J(t)$. Lemma 1 confirms the intuition that the B-values upper-bound the regret of the corresponding set of arms (with high probability). Unfortunately, this is not enough to claim that selecting $J(t)$ as the set of arms with smallest B-values actually correspond to arms with small regret, since $B_{J(t)}$ could be an arbitrary loose bound on the regret. Lemma 2 provides this complementary guarantee specifically for the set $J(t)$, in the form of an upper-bound on $B_{J(t)}$ w.r.t. the gap of $k \in \{u_t, l_t\}$. This implies that as the algorithm runs, the choice of $J(t)$ becomes more and more accurate since $B_{J(t)}$ is constrained between $r_{J(t)}$ and a quantity (Eq. 8) that gets smaller and smaller, thus implying that selecting the arms with the smaller B-value, i.e., the set $J(t)$, corresponds to those which actually have the smallest regret, i.e., the arms in S^*. This argument will be implicitly at the basis of the proofs of the two following theorems.

4.2 Regret Bound for the Fixed-Budget Setting

Here we prove an upper-bound on the simple-regret of UGapEb. Since the setting considered by the algorithm is fixed-budget, we may set $T = n$. From the definition of the confidence interval $\beta_i(t)$

in Eq. 4 and a union bound, we have that $\mathbb{P}(\mathcal{E}) \geq 1 - 2Kn\exp(-2a)$.[4] We now have all the tools needed to prove the performance of UGapEb for the m (ϵ,m)-best arm identification problem.

Theorem 1. *If we run UGapEb with parameter $0 < a \leq \frac{n-K}{4H_\epsilon}$, its simple regret $r_{\Omega(n)}$ satisfies*

$$\widetilde{\delta} = \mathbb{P}\big(r_{\Omega(n)} \geq \epsilon\big) \leq 2Kn\exp(-2a),$$

and in particular this probability is minimized for $a = \frac{n-K}{4H_\epsilon}$.

Proof. The proof is by contradiction. We assume that $r_{\Omega(n)} > \epsilon$ on event $\mathcal{E}$ and consider the following two steps:

Step 1: Here we show that on event $\mathcal{E}$, we have the following upper-bound on the number of pulls of any arm $i \in A$:

$$T_i(n) < \frac{4ab^2}{\max\left(\frac{\Delta_i+\epsilon}{2}, \epsilon\right)^2} + 1. \tag{13}$$

Let t_i be the last time that arm i is pulled. If arm i has been pulled only during the initialization phase, $T_i(n) = 1$ and Eq. 13 trivially holds. If i has been selected by SELECT-ARM, then we have

$$\min\big(-\Delta_i + 2\beta_i(t_i - 1), 0\big) + 2\beta_i(t_i - 1) \overset{\text{(a)}}{\geq} B(t_i) \overset{\text{(b)}}{\geq} B_{J(t_i)}(t_i) \overset{\text{(c)}}{\geq} B_{\Omega(n)}(t_\ell) \overset{\text{(d)}}{>} \epsilon, \tag{14}$$

where $t_\ell \in \{1, \ldots, n\}$ is the time such that $\Omega(n) = J(t_\ell)$. **(a)** and **(b)** are the results of Lemmas 2 and 3, **(c)** is by the definition of $\Omega(n)$, and **(d)** holds because using Lemma 1, we know that if the algorithm suffers a simple regret $r_{\Omega(n)} > \epsilon$ (as assumed at the beginning of the proof), then $\forall t = 1, \ldots, n+1,\ B_{\Omega(n)}(t) > \epsilon$. By the definition of t_i, we know $T_i(n) = T_i(t_i - 1) + 1$. Using this fact, the definition of $\beta_i(t_i - 1)$, and Eq. 14, it is straightforward to show that Eq. 13 holds.

Step 2: We know that $\sum_{i=1}^K T_i(n) = n$. Using Eq. 13, we have $\sum_{i=1}^K \frac{4ab^2}{\max\left(\frac{\Delta_i+\epsilon}{2}, \epsilon\right)^2} + K > n$ on event $\mathcal{E}$. It is easy to see that by selecting $a \leq \frac{n-K}{4H_\epsilon}$, the left-hand-side of this inequality will be smaller than or equal to n, which is a contradiction. Thus, we conclude that $r_{\Omega(n)} \leq \epsilon$ on event $\mathcal{E}$. The final result follows from the probability of event $\mathcal{E}$ defined at the beginning of this section. □

4.3 Regret Bound for the Fixed-Confidence Setting

Here we prove an upper-bound on the simple-regret of UGapEc. Since the setting considered by the algorithm is fixed-confidence, we may set $T = +\infty$. From the definition of the confidence interval $\beta_i(t)$ in Eq. 4 and a union bound on $T_k(t) \in \{0, \ldots, t\},\ t = 1, \ldots, \infty$, we have that $\mathbb{P}(\mathcal{E}) \geq 1 - \delta$.

Theorem 2. *The UGapEc algorithm stops after $\widetilde{n}$ rounds and returns a set of m arms, $\Omega(\widetilde{n})$, that satisfies*

$$\mathbb{P}\big(r_{\Omega(\widetilde{n}+1)} \leq \epsilon \wedge \widetilde{n} \leq N\big) \geq 1 - \delta,$$

where $N = K + O(H_\epsilon \log \frac{H_\epsilon}{\delta})$ and c has been set to $1/2$.

Proof. We first prove the bound on the simple regret of UGapEc. Using Lemma 1, we have that on event $\mathcal{E}$, the simple regret of UGapEc upon stopping satisfies $B_{J(\widetilde{n}+1)}(\widetilde{n}+1) = B_{\Omega(\widetilde{n}+1)}(\widetilde{n}+1) \geq r_{\Omega(\widetilde{n}+1)}$. As a result, on event $\mathcal{E}$, the regret of UGapEc cannot be bigger than ϵ, because then it contradicts the stopping condition of the algorithm, i.e., $B_{J(\widetilde{n}+1)}(\widetilde{n}+1) < \epsilon$. Therefore, we have $\mathbb{P}\big(r_{\Omega(\widetilde{n}+1)} \leq \epsilon\big) \geq 1 - \delta$. Now we prove the bound for the sample complexity. Similar to the proof of Theorem 1, we consider the following two steps:

[4] The extension to a confidence interval that takes into account the variance of the arms is discussed in Appendix B.

Step 1: Here we show that on event $\mathcal{E}$, we have the following upper-bound on the number of pulls of any arm $i \in A$:

$$T_i(\widetilde{n}) \leq \frac{2b^2 \log(4K(\widetilde{n}-1)^3/\delta)}{\max\left(\frac{\Delta_i+\epsilon}{2}, \epsilon\right)^2} + 1. \tag{15}$$

Let t_i be the last time that arm i is pulled. If arm i has been pulled only during the initialization phase, $T_i(\widetilde{n}) = 1$ and Eq. 15 trivially holds. If i has been selected by SELECT-ARM, then we have $B_{J(t_i)}(t_i) \geq \epsilon$. Now using Lemma 2, we may write

$$B_{J(t_i)}(t_i) \leq \min\left(0, -\Delta_i + 2\beta_i(t_i - 1)\right) + 2\beta_i(t_i - 1). \tag{16}$$

We can prove Eq. 15 by plugging in the value of $\beta_i(t_i - 1)$ from Eq. 4 and solving Eq. 16 for $T_i(t_i)$ taking into account that $T_i(t_i - 1) + 1 = T_i(t_i)$.

Step 2: We know that $\sum_{i=1}^{K} T_i(\widetilde{n}) = \widetilde{n}$. Using Eq. 15, on event $\mathcal{E}$, we have $2H_\epsilon \log\left(K(\widetilde{n} - 1)^3/\delta\right) + K \geq \widetilde{n}$. Solving this inequality gives us $\widetilde{n} \leq N$. □

4.4 Problem Complexity

Theorems 1 and 2 indicate that both the probability of success and sample complexity of UGapE are directly related to the complexity H_ϵ defined by Eq. 6. This implies that H_ϵ captures the intrinsic difficulty of the (ϵ,m)-best arm(s) identification problem independently from the specific setting considered. Furthermore, note that this definition generalizes existing notions of complexity. For example, for $\epsilon = 0$ and $m = 1$ we recover the complexity used in the definition of UCB-E [1] for the fixed budget setting and the one defined in [6] for the fixed accuracy problem. Let us analyze H_ϵ in the general case of $\epsilon > 0$. We define the complexity of a single arm $i \in A$, $H_{\epsilon,i} = b^2/\max(\frac{\Delta_i+\epsilon}{2}, \epsilon)^2$. When the gap Δ_i is smaller than the desired accuracy ϵ, i.e., $\Delta_i \leq \epsilon$, then the complexity reduces to $H_{\epsilon,i} = 1/\epsilon^2$. In fact, the algorithm can stop as soon as the desired accuracy ϵ is achieved, which means that there is no need to exactly discriminate between arm i and the best arm. On the other hand, when $\Delta_i > \epsilon$, then the complexity becomes $H_{\epsilon,i} = 4b^2/(\Delta_i + \epsilon)^2$. This shows that when the desired accuracy is smaller than the gap, the complexity of the problem is smaller than the case of $\epsilon = 0$, for which we have $H_{0,i} = 4b^2/\Delta_i^2$.

More in general, the analysis reported in the paper suggests that the performance of a confidence-bound-based algorithm such as UGapE is characterized by the same notion of complexity in both settings. Thus, whenever the complexity is known, it is possible to exploit the theoretical analysis (bounds on the performance) to easily switch from one setting to the other. For instance, as suggested in section 5.4 of [8], if the complexity H is known, an algorithm as UGapEc can be adapted to run in the fixed budget setting by inverting the bound on its sample complexity (equation xxx). This would lead to an algorithm very similar to UGapEb with similar performance, although the parameter tuning could be more difficult because of the intrinsic poor accuracy in the constants in the bound. On the other hand, it is an open question whether it is possible to find an "equivalence" between algorithms for the two different settings when the complexity is not known. In particular, it would be important to derive a distribution–dependent lower bound in the form of the one reported in [1] for the general case of $\epsilon \geq 0$ and $m \geq 1$ for both the fixed budget and fixed confidence settings.

5 Summary and Discussion

We proposed a meta-algorithm, called unified gap-based exploration (UGapE), that unifies the two settings of the best arm(s) identification problem in stochastic multi-armed bandit: *fixed budget* and *fixed confidence*. UGapE can be instantiated as two algorithms with a common structure (the same arm-selection and arm-return strategies) corresponding to these two settings, whose performance can be analyzed in a unified way, i.e., a large portion of their theoretical analysis can be unified in a series of lemmas. We proved a performance bound for the UGapE algorithm in the two settings. We also showed how UGapE and its theoretical analysis can be extended to take into account the variance of the arms and to multiple bandits. Finally, we evaluated the performance of UGapE and compare it with a number of existing fixed budget and fixed confidence algorithms.

This unification is important for both theoretical and algorithmic reasons. Despite their similarities, fixed budget and fixed confidence settings have been treated differently in the literature. We believe

that this unification provides a better understanding of the intrinsic difficulties of the best arm(s) identification problem. In particular, our analysis showed that the same complexity term characterizes the hardness of both settings. As mentioned in the introduction, there was no algorithm available for several settings considered in this paper, e.g., (ϵ,m)-best arm identification with fixed budget. With UGapE, we introduced an algorithm that can be easily adapted to all these settings.

References

[1] J.-Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In *Proceedings of the Twenty-Third Annual Conference on Learning Theory*, pages 41–53, 2010.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, 47:235–256, 2002.

[3] S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in multi-armed bandit problems. In *Proceedings of the Twentieth International Conference on Algorithmic Learning Theory*, pages 23–37, 2009.

[4] S. Bubeck, T. Wang, and N. Viswanathan. Multiple identifications in multi-armed bandits. *CoRR*, abs/1205.3181, 2012.

[5] K. Deng, J. Pineau, and S. Murphy. Active learning for developing personalized treatment. In *Proceedings of the Twenty-Seventh International Conference on Uncertainty in Artificial Intelligence*, pages 161–168, 2011.

[6] E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

[7] V. Gabillon, M. Ghavamzadeh, A. Lazaric, and S. Bubeck. Multi-bandit best arm identification. In *Proceedings of Advances in Neural Information Processing Systems 25*, pages 2222–2230, 2011.

[8] S. Kalyanakrishnan. *Learning Methods for Sequential Decision Making with Imperfect Representations*. PhD thesis, Department of Computer Science, The University of Texas at Austin, Austin, Texas, USA, December 2011. Published as UT Austin Computer Science Technical Report TR-11-41.

[9] S. Kalyanakrishnan and P. Stone. Efficient selection of multiple bandit arms: Theory and practice. In *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, pages 511–518, 2010.

[10] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. Pac subset selection in stochastic multi-armed bandits. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2012.

[11] O. Maron and A. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. In *Proceedings of Advances in Neural Information Processing Systems 6*, pages 59–66, 1993.

[12] A. Maurer and M. Pontil. Empirical bernstein bounds and sample-variance penalization. In *22th annual conference on learning theory*, 2009.

[13] V. Mnih, Cs. Szepesvári, and J.-Y. Audibert. Empirical Bernstein stopping. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning*, pages 672–679, 2008.

[14] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematics Society*, 58:527–535, 1952.

A Technical Lemmas

In this section, we report three results (Lemmas 3 and 4 and Corollary 1) that have been used in the proof of Lemmas 1 and 2. Let us first redefine $B(t)$ for each time $t \in \{1, \dots, T\}$ as

$$B(t) = U_{u_t}(t) - L_{l_t}(t) = \max_{j \notin J(t)} U_j(t) - \min_{i \in J(t)} L_i(t). \tag{17}$$

Lemma 3. *For each $t \in \{1, \dots, T\}$, we have $B_{J(t)}(t) \leq B(t)$.*

Proof. For $t \in \{1, \dots, T\}$, we may write

$$\begin{aligned} B_{J(t)}(t) &= \max_{i \in J(t)} \left(\overset{m}{\max_{j \neq i}} U_j(t) - L_i(t) \right) \leq \max_{i \in J(t)} \overset{m}{\max_{j \neq i}} U_j(t) - \min_{i \in J(t)} L_i(t) \\ &\overset{\text{(a)}}{\leq} \max_{j \notin J(t)} U_j(t) - \min_{i \in J(t)} L_i(t) = U_{u_t}(t) - L_{l_t}(t) = B(t). \end{aligned}$$

(a) Let set $Z = \max_{i \in J(t)} \overset{m}{\max_{j \neq i}} U_j(t)$ and define $\widehat{S}(t) = \arg \overset{1..m}{\max_{i \in A}} U_i(t)$ to be the set of m arms with the largest $U(t)$. If $J(t) = \widehat{S}(t)$ then $Z = U_{(m+1)}(t) = \max_{j \notin J(t)} U_j(t)$, and if $J(t) \neq \widehat{S}(t)$ then $Z = U_{(m)}(t) \leq \max_{j \notin J(t)} U_j(t)$. □

Lemma 4. *At each time $t \in \{1, \dots, T\}$, we have that*

$$\text{if } u_t \text{ is pulled,} \qquad L_{u_t}(t) \leq L_{l_t}(t) \tag{18}$$

$$\text{if } l_t \text{ is pulled,} \qquad U_{u_t}(t) \leq U_{l_t}(t). \tag{19}$$

Proof. We consider the following two cases:
Case 1. $\widehat{\mu}_{u_t}(t-1) \leq \widehat{\mu}_{l_t}(t-1)$**:** If we pull u_t, by definition we have $\beta_{u_t}(t-1) \geq \beta_{l_t}(t-1)$, and thus, Eq. 18 holds. A similar reasoning gives Eq. 19 when l_t is pulled.

Case 2. $\widehat{\mu}_{u_t}(t-1) > \widehat{\mu}_{l_t}(t-1)$**:** We consider the following two sub-cases:

Case 2.1. u_t is pulled**:** We prove this case by contradiction. Let us assume that $L_{u_t}(t) > L_{l_t}(t)$. Since arm u_t is pulled, by definition we have $\beta_{u_t}(t-1) \geq \beta_{l_t}(t-1)$, and thus, $U_{u_t}(t) > U_{l_t}(t)$. From this, it is easy to see that $\overset{m}{\max_{j \neq l_t}} U_j(t) \geq \overset{m}{\max_{j \neq u_t}} U_j(t)$. Using these results, we may write

$$B_{u_t}(t) = \overset{m}{\max_{j \neq u_t}} U_j(t) - L_{u_t}(t) < \overset{m}{\max_{j \neq l_t}} U_j(t) - L_{l_t}(t) = B_{l_t}(t).$$

Since $u_t \notin J(t)$ and $l_t \in J(t)$ and $J(t)$ collects all the arms with the smallest B_i values, we have $B_{u_t}(t) \geq B_{l_t}(t)$ by definition. This contradicts the previous statement, and thus, Eq. 18 holds.

Case 2.2. l_t is pulled**:** With a similar reasoning to Case 2.1. and by contradiction, we can show that Eq. 19 holds in this case. □

Corollary 1. *If arm k is pulled at time $t \in \{1, \dots, T\}$, then we have $B(t) \leq 2\beta_k(t-1)$.*

Proof. The proof is a direct application of Lemma 4. We know that the pulled arm k is either u_t or l_t. If $k = u_t$, we have

$$B(t) = U_{u_t}(t) - L_{l_t}(t) \overset{\text{(a)}}{\leq} U_{u_t}(t) - L_{u_t}(t) = 2\beta_{u_t}(t-1) = 2\beta_k(t-1),$$

where **(a)** is from Eq. 18 in Lemma 4. Similarly if $k = l_t$, we have

$$B(t) = U_{u_t}(t) - L_{l_t}(t) \overset{\text{(b)}}{\leq} U_{l_t}(t) - L_{l_t}(t) = 2\beta_{l_t}(t-1) = 2\beta_k(t-1),$$

where **(b)** is from Eq. 19 in Lemma 4. □

B Extensions

In this section, we propose three variants of the UGapE algorithm with the objective of extending its applicability and improving its performance.

UGapE-Variance (UGapE-V). As discussed in Section 3, UGapE pulls arms according to their B index, which is a high probability upper-bound on their simple regret. This gives us the flexibility of using any high probability bound in the definition of index in UGapE and to extend the algorithm. As discussed earlier, the algorithm and analysis (for both settings) are also modular enough to allow such extension. One natural extension is to replace the Chernoff-Hoeffding bounds of Eq. 4 with the following Bernstein bounds in order to take into account the variances of the arms:

$$\text{UGapEb-V:} \qquad \beta_k(t) = \sqrt{\frac{2a\ \widehat{\sigma}_k^2(t)}{T_k(t)}} + \frac{(7/3)ab}{T_k(t)-1},$$

$$\text{UGapEc-V:} \qquad \beta_k(t) = \sqrt{\frac{2c \log \frac{Kt^3}{\delta}\ \widehat{\sigma}_k^2(t)}{T_k(t)}} + \frac{(7/3)bc \log \frac{Kt^3}{\delta}}{T_k(t)-1},$$

where $\widehat{\sigma}_k^2(t) = \frac{1}{T_k(t)-1}\sum_{s=1}^{T_k(t)} \left(X_k(s) - \widehat{\mu}_k(t)\right)^2$ is the estimated variance of arm k at the end of round t. We call the resulting algorithm UGapE-variance (UGapE-V). Using Theorem 11 in [12], it is easy to show that Theorems 1 and 2, bounding the simple regret of the fixed budget and fixed confidence settings, still hold (without major change in their proofs) with a new definition of complexity, i.e.,

$$H_\epsilon^\sigma = \sum_{i=1}^{K} \frac{\left(\sigma_i + \sqrt{(13/3)b\Delta_i}\right)^2}{\max\left(\frac{\Delta_i+\epsilon}{2}, \epsilon\right)^2}. \tag{20}$$

This variance-complexity H_ϵ^σ is expected to better capture the complexity of the arms and to be smaller than H_ϵ defined by Eq. 6 whenever the variances of the arms are small compared to the range b of the distribution.

As mentioned in the introduction, Mnih et al. [13] proposed Bernstein Race, a best arm identification algorithm based on the Bernstein inequality for the fixed confidence setting. The term bounding the number of pulls of a sub-optimal arm i in their analysis is of the form $(\sigma_{(1)}^2 + \sigma_i^2)/\Delta_i^2$, where $\sigma_{(1)}^2$ is the variance of the best arm. This causes Bernstein Race to allocate the pulls equally over the arms, when the task is to discriminate between two arms ($K = 2$), while intuitively the arms should be pulled proportionally to their variances. UGapE-V on the other hand is able to handle this case properly (i.e., to pull the arms proportionally to their variances), because its bound on the number of pulls of a sub-optimal arm i is of the form σ_i^2/Δ_i^2 (see the definition of H_ϵ^σ in Eq. 20). In Appendix C.1, we report numerical results showing that UGapE-V has better performance than Bernstein Race when variance of the optimal arm is larger than those of the sub-optimal arms.

Gabillon et al. [7] proposed a similar variance extension in the fixed budget multi-bandit setting studied in their paper, with the price of a separate and tedious proof. The UGapE extension to multiple bandits (see the next paragraph), for the case $\epsilon = 0$, recovers the same variance complexity as in [7] not only with improved numerical constants, but also without a separate proof.

Multi-bandit setting. In this setting the forecaster faces M distinct best arm identification problems. The goal is to return the set of (m_p, ϵ)-optimal arms from the set of available arms A_p for each bandit $p \in \{1, \ldots, M\}$. As discussed in the introduction, this settings has been studied in the fixed budget setting for the case of $m_p = 1,\ p = 1, \ldots, M$ [7, 5] and for arbitrary values of m_p [4]. In this section, we briefly outline how our UGapE algorithm can be extended to the multi-bandit multi-armed problem for both fixed budget and fixed confidence settings.

For UGapE to be applied to the multi-bandit problem we first need to define the index $B_{pk}(t)$ over all bandits p and arms k at time t. For each bandit p and arm k, we define the simple regret as $r_{pk} = \mu_{p(m_p)} - \mu_{pk}$. Similar to the single bandit case, $B_{pk}(t)$ is defined as an upper-bound on the simple regret of the kth arm of bandit p, i.e.,

$$B_{pk}(t) = \max_{i \neq k}^{m_p} U_{pi}(t) - L_{pk}(t).$$

At each time t, SELECT-ARM first finds for each bandit p a set of m_p arms $J_p(t)$ with minimum upper-bound on their simple regrets, i.e., $J_p(t) = \arg\min_{k \in A_p}^{1..m_p} B_{pk}(t)$, and calculates $B_{J_p(t)}(t) = \min_{k \in A_p}^{1..m_p} B_{pk}(t)$, where $B_{J_p(t)}(t)$ is the upper bound on the simple regret of the set of arms $J_p(t)$. It then selects the bandit with maximal upper bound on its simple regret to explore. Finally, it selects the arm to be pulled in the exact same manner as in the single bandit case (see Figure 2).

The analysis of the resulting algorithm is quite similar, up to some new notations, to the one for single bandit. Theorems 1 and 2 will be stated similarly, only with new definition of complexity:

$$H_\epsilon = \sum_{p=1}^{M} H_\epsilon(p) = \sum_{p=1}^{M} \sum_{i=1}^{K} \frac{b^2}{\max(\frac{\Delta_{pi}+\epsilon}{2}, \epsilon)^2},$$
$$H_\epsilon^\sigma = \sum_{p=1}^{M} H_\epsilon^\sigma(p) = \sum_{p=1}^{M} \sum_{i=1}^{K} \frac{\left(\sigma_{pi} + \sqrt{(13/3)b\Delta_{pi}}\right)^2}{\max\left(\frac{\Delta_{pi}+\epsilon}{2}, \epsilon\right)^2},$$

where $H_\epsilon(p)$ and $H_\epsilon^\sigma(p)$ are the complexities of the single bandit problem p, without and with variance, defined by Eqs. 6 and 20. Note that the above complexities resemble those in [7] in the sense that the complexity of the multi-bandit problem is the sum of the single bandit's complexities.

Finally it is important to note that our approach can be easily modified to tackle the problem of finding the set of (m_p, ϵ_p)-optimal arms in each bandit p, where the difference w.r.t. the previous version is that now ϵ depends on p. For this modification, it is sufficient to select the bandit with maximal value of the quantity $B_{J_p(t)}(t) - \epsilon_p$ at time t. This quantity is the difference between the upper-bound on the simple regret of bandit p and its maximum accepted simple regret.

Adaptive UGapEb and UGapEb-V. A drawback of UGapE and UGapE-V in the fixed budget setting is that the exploration parameter a should be tuned according to the complexities H_ϵ and H_ϵ^σ of the problem, which are rarely known in advance. A straightforward solution to this problem is to move to an *adaptive* version of the algorithms by substituting H_ϵ and H_ϵ^σ with suitable estimates, often lower bounds on the true complexities. Such adaptive procedure has been implemented for UCB-E, GapE, and GapE-V algorithms [1, 7]. Similar implementations can be used here.

C Experimental Results

In this section, we compare UGapE and UGapE-V to the state-of-the-art algorithms in the fixed budget and fixed confidence settings. The objective is to verify that designing a meta-algorithm for the two settings does not come at the cost of a worse performance.

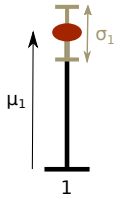

Figure 3: This figure shows how we illustrate the mean and variance of an arm.

C.1 Fixed Confidence Setting

Experimental setting. We define the following two problems, where (x, y) represents a uniform distribution in $[x, y]$.

- *Problem 1.* We have $K = 5$ arms with $\big((0, 1), (0.4, 0.5), (0.4, 0.5), (0.4, 0.5), (0.4, 0.5)\big)$.
- *Problem 2.* We have $K = 5$ arms with $\big((0, 1), (0, 0.8), (0, 0.8), (0, 0.6), (0, 0.6)\big)$.

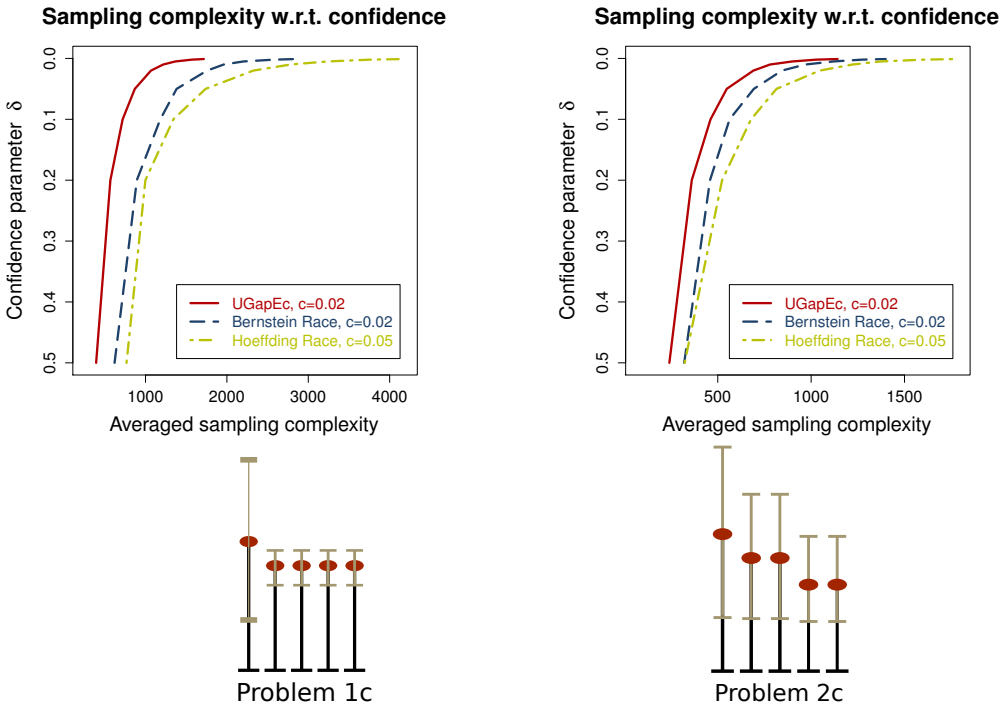

Figure 4: Comparison between UGapEc-V, Bernstein Race, and Hoeffding Race algorithms on Problem 1 *(left)* and Problem 2: *(right)*

We compare UGapEc-V with Bernstein race and Hoeffding race algoerithms. All the algorithms have an exploration parameter c. Although, the theoretical analysis suggests $c = 1/2$, we tuned c empirically. For each algorithm and each confidence parameter δ, we compute the average sample complexity over 1000 runs for different values of $c \in \{1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01, 0.005\}$. For each algorithm, we only consider the results corresponding to the value of c for which the required confidence level δ is satisfied (i.e., the values of c for which the algorithm satisfies $\mathbb{P}[r_{\Omega(\widetilde{n})} > \epsilon =$

$0] \leq \delta$, for all the values of δ considered). Finally in Figure 4, for each algorithm, we report the results for the value of c with the smallest sample complexity.

In the left panel of Figure 4, we report the results for Problem 1 with $m = 1$ and $\epsilon = 0$. In this problem, the optimal arm has significantly higher variance than the other arms. This problem has been designed to highlight the difference between the three algorithms and to illustrate the advantage of UGapEc-V over the Racing algorithms, as discussed in Appendix B. Since the suboptimal arms have the same mean and variance, the analysis of Bernstein Race and Hoeffding Race suggests that all the arms (including the best arm) should be pulled the same number of times. However, since the Bernstein Race takes into account the variance, it has a tighter bound and the stopping condition is met earlier than for Hoeffding Race. For instance for $\delta = 0.1$, Bernstein Race has an expected sample complexity of 1181 pulls while the Hoeffding Race stops after 1342 pulls on average. On the other hand, as expected from the theoretical analysis (see Appendix B), UGapEc-V stops after only 719 pulls. Note that UGapEc-V, on average, distributes the number of pulls as (72%,7%,7%,7%,7%) over the arms. This indicates that the algorithm successfully adapts to the variance of the arms. The parameter c for which the algorithms have a minimal sample complexity are $c = 0.02$ for Bernstein Race and UGapEc-V and $c = 0.05$ for Hoeffding Race.

Finally, in the right panel of Figure 4 we consider Problem 2. Although this problem has not been specifically designed to illustrate the advantage of UGapEc over the Racing algorithms, UGapEc-V still outperforms the Racing algorithms.

C.2 Fixed Budget Setting

In this section, we compare UGapEb with the state-of-the-art fixed budget algorithms: UCBE, UCBE-V, GapE, and GapE-V. Since all these algorithms share a very similar structure, we expect them to have similar performance. All the algorithms have an exploration parameter a. The theoretical analysis suggests that a should be proportional to $\frac{n}{H}$. Although a could be optimized according to the bound, since the constants in the analysis are not accurate, we will run the algorithms with $a = \eta \frac{n}{H}$, where η is a parameter which is empirically tuned (in the experiments we use four different values of η). The results are averaged over 1000 runs and the error bars correspond to three times the estimated standard deviation.

Experimental setting. We use the following two problems in our experiments.

- *Problem 1.* $n = 2000$, $K = 20$. The arms have Bernoulli distribution, the best arm has a mean of $1/2$ and the sub-optimal arms have a mean of 0.4.
- *Problem 2.* $n = 4000$, $K = 15$. The arms have Rademacher distribution with parameters $\mu_i = 0.5 - 0.025(i-1),\ i \in \{1, \dots, 15\}$.

Note that $b = 1$ in these problems. In Figures 5 and 6, we report the performance, calculated as the probability to identify the best arm after n rounds, of UCBE, UCBE-V, GapE, GapE-V, UGapEb, and UGapEb-V algorithms. The results indicate that the best performance of each algorithm is achieved for similar values of the parameter η. As expected, all the algorithms achieve similar performance, no one has clear advantage over the others. Investigating the allocation over the budget n over arms, we also notice that for all the algorithms the number of pulls is inversely proportional to the gaps.

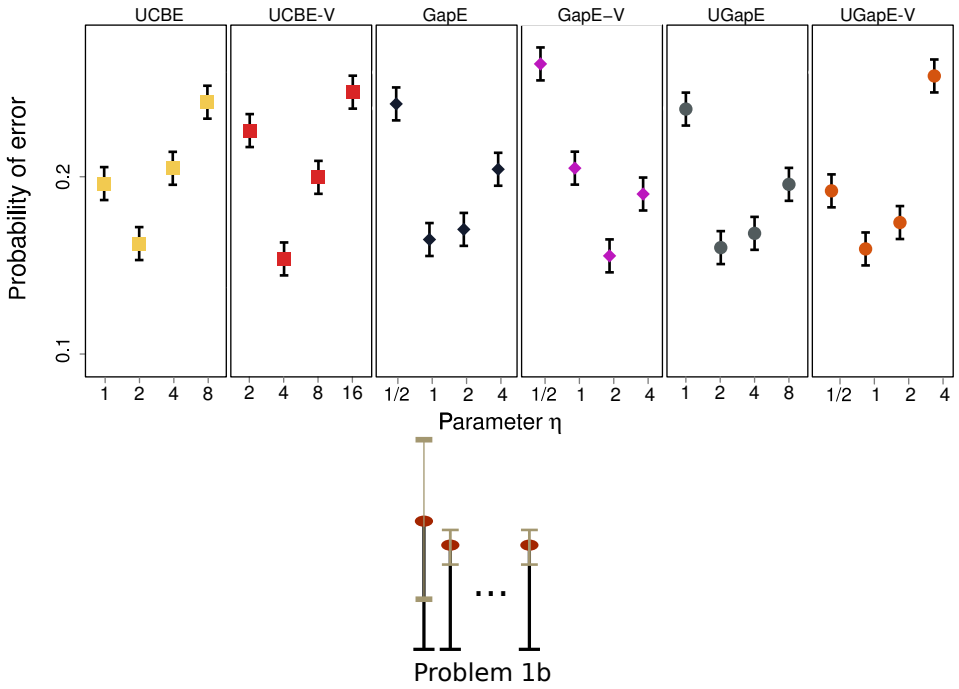

Figure 5: Comparison between UCB-E, UCBE-V, GapE, GapE-V, UGapE, and UGapE-V algorithms in Problem 3.

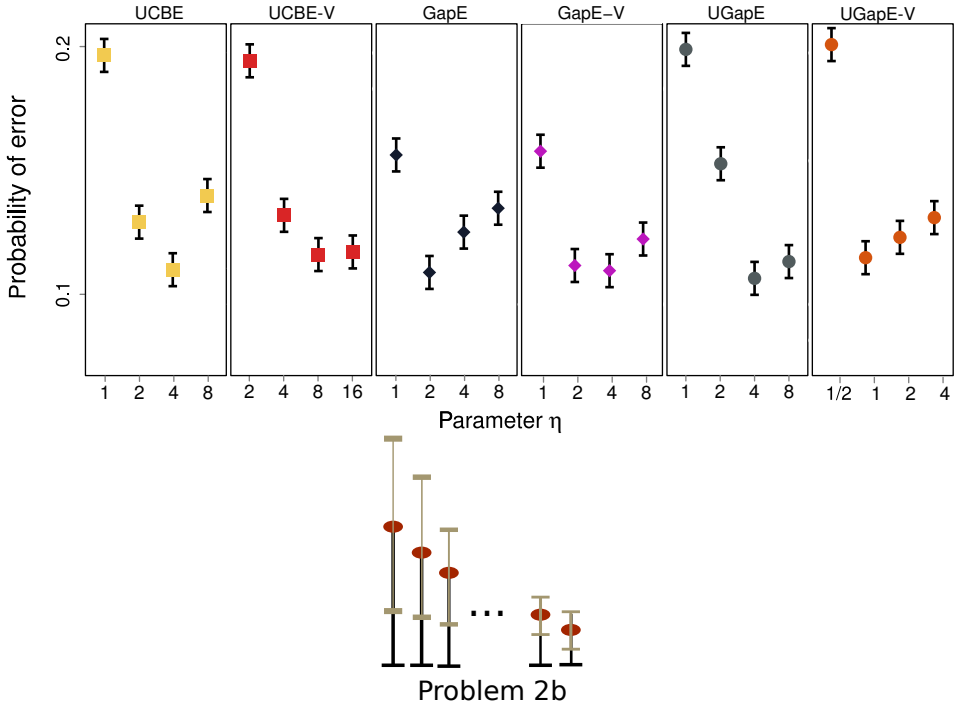

Figure 6: Comparison between UCB-E, UCBE-V, GapE, GapE-V, UGapEb, and UGapEb-V algorithms in Problem 4.