Fixed-Budget Best-Arm Identification with Heterogeneous Reward Variances

Abstract

We study the problem of best-arm identification (BAI) in the fixed-budget setting with heterogeneous reward variances. We propose two variance-adaptive BAI algorithms for this setting: `SHVar` for known reward variances and `SHAdaVar` for unknown reward variances. Our algorithms rely on non-uniform budget allocations among the arms where the arms with higher reward variances are pulled more often than those with lower variances. The main algorithmic novelty is in the design of `SHAdaVar`, which allocates budget greedily based on overestimating the unknown reward variances. We bound probabilities of misidentifying the best arms in both `SHVar` and `SHAdaVar`. Our analyses rely on novel lower bounds on the number of pulls of arms that do not require closed-form solutions to the budget allocation problem. Since one of our budget allocation problems is analogous to the optimal experiment design with unknown variances, we believe that our results are of a broad interest. Our experiments validate our theory, and show that `SHVar` and `SHAdaVar` outperform algorithms from prior works with analytical guarantees.

1 INTRODUCTION

The problem of *best-arm identification (BAI)* in the *fixed-budget* setting is a *pure exploration* bandit problem which can be briefly described as follows. A learner interacts with a stochastic multi-armed bandit with K arms, where the goal is to identify the best arm with the highest mean reward within a fixed budget of at most n arm pulls [Bubeck et al., 2009, Audibert et al., 2010]. This arises naturally in many applications in practice, such as online advertising, recommender systems, and vaccine tests [Lattimore and Szepesvari, 2019]. Furthermore, the fixed-budget BAI setting [Bubeck et al., 2009, Audibert et al., 2010] is common in applications where the observations are costly, such as in Bayesian optimization [Krause et al., 2008]. The BAI problem has also been traditionally studied in the *fixed-confidence* setting [Even-Dar et al., 2006, Soare et al., 2014]. Here the goal is to find the best arm with a given confidence level, while minimizing the budget. Some works have studied both settings [Gabillon et al., 2012, Karnin et al., 2013, Kaufmann et al., 2016].

Consider an A/B test where the goal is to identify the movie with the highest average user rating from a set of K movies. This problem can be formulated as BAI by treating each movie as an arm and the rating given by a user as its stochastic reward. Some movies get either unanimously good or bad ratings, leading to a low variance in the ratings. Others get a wide range of ratings, because they are rated highly by their target audience and poorly by others; leading to a high variance in the ratings. For this setting, we can design better BAI strategies that take the variance of the movie ratings into account. Specifically, we can reduce the number of times that low-variance movies are recommended in the A/B test and allocate more budget to high-variance movies. A more extreme but synthetic example of this scenario is shown in Figure 1. In this example, the reward variance increases with the mean reward for a half of the arms, while the remaining arms have very low variances. We can again use the knowledge of the reward variances to reduce the number of times that low-variance arms are pulled. However, in practice, the reward variances are rarely known in advance, such as in the discussed A/B test, and this makes the design and analysis of variance-adaptive BAI algorithms challenging. We revisit these two examples in our empirical studies in Section 5.

We propose and analyze two variance-adaptive BAI algorithms: `SHVar` and `SHAdaVar`. `SHVar` assumes that the reward variances are known and is a stepping stone for our fully-adaptive BAI algorithm `SHAdaVar`, which estimates them. `SHAdaVar` utilizes upper confidence bounds on the reward variances that hold with a high probability. Both al-

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

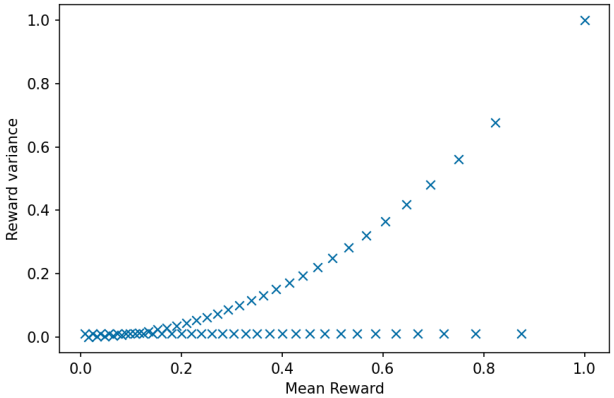

Figure 1: Mean rewards and variances for $K = 64$ arms in the Gaussian bandit problem in Section 5.1.

gorithms are motivated by sequential halving (SH) of Karnin et al. [2013], a near-optimal solution for fixed-budget BAI with homogeneous reward variances.

Our main contributions are:

- We propose two variance-adaptive algorithms for fixed-budget BAI: SHVar for known reward variances and SHAdaVar for unknown reward variances. SHAdaVar is only a third algorithm for this setting [Gabillon et al., 2011, Faella et al., 2020] and only a second that can be implemented as analyzed [Faella et al., 2020]. The key idea in SHAdaVar is to solve a budget allocation problem with unknown reward variances by a greedy algorithm that overestimates them. This idea can be applied to elimination bandit algorithms in the cumulative regret setting [Auer and Ortner, 2010] and is of independent interest to the field of optimal experiment design [Pukelsheim, 1993].
- We prove upper bounds on the probability of misidentifying the best arm for both SHVar and SHAdaVar. The analysis of SHVar extends that of Karnin et al. [2013] to sub-Gaussian reward distributions with heterogeneous variances. The analysis of SHAdaVar relies on a novel lower bound on the number of pulls of an arm that scales linearly with the unknown reward variance of the arm. This permits an analysis of sequential halving without requiring a closed form for the number of pulls of each arm.
- We evaluate the empirical performance of our algorithms on Gaussian bandit problems and the MovieLens dataset [Lam and Herlocker, 2016]. Our results validate our theoretical findings, and show that SHVar and SHAdaVar outperform all algorithms from prior works with error bounds.

The paper is organized as follows. In Section 2, we present the fixed-budget BAI problem. We present our algorithms in Section 3 and analyze them in Section 4. The algorithms are empirically evaluated in Section 5. We review prior works in Section 6 and conclude in Section 7.

2 SETTING

We use the following notation. Random variables are capitalized, except for Greek letters like μ. For any positive integer n, we define $[n] = \{1, \ldots, n\}$. The indicator function is denoted by $\mathbb{1}\{\cdot\}$. The i-th entry of vector v is v_i. If the vector is already indexed, such as v_j, we write $v_{j,i}$. The big O notation up to logarithmic factors is $\tilde{O}$.

We have a stochastic bandit with K arms and denote the set of arms by $\mathcal{A} = [K]$. When the arm is pulled, its reward is sampled from its distribution. The reward of arm $i \in \mathcal{A}$ is sampled from a sub-Gaussian distribution with mean μ_i and variance proxy σ_i^2. The *best arm* is $i_* = \arg\max_{i \in \mathcal{A}} \mu_i$. Without loss of generality, we assume that the mean arm rewards are ordered and that there is only one best arm, meaning that $\mu_1 > \mu_2 \geq \ldots \geq \mu_K$. The learning agent is given a budget of n observations and the goal is to identify i_* as accurately as possible after pulling all arms up to n times. Specifically, let $\hat{I}$ be the arm returned by the agent after n arm pulls. Then our goal is to design an agent that minimizes $\mathbb{P}\left(\hat{I} \neq i_*\right)$ for a bandit instance $\mu = (\mu_i)_{i \in [K]}$. This setting is called the fixed-budget setting [Bubeck et al., 2009, Audibert et al., 2010]. When observations are costly, it is natural to limit their number by a fixed budget.

We study only the fixed-budget setting. Another commonly studied setting is fixed-confidence BAI [Even-Dar et al., 2006, Soare et al., 2014], where the agent is given δ as an input and the goal is to guarantee $\mathbb{P}\left(\hat{I} \neq i_*\right) \leq \delta$ while minimizing budget n. Some works have studied both the fixed-budget and fixed-confidence settings [Gabillon et al., 2012, Karnin et al., 2013, Kaufmann et al., 2016].

3 ALGORITHMS

A near-optimal solution for fixed-budget BAI is sequential halving [Karnin et al., 2013]. The key idea is to sequentially eliminate suboptimal arms in $\log_2 K$ stages. In each stage, all arms are pulled uniformly and the worst half of the arms are eliminated at the end of the stage. At the end of the last stage, only one arm $\hat{I}$ remains and that arm is returned as the estimated best arm.

The main algorithmic contribution of our work is that we generalize sequential halving of Karnin et al. [2013] to heterogeneous reward variances. All of our proposed algorithms can be viewed as instances of a meta-algorithm (Algorithm 1), which we present next. The input to the meta-algorithm is a *budget* on the number of observations n. The meta-algorithm has m stages (line 2) and the budget is divided equally across the stages, with a *per-stage bud-*

Algorithm 1 Meta-algorithm for sequential halving.

1: **Input:** Budget n, base algorithm `Alg`

2: Number of stages $m \leftarrow \lceil \log_2 K \rceil$
3: $\mathcal{A}_1 \leftarrow \mathcal{A}$
4: **for** $s = 1, \ldots, m$ **do**
5: Per-stage budget $n_s \leftarrow \lfloor n/m \rfloor$
6: **for** $t = 1, \ldots, n_s$ **do**
7: $I_{s,t} \leftarrow \texttt{Alg(s, t)}$
8: Observe reward of arm $I_{s,t}$, $Y_{s,t,I_{s,t}}$
9: **for** $i \in \mathcal{A}_s$ **do**
10: $N_{s,i} \leftarrow \sum_{t=1}^{n_s} \mathbb{1}\{I_{s,t} = i\}$
11: $\hat{\mu}_{s,i} \leftarrow \dfrac{1}{N_{s,i}} \displaystyle\sum_{t=1}^{n_s} \mathbb{1}\{I_{s,t} = i\}\, Y_{s,t,i}$
12: $\mathcal{A}_{s+1} \leftarrow \{\lceil |\mathcal{A}_s| / 2 \rceil$ arms $i \in \mathcal{A}_s$ with highest $\hat{\mu}_{s,i}\}$

13: **Output:** The last remaining arm $\hat{I}$ in $\mathcal{A}_{m+1}$

Algorithm 2 `SH`: Pulled arm in sequential halving.

1: **Input:** Stage s, round t

2: $k \leftarrow (t-1) \bmod |\mathcal{A}_s| + 1$
3: $I_{s,t} \leftarrow k$-th arm in $\mathcal{A}_s$

4: **Output:** Arm to pull $I_{s,t}$

get $n_s = \lfloor n/m \rfloor$ (line 5). In stage s, all *remaining arms* $\mathcal{A}_s$ are pulled (lines 6–8). At the end of stage s, the worst half of the remaining arms, as measured by their estimated mean rewards, is eliminated (lines 9–12). Here $Y_{s,t,i}$ is the *stochastic reward* of arm i in round t of stage s, $I_{s,t} \in \mathcal{A}_s$ is the *pulled arm* in round t of stage s, $N_{s,i}$ is the *number of pulls* of arm i in stage s, and $\hat{\mu}_{s,i}$ is its *mean reward estimate* from all observations in stage s.

The sequential halving of Karnin et al. [2013] is an instance of Algorithm 1 for `Alg` = `SH`. The pseudocode of `SH`, which pulls all arms in stage s in a round robin, is in Algorithm 2. We call the resulting algorithm `SH`. Based on Karnin et al. [2013], `SH` misidentifies the best arm with probability

$$\mathbb{P}\left(\hat{I} \neq 1\right) \leq 3 \log_2 K \exp\left[-\frac{n}{8 H_2 \log_2 K}\right], \quad (1)$$

where

$$H_2 = \max_{i \in \mathcal{A} \setminus \{1\}} i / \Delta_i^2 \quad (2)$$

is the so-called *complexity parameter*. The key observation is that the probability decreases as the budget n increases and the difficulty of the problem H_2 decreases.

`SH` is near optimal only in the setting of homogeneous reward variances. In this work, we study the general setting

Algorithm 3 `SHVar`: Pulled arm in sequential halving with known heterogeneous reward variances.

1: **Input:** Stage s, round t

2: $\forall i \in \mathcal{A}_s : N_{s,t,i} \leftarrow \sum_{\ell=1}^{t-1} \mathbb{1}\{I_{s,\ell} = i\}$
3: $I_{s,t} \leftarrow \arg\max_{i \in \mathcal{A}_s} \dfrac{\sigma_i^2}{N_{s,t,i}}$
4: **Output:** Arm to pull $I_{s,t}$

where the reward variances of arms vary, potentially as extremely as in our motivating example in Figure 1. In this example, `SH` would face arms with both low and high variances in each stage. A variance-adaptive `SH` could adapt its budget allocation in each stage based on the variances and thus eliminate suboptimal arms with a higher confidence.

3.1 KNOWN HETEROGENEOUS REWARD VARIANCES

We start with the setting of known reward variances. Let

$$\sigma_i^2 = \operatorname{var}\left[Y_{s,t,i}\right] = \mathbb{E}\left[(Y_{s,t,i} - \mu_i)^2\right] \quad (3)$$

be a known reward variance of arm i. Our proposed algorithm is an instance of Algorithm 1 for `Alg` = `SHVar`. The pseudocode of `SHVar` is in Algorithm 3. The key idea is to pull the arm with the highest variance of its mean reward estimate. Specifically, in round t of stage s, the variance of the mean reward estimate of arm i is $\sigma_i^2 / N_{s,t,i}$, where σ_i^2 is the reward variance of arm i and $N_{s,t,i}$ is the number of pulls of arm i up to round t of stage s. We call the resulting algorithm `SHVar`.

Note that `SH` is an instance of `SHVar`. Specifically, when all $\sigma_i = \sigma$ for some $\sigma > 0$, `SHVar` pulls arms in a round robin, exactly as `SH`. `SHVar` can be also viewed as an algorithm that pulls arm i in stage s for

$$N_{s,i} = \frac{\sigma_i^2}{\sum_{i \in \mathcal{A}_s} \sigma_i^2} n_s \quad (4)$$

times. This is stated formally and proved below.

Lemma 1. *Fix stage s and let*

$$\lambda_{s,i} = \frac{\sigma_i^2}{\sum_{j \in \mathcal{A}_s} \sigma_j^2} n_s$$

be an integer for all arms $i \in \mathcal{A}_s$. Then `SHVar` *pulls arm i in stage s exactly $\lambda_{s,i}$ times.*

Proof. First, suppose that `SHVar` pulls each arm i exactly $\lambda_{s,i}$ times. Then the variances of all mean reward estimates at the end of stage s are identical, because

$$\frac{\sigma_i^2}{N_{s,i}} = \frac{\sigma_i^2}{\lambda_{s,i}} = \frac{\sigma_i^2}{\frac{\sigma_i^2}{\sum_{j \in \mathcal{A}_s} \sigma_j^2} n_s} = \frac{\sum_{j \in \mathcal{A}_s} \sigma_j^2}{n_s}.$$

Now suppose that this is not true. Then there must exist arms $i, k \in \mathcal{A}_s$ such that

$$\frac{\sigma_i^2}{N_{s,i}} < \frac{\sum_{j \in \mathcal{A}_s} \sigma_j^2}{n_s} < \frac{\sigma_k^2}{N_{s,k}} \,. \tag{5}$$

In fact, let $t \in [n_s]$ be the round where arm i is pulled the last time. Then it must be true that

$$\frac{\sigma_i^2}{N_{s,i}} < \frac{\sigma_i^2}{N_{s,t,i}} \leq \frac{\sum_{j \in \mathcal{A}_s} \sigma_j^2}{n_s} < \frac{\sigma_k^2}{N_{s,k}} \leq \frac{\sigma_k^2}{N_{s,t,k}} \,.$$

The first inequality follows from $N_{s,i} = N_{s,t,i} + 1$, because round t is the last round where arm i is pulled. The second inequality follows from (5) and that the optimal allocation are integers. The third inequality follows from (5) and the last holds because $N_{s,k} \geq N_{s,t,k}$.

Note that the above leads to a contradiction, since `SHVar` cannot pull arm i in round t if $\sigma_i^2/N_{s,t,i} < \sigma_k^2/N_{s,t,k}$. This completes the proof. □

Lemma 1 says that each arm $i \in \mathcal{A}_s$ is pulled $O(\sigma_i^2)$ times. Since the mean reward estimate of arm i at the end of stage s has variance $\sigma_i^2/N_{s,i}$, the variances of all estimates at the end of stage s are identical, $\left(\sum_{i \in \mathcal{A}_s} \sigma_i^2\right)/n_s$. This relates our problem to the G-optimal design [Pukelsheim, 1993]. In the G-optimal design, the goal is to allocate observations to minimize the maximum variance in the estimated values of K experiments. In our setting, the arms are experiments and the optimal allocation is given in Lemma 1. Because the variances of all estimates are the same, the arm elimination at the end of stage s can be done based on the mean reward estimates only.

3.2 UNKNOWN HETEROGENEOUS REWARD VARIANCES

Our second proposal is an algorithm for unknown reward variances. One natural idea, which is expected to be practical but hard to analyze, is to replace σ_i^2 in `SHVar` with its empirical estimate from the past $t-1$ rounds in stage s,

$$\hat{\sigma}_{s,t,i}^2 = \frac{1}{N_{s,t,i} - 1} \sum_{\ell=1}^{t-1} \mathbb{1}\{I_{s,\ell} = i\} \left(Y_{s,\ell,i} - \hat{\mu}_{s,t,i}\right)^2 ,$$

where

$$\hat{\mu}_{s,t,i} = \frac{1}{N_{s,t,i}} \sum_{\ell=1}^{t-1} \mathbb{1}\{I_{s,\ell} = i\} \, Y_{s,\ell,i}$$

is the empirical estimate of the mean reward of arm i in round t of stage s. The difficulty in the analysis stems from the fact that $\hat{\sigma}_{s,t,i}$ can underestimate σ_i and thus does not lead to an optimistic design.

The key idea in our solution is to act optimistically using an *upper confidence bound (UCB)* on the reward variance.

Algorithm 4 `SHAdaVar`: Pulled arm in sequential halving with unknown heterogeneous reward variances.

Input: Stage s, round t

if $t \leq |\mathcal{A}_s| \left(4 \log(1/\delta) + 1\right)$ **then**
  $k \leftarrow (t-1) \bmod |\mathcal{A}_s| + 1$
  $I_{s,t} \leftarrow k$-th arm in $\mathcal{A}_s$
else
  $\forall i \in \mathcal{A}_s : N_{s,t,i} \leftarrow \sum_{\ell=1}^{t-1} \mathbb{1}\{I_{s,\ell} = i\}$
  $I_{s,t} \leftarrow \arg\max_{i \in \mathcal{A}_s} \dfrac{U_{s,t,i}}{N_{s,t,i}}$

Output: Arm to pull $I_{s,t}$

To derive the upper bound, we make an assumption that the reward noise is Gaussian. Specifically, for any arm i, and round t in stage s, we assume that $Y_{s,t,i} \sim \mathcal{N}(\mu_i, \sigma_i^2)$. This allows us to derive the following upper and lower bounds on the unkown variance σ_i^2.

Lemma 2. *Fix stage s, round $t \in [n_s]$, arm $i \in \mathcal{A}_s$, and failure probability $\delta \in (0,1)$. Let $m = N_{s,t,i} - 1$ and suppose that $m > 4\log(1/\delta)$. Then*

$$\mathbb{P}\left(\sigma_i^2 \geq \frac{\hat{\sigma}_{s,t,i}^2}{1 - 2\sqrt{\frac{\log(1/\delta)}{m}}}\right) \leq \delta$$

holds with probability at least $1-\delta$. Analogously,

$$\mathbb{P}\left(\hat{\sigma}_{s,t,i}^2 \geq \sigma_i^2 \left[1 + 2\sqrt{\frac{\log(1/\delta)}{m}} + \frac{2\log(1/\delta)}{m}\right]\right) \leq \delta$$

holds with probability at least $1-\delta$.

Proof. The first claim is proved as follows. By Cochran's theorem, $\hat{\sigma}_{s,t,i}^2 m/\sigma_i^2$ is a χ^2 random variable with m degrees of freedom. The concentration of this variable was analyzed in Laurent and Massart [2000]. Specifically, by (4.4) in Laurent and Massart [2000], an immediate corollary of their Lemma 1, we have

$$\mathbb{P}\left(m - \frac{\hat{\sigma}_{s,t,i}^2 m}{\sigma_i^2} \geq 2\sqrt{m \log(1/\delta)}\right) \leq \delta \,.$$

Now we divide both sides in the probability by m, multiply by σ_i^2, and rearrange the formula as

$$\mathbb{P}\left(\sigma_i^2 \left(1 - 2\sqrt{\log(1/\delta)/m}\right) \geq \hat{\sigma}_{s,t,i}^2\right) \leq \delta \,.$$

Finally, the first claim in Lemma 2 follows from dividing by $1 - 2\sqrt{\log(1/\delta)/m}$, under the assumption that this quantity is positive.

The second claim is proved analogously. Specifically, by (4.3) in Laurent and Massart [2000], an immediate corollary of their Lemma 1, we have

$$\mathbb{P}\left(\frac{\hat{\sigma}^2_{s,t,i} m}{\sigma_i^2} - m \geq 2\sqrt{m\log(1/\delta)} + 2\log(1/\delta)\right) \leq \delta\,.$$

Now we divide both sides in the probability by m, multiply by σ_i^2, and get the second claim in Lemma 2. This concludes the proof. □

Based on Lemma 2, when $N_{s,t,i} > 4\log(1/\delta) + 1$,

$$U_{s,t,i} = \frac{\hat{\sigma}^2_{s,t,i}}{1 - 2\sqrt{\frac{\log(1/\delta)}{N_{s,t,i}-1}}} \tag{6}$$

is an upper bound on variance σ_i^2 that holds with probability at least $1-\delta$, for a given stage s, round t, and arm i. This bound decreases as the number of observations $N_{s,t,i}$ increases and confidence δ decreases. To apply the bound across multiple stages, rounds, or arms, the union bound is needed (Section 4.2).

The bound in (6) leads to our algorithm that overestimates the variance. The algorithm is an instance of Algorithm 1 for `Alg` = `SHAdaVar`. The pseudocode of `SHAdaVar` is in Algorithm 4. To guarantee $N_{s,t,i} > 4\log(1/\delta) + 1$, we pull all arms $\mathcal{A}_s$ in any stage s for $4\log(1/\delta) + 1$ times initially. We call the resulting algorithm `SHAdaVar`.

Note that `SHAdaVar` can be viewed as a variant of `SHVar` where $U_{s,t,i}$ replaces σ_i^2. As a result, it can also be viewed as a solution to a variant of the G-optimal design with unknown observation variances. Therefore, we believe that `SHAdaVar` is of a broader interest to the optimal experiment design community [Pukelsheim, 1993]. We also note that the assumption of Gaussian noise, which is used in the design of `SHAdaVar`, is limiting. In Section 5.2, we conduct experiments with a non-Gaussian noise to assess the robustness of `SHAdaVar` to model misspecification.

4 ANALYSIS

This section has three parts. In Section 4.1, we bound the probability that `SHVar`, which knows the variances of arm rewards, misidentifies the best arm. In Section 4.2, we provide an alternative proof of this bound that does not rely on the closed form in (4). Finally, in Section 4.3, we bound the probability that `SHAdaVar`, which learns the variances of arm rewards, misidentifies the best arm.

4.1 ERROR BOUND OF `SHVar`

We start with analyzing `SHVar`, which is a stepping stone for analyzing `SHAdaVar`. To simplify the proof, we assume that m and n_s are integers. We also assume that all budget allocations have integral solutions in Lemma 1.

Theorem 3. `SHVar` *misidentifies the best arm with probability*

$$\mathbb{P}\left(\hat{I} \neq 1\right) \leq 2\log_2 K \exp\left[-\frac{n\Delta^2_{\min}}{4\log_2 K \sum_{j\in\mathcal{A}} \sigma_j^2}\right],$$

where $\Delta_{\min} = \mu_1 - \mu_2$ *is the minimum gap.*

Proof. The claim is proved in Appendix A. The proof follows the outline in Karnin et al. [2013]. The novelty is in extending the proof to sub-Gaussian reward distributions with heterogeneous reward variances. These require a non-uniform budget allocation, where the arms with higher reward variances are pulled more than those with lower variances (Lemma 1). □

The bound in Theorem 3 scales with all quantities of interest as expected. It decreases as the budget n and minimum gap $\Delta_{\min}$ increase, and the number of arms K and variances σ_j^2 decrease. `SHVar` reduces to `SH` in Karnin et al. [2013] when $\sigma_i^2 = 1/4$ for all arms $i \in \mathcal{A}$. Our error bound becomes comparable when we replace $\sum_{j\in\mathcal{A}} \sigma_j^2$ in Theorem 3 with $K/4$ and apply $H_2 \leq K/\Delta^2_{\min}$ in (1). The additional factor of 8 in the exponent of (1) is due to a slightly different proof, which yields a finer dependence on gaps than in our bound.

4.2 ALTERNATIVE ERROR BOUND OF `SHVar`

Now we provide an alternative analysis of `SHVar`. While the resulting bound is weaker, the proof technique can be easily extended to `SHAdaVar`.

Theorem 4. `SHVar` *misidentifies the best arm with probability*

$$\mathbb{P}\left(\hat{I} \neq 1\right) \leq 2\log_2 K \exp\left[-\frac{(n - K\log K)\Delta^2_{\min}}{4\sigma^2_{\max} K \log_2 K}\right].$$

where $\Delta_{\min} = \mu_1 - \mu_2$ *is the minimum gap and* $\sigma^2_{\max} = \max_{i\in\mathcal{A}} \sigma_i^2$ *is the maximum reward variance.*

Proof. The claim is proved in Appendix B. The key idea in the proof is to derive a lower bound on the number of pulls of any arm i in stage s, instead of using the closed form of $N_{s,i}$ in (4). The lower bound is

$$N_{s,i} \geq \frac{\sigma_i^2}{\sigma^2_{\max}}\left(\frac{n_s}{|\mathcal{A}_s|} - 1\right).$$

An important property of the bound is that it is $\Omega(\sigma_i^2 n_s)$, similarly to $N_{s,i}$ in (4). Therefore, the rest of the proof can be conducted similarly to that of Theorem 3. □

As in Theorem 3, the bound in Theorem 4 scales with all quantities of interest as expected. It decreases as the budget n and minimum gap $\Delta_{\min}$ increase, and the number of arms K and maximum variance $\sigma^2_{\max}$ decrease. When all reward variances are the same, the bound reduces to that in Theorem 3.

4.3 ERROR BOUND OF SHAdaVar

Now we analyze SHAdaVar under the assumption of Gaussian reward noise. Specifically, for any arm i, and round t in stage s, we assume that $Y_{s,t,i} \sim \mathcal{N}(\mu_i, \sigma_i^2)$.

Theorem 5. *Suppose that $\delta < 1/(Kn)$ and*

$$n \geq K \log_2 K (4 \log(Kn/\delta) + 1)\,.$$

Then SHAdaVar *misidentifies the best arm with probability*

$$\mathbb{P}\left(\hat{I} \neq 1\right) \leq 2 \log_2 K \exp\left[-\alpha \frac{(n - K \log K)\Delta^2_{\min}}{4\sigma^2_{\max} K \log_2 K}\right],$$

where $\Delta_{\min}$ and $\sigma^2_{\max}$ are defined in Theorem 4, and

$$\alpha = \frac{1 - 2\sqrt{\frac{\log(Kn/\delta)}{n/K-2}}}{1 + 2\sqrt{\frac{\log(Kn/\delta)}{n/K-2}} + \frac{2\log(Kn/\delta)}{n/K-2}}\,.$$

Proof. The claim is proved in Appendix C. The key idea in the proof is to derive a lower bound on the number of pulls of any arm i in stage s, similarly to that in Theorem 4. The lower bound is

$$N_{s,i} \geq \frac{\sigma_i^2}{\sigma^2_{\max}} \alpha(|\mathcal{A}_s|, n_s, \delta) \left(\frac{n_s}{|\mathcal{A}_s|} - 1\right)$$

and holds with probability at least $1 - \delta$. Since the bound is $\Omega(\sigma_i^2 n_s)$, as in the proof of Theorem 4, the rest of the proof can be conducted similarly. The only major difference from Theorem 4 is in factor $\alpha(|\mathcal{A}_s|, n_s, \delta)$, which converges to 1 as $n_s \to \infty$. □

Similarly to Theorem 4, the bound scales with all quantities of interest as expected. It decreases as the budget n and minimum gap $\Delta_{\min}$ increase, and the number of arms K and maximum variance $\sigma^2_{\max}$ decrease. As $n \to \infty$, we note that $\alpha \to 1$. In this case, the bound has the same shape as that in Theorem 4.

5 EXPERIMENTS

In this section, we empirically evaluate our proposed algorithms, SHVar and SHAdaVar, and compare them to algorithms from prior works. We choose the following baselines: uniform allocation (Unif), sequential halving (SH) [Karnin et al., 2013], gap-based exploration (GapE) [Gabillon et al., 2011], gap-based exploration with variance (GapE-V) [Gabillon et al., 2011], and variance-based rejects (VBR) [Faella et al., 2020]. Unif is a naive budget allocation strategy that allocates equal budget to all arms. SH was originally proposed and analyzed for homogeneous reward variances and it is a special case of SHVar. Neither Unif nor SH adapt to heterogenuous reward variances. GapE, GapE-V and VBR are variance-adaptive BAI algorithms and the closed related works (Section 6).

Both GapE and GapE-V assume that the reward distributions have a bounded support of $[0, b]$ and b is their input. When we apply these algorithms to Gaussian reward distributions, we set $b = \max_{i \in \mathcal{A}} \mu_i + \sigma_i \sqrt{\log n}$, which is a high-probability upper bound on the absolute value of n independent observations from the reward distributions. Both GapE and GapE-V have $O(\exp[-cn/H])$ error bounds on the probability of misidentifying the best arm, where n is the budget, H is the complexity parameter, and $c = 1/144$ for GapE and $c = 1/512$ for GapE-V. On the other hand, our bounds are $O(\exp[-c'n/H'])$, where H' is another complexity parameter and $c' = 1/(4 \log_2 K)$. Note that $c \ll c'$ even for moderately large K. To run GapE and GapE-V with a comparable error bound to our methods, we replace their H with Hc'/c. Since the complexity parameter is an input to both GapE and GapE-V, note that these methods have an advantage over our methods because they have access to additional information about the problem. In SHAdaVar, we choose $\delta = 1/n$ to have a high-probability upper bound on all variances over n rounds. All reported results are averaged over 5 000 runs.

5.1 SYNTHETIC EXPERIMENTS

We start with a K-armed Gaussian bandit where the mean reward of arm i is $\mu_i = 1 - \sqrt{(i-1)/K}$. We choose this setting since sequential elimination with halving is known to perform well in it. Specifically, note that the complexity parameter H_2 in (2) is minimized when i/Δ_i^2 is the same for all $i \in \mathcal{A} \setminus \{1\}$. For our μ_i, $\Delta_i^2 = (i-1)/K \approx i/K$ and thus $i/\Delta_i^2 \approx K$. Moreover, we set $\sigma_i = \mu_i$ when arm i is even and $\sigma_i = 0.1$ when arm i is odd. We visualize the mean rewards μ_i and the corresponding variances σ_i^2, for $K = 64$ arms, in Figure 1. The variances are chosen so that every stage of sequential halving contains arms with both high and low variances. This means that, at every stage, the algorithm that adapts its budget allocation to the variances of the remaining arms eliminates the best arm with a lower probability than the algorithm that does not.

In Figure 2, we report the probability of misidentifying the best arm among $K = 64$ arms (Figure 1) as a function of increasing budget n for all algorithms. As expected, the naive algorithm Unif performs the worst. The next two algorithms are GapE and GapE-V. When the algorithms are parameterized to have comparable error bounds to SHVar and

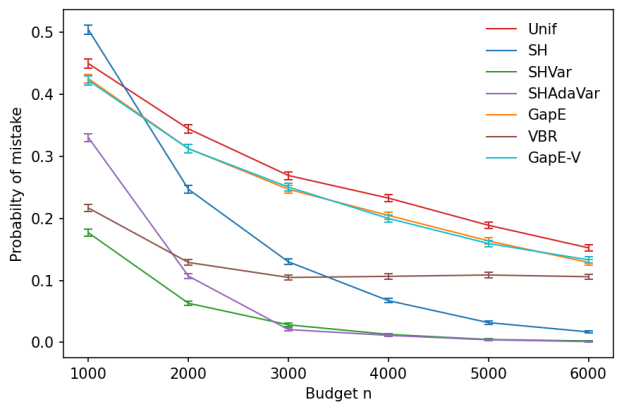

Figure 2: Probability of misidentifying the best arm in the Gaussian bandit problem in Figure 1, as a function of increasing budget n. The number of arms is $K = 64$ and the results are averaged over 5 000 runs.

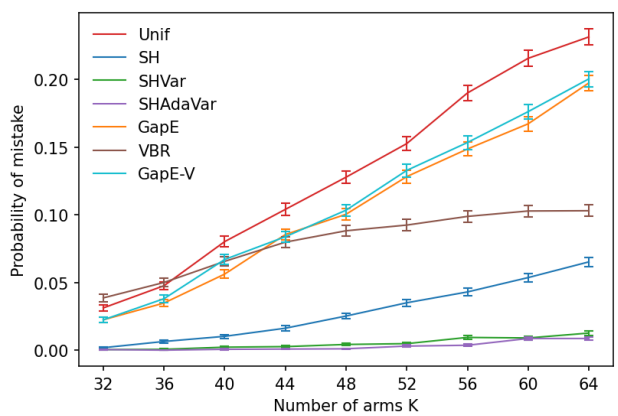

Figure 3: Probability of misidentifying the best arm in the Gaussian bandit problem in Figure 1, as a function of the number of arms K. The budget is fixed at $n = 4\,000$ and the results are averaged over 5 000 runs.

`SHAdaVar`, their confidence intervals are too wide and the algorithms perform only slightly better than uniform allocation. `SHVar` and `SHAdaVar` outperform all other algorithms while not needing any additional information about the problem instance. We note that the probability of misidentifying the best arm in `SHAdaVar` approaches that of `SHVar`, which knows the reward variances, as the budget n increases. This shows that our variance upper bounds improve quickly with a larger budget, as is expected based on the algebraic form in (6). Moreover, the upper bounds are very practical, to the point that `SHAdaVar` can perform as well as `SHVar` when the budget is large.

In the next experiment, we have the same Gaussian bandit as in Figure 2. However, the budget is fixed at $n = 4\,000$, and the number of arms K varies from 32 to 64. In Figure 3, we report the probability of misidentifying the best arm as a function of the number of arms K. We observe three major trends. First, the relative order of the algorithms, as measured by the probability of misidentifying the best arm, is similar to Figure 2. Second, all algorithms perform worse as the number of arms K increases because the problem instance becomes harder. Finally, the rate at which the probability of misidentifying the best arm increases is the lowest for `SHVar` and `SHAdaVar`. This shows that `SHVar` and `SHAdaVar` can perform well for a wide range of K, unlike the other algorithms. Specifically, both `SHVar` and `SHAdaVar` have low probabilities of misidentifying the best arm for any K from 32 to 64. On the other hand, the other algorithms perform well at $K = 32$, the probability of a mistake is below 0.05; and perform poorly at $K = 64$, the probability of a mistake is above 0.05.

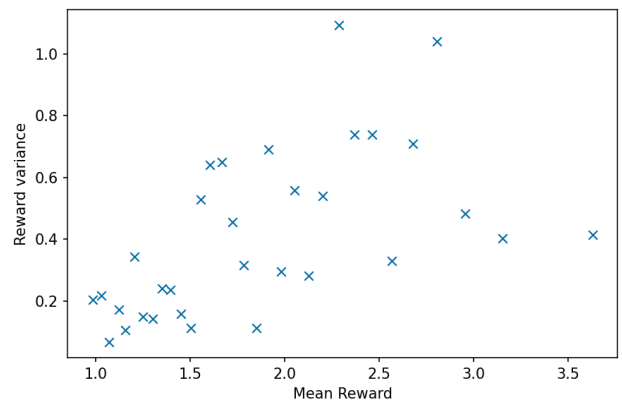

Figure 4: Mean ratings and variances for $K = 32$ movies in the MovieLens dataset. This sample is generated freshly in each run of the experiment, as described in Section 5.2.

5.2 MOVIELENS EXPERIMENTS

Our next experiment is motivated by A/B testing. The goal is to identify the movie with the highest mean rating from a pool of K movies, where each movie represents an arm and its rating by a user is the stochastic reward of the arm. The movies, users, and ratings are simulated using the MovieLens 1M dataset [Lam and Herlocker, 2016]. This dataset contains one million ratings given by 6 040 users to 3 952 movies. We complete the missing ratings using low-rank matrix factorization with rank 5, which is done using alternating least squares [Davenport and Romberg, 2016]. The result is a $6\,040 \times 3\,952$ matrix M where $M_{i,j}$ is the rating given by user i to movie j.

We experiment with $K = 32$ movies. The probability of misidentifying the best movie is averaged over 5 000 runs. In each run, we randomly select new movies according to the following procedure. All movie ratings are normalized to $[0, 1]$. For each movie i, we compute its mean rating over all users, $\frac{1}{6040}\sum_{j=1}^{6040} M_{j,i}$. The best movie is chosen

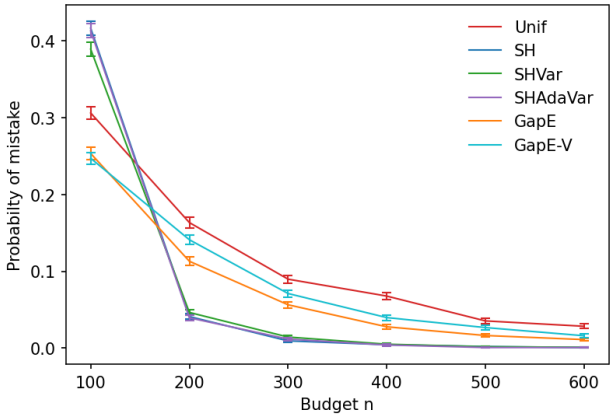

Figure 5: Probability of misidentifying the best movie in the MovieLens bandit problem in Section 5.2, as a function of increasing budget n. The number of movies is $K = 32$ and the results are averaged over $5\,000$ runs.

randomly from movies with a mean rating of at least 0.8. The i-th movie, for $i \in \mathcal{A} \setminus \{1\}$, is chosen randomly from movies with a mean rating of at most $\mu_1 - 0.6\sqrt{(i-1)/K}$, where μ_1 is the mean rating of the best movie. Figure 4 shows the mean and variance of user ratings for $K = 32$ movies selected in one run. As in Section 5.1, the movies are chosen so that sequential elimination with halving is expected to perform well. The variance of movie ratings in Figure 4 is intrinsic to our domain: many movies are made for specific audiences and thus can have a huge variance in their ratings. For instance, a child may not like a horror movie, while a horror enthusiast would enjoy it. For this reason, the algorithm that adapts its budget allocation to the variances of the remaining movies eliminates the best movie with a lower probability than the algorithm that does not. The last notable difference from Section 5.1 is that the movie rating distributions are realistic and not Gaussian. Therefore, this experiment showcases the robustness of our algorithms to model misspecification.

Figure 5 shows the probability of misidentifying the best movie from $K = 32$ movies as a function of increasing budget n for all algorithms. When the budget is small, all the baselines perform better. When the budget is larger, `SHVar` and `SHAdaVar` perform the best, although the reward distributions are not Gaussian. The relative performance of the baselines is similar to Section 5.1: `Unif` is the worst, and `GapE-V` and `GapE` improve upon `Unif`. The only exception is that `VBR` performs poorly and thus we do not plot it.

6 RELATED WORK

Best-arm identification has a long history, in both fixed-budget [Bubeck et al., 2009, Audibert et al., 2010] and fixed-confidence settings [Even-Dar et al., 2006]. The first work on variance-adaptive BAI was in the fixed-budget setting [Gabillon et al., 2011]. This paper proposed an algorithm `GapE-V` and showed that its error probability decreases exponentially with budget n when the complexity parameter is known and used by `GapE-V`. The complexity parameter is rarely known in practice, because it depends on unknown gaps and reward variances of the arms. To address this, Gabillon et al. [2011] introduced adaptive variants of their algorithms, `A-GapE` and `A-GapE-V`, where the complexity parameter is estimated. These algorithms do not come with regret bounds. The only other work that studied variance-adaptive fixed-budget BAI is Faella et al. [2020]. This paper proposed and analyzed a variant of successive rejects algorithm [Audibert et al., 2010]. All other works on variance-adaptive BAI are in the fixed-confidence setting [Lu et al., 2021, Zhou and Tian, 2022, Jourdan et al., 2022]. We propose variance-adaptive sequential halving for fixed-budget BAI. Our algorithms have state-of-the-art performance on our benchmarks (Section 5). They are conceptually simpler than prior works [Gabillon et al., 2011, Faella et al., 2020] because they mostly re-estimate the variance. Unlike Gabillon et al. [2011], they can be implemented as analyzed.

7 CONCLUSIONS

We study the problem of best-arm identification in the fixed-budget setting where the reward variances can vary across the arms. We propose two variance-adaptive elimination algorithms for this problem: `SHVar` for known reward variances and `SHAdaVar` for unknown reward variances. Both algorithms proceed in stages and pull arms with higher reward variances more often than those with lower variances. While the design and analysis of `SHVar` are interesting, they are mostly a stepping stone for `SHAdaVar`, which adapts to the unknown reward variances. The novelty in the design of `SHAdaVar` is that is solves an optimal experiment design problem without knowing the variances of observations. Its analysis relies on a novel lower bound on the number of pulls of arms that does not require closed-form solutions to the budget allocation problem. Our numerical simulations show that `SHVar` and `SHAdaVar` are not only theoretically sound, but also competitive with state-of-the-art baselines.

Our work leaves open several questions of interest. First, the design of `SHAdaVar` is for Gaussian reward noise. To alleviate this issue, we conduct experiments with a non-Gaussian noise. Second, while our error bounds have the expected scaling in all parameters of interest, we do not prove a matching lower bound. When the reward variances are known, we believe that a lower bound can be proved by building on the work of Carpentier and Locatelli [2016]. Finally, our algorithms are not contextual, which limits their application to practical problems.

References

Jean-Yves Audibert, Sebastien Bubeck, and Remi Munos. Best arm identification in multi-armed bandits. In *Proceeding of the 23rd Annual Conference on Learning Theory*, pages 41–53, 2010.

Peter Auer and Ronald Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

Stephane Boucheron, Gabor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Sebastien Bubeck, Remi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory*, pages 23–37, 2009.

Alexandra Carpentier and Andrea Locatelli. Tight (lower) bounds for the fixed budget best arm identification bandit problem. In *Proceeding of the 29th Annual Conference on Learning Theory*, pages 590–604, 2016.

Mark Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4): 608–622, 2016.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

Marco Faella, Alberto Finzi, and Luigi Sauro. Rapidly finding the best arm using variance. In *Proceedings of the 24th European Conference on Artificial Intelligence*, 2020.

Victor Gabillon, Mohammad Ghavamzadeh, Alessandro Lazaric, and Sebastien Bubeck. Multi-bandit best arm identification. In *Advances in Neural Information Processing Systems 24*, 2011.

Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in Neural Information Processing Systems*, pages 3212–3220, 2012.

Marc Jourdan, Remy Degenne, and Emilie Kaufmann. Dealing with unknown variances in best-arm identification. *CoRR*, abs/2210.00974, 2022. URL `https://arxiv.org/abs/2210.00974`.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1238–1246, 2013.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 2016.

Andreas Krause, Ajit Paul Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

Shyong Lam and Jon Herlocker. MovieLens Dataset. http://grouplens.org/datasets/movielens/, 2016.

Tor Lattimore and Csaba Szepesvari. *Bandit Algorithms*. Cambridge University Press, 2019.

B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000.

Pinyan Lu, Chao Tao, and Xiaojin Zhang. Variance-dependent best arm identification. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence*, 2021.

Friedrich Pukelsheim. *Optimal Design of Experiments*. John Wiley & Sons, 1993.

Marta Soare, Alessandro Lazaric, and Remi Munos. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems 27*, pages 828–836, 2014.

Ruida Zhou and Chao Tian. Approximate top-m arm identification with heterogeneous reward variances. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, 2022.