

# Max-Min Grouped Bandits

Zhenlin Wang,<sup>1</sup> Jonathan Scarlett<sup>1,2</sup>

<sup>1</sup>School of Computing, National University of Singapore

<sup>2</sup>Department of Mathematics & Institute of Data Science, National University of Singapore  
wang\_zhenlin@u.nus.edu, scarlett@comp.nus.edu.sg

## Abstract

In this paper, we introduce a multi-armed bandit problem termed max-min grouped bandits, in which the arms are arranged in possibly-overlapping groups, and the goal is to find a group whose worst arm has the highest mean reward. This problem is of interest in applications such as recommendation systems, and is also closely related to widely-studied robust optimization problems. We present two algorithms based successive elimination and robust optimization, and derive upper bounds on the number of samples to guarantee finding a max-min optimal or near-optimal group, as well as an algorithm-independent lower bound. We discuss the degree of tightness of our bounds in various cases of interest, and the difficulties in deriving uniformly tight bounds.

## 1 Introduction

Multi-armed bandit (MAB) algorithms are widely adopted in scenarios of decision-making under uncertainty (Lattimore and Szepesvári 2020). In theoretical MAB studies, two particularly common performance goals are *regret minimization* and *best-arm identification*, and this paper is more closely related to the latter.

The most basic form of best-arm identification seeks to identify the arm with the highest mean (e.g., see (Kaufmann, Cappé, and Garivier 2016)). Other variations are instead based on returning *multiple arms*, such as the  $k$  believed to have the highest  $k$  means (Kalyanakrishnan et al. 2012), or the individual highest-mean arms within a pre-defined set of groups that may be non-overlapping (Gabillon et al. 2011) or overlapping (Scarlett, Bogunovic, and Cevher 2019).

In this paper, we introduce a distinct problem setup in which we are again given a collection of (possibly overlapping) groups of arms, but the goal is to identify the group whose *worst arm* (in terms of the mean reward) is as high as possible. To motivate this problem setup, we list two potential applications:

- In recommendation systems, the groups may correspond to packages of items that can be offered or advertised together. If the users are highly averse to poor items, then it is natural to model the likelihood of clicking/purchasing as being dictated by the worst item.
- In a resource allocation setting, suppose that we would like to choose the best group of computing machines (or other resources), but we require robustness because the slowest machine will be bottleneck when it comes to running jobs. Then, we would like to find the group whose worst-case machine is the best.

More generally, this problem captures the notion of a group *only being as strong as its weakest link*, and is closely related to widely-studied robust optimization problems (e.g., (Bertsimas, Nohadani, and Teo 2010)).

Before describing our main contributions, we provide a more detailed overview of the most related existing works.

### 1.1 Related Work

The related work on multi-armed bandits and best-arm identification is extensive; we only provide a brief outline here with an emphasis on the most closely related works.

The standard best-arm identification problem was studied in (Audibert, Bubeck, and Munos 2010; Gabillon et al. 2012; Jamieson and Nowak 2014; Kaufmann, Cappé, and Garivier 2016; Garivier and Kaufmann 2016), among others. These works are commonly distinguished according to whether the time horizon is fixed (fixed-budget setting) or the target error probability is fixed (fixed-confidence setting), and the latter is more relevant to our work. In particular, we will utilize anytime confidence bounds from (Jamieson and Nowak 2014) in our upper bounds, as well as a fundamental change-of-measure technique from (Kaufmann, Cappé, and Garivier 2016) in our lower bounds.

A notable *grouped* best-arm identification problem was studied in (Gabillon et al. 2011; Bubeck, Wang, and Viswanathan 2013), where the arms are partitioned into disjoint groups, and the goal is to find the best arm in each group. A generalization of this problem to the case of overlapping groups was provided in (Scarlett, Bogunovic, and Cevher 2019). Another notable setting in which multiple arms are returned is that of subset selection, where one seeks to find a subset of  $k$  arms attaining the highest mean rewards (Kalyanakrishnan et al. 2012; Kaufmann and Kalyanakrishnan 2013; Kaufmann, Cappé, and Garivier 2016). In our understanding, all of these works are substantially different from the max-min grouped bandit problem that we consider.

Another setup of interest is the recently-proposed categorized bandit problem (Jedor, Perchet, and Louedec 2019). This setting consists of disjoint groups with a partial ordering, and the knowledge of the group structure (but not their order) is given as prior information. However, different from our setting, the goal is still to find the best overall arm (or more precisely, minimize the corresponding regret notion). In addition, the results of (Jedor, Perchet, and Louedec 2019) are based on the arm means satisfying certain partial ordering assumptions between the groups (e.g., all arms in a better group beat all arms in a worse group), whereas we consider general instances without such restrictions. See also (Bouneffouf et al. 2019; Ban and He 2021; Singh et al. 2020) and the references therein for other MAB settings with a clustering structure.

Finally, we note that our setup can be viewed as a MAB counterpart to *robust optimization*, which has received considerable attention on continuous domains (Bertsimas, Nohadani, and Teo 2010; Chen et al. 2017; Bogunovic et al. 2018), as well as set-valued domains with submodular functions (Krause et al. 2008; Orlin, Schulz, and Udwani 2018; Bogunovic et al. 2017). Robust maximization problems generically take the form  $\max_{x \in D_x} \min_{c \in D_c} f(x, c)$ , and in Sec. 4 we will explicitly connect our setup to the kernelized robust optimization setting studied in (Bogunovic et al. 2018). However, based on what is currently known, this connection will only provide relatively loose instance-independent bounds when applied to our setting, and the bounds derived in our work (both instance-dependent and instance-independent) will require a separate treatment.

## 1.2 Contributions and Paper Structure

The paper is outlined as follows:

- In Sec. 2, we formally introduce the max-min grouped bandit problem, and briefly discuss a naive approach.
- In Sec. 3, we present an algorithm based on successive elimination, and derive an instance-dependent upper bound on time required to find the optimal group.
- In Sec. 4, we show that our setup can be cast under the framework of kernel-based robust optimization, and use this connection to adapt an algorithm from (Bogunovic et al. 2018). We additionally derive an instance-independent regret bound (i.e., a bound on the suboptimality of the declared group relative to the best).
- In Sec. 5, we return to considering instance-dependent bounds, and complement our upper bound with an algorithm-independent lower bound.
- In Sec. 6, we further discuss our bounds, including highlighting cases where they are tight vs. loose, and the difficulty in deriving uniformly tight bounds.
- In Sec. 7, we present numerical experiments investigating the relative performance of the algorithms considered.

## 2 Problem Setup

We first describe the problem aspects that are the same as the regular MAB problem. We consider a collection  $\mathcal{A} = \{1, \dots, n\}$  of  $n$  arms/actions. In each round, indexed by  $t$ , the MAB algorithm selects an arm  $j_t \in \mathcal{A}$ , and observes the corresponding reward  $X_{j_t, T_{j_t}(t)}$ , where  $T_j(t)$  is the number of pulls of arm  $a_j$  up to time  $t$ . We consider stochastic rewards, in which for each  $j \in \mathcal{A}$ , the random variables  $\{X_{j, \tau}\}_{\tau \geq 1}$  are independently drawn from some unknown distribution with mean  $\mu_j$ . We will consider algorithms that make use of the empirical mean, defined as

$$\hat{\mu}_{j, T_j(t)} = \frac{1}{T_j(t)} \sum_{s=1}^{T_j(t)} X_{j, s}.$$

Different from the standard MAB setup, we assume that there is a known set of groups  $\mathcal{G}$ , where each group  $G \in \mathcal{G}$  is a non-empty subset of  $\mathcal{A}$ . We allow overlaps between groups, i.e., a given arm may appear in multiple groups. Without loss of generality, we assume that each arm is in at least one group. We are interested in identifying the *max-min optimal group*, defined as follows:

$$G^* = \arg \max_{G \in \mathcal{G}} \min_{j \in G} \mu_j. \quad (1)$$

To reduce notation, we define  $j_{\text{worst}}(G) = \arg \min_{j \in G} \mu_j$  to be the arm in  $G$  with the lowest mean; if this is non-unique, we simply take any one of them chosen arbitrarily.

After  $T$  rounds (where  $T$  may be fixed in advance or adaptively chosen based on the rewards), the algorithm outputs a recommendation  $G^{(T)}$  representing its guess of the optimal group. We consider two closely-related performance measures, namely, the error probability

$$P_e = \mathbb{P}[G^{(T)} \neq G^*], \quad (2)$$

and the simple regret

$$r(G^{(T)}) = \mu_{j_{\text{worst}}(G^*)} - \mu_{j_{\text{worst}}(G^{(T)})}. \quad (3)$$

Naturally, we would like  $P_e$  and/or  $r(G^{(T)})$  to be as small as possible, while also using as few arm pulls as possible.

## 2.1 Assumptions

Throughout the paper, we will make use of several assumptions that are either standard in the literature, or simple variations thereof. We start with the following.

**Assumption 1.** We assume that the arm means are bounded in  $[0, 1]$ ,<sup>1</sup> and that the reward distributions are sub-Gaussian with parameter  $R$ , i.e., if  $X_j$  is a random variable drawn from the  $j$ -th arm's distribution, then  $\mathbb{E}[X_j] = \mu_j$  and  $\mathbb{E}[e^{\lambda(X_j - \mu_j)}] \leq \exp(\lambda^2 R^2 / 2)$  for all  $\lambda \in \mathbb{R}$ .

We will consider Gaussian and Bernoulli rewards as canonical examples of distributions satisfying Assump. 1.

The next assumption serves as a natural counterpart to that of having a unique best arm in the standard best-arm identification problem, i.e., an *identifiability* assumption.

**Assumption 2.** There exists a unique group  $G^*$  with the highest worst arm, i.e.,

$$\min_{j \in G^*} \mu_j > \max_{G \in \mathcal{G}: G \neq G^*} \min_{j \in G} \mu_j. \quad (4)$$

With this assumption in mind, we now turn to defining *fundamental gaps* between the arm means. These are also ubiquitous in instance-dependent studies of MAB problems, but are somewhat different here compared to other settings.

Recall that  $j_{\text{worst}}(G)$  is the worst arm in a group  $G \in \mathcal{G}$ . We define the difference between the worst arm of  $G^*$  and the worst arm of group  $G$  as  $\Delta_G = \mu_{j_{\text{worst}}(G^*)} - \mu_{j_{\text{worst}}(G)}$ . Then, for each arm indexed by  $j$ , the following quantities will play a key role in our analysis:

- $\Delta'_j := \min_{G: j \in G} (\mu_j - \mu_{j_{\text{worst}}(G)})$  is the minimum distance between (the mean reward of)  $j$  and the worst arm  $j_{\text{worst}}(G)$  in any of the groups containing  $j$ . This gap determines when  $j$  can be removed (i.e., no longer pulled) if it is not a worst arm in any group.
- $\Delta''_j := \min_{G: j \in G} \Delta_G$  is the minimum distance between the worst arm in the optimal group  $G^*$  and the worst arm in any of the groups containing  $j$ . This gap determines when all the groups that  $j$  is in can be ruled out as being suboptimal. If  $j$  is not in the optimal group  $G^*$ , the removal of these groups also amounts to  $j$  being removed, whereas if  $j$  is in  $G^*$ , this value becomes zero.
- $\Delta_0 := \min_{G: G \neq G^*} \Delta_G$  is a fixed constant indicating the distance between the worst arm in the optimal group  $G^*$  and the best among the worst arms in the remaining suboptimal groups. This gap determines when the optimal group is found (and the algorithm terminates).

Following the definitions above, we define the overall gap associated with each arm  $j$  as follows:

$$\Delta_j = \max\{\Delta'_j, \Delta''_j, \Delta_0\} > 0. \quad (5)$$

In Sec. 3, we will present an algorithm such that, with high probability, arm  $j$  stops being sampled after a certain number of pulls dependent on  $\Delta_j$ .

---

<sup>1</sup>Any finite interval can be shifted and scaled to this range.

## 2.2 Failure of Naive Approach

A simple algorithm to solve the max-min grouped bandit problem is to treat it as a combination of  $|\mathcal{G}|$  worst arm search problems. We can consider each group separately, and identify the worst arm for each group via a “best”-arm identification algorithm (trivially adapted to find the *worst* arm instead of the best) such as UCB or LUCB (Jamieson and Nowak 2014). We can then rank the worst arms in the various groups to find the optimal group.

However, this method may be highly suboptimal, as it ignores the comparisons between arms from different groups during the search. For instance, consider a setting in which  $\mathcal{G} = \{G_1, G_2\}$  with  $G_1 = \{1, \dots, k\}$  and  $G_2 = \{k+1, \dots, n\}$ . Suppose that  $\mu_1 = 1 > \dots > \mu_{k-1} = 0.9 > \mu_k = 0.8$  and  $\mu_{k+1} = 0.1 > \dots > \mu_{n-1} = 0.01 > \mu_n = 0.00999$ . We observe that finding the worst arm in  $G_2$  is highly inefficient due to the narrow gap of  $0.01 - 0.00999$  between arms  $n-1$  and  $n$ . On the other hand, a simple comparison between the observed values of arms from  $G_1$  and  $G_2$  can relatively quickly verify that all arms in  $G_1$  are better than all arms in  $G_2$ , without needing to know the precise ordering of arms within either group.

## 2.3 Auxiliary Results

As is ubiquitous in MAB problems, our analysis relies on confidence bounds. Despite our distinct objective, our setup still consists of regular arm pulls, and accordingly, we can utilize well-established confidence bounds for stochastic bandits. Many such bounds exist with varying degrees of simplicity vs. tightness, and to ease the exposition, we do not seek to optimize this trade-off, but instead focus on one representative example given as follows.

**Lemma 1.** (Law of Iterated Logarithm (Jamieson and Nowak 2014)) *Let  $Z_1, Z_2, \dots$  be i.i.d sub-Gaussian random variables with mean  $\mu \in \mathbb{R}$  and parameter  $\sigma \leq \frac{1}{2}$ . For any  $\epsilon \in (0, 1)$  and  $\delta \in (0, \frac{1}{e} \log(1 + \epsilon))$ , with probability at least  $1 - \frac{2+\epsilon}{\epsilon/2} (\frac{\delta}{\log(1+\epsilon)})^{1+\epsilon}$ , we have*

$$\left| \frac{1}{t} \sum_{s=1}^t Z_s - \mu \right| \leq U(t, \delta) \quad \forall t \geq 1, \quad (6)$$

where

$$U(t, \delta) = (1 + \sqrt{\epsilon}) \sqrt{\frac{1 + \epsilon}{2t} \log \frac{\log(1 + \epsilon)t}{\delta}}. \quad (7)$$

In accordance with this result, we henceforth assume that  $R \leq \frac{1}{2}$  in Assump. 1, which notably always holds for Bernoulli rewards.

Since the error probability is dependent on the entire set of  $n$  arms, we further replace  $\delta$  by  $\frac{\delta}{n}$  in Lemma 1 and apply a union bound, which leads to the following upper/lower confidence bound of arm  $j$  in round  $t$ :

$$\text{UCB}_t(j) = \hat{\mu}_{j, T_j(t)} + U\left(T_j(t), \frac{\delta}{n}\right) \quad (8)$$

$$\text{LCB}_t(j) = \hat{\mu}_{j, T_j(t)} - U\left(T_j(t), \frac{\delta}{n}\right). \quad (9)$$

Hence, with probability at least  $1 - \frac{2+\epsilon}{\epsilon/2} (\frac{\delta}{\log(1+\epsilon)})^{1+\epsilon}$ ,

$$\text{LCB}_t(j) \leq \mu_j \leq \text{UCB}_t(j), \quad \forall j \in \{1, \dots, n\}, t \geq 1. \quad (10)$$

To derive the performance bounds for our algorithms, we further require the following lemma:

**Lemma 2.** (Inversion of  $U(t, \delta)$  (Jamieson and Nowak 2014)) *For any  $\epsilon \in (0, 1)$ ,  $\delta \in (0, \frac{1}{e} \log(1 + \epsilon))$ , and  $\Delta \in (0, 1)$ , we have*

$$\min \left\{ k : U\left(k, \frac{\delta}{n}\right) \leq \frac{\Delta}{4} \right\} \leq \frac{2\gamma}{\Delta^2} \log \frac{2 \log(\gamma(1 + \epsilon)\Delta^{-2})}{\delta/n}, \quad (11)$$

where  $\gamma = 8(1 + \sqrt{\epsilon})^2(1 + \epsilon)$ .

## 3 Successive Elimination

Elimination-based algorithms are widely used in MAB problems. In the standard best-arm identification setting, the idea is to sample arms in batches and then eliminate those known to be suboptimal based on confidence bounds,

until only one arm remains. In the max-min setting that we consider, we can use a similar idea, but we need to carefully consider the conditions under which an arm no longer needs to be pulled. We proceed by describing this and giving the relevant definitions.

We will work in epochs indexed by  $i$ , and let  $t_i$  denote the number of arm pulls up to the start of the  $i$ -th epoch. For each group  $G$ , we define the set of potential worst arms as

$$m_i^{(G)} := \left\{ j \in G : \text{LCB}_{t_i}(j) \leq \min_{j' \in G} \text{UCB}_{t_i}(j') \right\}. \quad (12)$$

This definition will allow us to eliminate arms that are no longer potentially worst in any group. We additionally consider a set of candidate *potentially optimal* groups  $\mathcal{C}_i$ , initialized  $\mathcal{C}_0 = \mathcal{G}$  and subsequently updated as follows:

$$\mathcal{C}_{i+1} := \left\{ G \in \mathcal{C}_i : \min_{j' \in m_i^{(G')}} \text{LCB}_{t_i}(j') \leq \min_{j \in m_i^{(G)}} \text{UCB}_{t_i}(j), \forall G' \in \mathcal{C}_i \right\}. \quad (13)$$

This definition allows us to stop considering any groups that are already believed to be suboptimal.

Finally, the set of candidate arms (i.e., arms that we still need to continue pulling) is given by

$$\mathcal{A}_i := \{ j : j \in m_i^{(G)} \text{ for at least one } G \in \mathcal{C}_i \}. \quad (14)$$

With these definitions in place, pseudo-code for the successive elimination algorithm is shown in Alg. 1.

---

**Algorithm 1** Successive Elimination algorithm

---

**Require:** Arms  $(a_1, \dots, a_n)$ , set of groups  $\mathcal{G}$ , parameters  $\delta, \epsilon > 0$

- 1: Initialize  $i = 0, t = 0$  and  $T_j(t) = 0$  for all  $j$
  - 2: Set  $m_0^{(G)} = G$  for all  $G \in \mathcal{G}, \mathcal{C}_0 = \mathcal{G}, \mathcal{A}_0 = \{1, 2, \dots, n\}$
  - 3: **while**  $|\mathcal{C}_i| > 1$  **do**
  - 4:     Pull every arm  $j$  in  $\mathcal{A}_i$  once, incrementing  $t$  after each pull and updating all  $T_j(t)$
  - 5:     Compute  $m_{i+1}^{(G)}, \mathcal{C}_{i+1}$  and  $\mathcal{A}_{i+1}$  via expressions (12), (13) and (14)
  - 6:     Increment round index  $i$  by 1
  - 7: **return**  $\hat{\mathcal{C}} = \mathcal{C}_i$
- 

We now state our main result regarding this algorithm.

**Theorem 1.** (Upper Bound for Successive Elimination) *For any max-min grouped bandit instance as defined in Sec. 2, given  $\epsilon \in (0, 1)$  and  $\delta \in (0, \frac{1}{e} \log(1 + \epsilon))$ , with probability at least  $1 - \frac{2+\epsilon}{\epsilon/2} \left( \frac{\delta}{\log(1+\epsilon)} \right)^{1+\epsilon}$ , Alg. 1 identifies the optimal group and uses a number of arm pulls satisfying*

$$T(\delta, \epsilon) \leq \sum_{j=1}^n \frac{2\gamma}{\Delta_j^2} \log \frac{2 \log(\gamma(1 + \epsilon) \Delta_j^{-2})}{\delta/n}, \quad (15)$$

where  $\gamma = 8(1 + \sqrt{\epsilon})^2(1 + \epsilon)$ .

The proof is given in Appendix A, and is based on considering the gap  $\Delta_j$  associated with each arm; we show that after the confidence width falls below  $\frac{\Delta_j}{4}$ , any such arm will be eliminated (or the algorithm will terminate), as long as the confidence bounds are valid. Applying Lemma 2 and summing over the arms then gives the desired result.

While the error term  $\delta_0 = \frac{2+\epsilon}{\epsilon/2} \left( \frac{\delta}{\log(1+\epsilon)} \right)^{1+\epsilon}$  in Thm. 1 is somewhat complicated, one can fix  $\epsilon = \frac{1}{2}$  (say) and solve for  $\delta$ , and it readily follows that the right-hand side of (15) has the standard  $O(\log \frac{1}{\delta_0})$  dependence.

## 4 A Variant of STABLEOPT

In this section, we first discuss how the max-min grouped bandit problem is related to the problem of adversarially robust optimization. We then demonstrate that a robust optimization algorithm known as STABLEOPT (Bogunovic et al. 2018) can be adapted to our setting with instance-independent regret guarantees.

**Connection to adversarially robust optimization** In general, adversarially robust optimization problems take the form  $\max_{x \in D_x} \min_{c \in D_c} f(x, c)$ , where  $x$  is chosen by the algorithm, and  $c$  can be viewed as being chosen by an adversary.

The main focus in (Bogunovic et al. 2018) is finding an  $\epsilon$ -stable optimal input for some function  $f$ :

$$x_\epsilon^* \in \arg \max_{x \in D_x} \min_{\delta \in \Delta_\epsilon(x)} f(x + \delta), \quad (16)$$

where  $\Delta_\epsilon(x) = \{x' - x : x \in D_x \text{ and } d(x, x') \leq \epsilon\}$  is the perturbed region around  $x$ , and  $d(\cdot, \cdot)$  is a generic “distance” function (but need not be a true distance measure).

In Appendix D, we discuss a partial reduction to a grouped max-min problem presented in (Bogunovic et al. 2018), while also highlighting the looseness in directly applying the results therein to our setting.

**Adapting STABLEOPT** The original STABLEOPT algorithm in (Bogunovic et al. 2018) corresponding to the problem (16) selects  $x_t = \arg \max_{x \in D_x} \min_{\delta \in \Delta_\epsilon(x)} \text{UCB}_{t-1}(x + \delta)$ , where  $\delta_t = \arg \min_{\delta \in \Delta_\epsilon(x_t)} \text{LCB}_{t-1}(x_t + \delta)$ , for suitably defined confidence bounds  $\text{UCB}_t$  and  $\text{LCB}_t$ . When adapted to our formulation (1), the algorithm becomes the following:

$$G_t = \arg \max_{G \in \mathcal{G}} \min_{j \in G} \text{UCB}_{t-1}(j) \quad (17)$$

$$j_t = \arg \min_{j \in G_t} \text{LCB}_{t-1}(j), \quad (18)$$

and the algorithm samples arm  $j_t$  in round  $t$ .

Intuitively, this criterion selects the optimistic estimate of the best group and its pessimistic estimate for the worst arm via the UCB and LCB values computed in each round. Instead of using the general RKHS-based confidence bounds in (Bogunovic et al. 2018), we use those in (8)–(9).

The method for breaking ties in (17)–(18) does not impact our analysis. For instance, one could break ties uniformly at random, or one may prefer to break ties in (17) by taking the group that attains the lower LCB score in (18).

**Instance-independent regret bound** Deriving instance-dependent regret bounds for STABLEOPT appears to be challenging, though would be of interest for future work. We instead focus on *instance-independent* bounds. Since there always exist instances for which finding the best group requires an arbitrarily long time (e.g., see Sec. 5), we instead measure the performance using the simple regret, whose definition is repeated from (3) as follows:

$$r(G^{(T)}) = \max_{G \in \mathcal{G}} \min_{j \in G} \mu_j - \min_{j \in G^{(T)}} \mu_j, \quad (19)$$

where  $T$  is the time horizon, and  $G^{(T)}$  is the group returned after  $T$  rounds. For STABLEOPT, the theoretical choice of  $G^{(T)}$  is given by (Bogunovic et al. 2018)

$$G^{(T)} = G_{t^*}, \quad t^* = \arg \max_{t \in \{1, \dots, N\}} \min_{j \in G_t} \text{LCB}_{t-1}(j). \quad (20)$$

Here and subsequently, we define  $\text{LCB}_0(j) = 0$  and  $\text{UCB}_0(j) = 1$  in accordance with the fact that the arm means are in  $[0, 1]$ , and for later convenience we similarly define  $T_j(0) = 0$  and  $U(0, \delta) = 1$ .

With these definitions in place, we have the following result; we state a simplified form with fixed  $\epsilon$  and an implicit constant factor, but give the precise constants in the proof.

**Theorem 2.** (Instance-Independent Regret Bound) *Suppose that Assump. 1 holds. Given  $\delta \in (0, \frac{\log 2}{e})$ , the above variant of STABLEOPT yields with probability at least  $1 - O(\delta)$  that*

$$r(G^{(T)}) \leq O\left(\sqrt{\frac{n}{T}} \left(\sqrt{\log \frac{n}{\delta}} + \log \log T\right)\right). \quad (21)$$

The proof is given in Appendix B, and is based on initially following the max-min analysis of (Bogunovic et al. 2017) to deduce an upper bound of  $\frac{1}{T} \sum_{t=1}^T 2U(T_{j_t}(t-1), \frac{\delta}{n})$ , but then proceeding differently to further upper bound the right-hand side, in particular relying on Lemma 2.

To compare Thm. 2 with Thm. 1, it helps to consider the following corollary.

**Corollary 1.** *Under the setup of Thm. 2, if we additionally have that Assump. 2 holds and the gaps defined in Sec. 2 satisfy  $\Delta_j \geq \Delta_{\min}$  for all  $j = 1, \dots, n$  and some  $\Delta_{\min} > 0$ . Then, with probability at least  $1 - O(\delta)$ , the algorithm outputs  $G^{(T)} = G^*$  provided that  $T \geq \Omega^*\left(\frac{n \log \frac{n}{\delta}}{\Delta_{\min}^2}\right)$  where  $\Omega^*(\cdot)$  hides  $\log \log(\cdot)$  factors in the argument.*

This result matches the scaling in Thm. 1 whenever  $\Delta_j = \Delta_{\min}$  for all  $j$ , which can be viewed as a minimax worst-case instance. Moreover, a standard reduction to finding a biased coin (e.g., (Mannor and Tsitsiklis 2004)) reveals that any algorithm must use the preceding number of arm pulls (or more) on worst-case instances, at least up to the replacement of  $\log \frac{n}{\delta}$  by  $\log \frac{1}{\delta}$ ; hence, there is no significant room for improvement in the minimax sense.

On the other hand, it is also of interest to understand instances that are not of the worst-case kind, in which case the number of arm pulls given in Thm. 1 can be significantly smaller. We expect that STABLEOPT also enjoys instance-dependent guarantees (see Sec. 7 for some numerical evidence), though Thm. 2 does not show it.

## 5 Algorithm-Independent Lower Bound

### 5.1 Preliminaries

We follow the high-level approach of (Kaufmann, Cappé, and Garivier 2016), and make use of the following assumption adopted therein.

**Assumption 3.** *The reward distribution  $P_j$  for any arm  $j$  belongs to family  $\mathcal{P}$  of distributions parametrized by their mean  $\mu_j \in (0, 1)$ . Any two distributions  $P_j, P_{j'} \in \mathcal{P}$  are mutually absolutely continuous, and  $D(P_j \| P_{j'}) \rightarrow 0$  in the limit as  $\mu_{j'}$  approaches  $\mu_j$ .*

The following assumption is not necessary for the bulk of our analysis, but will allow us to state our results in a cleaner form that can be compared directly to the upper bounds.

**Assumption 4.** *There exists a constant  $\tilde{C} > 0$  such that, for any arm distributions  $P_j$  and  $P_{j'}$  having corresponding means  $\mu_j$  and  $\mu_{j'}$ , it holds that  $D(P_j \| P_{j'}) \leq \tilde{C}(\mu_j - \mu_{j'})^2$ .*

As is well-known from previous works, the above assumptions are satisfied in the case of Gaussian rewards, and also Bernoulli rewards under the additional requirement of means strictly bounded away from zero and one (e.g.,  $\mu_j \in (0.01, 0.99)$  for all  $j$ ).

We use the widely-adopted approach of considering a base instance, and perturbing the arm means (ideally only a small amount) to form a different instance with a different optimal group; see Lemma 4 in the appendix. An additional technical challenge here is that even if  $\mathcal{A}$  is identifiable (i.e., satisfies Assump. 2), it can easily happen that the perturbed instance is non-identifiable due to the new max-min arm appearing in multiple groups. To circumvent this issue, we introduce the following definition for the algorithm's success.

**Definition 1.** We say that a max-min grouped bandit algorithm is *uniformly  $\delta$ -successful* with respect to a class of instances if it satisfies the following:

- For any identifiable instance in the class, the algorithm almost surely terminates, and returns the max-min optimal group (i.e.,  $G^*$ ) with probability at least  $1 - \delta$ .
- For any non-identifiable instance in the class, the algorithm may or may not terminate, but has a probability at most  $\delta$  of returning a group that is not max-min optimal.

We note that successive elimination in Sec. 3 satisfies these conditions; in the non-identifiable case, as long as the confidence bounds are valid, the algorithm never terminates.

### 5.2 Statement of Result

In the following, we let  $N_j$  denote the total number of times arm  $j$  is pulled.

**Theorem 3.** (Algorithm-Independent Lower Bound) *Consider any algorithm that is uniformly  $\delta$ -successful with respect to the instances satisfying Assump. 1, Assump. 3, and Assump. 4. Fix any identifiable instance  $\mathcal{A} = (a_1, \dots, a_n)$  with distributions  $(P_1, \dots, P_n)$  and a specified grouping  $\mathcal{G} = (G_1, \dots, G_m)$ . Then, when the algorithm is run on instance  $\mathcal{A}$ , we have the following:*

- For each  $j \in G^*$ , the average number of pulls satisfies

$$\mathbb{E}[N_j] \geq \frac{\log \frac{1}{2.4\delta}}{\tilde{C}(\Delta'_j + \Delta_0)^2}, \quad (22)$$

where  $\tilde{C}$  appears in Assump. 4, and  $\Delta'_j$  and  $\Delta_0$  are defined in Sec. 2.1.

- For each  $G \neq G^*$ , we have

$$\begin{aligned} \sum_{j \in G : \mu_j < \mu_{\text{worst}}(G^*)} \mathbb{E}[N_j(\sigma)] \cdot \tilde{C}(\mu_{j_{\text{worst}}(G^*)} - \mu_j)^2 \\ \geq \log \frac{1}{2.4\delta}. \end{aligned} \quad (23)$$

The proof is given in Appendix C, and is based on shifting the given instance to create a new instance with a different optimal group, and then quantifying the number of arm pulls required to distinguish the two instances. This turns out to be straightforward when  $j \in G^*$ , but less standard (requiring multiple arms to be shifted) when  $j \notin G^*$ .

While Thm. 3 does not directly state a lower bound on the total number of pulls, we can perform some further steps to deduce such a bound depending on the number of groups-per-arm (which could alternatively be upper bounded trivially by  $|\mathcal{G}|$ ), stated as follows and proved in Appendix C.

**Corollary 2.** (Simplified Algorithm-Independent Lower Bound) *Consider the setup of Thm. 3, and suppose that there exists an integer  $m > 0$  such that every arm appears in at most  $m$  groups. Then, the expected number of arm pulls is lower bounded by*

$$T_{\text{lower}}(\delta) = \sum_{j \in G^*} \frac{\log \frac{1}{2.4\delta}}{\tilde{C}(\Delta'_j + \Delta_0)^2} + \frac{1}{m} \sum_{G \in \mathcal{G} \setminus \{G^*\}} \frac{\log \frac{1}{2.4\delta}}{\tilde{C}\Delta_G^2}, \quad (24)$$

where  $\Delta_G = \mu_{j_{\text{worst}}(G^*)} - \mu_{j_{\text{worst}}(G)}$ .

In the following section, we discuss the strengths and weaknesses of our bounds in detail.

## 6 Discussion

**Cases with near-matching behavior.** We first note that for  $j \in G^*$ , the number of pulls of that particular arm dictated by our upper and lower bounds are near-matching. This is because any  $j \in G^*$  has  $\Delta'_j = 0$  and hence  $\Delta_j = \max\{\Delta'_j, \Delta_0\}$ , which matches  $\Delta'_j + \Delta_0$  (see the lower bound) to within a factor of two.

Regarding  $j \notin G^*$ , it is useful to note that  $\Delta'_j = \min_{G: j \in G} \Delta_G$ , and  $\Delta_G$  appears in Cor. 2. Hence, the gaps  $\Delta_G$  between worst arms play a fundamental role in both the upper and lower bounds. However, near-matching behavior is not necessarily observed, as we discuss below.

As an initial positive case, under the trivial grouping  $\mathcal{G} = \{\{1\}, \{2\}, \dots, \{n\}\}$ , our bounds reduce to near-tight bounds for the standard best-arm identification problem (Jamieson and Nowak 2014; Kaufmann, Cappé, and Garivier 2016), with  $\sum_{j=1}^n \frac{1}{\Delta_j^2}$  dependence on the gaps  $\{\Delta_j\}$  between suboptimal arms and the optimal arm.

More generally, our upper and lower bounds have near-matching dependencies when both the number of items-per-group and groups-per-item are bounded, say by some absolute constant. In this case, the second term in (24) is dictated by  $\sum_{G \neq G^*} \frac{1}{\Delta_G^2}$  (since  $m$  is bounded), and we claim that the same is true for the contribution of  $j \notin G^*$  in the upper bound. To see this, first note that within each group, the arm with the smallest  $\Delta_j$  is the one with the smallest  $\Delta'_j$  (the other two quantities  $\Delta'_j$  and  $\Delta_0$  do not vary within  $G$ ), which is  $j_{\text{worst}}(G)$  (having  $\Delta'_j = 0$ ). Thus,  $j = j_{\text{worst}}(G)$  incurs the most pulls in  $G$ , and has  $\Delta_j = \max\{\Delta'_j, \Delta_0\}$ . The definition of  $\Delta_0$  combined with  $j = j_{\text{worst}}(G)$  imply that this simplifies to  $\Delta_j = \Delta_G$ . When we have bounded items-per-group and groups-per-item, it follows that  $\sum_{j \notin G^*} \frac{1}{\Delta_j^2}$  reduces to  $\sum_{G \neq G^*} \frac{1}{\Delta_G^2}$  up to constant factors, as desired.

**Cases where the bounds are not tight.** Perhaps most notably, the lower bound only dictates a minimum *total* number of pulls for arms in a given group  $G \neq G^*$ , whereas the upper bound is based on each *individual* arm being pulled enough. It turns out that we can identify weaknesses in both of these, and it is likely that neither bound can consistently be identified as “tighter” than the other.

To see a potential weakness in the upper bound in Thm. 1, consider an instance with  $|\mathcal{G}| = 2$  and only a single arm  $j^*$  in the optimal group  $G^*$ , and a large number of arms in the suboptimal group. For the arms in the suboptimal



group  $G_2$ , suppose that half of them have a mean reward significantly above that of  $j^*$ , and the other half have a significantly smaller mean reward. In this case, it is feasible to quickly identify  $G_2$  as suboptimal by randomly selecting a small number of arms (namely,  $O(\log \frac{1}{\delta})$  of them if we require an error probability of at most  $\delta$ ) and sampling them a relatively small number of times. Hence, it is not necessary to sample every arm in  $G_2$ . On the other hand, our proposed elimination algorithm immediately starts by pulling every arm, which may be highly suboptimal if  $|G_2|$  is very large.

In contrast, the main looseness is clearly in the lower bound if we modify the above example so that  $G_2$  only has *one* low-mean arm. In this case, the total number of arm pulls should clearly increase linearly with the number of arms, but the lower bound in Thm. 3 does not capture this fact; it only states that both  $j^*$  and the low-mean arm in  $G_2$  should be pulled sufficiently many times.

**Difficulties in obtaining uniformly tight bounds** The examples in Sec. 6 indicate that the general max-min grouped bandit problem is connected to problem of *good-arm identification* (Katz-Samuels and Jamieson 2020), and that improved algorithms might randomly select subsets of arms within the groups (possibly starting with a small subset and expanding it when further information is needed). In fact, in the examples we gave, if the mean reward of  $j^*$  were to be revealed to the algorithm, the remaining task of determining whether  $G_2$  contains an arm with a mean below that of  $j^*$  would be exactly equivalent to the problem studied in (Katz-Samuels and Jamieson 2020). Even this sub-problem required a lengthy and highly technical analysis in (Katz-Samuels and Jamieson 2020), and the difficulty appears to compound further when there are multiple non-overlapping groups, and even further when overlap is introduced. Thus, we believe that the development of near-matching upper and lower bounds is likely to be challenging in general.

**Discussion on identifiability assumptions** Recall from Assump. 2 that we assume  $G^*$  to be uniquely defined. In contrast, we do not require a unique worst arm within *every* group; multiple such arms only amounts to more ways in which the suboptimal group can be identified as such.

The unique optimal group assumption could be removed by introducing a tolerance parameter  $\epsilon$ , and only requiring to identify a group whose worst arm mean is within  $\epsilon$  of the highest possible. In this case, any gap values  $\Delta_j$  that are below  $\epsilon$  get capped to  $\epsilon$  in the upper bound Thm. 1. Our lower bounds can also be adapted accordingly. The changes in the analysis are entirely analogous to similar findings in the standard best-arm identification problem (e.g., see (Gabillon et al. 2012)), so we do not go into detail.

## 7 Experiments

In this section, we present some basic experimental results comparing the algorithms considered in the previous sections.

### 7.1 Experimental Setup

In each experiment, we generate 10 instances, each containing 100 arms and 10 possibly overlapping groups. The arm rewards are Bernoulli distributed, and the instances are generated to ensure a pre-specified minimum gap value ( $\Delta$ ) as per (5), and we consider  $\Delta \in \{0, 1.0.2, 0.4\}$ . The precise details of the (randomly-chosen) groupings and arm means are given in Appendix E. Empirical error rates (or simple regret values) are computed by performing 10 trials per instance, for a total of 100 trials.

**Confidence bounds.** Both Successive Elimination and STABLEOPT rely on the confidence bounds (8)–(9). These bounds are primarily adopted for theoretical convenience, so in our experiments we adopt the more practical choice of  $\hat{\mu}_{j, T_j(t)} \pm \frac{c}{\sqrt{T_j(t)}}$  with  $c = 1$ . The theoretical bounds, as well as difference choices of  $c$ , are explored in Appendix E.

**Stopping conditions.** Successive Elimination is defined with a stopping condition, but STABLEOPT is not. A natural stopping condition for STABLEOPT is to stop when highest max-min LCB value exceeds all other groups' max-min UCB values. However, this often requires an unreasonably large number of pulls, due to the existence of UCB values that are only slightly too low for the algorithm to pull based on. We therefore relax this rule by only requiring it to hold to within a specified tolerance, denoted by  $\eta$ . We set  $\eta = 0.01$  by default, and explore other values in Appendix E.

We will also explore the notion of simple regret, and to do so, both algorithms require a method for declaring the *current best guess* of the max-min optimal group. We choose to return the group with the best max-min LCB score, though the max-min empirical mean would also be reasonable.

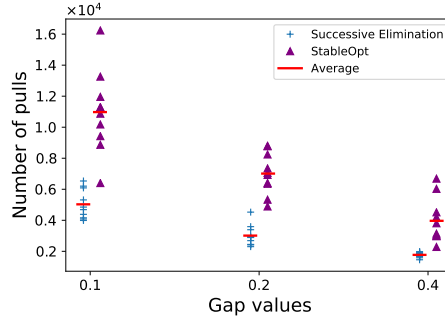


Figure 1: Plot of total arm pulls used for Successive Elimination and STABLEOPT for  $\Delta \in \{0.1, 0.2, 0.4\}$ .

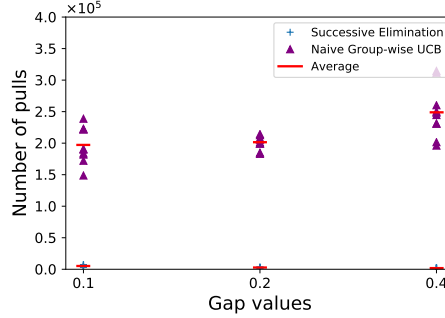


Figure 2: Comparison of Successive Elimination and the naive group-wise strategy for  $\Delta \in \{0.1, 0.2, 0.4\}$ .

## 7.2 Results

**Effect of  $\Delta$ .** From Fig. 1, we observe that the number of arm pulls decreases when the gap  $\Delta$  increases, particularly for Successive Elimination.<sup>2</sup> This is intuitive, and consistent with our theoretical results. These results also suggest that STABLEOPT can adapt to easier instances in the same way as Successive Elimination; obtaining theory to support this would be interesting for future work.

**Comparison to the Naive Approach.** We demonstrate that the simple group-wise approach is indeed suboptimal by comparing its empirical performance with Successive Elimination. Within each group, we identify the worst arm using the UCB algorithm with the same stopping rule as that of STABLEOPT described above, and among the arms identified, the one with the highest LCB score is returned. Fig. 2 supports our discussion in Sec. 2.2, as we observe that this naive approach requires considerably more arm pulls, and does not appear to improve even as  $\Delta$  increases.

**Simple Regret.** As seen above, the total number of pulls comes out to be fairly high for both algorithms. This is due to stringent stopping conditions, and an investigation of the *average simple regret* reveals that the algorithms in fact learn much faster despite not yet terminating, especially for STABLEOPT, and especially when  $\Delta$  is larger; see Fig. 3 (error bars show half a standard deviation). These results again support the hypothesis that STABLEOPT naturally adapts to easier instances, though our theory only handles the instance-independent case.

A possible reason why StableOpt performs better in Fig. 3 is that it more quickly steers towards the more promising groups, whereas Successive Elimination always treats every non-eliminated arm equally.

**Effect of Confidence Width.** In Appendix E, we present further experiments exploring the theoretical choice of confidence width vs. our practical choice of  $\frac{c}{\sqrt{t_j}}$  with  $c = 1$ , as well as considering difference choices of  $c$  (and also the STABLEOPT stopping parameter  $\eta$ ).

## 8 Conclusion

We have introduced the problem of max-min grouped bandits, and studied the number of arm pulls for both an elimination-based algorithm and a variation of StableOpt (Bogunovic et al. 2018). In addition, we provided an algorithm-independent lower bound, and identified some of the potential weaknesses in the bounds and discussed the difficulties in overcoming them. We believe that this leaves open several interesting avenues for further work.

<sup>2</sup>Error probabilities are not shown, because there were no failures in any of the trials here.

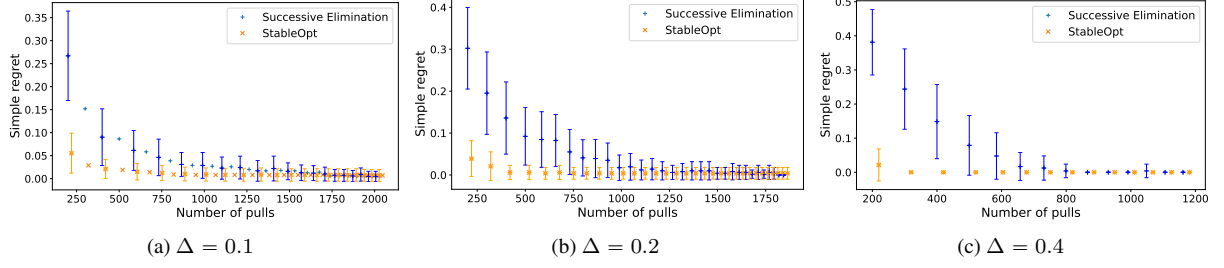


Figure 3: Simple regret plots with various gap values.

## A Proof of Thm. 1 (Upper Bound for Successive Elimination)

We first formally state the correctness of the algorithm.

**Lemma 3.** (Correctness of SE) *Suppose that Assump. 1 and Assump. 2 hold. Given  $\epsilon \in (0, 1)$  and  $\delta \in (0, \frac{1}{e} \log(1 + \epsilon))$ , with probability at least  $1 - \frac{2+\epsilon}{\epsilon/2} (\frac{\delta}{\log(1+\epsilon)})^{1+\epsilon}$ , we have that Alg. 1 returns  $\hat{\mathcal{C}} = \{G^*\}$ .*

*Proof.* We define an event under which the optimal group  $G^*$  remains a candidate group, and its worst arm  $j_{\text{worst}}(G^*)$  remains a candidate worst arm in  $G^*$  in epoch  $i$ :

$$\mathcal{E}_i := \{G^* \in \mathcal{C}_i\} \cap \{j_{\text{worst}}(G^*) \in \mathcal{A}_i\}.$$

We show that  $\mathcal{E}_i$  holds for all  $i$  whenever the confidence bounds introduced Sec. 2.3 are valid, which we know holds with probability at least  $1 - \frac{2+\epsilon}{\epsilon/2} (\frac{\delta}{\log(1+\epsilon)})^{1+\epsilon}$ .

At the beginning of Alg. 1,  $m_0^{(G)} = G$  for all  $G \in \mathcal{G}$  and  $G^* \in \mathcal{G} = \mathcal{C}_0$ . Hence  $j_{\text{worst}}(G^*) \in \{1, 2, \dots, n\} = \mathcal{A}_0$ , and the base case  $\mathcal{E}_0$  holds. We proceed by showing that when  $\mathcal{E}_{i-1}$  holds, so does  $\mathcal{E}_i$ .

First, the validity of the confidence bounds gives for all  $G$  that

$$\text{LCB}_{t_i}(j_{\text{worst}}(G)) \leq \mu_{j_{\text{worst}}(G)} \quad (25)$$

$$= \min_{j' \in G} \mu_{j'} \quad (26)$$

$$\leq \min_{j' \in G} \text{UCB}_{t_i}(j'), \quad (27)$$

implying that  $j_{\text{worst}}(G) \in m_i^{(G)}$  for all  $G$ . That is, each group's truly worst arm always remains a candidate worst arm, and this holds in particular for  $j_{\text{worst}}(G^*)$  in  $G^*$ . For later use, it will also be useful to note that this property implies the final minimum in (27) can be restricted to  $m_i^{(G)}$  instead of  $G$ , yielding

$$\min_{j \in m_i^{(G)}} \text{UCB}_{t_i}(j) \geq \mu_{j_{\text{worst}}(G)}. \quad (28)$$

It remains to show that  $G^*$  always remains a candidate potentially optimal group. Denoting the set of worst arms among all groups as  $\mathcal{J}_{\text{worst}}(\mathcal{G}) := \{j : \mu_j = \min_{j' \in G} \mu_{j'} \text{ for some } G \in \mathcal{G}\}$ , we have

$$\min_{j \in m_i^{(G^*)}} \text{UCB}_{t_i}(j) \geq \mu_{j_{\text{worst}}(G^*)} \quad (29)$$

$$= \max_{j \in \mathcal{J}_{\text{worst}}(\mathcal{G})} \mu_j \quad (30)$$

$$\geq \max_{j \in \mathcal{J}_{\text{worst}}(\mathcal{G})} \text{LCB}_{t_i}(j) \quad (31)$$

$$\geq \min_{j \in m_i^{(G)}} \text{LCB}_{t_i}(j) \quad \forall G \in \mathcal{C}_i, \quad (32)$$

where (29) is an application of (28) to  $G^*$ , (30) holds by the definitions of  $G^*$  and  $\mathcal{J}_{\text{worst}}(\mathcal{G})$ , (31) uses the validity of the confidence bounds, and (32) holds because for each candidate group  $G \in \mathcal{C}_i$ , we have

$$\begin{aligned} \min_{j \in m_i^{(G)}} \text{LCB}_{t_i}(j) &\leq \text{LCB}_{t_i}(j_{\text{worst}}(G)) \\ &\leq \max_{j \in \mathcal{J}_{\text{worst}}(\mathcal{G})} \text{LCB}_{t_i}(j) \end{aligned} \quad (33)$$

due to the fact that  $j_{\text{worst}}(G) \in \mathcal{J}_{\text{worst}}(\mathcal{G})$  and  $j_{\text{worst}}(G) \in m_i^{(G)}$ .

By the definition of  $\mathcal{C}_i$  and (32), we conclude that  $G^* \in \mathcal{C}_i$  after the  $i$ -th epoch. Combining this with  $j_{\text{worst}}(G^*) \in m_i^{(G^*)}$ , we then have  $j_{\text{worst}}(G^*) \in \mathcal{A}_i$ , implying that  $\mathcal{E}_i$  holds.

Since the algorithm stops when  $|\mathcal{C}_i| = 1$  and  $G^* \in \mathcal{C}_i$  for all  $i$ , we conclude that the returned set  $\hat{\mathcal{C}} = \mathcal{C}_i$  must contain only the optimal group  $G^*$ . Note that by the identifiability assumption (Assump. 2) and the fact that  $U(t, \delta) \rightarrow 0$  as  $t \rightarrow \infty$ , the algorithm will never continue running forever when the confidence bounds remain valid.  $\square$

Having established high-probability correctness in Lemma 3, it remains to bound the number of arm pulls. We bound the number of pulls separately for each arm, considering the cases  $j \in G^*$  and  $j \notin G^*$  separately, and showing that  $U(T_j(t), \frac{\delta}{n}) < \frac{\Delta_j}{4}$  is a sufficient condition for the arm to be eliminated in all cases. We then apply Lemma 2 and sum over the arms to obtain the result.

We henceforth suppose that the confidence bounds are valid, as we already considered in the proof of Lemma 3. First observe that if  $U(T_j(t), \frac{\delta}{n}) < \frac{\Delta_j}{4}$ , then we have  $|\text{UCB}_{t_i}(j) - \text{LCB}_{t_i}(j)| < \frac{\Delta_j}{2}$ . In the following, we assume that this is the case for all  $j$  indexing non-eliminated arms; note that by construction, all such arms have been pulled exactly the same number of times after each epoch.

**Case 1** ( $j \in G^*$ ) In this case, we immediately have  $\mu_j \geq \mu_{j_{\text{worst}}(G^*)}$ . By design in the algorithm,  $j$  will stop being pulled in either of the following scenarios:

1.  $j$  is no longer a potential worst arm in any group;
2.  $G^*$  is found to be the optimal group, and the algorithm terminates.

Recall the definitions of  $\Delta'_j$ ,  $\Delta''_j$ ,  $\Delta_0$ , and  $\Delta_j$  in Sec. 2.1. For  $j \in G^*$ , we have  $\Delta''_j = \mu_{j^*} - \mu_{j^*} = 0$ , and hence  $\Delta_j = \min\{\Delta'_j, \Delta_0\}$ . From the intuition behind each gap value defined in Sec. 2, we note that scenario 1 above is related to  $\Delta'_j$ , and scenario 2 is related to  $\Delta_0$ . We now consider each scenario separately as follows:

1. If  $\Delta_j = \Delta'_j$ , then for all  $G$  with  $j \in G$ , we have

$$\text{LCB}_{t_i}(j) > \text{UCB}_{t_i}(j) - \frac{\Delta'_j}{2} \quad (34)$$

$$\geq \mu_j - \frac{\Delta'_j}{2} \quad (35)$$

$$\geq \mu_j - \frac{\mu_j - \mu_{j_{\text{worst}}(G)}}{2} \quad (36)$$

$$= \mu_{j_{\text{worst}}(G)} + \frac{\mu_j - \mu_{j_{\text{worst}}(G)}}{2} \quad (37)$$

$$\geq \mu_{j_{\text{worst}}(G)} + \frac{\Delta'_j}{2} \quad (38)$$

$$\geq \text{LCB}_{t_i}(j_{\text{worst}}(G)) + \frac{\Delta'_j}{2} \quad (39)$$

$$> \text{UCB}_{t_i}(j_{\text{worst}}(G)), \quad (40)$$

where (34) and (40) apply the assumption  $|\text{UCB}_{t_i}(j) - \text{LCB}_{t_i}(j)| < \frac{\Delta_j}{2}$ , (35) and (39) use the confidence bounds, and (36) and (38) use the definition of  $\Delta'_j$ . From (40), we have that  $j$  is removed from  $\mathcal{A}_i$  and is no longer pulled.

2. If  $\Delta_j = \Delta_0$ , then for  $G \neq G^*$ , we have

$$\begin{aligned} & \text{LCB}_{t_i}(j_{\text{worst}}(G^*)) \\ & > \text{UCB}_{t_i}(j_{\text{worst}}(G^*)) - \frac{\Delta_0}{2} \end{aligned} \quad (41)$$

$$\geq \mu_{j_{\text{worst}}(G^*)} - \frac{\Delta_0}{2} \quad (42)$$

$$\geq \mu_{j_{\text{worst}}(G^*)} - \frac{\mu_{j_{\text{worst}}(G^*)} - \mu_{j_{\text{worst}}(G)}}{2} \quad (43)$$

$$= \mu_{j_{\text{worst}}(G)} + \frac{\mu_{j_{\text{worst}}(G^*)} - \mu_{j_{\text{worst}}(G)}}{2} \quad (44)$$

$$\geq \mu_{j_{\text{worst}}(G)} + \frac{\Delta_0}{2} \quad (45)$$

$$\geq \text{LCB}_{t_i}(j_{\text{worst}}(G)) + \frac{\Delta_0}{2} \quad (46)$$

$$> \text{UCB}_{t_i}(j_{\text{worst}}(G)), \quad (47)$$

where (41) and (47) apply the assumption  $|\text{UCB}_{t_i}(j) - \text{LCB}_{t_i}(j)| < \frac{\Delta_j}{2}$ , (42) and (46) use the confidence bounds, and (43) and (45) use the definition of  $\Delta'_j$ . Then, (47) implies that all of the non-optimal groups are removed from  $\mathcal{C}_i$ , so the algorithm terminates and  $j$  is no longer pulled.

**Case 2** ( $j \notin G^*$ ) In this case,  $j$  will stop being pulled if any of the following scenarios are satisfied:

1.  $j$  is no longer a potential worst arm in any group;
2.  $G^*$  is found to be the optimal group, and the algorithm terminates;
3. All the groups  $G$  where  $j \in G$  are no longer candidate groups.

The gap values associated with these conditions are  $\Delta'_j$ ,  $\Delta_0$ , and  $\Delta''_j$ , respectively. In our elimination algorithm,  $G^*$  is found only after all suboptimal groups are removed. Therefore, the second scenario will never be satisfied before the third condition is satisfied, and we have  $\Delta_j = \min\{\Delta'_j, \Delta''_j\}$ .

For brevity, we use the shorthand  $j^* = j_{\text{worst}}(G^*)$  in the remainder of this section. If the arm  $j$  has a mean reward satisfying  $\mu_j > \mu_{j^*}$ , then

$$\begin{aligned} \Delta'_j &= \min_{G: j \in G} (\mu_j - \mu_{j_{\text{worst}}(G)}) \\ &> \min_{G: j \in G} (\mu_{j^*} - \mu_{j_{\text{worst}}(G)}) = \Delta''_j > 0. \end{aligned} \quad (48)$$

Hence,  $\Delta_j \equiv \Delta'_j$ . In this case, by the same reasoning as (34)–(40), the first condition is satisfied and  $j$  is removed from  $\mathcal{A}_i$ .

By the same reasoning, if  $\mu_j < \mu_{j^*}$ , then  $\Delta_j = \Delta''_j$ . In this case, for all  $G \neq G^*$  with  $j \in G$ , we have

$$\begin{aligned} & \text{LCB}_{t_i}(j_{\text{worst}}(G^*)) \\ & > \text{UCB}_{t_i}(j_{\text{worst}}(G^*)) - \frac{\Delta''_j}{2} \end{aligned} \quad (49)$$

$$\geq \mu_{j_{\text{worst}}(G^*)} - \frac{\Delta''_j}{2} \quad (50)$$

$$\geq \mu_{j^*} - \frac{\mu_{j_{\text{worst}}(G^*)} - \mu_{j_{\text{worst}}(G)}}{2} \quad (51)$$

$$= \mu_{j_{\text{worst}}(G)} + \frac{\mu_{j^*} - \mu_{j_{\text{worst}}(G)}}{2} \quad (52)$$

$$\geq \mu_{j_{\text{worst}}(G)} + \frac{\Delta''_j}{2} \quad (53)$$

$$\geq \text{LCB}_{t_i}(j_{\text{worst}}(G)) + \frac{\Delta''_j}{2} \quad (54)$$

$$> \text{UCB}_{t_i}(j_{\text{worst}}(G)), \quad (55)$$

where (49) and (55) apply the assumption  $|\text{UCB}_{t_i}(j) - \text{LCB}_{t_i}(j)| < \frac{\Delta_j}{2}$ , (50) and (54) use the confidence bounds, (51) and (53) use the definition of  $\Delta'_j$ , and (52) uses the definition of  $j^*$ . Since (55) implies the removal of all  $G$  where  $j \in G$ , we obtain that all of these suboptimal groups are eliminated, and hence scenario 3 is satisfied and  $j$  is no longer pulled.

Having handled both cases, we conclude that arm  $j$  is no longer pulled when  $U(T_j(t), \frac{\delta}{n}) < \frac{\Delta_j}{4}$ . Combining this with Lemma 2, we obtain the bound on number of arm pulls  $T_j(t)$  for each individual arm  $j$ :

$$T_j(t) \leq \frac{2\gamma}{\Delta_j^2} \log \frac{2 \log(\gamma(1+\epsilon)\Delta_j^{-2})}{\delta/n}. \quad (56)$$

Summing over the  $n$  arms, we obtain the bound on the total number of arm pulls in (15).

## B Proof of Thm. 2 (Regret Bound for STABLEOPT)

The first steps of the proof follow those of (Bogunovic et al. 2018). With probability at least  $1 - \frac{2+\epsilon}{\epsilon/2} \left( \frac{\delta}{\log(1+\epsilon)} \right)^{1+\epsilon}$ , the confidence bounds in (8)–(9) are uniformly valid, and we henceforth condition on this being the case. For the group  $G_t$  and corresponding arm  $j_t \in G_t$  selected in round  $t$ , we have:

$$r(G_t) = \max_{G \in \mathcal{G}} \min_{j \in G} \mu_j - \min_{j \in G_t} \mu_j \quad (57)$$

$$\leq \max_{G \in \mathcal{G}} \min_{j \in G} \mu_j - \min_{j \in G_t} \text{LCB}_{t-1}(j) \quad (58)$$

$$= \max_{G \in \mathcal{G}} \min_{j \in G} \mu_j - \text{LCB}_{t-1}(j_t) \quad (59)$$

$$\leq \max_{G \in \mathcal{G}} \min_{j \in G} \text{UCB}_{t-1}(j) - \text{LCB}_{t-1}(j_t) \quad (60)$$

$$= \min_{j \in G_t} \text{UCB}_{t-1}(j) - \text{LCB}_{t-1}(j_t) \quad (61)$$

$$\leq \text{UCB}_{t-1}(j_t) - \text{LCB}_{t-1}(j_t) \quad (62)$$

$$= 2U\left(T_{j_t}(t-1), \frac{\delta}{n}\right) \quad (63)$$

where (58) and (60) use the validity of the confidence bounds, (59) and (61) use the selection rules for  $j_t$  and  $G_t$ , and (63) uses the definitions of the confidence bounds.

Using the choice of  $G^{(T)}$  in (20), we further have:

$$r(G^{(T)}) \leq \max_{G \in \mathcal{G}} \min_{j \in G} \mu_j - \text{LCB}_{T-1}(j_T) \quad (64)$$

$$\leq \frac{1}{T} \sum_{t=1}^T \left( \max_{G \in \mathcal{G}} \min_{j \in G} \mu_j - \text{LCB}_{t-1}(j_t) \right) \quad (65)$$

$$\leq \frac{1}{T} \sum_{t=1}^T 2U(T_{j_t}(t-1), \frac{\delta}{n}), \quad (66)$$

where (64) follows from (58), (65) bounds the minimum by the average, and (66) follows from the argument leading to (63).

We observe from (7) that  $U(t, \frac{\delta}{n})$  decreases monotonically with respect to  $t$ .<sup>3</sup> Therefore,  $\sum_{t=1}^T U(T_{j_t}(t-1), \frac{\delta}{n})$  is highest when each arm is pulled the same number of times (up to rounding), i.e.,  $\sum_{t=1}^T U(T_{j_t}(t-1), \frac{\delta}{n}) \leq$

---

<sup>3</sup>An exception may be moving from  $t = 1$  to  $t = 2$ , but this does not affect our argument here.

$n(1 + \sum_{t=1}^{\lfloor \frac{T}{n} \rfloor} U(t, \frac{\delta}{n}))$ , where we recall that  $U(0, \delta) = 1$ . Hence:

$$\begin{aligned} & \frac{1}{T} \sum_{t=1}^N 2U\left(T_{j_t}(t-1), \frac{\delta}{n}\right) \\ & \leq \frac{2n}{T} \left(1 + \sum_{t=1}^{\lfloor \frac{T}{n} \rfloor} U\left(t, \frac{\delta}{n}\right)\right) \end{aligned} \quad (67)$$

$$\leq \frac{2n}{T} + \frac{2n}{T} \cdot C_1(\epsilon) \sum_{t=1}^{\lfloor \frac{T}{n} \rfloor} \sqrt{\frac{1}{2t} \log \frac{\log(1+\epsilon)t}{\frac{\delta}{n}}} \quad (68)$$

$$\begin{aligned} & \leq \frac{2n}{T} + \frac{2C_1(\epsilon)n}{T} \sum_{t=2}^{\lfloor \frac{T}{n} \rfloor} \left( \sqrt{\frac{1}{2t} \log \frac{n}{\delta}} \right. \\ & \quad \left. + \sqrt{\frac{1}{2t} \log(\log((1+\epsilon)t))} \right) \end{aligned} \quad (69)$$

$$\leq \frac{2n}{T} + \frac{2C_2(n, \delta, \epsilon, T)n}{T} \sum_{t=1}^{\lfloor \frac{T}{n} \rfloor} \sqrt{\frac{1}{t}} \quad (70)$$

$$\leq \frac{2n}{T} + \frac{4C_2(n, \delta, \epsilon, T)n}{T} \sqrt{\frac{T}{n}} \quad (71)$$

$$= \frac{2n}{T} + 4C_2(n, \delta, \epsilon, T) \sqrt{\frac{n}{T}}, \quad (72)$$

where (68) uses the definition of  $U$  and defines  $C_1(\epsilon) = (1 + \sqrt{\epsilon})\sqrt{1 + \epsilon}$ , (69) holds since  $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ , (70) defines  $C_2(n, \delta, \epsilon, T) = \frac{C_1(\epsilon)}{\sqrt{2}} (\sqrt{\log \frac{n}{\delta}} + \sqrt{\log(\log((1+\epsilon)T))})$ , and (71) uses the fact that  $\sum_{i=1}^k \frac{1}{\sqrt{i}} \leq 2\sqrt{k}$ .

Combining (66) and (72) and letting  $\epsilon \in (0, 1)$  be an arbitrary fixed constant gives the desired  $O(\sqrt{\frac{n}{T}} (\sqrt{\log \frac{n}{\delta}} + \log \log T))$  regret bound. The term  $\frac{2+\epsilon}{\epsilon/2} (\frac{\delta}{\log(1+\epsilon)})^{1+\epsilon}$  in the error probability is at most  $O(\delta)$  regardless of the choice of  $\epsilon \in (0, 1)$ .

## C Proofs of Algorithm-Independent Lower Bounds

### C.1 A Fundamental Auxiliary Lemma

We make use of a fundamental result introduced in (Kaufmann, Cappé, and Garivier 2016), which has subsequently been applied to numerous bandit settings. The following statement is somewhat different from that in (Kaufmann, Cappé, and Garivier 2016), and the differences are explained in Sec. C.2.

**Lemma 4.** (Implicit in (Kaufmann, Cappé, and Garivier 2016)) *Let  $\mathcal{A} = (a_1, \dots, a_n)$  and  $\mathcal{A}' = (a'_1, \dots, a'_n)$  be two distinct bandit instances such that for any arm pair  $(a_j, a'_j)$ , the corresponding distributions  $P_j$  and  $P'_j$  are mutually absolutely continuous. For any stopping time  $\sigma$  which is almost surely finite under instance  $\mathcal{A}$ , and any event  $\mathcal{E}$  depending only on the reward history up to the stopping time and satisfying  $\mathbb{P}_{\mathcal{A}}[\mathcal{E}] \in (0, 1)$ , we have*

$$\sum_{j=1}^n \mathbb{E}_{\mathcal{A}}[N_j(\sigma)] D(P_j \| P'_j) \geq d(\mathbb{P}_{\mathcal{A}}[\mathcal{E}], \mathbb{P}_{\mathcal{A}'}[\mathcal{E}]), \quad (73)$$

where  $d(x_1, x_2) = x_1 \log \frac{x_1}{x_2} + (1-x_1) \log \frac{1-x_1}{1-x_2}$  is the binary relative entropy function, with  $d(0, 0) = d(1, 1) = 0$ , and  $\mathbb{P}_{\mathcal{A}}$  denotes the probability under instance  $\mathcal{A}$ .

High-probability guarantees for MAB problems are based on attaining a small error probability for a suitably-defined notion of success; in our case, this is the identification of  $G^*$ . Hence, if  $\mathcal{E}$  is the event that the returned group is the best according to instance  $\mathcal{A}$ , then we should have  $\mathbb{P}_{\mathcal{A}}[\mathcal{E}] \geq 1 - \delta$  and  $\mathbb{P}_{\mathcal{A}'}[\mathcal{E}] \leq \delta$  given a target error probability  $\delta \in (0, \frac{1}{2})$ , as long as the best group in  $\mathcal{A}$  is not max-min optimal in  $\mathcal{A}'$ . Since  $d(x, 1-x) \geq \log \frac{1}{2.4x}$  for all  $x \in [0, 1]$  (Kaufmann, Cappé, and Garivier 2016), we can then simplify (73) to

$$\sum_{j=1}^n \mathbb{E}_{\mathcal{A}}[N_j(\sigma)] D(P_j \| P_{j'}) \geq \log \frac{1}{2.4\delta}. \quad (74)$$

Then, given a “base” instance  $\mathcal{A}$  with optimal group  $G^*$ , we are left to design another instance  $\mathcal{A}'$  such that  $G^*$  is suboptimal, ideally with each  $D(P_j \| P_{j'})$  being small so that (74) leads to a stronger lower bound on the number of arm pulls.

## C.2 Note on Lemma 4

Lemma 4 is slightly different from that in (Kaufmann, Cappé, and Garivier 2016), in that (i) the stopping time is only assumed to be almost-surely finite under  $\mathcal{A}$  but not necessarily under  $\mathcal{A}'$ , and (ii) we assume that  $\mathbb{P}_{\mathcal{A}}[\mathcal{E}] \in (0, 1)$ , rather than allowing all of  $[0, 1]$ .

To understand this difference, we note that in (Kaufmann, Cappé, and Garivier 2016), the almost-sure finite stopping time is used for two purposes: To apply Wald’s lemma to a sum of log-likelihood ratios under instance  $\mathcal{A}$ , and to prove that  $\mathbb{P}_{\mathcal{A}}[\mathcal{E}] = 0 \iff \mathbb{P}_{\mathcal{A}'}[\mathcal{E}] = 0$  (and similarly if both 0s are replaced by 1s). The former only requires the stopping time to be almost-surely finite under  $\mathcal{A}$ . As for the latter, the proof in (Kaufmann, Cappé, and Garivier 2016) establishes that if  $\sigma$  is almost-surely finite under  $\mathcal{A}$ , then it holds that  $\mathbb{P}_{\mathcal{A}'}[\mathcal{E}] = 0 \implies \mathbb{P}_{\mathcal{A}}[\mathcal{E}] = 0$ , or equivalently  $\mathbb{P}_{\mathcal{A}}[\mathcal{E}] \in (0, 1) \implies \mathbb{P}_{\mathcal{A}'}[\mathcal{E}] \in (0, 1)$ . We do not require the reverse implication, because we already explicitly assume that  $\mathbb{P}_{\mathcal{A}}[\mathcal{E}] \in (0, 1)$ .

## C.3 Proof of Thm. 3

As suggested by Lemma 4, we prove Thm. 3 by taking the given instance with optimal group  $G^*$ , and shifting one or more of its arms (from  $\mu_j$  to  $\mu'_j$ ) in a way that ensures that  $G^*$  is suboptimal in the new instance.

Without loss of generality, assume that in the original instance,  $G_1 = G^*$  is the optimal group, and  $G_2$  is the second best group. We consider the two cases in the theorem statement as follows.

**Case 1** ( $j \in G_1$ ) For a fixed arm  $j$ , we define an instance  $\mathcal{A}^{(j)}$  such that the arm means  $\mu_i$  are unchanged for all  $i \neq j$ , and where  $\mu_j$  changes to another value  $\mu'_j$ ; the corresponding distributions are denoted by  $P_j$  and  $P'_j$ . Specifically, we choose  $\mu'_j = \mu_j - (1 + \alpha)(\Delta'_j + \Delta_0)$  for some arbitrarily small  $\alpha > 0$ . By the definitions of  $\Delta'_j$  and  $\Delta_0$  in Sec. 2.1, the choice  $\alpha = 0$  would make  $\mu'_j$  exactly equal to  $\mu_{j_{\text{worst}}(G_2)}$  (the subtraction of  $\Delta'_j$  aligns the mean with  $\mu_{j_{\text{worst}}(G_1)}$ , and the subtraction of  $\Delta_0 = \mu_{j_{\text{worst}}(G_1)} - \mu_{j_{\text{worst}}(G_2)}$  further shifts this to  $\mu_{j_{\text{worst}}(G_2)}$ ). Hence, no matter how small  $\alpha > 0$ , we have that  $\mu'_j$  is strictly smaller than  $\mu_{j_{\text{worst}}(G_2)}$ , so that  $G_1$  is suboptimal in the new instance.

Hence, applying Lemma 4 with  $\mathcal{E}$  being the event of outputting  $G_1$ ,<sup>4</sup> we obtain the following lower bound for number of pulls of  $j \in G_1$ :

$$\mathbb{E}_{\mathcal{A}}[N_j(\sigma)] \geq \frac{\log \frac{1}{2.4\delta}}{D(P_j \| P'_j)} \quad (75)$$

since  $D(P_i \| P'_i) = 0$  for all  $i \neq j$ . Upper bounding the denominator via Assump. 4 and using the fact that  $\alpha$  can be arbitrarily small, we obtain the desired bound (22).

**Case 2** ( $j \notin G_1$ ). Let  $G$  be any suboptimal group. Due to the max-min nature of the problem, pushing a *single* arm’s mean up, even by an arbitrarily large amount, may fail to make  $G$  a better group than  $G_1$ . Instead, we need to shift *all arms with mean at most*  $\mu_{j_{\text{worst}}(G_1)}$  up to a value strictly above  $\mu_{j_{\text{worst}}(G_1)}$ . To achieve this, we set  $\mu'_j = \mu_j + (1 + \alpha)(\mu_{j_{\text{worst}}(G^*)} - \mu_j)$  for arbitrarily small  $\alpha > 0$ . For any arms in  $G$  with mean exactly  $\mu_{j_{\text{worst}}(G^*)}$ , we can perform an arbitrarily small perturbation similar to Appendix A of (Scarlett, Bogunovic, and Cevher 2019). As a result,  $G_1$  is no longer the best group in the new instance.

Applying Lemma 4, we obtain

$$\sum_{j \in G : \mu_j < \mu_{j_{\text{worst}}(G^*)}} \mathbb{E}[N_j(\sigma)] \cdot D(P_j \| P'_j) \geq \log \frac{1}{2.4\delta}, \quad (76)$$

where  $P_j$  and  $P'_j$  are the distributions in the original and modified instances. Applying Assump. 4 and the fact that  $\alpha$  is arbitrarily small, we obtain the lower bound (23) on the total number of arm pulls within group  $G$ .

<sup>4</sup>The condition  $\mathbb{P}_{\mathcal{A}}[\mathcal{E}] \in (0, 1)$  in the lemma is satisfied under our assumptions. Specifically, Assump. 3, and Assump. 4 ensure that the algorithm cannot have an error probability of zero. (The assumption of  $\mathcal{A}$  being identifiable rules out trivial cases such as only having one group, or all groups being identical.)



## C.4 Proof of Cor. 2

The first term in (24) follows immediately by summing over  $j \in G^*$  in the first case of Thm. 3, so it remains to establish the second term.

By the definition  $\Delta_G = \mu_{j_{\text{worst}}(G^*)} - \mu_{j_{\text{worst}}(G)}$ , the inequality (23) gives for any  $G \neq G^*$  that

$$\begin{aligned} \sum_{j \in G} \mathbb{E}_{\mathcal{A}}[N_j(\sigma)] &\geq \sum_{j \in G : \mu_j < \mu_{j_{\text{worst}}(G^*)}} \mathbb{E}_{\mathcal{A}}[N_j(\sigma)] \\ &\geq \frac{\log \frac{1}{2.4\delta}}{\tilde{C}\Delta_G^2}, \end{aligned} \quad (77)$$

since the definition of  $\Delta_G$  ensures that all gaps appearing in (23) are at most  $\Delta_G$ .

We observe that (77) provides a group-wise lower bound. In a disjoint grouping setup, a simple summation over each group-wise lower bound produces a valid lower bound on total arm pulls for the instance  $\mathcal{A}$ . However, in an instance with overlapping groups, we cannot simply sum the group-wise lower bounds in this way. This is because the overlaps between groups can cause potential double (or triple, etc.) counting of  $\mathbb{E}_{\mathcal{A}}[N_j(\sigma)]$  for some arms  $j$  in the summation.

To resolve this issue, we use the assumption that each arm can be in at most  $m$  groups (with  $m = 1$  amounting to disjoint groups). Dividing the group-wise bound by  $m$  accounts for any potential multiple-counting when computing the lower bound on total arm pulls upon adding up the group-wise bounds. Thus, we can weaken (77) to

$$\sum_{j \in G} \mathbb{E}_{\mathcal{A}}[N_j(\sigma)] \geq \frac{1}{m} \sum_{G: G \neq G^*} \frac{\log \frac{1}{2.4\delta}}{\tilde{C}\Delta_G^2}, \quad (78)$$

with the important difference that it is now valid to further sum over groups; doing so gives the second term in (24) as desired.

## D Note on the Original Version of STABLEOPT

Recall that the general STABLEOPT formulation is given in (16). A connection between (16) and a certain grouped max-min problem was already discussed in (Bogunovic et al. 2018), focusing on non-overlapping groups. In particular, it was noted that the interplay between  $x$  and  $\delta$  does not need to correspond to addition, and accordingly, we can replace  $(x, \delta)$  by  $(G, j)$  and transform (16) as follows:

$$G^* \in \arg \max_{G \in \mathcal{G}} \min_{j \in G} f(j). \quad (79)$$

In our setting, we take  $f(j) = \mu_j$ , i.e., the mean of the arm.

The theory in (Bogunovic et al. 2018) assumed that  $f(x)$  has a bounded norm in a Reproducing Kernel Hilbert Space (RKHS) corresponding to some kernel function  $k(x, x')$ . To produce our setting with independent arms, we can choose the 0-1 kernel  $k(j, j') = \mathbf{1}\{j = j'\}$ , and the RKHS norm reduces to  $\|f\|_k = \sqrt{\sum_{i=1}^n \mu_j^2} \leq \sqrt{n}$ .

While we can apply the main result of (Bogunovic et al. 2018) to deduce an instance-independent  $O(\frac{1}{\sqrt{T}})$  bound on the regret after  $T$  arm pulls, the dependence of the implied constants on the number of arms  $n$  is highly suboptimal. This is because both the squared RKHS norm  $\|f\|_k^2$  and the fundamental *information gain* quantity in (Bogunovic et al. 2018) scale linearly with  $n$ . Fortunately, we can sharpen the dependence of the regret on  $n$  by suitably adapting the analysis in a manner more directly targeted at our setup, as detailed in Sec. 4 and Appendix B.

## E Further Experimental Details and Results

### E.1 Details of Instance Generation

Here we describe more precisely how the groups are generated in our experiments, and how the arm means are assigned.

We allocate each arm into each group independently with probability  $1/10$ , so that each group contains 10 arms on average. For any subsequently non-allocated arms, we assign it to a uniformly random group. In addition, for any empty group, we assign 10 uniformly random arms into it.

We arbitrarily choose the first group to be the optimal one, and set its worst arm  $j_0$  to have a mean reward of 0.5 (here  $j_0$  is chosen to ensure that  $G^*$  is unique). We further choose the second group to be a suboptimal group whose

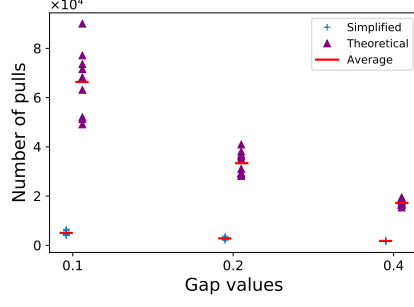


Figure 4: Comparison of theoretical and simplified choices of confidence bounds.

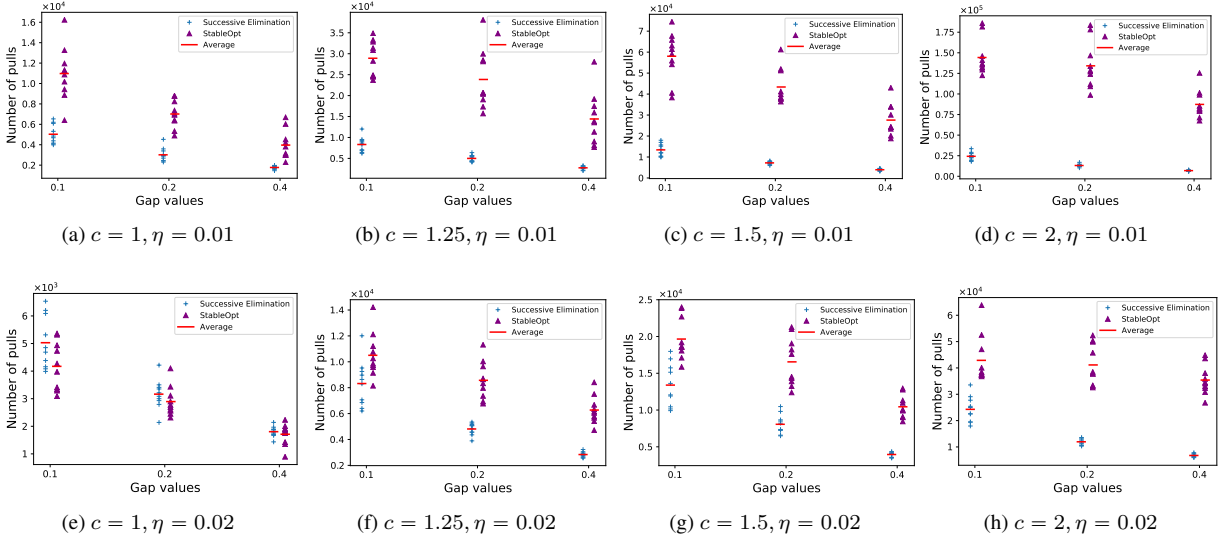


Figure 5: Comparisons of number of arm pulls for various  $c$  (controlling the confidence width),  $\eta$  (controlling the STABLEOPT stopping condition, and  $\Delta$  (gap value).)

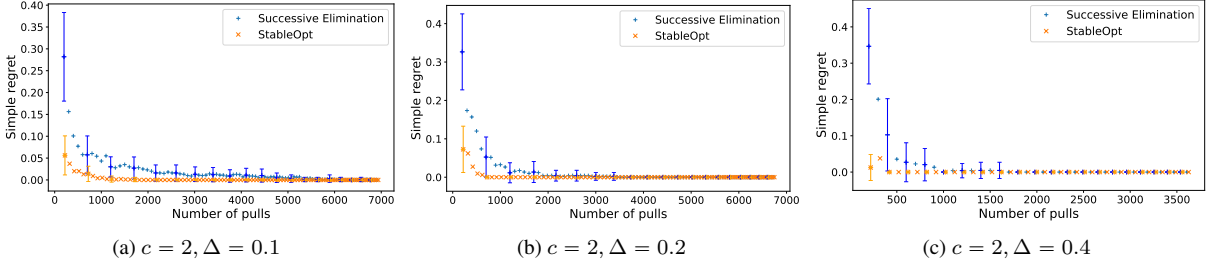


Figure 6: Simple regret plots with  $c = 2$  and various gap values.

worst arm  $j_1$  meets the gap value exactly, i.e.  $\mu_{j_0} - \mu_{j_1} = \Delta$ . Then, we generate other groups' worst arms to be uniform in  $[0, \mu_{j_1}]$ . Finally, we choose the means for the remaining arms to be uniform in  $[\mu_{j_G}, 1]$ , where  $j_G$  is the relevant worst arm in the relevant group  $G$ .

## E.2 Further Experiments

**Theoretical Confidence Bounds.** Here we compare our simplified choice of confidence width,  $\frac{1}{\sqrt{T_j(t)}}$ , to the theoretical choice in Sec. 2.3. The comparison is given in Fig. 4, where we observe that the former requires fewer arm pulls and is less prone to runs with an unusually high number of pulls, suggesting that the theoretical choice may be overly conservative. For both choices, there were no failures (i.e., returning the wrong group) in any of the runs performed, both here and in the experiments in Sec. 7.

Table 1: Empirical success rates for various gap values and  $(c, \eta)$ -value pairs

Model	$\Delta = 0.1$	$\Delta = 0.2$	$\Delta = 0.4$
Elimination ( $c = 1$ )	1.0	1.0	1.0
STABLEOPT ( $c = 1, \eta = 0.01$ )	0.98	0.99	1.0
STABLEOPT ( $c = 1, \eta = 0.02$ )	0.91	1.0	1.0
Elimination ( $c = 1.25$ )	1.0	1.0	1.0
STABLEOPT ( $c = 1.25, \eta = 0.01$ )	1.0	1.0	1.0
STABLEOPT ( $c = 1.25, \eta = 0.02$ )	1.0	1.0	1.0
Elimination ( $c = 1.5$ )	1.0	1.0	1.0
STABLEOPT ( $c = 1.5, \eta = 0.01$ )	1.0	1.0	1.0
STABLEOPT ( $c = 1.5, \eta = 0.02$ )	1.0	1.0	1.0
Elimination ( $c = 2$ )	1.0	1.0	1.0
STABLEOPT ( $c = 2, \eta = 0.01$ )	1.0	1.0	1.0
STABLEOPT ( $c = 2, \eta = 0.02$ )	1.0	1.0	1.0

**Varying  $c$  and  $\eta$ .** Here we explore the effect of varying  $c$ , the constant in the confidence width  $\frac{c}{\sqrt{T_j(t)}}$  (previously set to one), and  $\eta$ , the confidence width beyond which STABLEOPT terminates (previously set to 0.01).

In the top row of Fig. 5, we see that increasing  $c$  naturally increases the number of arm pulls for both algorithms (due to more conservative confidence bounds), but appears to impact STABLEOPT more. However, the second row indicates that this is at least partly due to the stringent stopping condition, since the less stringent choice  $\eta = 0.02$  brings the two algorithms back closer together.

A caveat here is that increasing  $\eta$  puts STABLEOPT at a higher risk of returning the wrong group; we investigate this in Table 1. For the most part, the algorithms return the correct group on all 100 trials, but STABLEOPT indeed starts to produce errors when both  $c$  and  $\eta$  are chosen too aggressively, particularly  $c = 1$  and  $\eta = 0.02$ .

Finally, in Fig. 6, we plot the simple regret with  $c = 2$ , in contrast to  $c = 1$  used in Fig. 3 and Fig. ???. Again, increasing  $c$  naturally increases the number of arm pulls for both algorithms, but we observe the same general behavior for both values of  $c$ . In general, our findings suggest that STABLEOPT is highly effective in providing small simple regret, but that more care is needed (compared to Successive Elimination) in choosing the confidence bounds and stopping rule when the goal is exact best-group identification.

**Acknowledgment.** This work was supported by the Singapore National Research Foundation (NRF) under grant number R-252-000-A74-281.

## References

- Audibert, J.-Y.; Bubeck, S.; and Munos, R. 2010. Best arm identification in multi-armed bandits. In *Conference on Learning Theory*, 41–53.
- Ban, Y.; and He, J. 2021. Local clustering in contextual multi-armed bandits. <https://arxiv.org/abs/2103.00063>.
- Bertsimas, D.; Nohadani, O.; and Teo, K. M. 2010. Nonconvex robust optimization for problems with constraints. *INFORMS journal on Computing*, 22(1): 44–58.
- Bogunovic, I.; Mitrović, S.; Scarlett, J.; and Cevher, V. 2017. Robust submodular maximization: A non-uniform partitioning approach. In *International Conference on Machine Learning*.
- Bogunovic, I.; Scarlett, J.; Jegelka, S.; and Cevher, V. 2018. Adversarially robust optimization with Gaussian processes. In *Conference on Neural Information Processing Systems*.
- Bouneffouf, D.; Parthasarathy, S.; Samulowitz, H.; and Wistub, M. 2019. Optimal exploitation of clustering and history information in multi-armed bandit. <https://arxiv.org/abs/1906.03979>.
- Bubeck, S.; Wang, T.; and Viswanathan, N. 2013. Multiple identifications in multi-armed bandits. In *International Conference on Machine Learning*, volume 28, 258–265. PMLR.
- Chen, R.; Lucier, B.; Singer, Y.; and Syrgkanis, V. 2017. Robust optimization for non-convex objectives. <https://arxiv.org/abs/1707.01047>.
- Gabillon; Victor; Ghavamzadeh; Mohammad; and Lazaric, A. 2012. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Conference on Neural Information Processing Systems*.
- Gabillon, V.; Ghavamzadeh, M.; Lazaric, A.; and Bubeck, S. 2011. Multi-bandit best arm identification. In *Conference on Neural Information Processing Systems*, volume 24.

- Garivier, A.; and Kaufmann, E. 2016. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, 998–1027.
- Jamieson, K.; and Nowak, R. 2014. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *Conference on Information Sciences and Systems*, 1–6.
- Jedor, M.; Perchet, V.; and Louedec, J. 2019. Categorized Bandits. In *Conference on Neural Information Processing Systems*.
- Kalyanakrishnan, S.; Tewari, A.; Auer, P.; and Stone, P. 2012. PAC subset selection in stochastic multi-armed bandits. In *International Conference on Machine Learning*, 227–234.
- Katz-Samuels, J.; and Jamieson, K. 2020. The true sample complexity of identifying good arms. In *International Conference on Artificial Intelligence and Statistics*.
- Kaufmann, E.; Cappé, O.; and Garivier, A. 2016. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1): 1–42.
- Kaufmann, E.; and Kalyanakrishnan, S. 2013. Information Complexity in Bandit Subset Selection. In *Conference on Learning Theory*, volume 30, 228–251. PMLR.
- Krause, A.; McMahan, H. B.; Guestrin, C.; and Gupta, A. 2008. Robust Submodular Observation Selection. *Journal of Machine Learning Research*, 9(12).
- Lattimore, T.; and Szepesvári, C. 2020. *Bandit Algorithms*. Cambridge University Press.
- Mannor, S.; and Tsitsiklis, J. N. 2004. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun): 623–648.
- Orlin, J. B.; Schulz, A. S.; and Udwani, R. 2018. Robust monotone submodular function maximization. *Mathematical Programming*, 172(1): 505–537.
- Scarlett, J.; Bogunovic, I.; and Cevher, V. 2019. Overlapping multi-bandit best arm identification. In *International Symposium on Information Theory*, 2544–2548. IEEE.
- Singh, R.; Liu, F.; Sun, Y.; and Shroff, N. 2020. Multi-armed bandits with dependent arms. <https://arxiv.org/abs/2010.09478>.