Max-Min Grouped Bandits

Zhenlin Wang,[1] **Jonathan Scarlett**[1,2]

[1]School of Computing, National University of Singapore
[2]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 the group whose worst arm has the highest mean reward. This problem is of interest in applications such as recommendation systems and resource allocation, 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 than 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.

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) (see also (Yang and Ren 2021) for a related follow-up work). 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 j up to time t. We consider stochastic rewards, in which for each $j \in \mathcal{A}$, the random variables $\{X_{j,\tau}\}_{\tau \geqslant 1}$ are independently drawn from some unknown distribution with mean μ_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:

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

To lighten notation, we define $j_{\rm worst}(G) = \arg\min_{j \in G} \mu_j$ 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

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

and the simple regret

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

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* $[0, 1]$,[1] *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* $\mathbb{E}[e^{\lambda(X_j - \mu_j)}] \leqslant \exp(\lambda^2 R^2/2)$ *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.,*

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

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.

[1] Any finite interval can be shifted and scaled to this range.

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:

- $\Delta'_j := \min_{G\,:\,j \in G} \big(\mu_j - \mu_{j_{\text{worst}}(G)}\big)$ 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:

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

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 Δ_j.

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 $\mu_1 = 1 > ... > \mu_{k-1} = 0.9 > \mu_k = 0.8$ and $\mu_{k+1} = 0.1 > ... > \mu_{n-1} = 0.01 > \mu_n = 0.00999$. 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 $\sigma \leqslant \frac{1}{2}$. For any $\epsilon \in (0,1)$ and $\delta \in (0, \frac{1}{e}\log(1+\epsilon))$, with probability at least $1 - \frac{2+\epsilon}{\epsilon/2}\big(\frac{\delta}{\log(1+\epsilon)}\big)^{1+\epsilon}$, we have*

$$\left|\frac{1}{t}\sum_{s=1}^{t} Z_s - \mu\right| \leqslant U(t,\delta) \;\; \forall t \geqslant 1, \tag{6}$$

where

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

In accordance with this result, we henceforth assume that $R \leqslant \frac{1}{2}$ 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 δ 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:

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

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

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

$$\text{LCB}_t(j) \leqslant \mu_j \leqslant \text{UCB}_t(j),\; \forall j \in \{1, ..., n\}, t \geqslant 1. \tag{10}$$

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*

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

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.

Algorithm 1: Successive Elimination algorithm

```
Require: Arms $(a_1, ..., a_n)$, set of groups $\mathcal{G}$, parameters $\delta,\ \epsilon > 0$
Initialize $i = 0, t = 0$ and $T_j(t) = 0$ for all $j$
Set $m_0^{(G)} = G$ for all $G \in \mathcal{G}$, $\mathcal{C}_0 = \mathcal{G}, \mathcal{A}_0 = \{1, 2, ..., n\}$
while $|\mathcal{C}_i| > 1$ do
    Pull every arm $j$ in $\mathcal{A}_i$ once, incrementing $t$ after each pull and updating all $T_j(t)$
    Compute $m_{i+1}^{(G)},\ \mathcal{C}_{i+1}$ and $\mathcal{A}_{i+1}$ via expressions (12), (13) and (14)
    Increment round index $i$ by 1
return $\hat{\mathcal{C}} = \mathcal{C}_i$
```

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

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

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:

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

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.

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* $1 - \frac{2+\epsilon}{\epsilon/2}\big(\frac{\delta}{\log(1+\epsilon)}\big)^{1+\epsilon}$*, Alg. 1 identifies the optimal group and uses a number of arm pulls satisfying*

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

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

The proof is given in Appendix A, and is based on considering the gap Δ_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 $\delta_0 = \frac{2+\epsilon}{\epsilon/2}\big(\frac{\delta}{\log(1+\epsilon)}\big)^{1+\epsilon}$ in Thm. 1 is somewhat complicated, one can fix $\epsilon = \frac{1}{2}$ (say) and solve for δ, and it readily follows that the right-hand side of (15) has the standard $O\big(\log\frac{1}{\delta_0}\big)$ 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 ϵ-stable optimal input for some function f:

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

where $\Delta_\epsilon(x) = \{x' - x : x \in D_x \text{ and } d(x, x') \leqslant \epsilon\}$ 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 $x_t = \operatorname*{arg\,max}_{x \in D_x} \min_{\delta \in \Delta_\epsilon(x)} \mathrm{UCB}_{t-1}(x + \delta)$, where $\delta_t = \operatorname*{arg\,min}_{\delta \in \Delta_\epsilon(x_t)} \mathrm{LCB}_{t-1}(x_t + \delta)$, for suitably defined confidence bounds UCB_t and LCB_t. When adapted to our formulation (1), the algorithm becomes the following:

$$G_t = \operatorname*{arg\,max}_{G \in \mathcal{G}} \min_{j \in G} \mathrm{UCB}_{t-1}(j) \quad (17)$$

$$j_t = \operatorname*{arg\,min}_{j \in G_t} \mathrm{LCB}_{t-1}(j), \quad (18)$$

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:

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

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)

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

Here and subsequently, we define $\mathrm{LCB}_0(j) = 0$ and $\mathrm{UCB}_0(j) = 1$ 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 ϵ 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*

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

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 $\frac{1}{T}\sum_{t=1}^{T} 2\,U(T_{j_t}(t-1), \frac{\delta}{n})$, 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 $\Delta_j \geqslant \Delta_{\min}$ 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 $T \geqslant \Omega^*\big(\frac{n \log\frac{n}{\delta}}{\Delta_{\min}^2}\big)$ 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 $D(P_j \| P_{j'}) \to 0$ in the limit as $\mu_{j'}$ approaches μ_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 μ_j and $\mu_{j'}$, it holds that $D(P_j \| P_{j'}) \leqslant \tilde{C}(\mu_j - \mu_{j'})^2$.*

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 δ-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 δ 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 δ-successful with respect to the instances satisfying Assump. 1, Assump. 3, and Assump. 4. Fix any identifiable instance $\mathcal{A} = (a_1, ..., a_n)$ with distributions $(P_1, ..., P_n)$ and a specified grouping $\mathcal{G} = (G_1, ..., G_m)$. 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*

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

where $\tilde{C}$ appears in Assump. 4, and Δ'_j and Δ_0 are defined in Sec. 2.1.

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

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

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*

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

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 Δ_G appears in Cor. 2. Hence, the gaps Δ_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 Δ_j is the one with the smallest Δ'_j (the other two quantities Δ''_j and Δ_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 Δ_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, $O\big(\log \frac{1}{\delta}\big)$ of them if we require an error probability of at most δ) 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 above 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

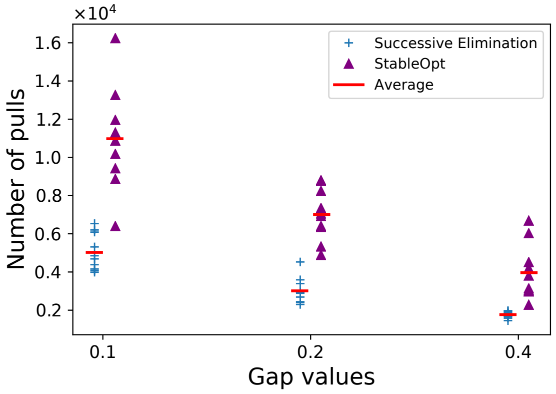


Figure 1: Plot of total arm pulls used for Successive Elimination and StableOpt for $\Delta \in \{0.1, 0.2, 0.4\}$.

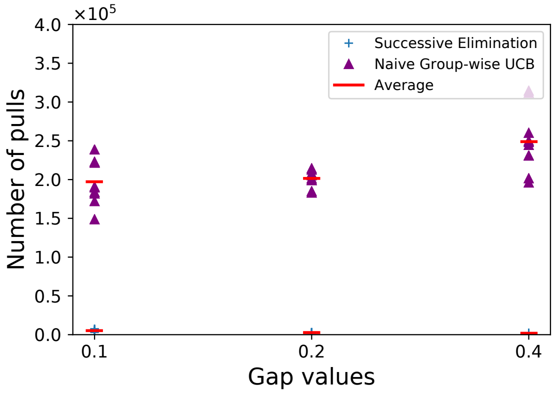


Figure 2: Comparison of Successive Elimination and the naive group-wise strategy for $\Delta \in \{0.1, 0.2, 0.4\}$.

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 ϵ, and only requiring to identify a group whose worst arm mean is within ϵ of the highest possible. In this case, any gap values Δ_j that are below ϵ get capped to ϵ in the upper bound in 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.

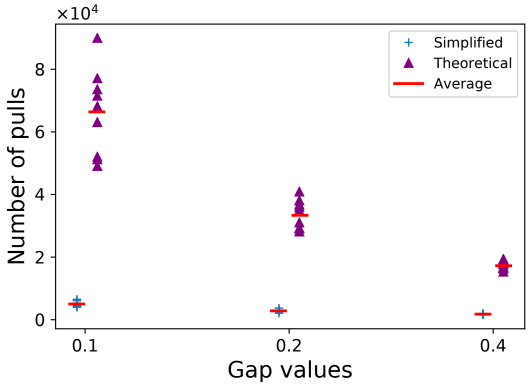


Figure 3: Comparison of theoretical and simplified choices of confidence bounds.

7.1 Experimental Setup

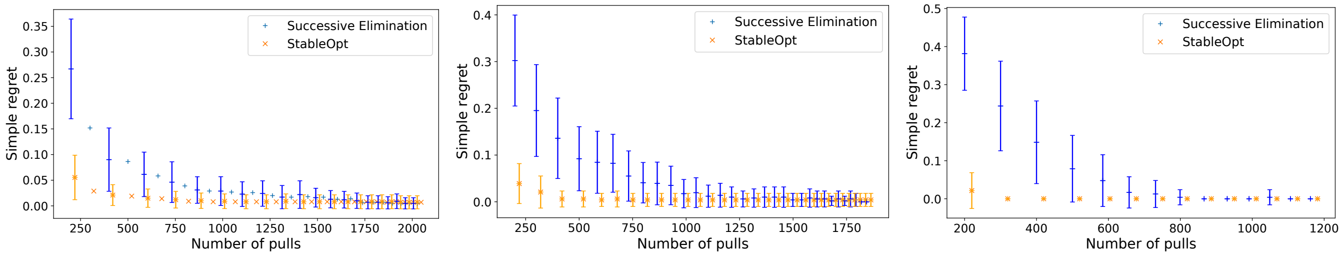
In each experiment, we generate 10 instances, each containing 100 arms and 10 possibly overlapping groups. The arm rewards are Bernoulli distributed, and the instances are generated to ensure a pre-specified minimum gap value (Δ) as per (5), and we consider $\Delta \in \{0, 1.0.2, 0.4\}$. The precise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

Details of groups and rewards. 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 arbitrary 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 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.

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 $c = 1$. Different 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 be only requiring it to hold to within a specified tolerance, denoted by η. We set $\eta = 0.01$



(a) $\Delta = 0.1$

(b) $\Delta = 0.2$

(c) $\Delta = 0.4$

Figure 4: Simple regret plots with various gap values.

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.

7.2 Results

Effect of Δ. From Fig. 1, we observe that the number of arm pulls decreases when the gap Δ increases, particularly for Successive Elimination.[2] 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 Δ increases.

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. 3, 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.

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 Δ is larger; see Fig. 4 (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 attains smaller simple regret in Fig. 4 is that it more quickly steers towards the more promising groups, due to its method of selecting the group with the highest upper confidence bound. In contrast, Successive Elimination always treats every non-eliminated arm equally, and the simple regret only decreases when eliminations occur.

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 η).

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 the StableOpt algorithm (Bogunovic et al. 2018). In addition, we provided an algorithm-independent lower bound, identified some of the potential weaknesses in the bounds, and discussed the difficulties in overcoming them.

We believe that this work leaves open several interesting avenues for further work. For instance:

- Following our discussion in Section 6, it would be of considerable interest to devise improved algorithms that do not necessarily pull every arm.
- Similarly, we expect that it should be possible to establish improved lower bounds that better capture certain difficulties, such as finding a single bad arm in a large group of otherwise good arms.
- Finally, one could move from the max-min setting to more general settings in which a single bad arm does not necessarily make the whole group bad, e.g., by considering a given quantile within the group.

Acknowledgments

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

[2] Error probabilities are not shown, because there were no failures in any of the trials here.

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*.

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.

Yang, J.; and Ren, S. 2021. Robust Bandit Learning with Imperfect Context. Https://arxiv.org/abs/2102.05018.