MULTI-LEVEL REGRET MINIMIZATION FOR MULTI-ARMED BANDITS

Sunday 29th November, 2020

ABSTRACT

In this paper, motivated by various applications, we study multi-level regret minimization in multi-armed bandits. In contrast to the classic single-objective multi-armed bandits model, the multi-level objective requires an agent to make sequential decisions based on multiple objectives ordered in decreasing priorities. Our main result is a near-complete characterization of the trade-off between keeping the regrets on all objectives small. In particular, we show a near-optimal algorithm that nearly realizes any regret target from the regret Pareto front underlying the componentwise worst-case regret of all algorithms. We also show that linear scalarization provably falls short of achieving the optimal tradeoffs, no matter the scalarization coefficients. We complement the theoretical results with experiments that illustrate the strength of the proposed method. Our results provide an alternative approach towards online learning with multi-objective bandit feedback and thus may be a useful complement to existing approaches.

1 Introduction

In this paper, we study multi-level regret minimization in the multi-armed bandits (MABs). In this problem, there are k arms, each of which is associated with l different objectives sorted in priority order. The goal is to maximize the total reward on each objective in a multi-level structure: maximizing higher priority objectives is the primary goal; the rewards of lower priority ones order only those arms that are equally good on all higher priority objectives. This problem strictly generalizes the classical multi-arm bandit problem which models a sequential decision-making problem about optimizing a single objective. Yet, in many real-world situations, people need to make decisions based on a number of different objectives (with different priority) simultaneously. For instance, when trading stocks, a trader not only wants to maximize the overall return but also minimize the probability of loss in each day [Huo and Fu, 2017]; when doing clinical trials, we not only care about the overall treating efficacy but also want to minimize side effects [Bastani and Bayati, 2015]. In games, the primary goal is to maximize the chance of winning the game, while a secondary objective is to keep the games short when the game is won. (Further applications are described by Tekin and Turğay [2018].) The multi-level objectives arise naturally and inspire the multi-level regret minimization formulation.

There is a long line of research in the literature about multi-objective optimization in bandits. For example, Drugan and Nowe [2013], Yahyaa et al. [2014b,a,b], Agarwal et al. [2019], Yahyaa and Manderick [2015], Yahyaa et al. [2015], Castejón et al. [2010], Yahyaa and Drugan [2015] consider a similar problem without putting priorities on the objectives. In this approach, if an algorithm maximizes the expected total cumulative reward for at least one objective, it will be considered *optimal*. However, it is often the case that priorities of each objective are specified. In the stock trading problem, maximizing profit is the primary objective and minimizing the probability of large daily losses is a secondary objective. Although these objectives are often conflicting, the priority ordering allows the decision maker to slightly relax the optimality requirement on the primary objective and do better on the second objective. Most importantly, in this problem, the multi-level regret minimization problem formulation is arguably more consistent with the goal of a decision maker.

In this paper, we pursue a systematic study of multi-level regret minimization in bandits. We show an interesting trade-off amongst the worst-case regrets of different objectives. For instance, in the case of two objectives, where the primary objective is to optimize the expected immediate reward and the secondary objective is to minimize the probability of each immediate reward being smaller than 0 over n rounds, we show that if an algorithm achieves a worst-case regret bound $O(r\sqrt{n})$ for the first objective for some $r \in (0, \sqrt{n})$, then its worst-case regret on the second objective is $\Omega(n/r^2 + \sqrt{n})$. To describe the trade-offs among worst-case regrets of multiple objectives, we introduce the notion of *regret Pareto front*. Each point on the regret Pareto front corresponds to some optimal trade-off among worst-case regrets. We also design an algorithm family with the property that every regret vector on the Pareto front, with an appropriate choice of the parameters, the algorithm's componentwise worst-case regret is provably close to the chosen regret vector.

As usual, lower bounds are shown via results on sample complexity to distinguish between distributions in terms of their KL divergence. To match the lower bound, we propose a new algorithm. This algorithm is a sequential phased elimination algorithm. In each stage of the algorithm, one objective is optimized and arms that are found to be sub-optimal according to that objective are eliminated. The number of rounds for each stage reflects the importance of the objective optimized in that stage. Input parameters of the algorithm determine the allocation of number of rounds for different stages. This algorithm differs significantly from existing scalarization algorithms, such as that of Drugan and Nowe [2013]. In contrast to scalarization algorithms, the new algorithm enables us to fully traverse the regret Pareto front. For any regret vector on the Pareto front, we show how to choose the input parameters so that the resulting worst-case regrets are close to that regret vector.

In fact, using another reduction to the problem of distinguishing between similar distributions, we further show that for some problems, linearly scalarization cannot achieve some regret vectors on the Pareto front.

In summary, we make the following contributions for the multi-level regret minimization problem for MAB: (1) A lower bound that characterizes the different necessary trade-offs among the worst-case regrets of objectives. (2) A class of algorithms that realizes all possible trade-offs (up to logarithmic factors) given by the lower bound. (3) A lower bound on the worst-case regrets of algorithms that optimize a linearly scalarized objective. This lower bound shows that the trade-off due to the linear scalarization approach is strictly worse than the best trade-off that is possible to achieve.

1.1 Related Works

MAB problems The multi-arm bandit (MAB) problem has been extensively studied in the literature, which dates back to the 1930s (Thompson [1933]). More recently, there are a number of papers studying the finite sample/regret bounds with various different strategies, e.g., upper confidence bounds (UCB; Bubeck and Cesa-Bianchi [2012], Cappé et al. [2013]) or Thompson sampling (Agrawal and Goyal [2012]). These methods give a nearly complete understanding of the complexity of a multi-arm bandit problem. A less related extension is the contextual MAB problem, which generalizes the MAB to the setting where actions can be infinite but possess certain structures. For instance, Li et al. [2010], Dani et al. [2008], Abbasi-Yadkori et al. [2011] give efficient algorithms with regret bounds not depending on the number of actions. For a more detailed survey, please refer to Lattimore and Szepesvári [2020] and references therein.

Multi-objective MAB The works that directly relate to this paper are those of multi-objective MAB. Bandits with knapsacks studies the special case when some objectives are constraining the optimization of the other objectives [Badanidiyuru et al., 2013, Agrawal et al., 2016]. There are also a number of papers studying the problem from the perspective of "Pareto regret", which treats all the arms on the Pareto front of the reward vectors equivalent. For instance, Drugan and Nowe [2013], Yahyaa et al. [2014b,a], Yahyaa and Manderick [2015], Yahyaa and Drugan [2015], Castejón et al. [2010], Pires et al. [2004], Lacerda [2017], Drugan and Nowé [2014], Durand et al. [2014] propose algorithms that achieve near-optimal Pareto regret. This setting is different from ours since the objectives in our case are of different priorities. A more closely related work is that of Tekin and Turğay [2018], who study the case when one objective is dominating in a nonparametric contextual stochastic multi-objective bandit. However, they are only studying two objectives. For this case they propose a "UCB"-type algorithm that switches to the secondary objective when the confidence width for the arm selected on the primary objective is small enough. They prove upper bounds on the regret for both the first and second objective, and also on the Pareto regret, but, unlike the main result this paper, they do not attempt to give a characterization of the trade-offs between objectives. Their main regret bound, when

simplified to fit our setting is that their algorithm achieves $O(T^{2/3})$ regret on both objectives. As we shall see, this is indeed in line with our results. We are not aware of other works studying a similar setting.

2 Problem Settings

2.1 Notation and definitions

We start with introducing some notations and definitions that we will use throughout the paper. We use $X \sim P$, to denote that the random element, X, follows distribution P. For a positive integer k, we let $[k] = \{1, \dots, k\}$. Somewhat unconventionally, for $x \in \mathbb{R}^l$, we use $x^{(i)}$ to denote the ith component of x.

2.2 Problem Definition

To formally state our problem, let k denote the number of arms and let l be the number of objectives. A bandit instance is given by $\nu = (\nu_a)_{a\in[k]}$ where ν_a is a probability distribution over $\mathbb{R}^l$. A bandit algorithm interacts with an instance ν in discrete steps. In step $t \in [n]$, the algorithm chooses $A_t \in [k]$ and receives a reward vector $X_t \sim \nu_{A_t}$. Note that the rewards underlying different objectives can be dependent, as is usually the case in applications. We let $\mu_a = (\mu_a^{(i)})_{i\in[l]} \in \mathbb{R}^l$ be the mean of ν_a.

For simplicity, we assume that the marginals of ν_a are 1-subgaussian.[1] We will also assume that all means $\mu_a^{(i)}$ lie in the interval $[-1, 1]$. We will denote the multi-level MAB instances satisfying these assumptions by $\mathcal{B}$. Below we will introduce two subsets of this set that will be of special interest.

The optimal expected reward vector $\mu^* = (\mu^{*(i)})_{i\in[l]}$ of an instance ν is defined as follows: We let $\mathcal{K}_1 = [k]$ and for $1 \le i \le l$ we let

$$\mu^{*(i)} = \max_{a\in\mathcal{K}_i} \mu_a^{(i)} \quad \text{while we let}$$
$$\mathcal{K}_{i+1} = \{a \in \mathcal{K}_i : \mu_a^{(i)} = \mu^{*(i)}\}\,.$$

In words, $\mathcal{K}_i$ represents all the equally good arms for the $(i-1)$-th objective. An omniscient decision maker only chooses arms from $\mathcal{K}_l$. The suboptimality of an arm $a \in [k]$ on objective $i \in [l]$ is defined as $\Delta_a^{(i)} = \mu^{*(i)} - \mu_a^{(i)}$. Note that in contrast to the single-objective case, in our problem for $i > 1$, some of these 'gaps' can be *negative*. When the dependence on ν is important, we will use $\nu^*(\nu)$, $\mathcal{K}_i(\nu)$ etc.

The n-round expected regret of an algorithm $\mathcal{A}$ on instance ν is

$$R_n(\nu, \mathcal{A}) = \mathbb{E}\left[n\mu^* - \sum_{j=1}^{n} \mu_{A_j}\right]\,.$$

Note again that as opposed to the single-objective problems, the regret of an algorithm may have negative components. The goal is to design algorithms that achieve a small worst-case regret over all objectives and all instances. To describe what can be achieved for this problem, we introduce the notion of the *regret Pareto front*. The regret Pareto front can be thought of as a generalization of the minimax regret for multiple objectives. We begin with introducing the standard partial ordering of the l-dimensional vectors:

Definition 2.1 (Partial ordering over regret vectors)**.** Given regret vectors $\rho, \rho' \in \mathbb{R}^l$ we say that ρ dominates ρ' (notation: $\rho \prec \rho'$) if for every $i \in [l]$, $\rho_i \le \rho'_i$ and for some $j \in [l]$, $\rho_j < \rho'_j$.

With the partial ordering defined above, we are now ready to define the Pareto front of a set (e.g.,Zitzler et al. [2003]).

Definition 2.2 (Pareto front of a set)**.** The Pareto front $\mathcal{F} \subseteq S$ of a set of regret vectors S is all members of S that are not dominated by any other vector in S: $\mathcal{F} = \{\rho \in S : \nexists \rho' \in S \text{ s.t. } \rho' \prec \rho\}$.

Given an algorithm $\mathcal{A}$ and a set $\mathcal{B}$ of l-objective bandit instances, we define the *objective-wise worst-case regret* of $\mathcal{A}$ as $R_n(\mathcal{A}) = (R_n^{(i)}(\mathcal{A}))_{i\in[l]}$ where

$$R_n^{(i)}(\mathcal{A}) = \sup_{\nu\in\mathcal{B}} R_n^{(i)}(\nu, \mathcal{A})\,.$$

[1] For $c > 0$, a distribution P over the reals is c-subgaussian if for $X \sim P$ and any $\lambda \in \mathbb{R}$, $\mathbb{E}[e^{\lambda X}] \le e^{c^2\lambda^2/2}$.

Note that $R_n(\mathcal{A})$ obviously depends on $\mathcal{B}$. This dependence is suppressed as the identity of $\mathcal{B}$ will always be clear from the context. Next, let

$$\mathcal{R}_n = \{R_n(\mathcal{A}) \,:\, \mathcal{A} \text{ is a bandit algorithm } \}$$

be the set of objective-wise worst-case regret vectors. The refined goal of algorithm design is to find all algorithms that produce regret vectors that are on the Pareto front of set $\mathcal{R}_n$. An algorithm is dubbed *Pareto-optimal* if its regret vector lies on the Pareto front of set $\mathcal{R}_n$: This means that the algorithm cannot be dominated by another algorithm in terms of its worst-case regret on all objectives: That is, if $\mathcal{A}$ is Pareto-optimal and $\mathcal{A}'$ is any other algorithm such that for some objective $i \in [l]$, $R_n^{(i)}(\mathcal{A}') < R_n^{(i)}(\mathcal{A})$ then for some other objective $j \neq i$, $j \in [l]$, $R_n^{(j)}(\mathcal{A}') \geq R_n^{(j)}(\mathcal{A})$.

Figure 1 illustrates these definitions. The figure shows regrets on instances for three algorithms whose identity is not important for the present discussion. The figure shows how the worst-case regret vector relates to the regrets on the instances. The figure also shows the regret Pareto front for the regret-vectors of these three algorithms.

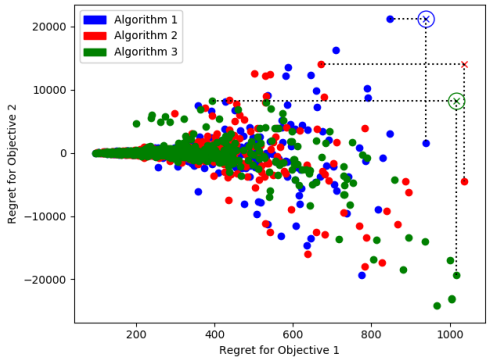

Figure 1: Regrets for three algorithms on two objectives are shown. Each algorithm is denoted by a different colour. Regret on an instance is shown by a dot (•), while a cross (×) denotes the worst-case regret of an algorithm. A circled cross denote elements of the regret Pareto front.

Since in the learning setting we cannot hope to identify exactly optimal algorithms we allow for a multiplicative slack $\lambda \geq 1$ when comparing algorithms. In particular, we say that $\mathcal{A}$ is λ-Pareto optimal if $R_n(\mathcal{A})/\lambda$ is not dominated by any vector in $\mathcal{R}_n$. With this we set the goal goal as finding a *complete set of λ-Pareto optimal algorithms*, where completeness means that for any Pareto optimal point ρ of the closure of $\mathcal{R}_n$ we want to find an algorithm $\mathcal{A}$ so that $R_n(\mathcal{A})/\lambda \prec \rho$.

2.2.1 Coupled objectives and exceedance probability

So far our results allow for arbitrary dependencies between the random rewards underlying different objectives. This allows bandit instances where the rewards underlying different objectives are independent (i.e., ν_a takes the form of product distributions). When proving lower bounds, more freedom in choosing the instances clearly helps with "pushing" the lower bound up. This raises the question whether the lower bounds continue to hold when the objectives are coupled in some nontrivial way. A common and practically relevant case which was also mentioned in the introduction is when the rewards underlying the two objectives are generated from a single common distribution over the reals and one of the objectives is just to maximize the mean reward, while the other objective is to maximize the probability that the reward exceeds (say) zero.

Formally, this leads to the instance spaces

$$\mathcal{B}_e = \{(e(P_a))_{a\in[k]} \,:\, P_a \in \mathcal{M}\}\,,$$
$$\mathcal{B}'_e = \{(e'(P_a))_{a\in[k]} \,:\, P_a \in \mathcal{M}\}\,,$$

where $\mathcal{M}$ contains all 1-subgaussian distributions whose means are in $[-1, 1]$ and for a distribution P over the reals, $e(P)$ denotes the distribution of the vector $(X, \mathbb{I}\{X \geq 0\})$ where $X \sim P$, while $e'(P)$ denotes the distribution of $(\mathbb{I}\{X \geq 0\}, X)$. Here, we choose zero as the exceedance target – but this choice has no significance for our results.

3 Results

The purpose of this section is to present the results of our theoretical analysis of the multi-level regret minimization problem. In Section 3.1 we start with our lower bounds that show that no algorithm can achieve low objective-wise worst-case regret on all objectives simultaneously. In Section 3.2 we introduce a class of algorithms and give upper-bounds on their worst-case regrets. Together with the lower bound, the upper-bounds show that these algorithms form a complete set of λ-Pareto optimal algorithms with $\lambda = O(k \log(kn))$, as also illustrated by Figure 2. Finally, in Section 3.3 we prove that algorithms that achieve minimax regret up to a constant factor on a linearly scalarized objective are provably dominated by the class of algorithms described in the previous section.

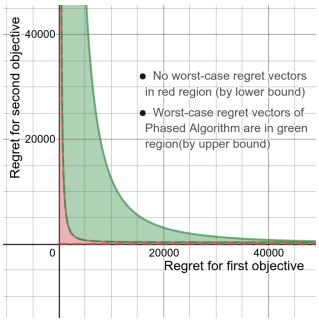

Figure 2: Illustration of the lower and upper bounds proved in the paper. When the number of rounds is 10^5 and number of arms is two, the red area is the region where no objective-wise worst-case regret vector can lie (no matter the algorithm), while the green area is where the objective-wise worst-case regret vector of the algorithms introduced in the paper lie. The upper bound shows that for any point on the red curve, there is a point in the green area that is at most a factor $\lambda \in O(\log(n))$-times larger. The regrets of this point can be reached by an appropriate tuning of the algorithm.

3.1 Lower Bounds

For ease of explanations, we start with the special case when there are $l = 2$ objectives and for the case when the rewards of arms under the two objectives are decoupled. We follow this up with a result that shows that the lower bound continues to hold when one of the objectives is to maximize the exceedance probability of the reward of the other objective (i.e., we prove that the lower bound continues to hold for the sets $\mathcal{B}_e$ and $\mathcal{B}'_e$). The section is finished with the lower bound that holds for an arbitrary number of uncoupled objectives. The proof of this result follows the same logic as the proof given for the two-objective case.

3.1.1 The Two-Objective Special Case

Our result for the two-objective case is as follows:

Theorem 3.1. *Let $n \in \mathbb{N}$ be the number of rounds and consider the set of instances $\mathcal{B}$ defined in Section 2.2 with $l = 2$ objectives. Then for all $0 < r < \frac{\sqrt{n}}{800}$, any algorithm $\mathcal{A}$ that satisfies $R_n^{(1)}(\mathcal{A}) \leq r\sqrt{n}$ has $R_n^{(2)}(\mathcal{A}) \geq \max\left(\frac{1}{27}\sqrt{n}, \frac{\ln 3}{16 \cdot 10^4} \cdot \frac{n}{r^2}\right)$.*

Proof. By the lower bound results on minimax regret (Theorem 15.2 of Lattimore and Szepesvári [2020]) we know that there is an instance on which any algorithm has at least $\frac{1}{27}\sqrt{n}$ regret on both objectives, and in particular the second objective. It remains to show that any algorithm $\mathcal{A}$ with $R_n^{(1)}(\nu, \mathcal{A}) \leq r\sqrt{n}$ on all bandit instances $\nu \in \mathcal{B}$ has $R_n^{(2)}(\nu', \mathcal{A}) \geq \frac{\ln 3}{16 \cdot 10^4} \cdot \frac{n}{r^2}$ for some bandit instance $\nu' \in \mathcal{B}$. The idea of the rest of the proof is as follows: We construct two two-armed "hard" instances such that either arm one, or arm two is optimal on the primary objective, depending on the instance. The payoff distributions for the secondary objective are the same in both instances for both arms so no information is leaked about the identity of the instance by the rewards received on this objective. Further, in both instances arm two is optimal on the secondary objective and the gap between the payoffs of arm one and arm two are large. The gap between the payoffs of arm one and arm two on the primary objective are carefully chosen so that any algorithm with small worst-case regret over these two instances on this objective needs to pull arm one many times,

which means, that by construction it will suffer a large regret on the secondary objective on the instance where arm one is suboptimal on the first objective. The details are as follows.

Construction of the hard instances We now construct the hard instances. Consider bandit instances ν_1, ν_2 with each instance having two arms. The reward distribution is a product distribution of unit variance Gaussians, which implies that we can fully describe the instances by specifying their means. Let $\Delta = \frac{200r}{\sqrt{n}}$.

Define the instances, using the means of the respective reward distributions, as follows:

$$\nu_1 = \begin{array}{c} \\ \text{arm 1} \\ \text{arm 2} \end{array} \begin{array}{c} \begin{array}{cc} \text{Criteria 1} & \text{Criteria 2} \end{array} \\ \begin{pmatrix} \frac{1}{2} - \Delta & 0 \\ \frac{1}{2} & 1 \end{pmatrix} \end{array}$$

$$\nu_2 = \begin{array}{c} \\ \text{arm 1} \\ \text{arm 2} \end{array} \begin{array}{c} \begin{array}{cc} \text{Criteria 1} & \text{Criteria 2} \end{array} \\ \begin{pmatrix} \frac{1}{2} + \Delta & 0 \\ \frac{1}{2} & 1 \end{pmatrix} \end{array}.$$

Note that by the restriction of r in the theorem, $\Delta < \frac{1}{4}$. The means of ν_1, ν_2 lie in $[-1, 1]$ and so $\nu_1, \nu_2 \in \mathcal{B}$. Note that $\mu^*(\nu_1) = (1, 1)^\top$ and $\mu^*(\nu_2) = (1 + \Delta, 0)^\top$: The optimal arm on the secondary objective in ν_1 is arm two, while in ν_2 it is arm one. In particular, the regret on objective two in instance ν_1 is at least as large as the number of expected pulls of arm one on this instance.

We will use the following lemmas to complete the proof. The first lemma (Lemma 3.2) shows that any algorithm with $R_n^{(1)} \leq r\sqrt{n}$ on both instances ν_1 and ν_2 can be used to distinguish between the two instances. The second lemma (Lemma 3.3) shows that to distinguish between ν_1, ν_2, any algorithm needs $\Omega(1/\Delta^2)$ pulls of each arm of each instance, in expectation. The proofs of these lemmas are deferred to the appendix.

Lemma 3.2. *From any algorithm $\mathcal{A}$ that has $R_n^{(1)}(\mathcal{A}) \leq r\sqrt{n}$, we can derive an algorithm $\mathcal{A}_d$ that distinguishes between bandit instances ν_1 and ν_2 with probability $\frac{11}{12}$. The expected number of times each arm of each instance is pulled by $\mathcal{A}_d$ is the same as the expected number of times it is pulled by $\mathcal{A}$.*

Lemma 3.3. *Any algorithm that distinguishes between bandit instances with reward distributions ν_1, ν_2 with probability at least $1 - \delta$ pulls the first arm at least $\frac{1}{4\Delta^2} \ln\left(\frac{1}{4\delta}\right)$ times on expectation, in either instance ν_1 or instance ν_2.*

From Lemmas 3.2, 3.3, we get that for any $\mathcal{A}$ with $R_n^{(1)}(\mathcal{A}) \leq r\sqrt{n}$, the expected number of times arm one is pulled by $\mathcal{A}$ on instance ν_1 is at least $\frac{\ln 3}{4} \cdot \frac{n}{4 \cdot 10^4 r^2}$, which implies that on this instance the regret is at least as large as this quantity.

□

Lower bound for exceedance and sum of rewards objectives: The above lower bound is shown to hold for a reward distribution that is a product distribution. So the lower bound of Theorem 3.1 does not automatically hold for the two special objectives - exceedance and sum-of-rewards (Section 2.2.1) which are objectives based on the same distribution. With a similar argument, we can actually show the lower bound still holds for these objectives. This is due to exceedance and mean being uncoupled in the sense that we can construct distributions with arbitrary values of mean and exceedance:

Lemma 3.4. *For any $\mu \geq 0$ and $e \in [1/2, 1)$, there is a Gaussian distribution with variance at most 1, having mean μ and exceedance e.*

Proof. Let $\bar{\Phi}_{\mu,\sigma}$ denote the tail distribution function of the Gaussian distribution with mean μ and σ: If $X \sim \mathcal{N}(\mu, \sigma)$, $\bar{\Phi}_{\mu,\sigma}(x) = \mathbb{P}(X \geq x)$. Fix $\mu \geq 0$. As is well known, $\bar{\Phi}_{\mu,\sigma}(0)$ is a continuous (and decreasing) function of σ, which takes the value of 1 at $\sigma = 0$ and takes a value of at least $1/2$ when $\sigma = 1$. The result follows from the intermediate value theorem. □

With the above instances and a very similar proof to that of Theorem 3.1, we show the following theorem. The formal proof is postponed to the Appendix.

Theorem 3.5. *Let $\mathcal{B} \in \{\mathcal{B}_e, \mathcal{B}'_e\}$ and let $n \in \mathbb{N}$ be the number of rounds. For any $0 < r < \frac{\sqrt{n}}{1600}$, any algorithm $\mathcal{A}$ that has $R_n^{(1)}(\nu) \leq r\sqrt{n}$ on all bandit instances ν in $\mathcal{B}$, has worst-case $R_n^{(2)}$ over $\mathcal{B}$ at least $\max\left(\frac{\ln 3}{64 \cdot 10^4} \cdot \frac{n}{r^2}, \frac{1}{27}\sqrt{n}\right)$.*

3.1.2 Lower bound for more than two objectives

Extending Theorem 3.1 for multiple objectives, we get the following result: Suppose an algorithm enjoys small worst-case regrets on the first p of the l objectives. Then, the worst-case regret of lower priority objectives will grow with the inverse (square) of the upper bounds on the regrets on the higher priority objectives. The proof of this result is similar to the previous proof given for two objectives (cf. Theorem 3.1). and is provided in the appendix.

Theorem 3.6. *Consider the instance class $\mathcal{B}$ defined in Section 2.2. Let $n \in \mathbb{N}$ be the number of rounds. For any $p < l$ and any $0 < \rho_1, \ldots, \rho_p < \frac{\sqrt{n}}{800}$, any algorithm $\mathcal{A}$ satisfying $R_n^{(i)}(\mathcal{A}) \leq \rho_i$ for every $i \in [p]$, has $R_n^{(j)}(\mathcal{A}) \geq \max\left(\frac{1}{27}\sqrt{n}, \frac{\ln 3}{16 \cdot 10^4} \sum_{i=1}^{p} \frac{n^2}{\rho_i^2}\right)$ for all $j \in [l] \setminus [p]$.*

3.2 Algorithm for multi-level regret minimization

In this section, we introduce a parameterized family of multi-level regret minimization algorithms that achieves near-optimal performance. Our algorithm runs in l stages - one stage for each objective (formally presented in Algorithm 1). The algorithm takes as input $\varepsilon_1, \ldots, \varepsilon_l$. These parameters will allow us to traverse the regret Pareto front.; in particular, later we will show how to choose these values to reach close to a specific point on the Pareto front. The ε_i's determine the number of rounds in each stage of the algorithm. A stage consists of a call to the sub-routine Phased Elimination (Algorithm 2), an algorithm borrowed from Auer and Ortner [2010]. The subroutine optimizes the objective of the phase and for the number of rounds determined by the input parameters. The subroutine also eliminates all arms determined to be sub-optimal according to the objective optimized.

Algorithm 1 Phased multi-level regret minimization

Input: $\varepsilon_1, \ldots, \varepsilon_l, n$, Bandit instance: ν
$\mathcal{A} := \{1, \ldots, k\}$
$j := 1$
while $n > 0$ and $j \leq l$ **do**
  $(\mathcal{A}, m) =$ **Phased Elimination**$(\varepsilon_j, n, \mathcal{A},$
    j^{th} reward distributions of arms in $\mathcal{A})$
  $n = n - m$
  $j = j + 1$
end while
For any remaining rounds, pull any arm from $\mathcal{A}$

3.2.1 Analysis of the Algorithm

First we state a worst-case bound on the regrets of running Algorithm 1. This worst case bound directly follows from an instance-dependent bound we prove later in Theorem 3.9.

Proposition 3.7. *For any bandit instance with k arms and l objectives, for every $i \in [l]$, $a \in [k]$, regrets from running Algorithm 1 for n rounds with parameters $\varepsilon_1, \ldots, \varepsilon_l$ satisfy, for each $i \in [l]$,*

$$R_n^{(i)} \leq n\varepsilon_i + 65\sqrt{nk}\log(2n^2k) + 16\log(2n^2k)\sum_{j=1}^{i-1}\frac{1}{\varepsilon_j^2}.$$

The goal we set out to solve was to find a complete set of λ-Pareto optimal algorithms. We will show through the following proposition that the above proposition (Proposition 3.7) implies that we meet this goal for $\lambda = C \cdot k\log(nk)$, for some universal constant C. The proof of the following proposition is presented in the appendix.

Proposition 3.8. *For $\varepsilon = (\varepsilon_i)_{i=1}^{l}$, let $\mathcal{A}_p(\varepsilon)$ be the phased algorithm 1 resulting from input parameters ε. Consider the set $\mathcal{F} = \{\mathcal{A}_p(\varepsilon)\}$ of all ε satisfying the constraints listed below. $\mathcal{F}$ is a complete set of λ-Pareto algorithms for $\lambda = C \cdot k\log(nk)$, for some universal constant C.*

1. *For each $1 \leq i < l$,*
   $\varepsilon_i n \geq 65\sqrt{nk}\log(2n^2k) + 16\log(2n^2k)\sum_{j=1}^{i-1}\frac{1}{\varepsilon_j^2}$,

Algorithm 2 Phased Elimination

Input: ε, n, Arm indices and reward distributions : $(\mathcal{A}, \mathcal{F})$
Output: Set of indices of arms that are ε-optimal, number of rounds played
$T := \lceil \log_2 \frac{2}{\varepsilon} \rceil$
$\mathcal{A}_1 := \mathcal{A}$
rounds $:= 0$
pulls := list of number of times each arm is pulled; all initialized to 0
for t = 1, . . . , T **do**
  $m(t) := \lceil 4 \cdot 2^{2t} \log\left(2n^2k\right) \rceil$
  for arm $a \in \mathcal{A}_t$ **do**
    if rounds $+ m(t) -$ pulls$[a] > n$ **then**
      Choose a for remaining rounds
      Return $(\mathcal{A}_t, n)$
    end if
    Choose a for $m(t) -$ pulls$[a]$ times and calculate
      average reward $\hat{\mu}_{a,t}$ based on all $m(t)$ pulls
    Update rounds $=$ rounds $+ m(t) -$ pulls$[a]$
    Update pulls$[a] = m(t)$
  end for
  $\mathcal{A}_{t+1} = \{a : \hat{\mu}_{a,t} + 2^{-t} \geq \max_{b \in \mathcal{A}_t} \hat{\mu}_{b,t}\}$
end for
Return $(\mathcal{A}_{T+1}, rounds)$

2. $\varepsilon_l n = 65\sqrt{nk}\log(2n^2k) + 16\log(2n^2k)\sum_{j=1}^{l-1}\frac{1}{\varepsilon_j^2}$.

Finally, let us state the promised instance-dependent regret bound:

Theorem 3.9. *For any bandit instance with k arms, for any $n \in \mathbb{N}, 0 < \varepsilon_1, \ldots, \varepsilon_l$, regrets from running Algorithm 1 satisfy, for each $i \in [l]$,*

$$R_i \leq n \cdot \max_{a \in \mathcal{K}_\varepsilon} \bar{\Delta}_a^{(i)} + \sum_{a \in [k] \setminus \mathcal{K}_\varepsilon} \left(\bar{\Delta}_a^{(i)} + \frac{64\log(2n^2k)}{\bar{\Delta}_a^{(i)}} + 16\log(2n^2k) \sum_{j=1}^{p(a)} \frac{\bar{\Delta}_a^{(i)}}{\varepsilon_j^2} \right)$$

where

$$\bar{\Delta}_a^{(i)} = \max\left(\Delta_a^{(i)}, 0\right)$$
$$\mathcal{K}_\varepsilon = \{a \in [k] : \forall i \in [l], \Delta_a^{(i)} \leq \varepsilon_i\}$$
$$p(a) = \begin{cases} \min\{p \in \mathbb{N} : \Delta_a^{(p)} > \varepsilon_p\} & \text{if } a \notin \mathcal{K}_\varepsilon \\ l & \text{otherwise} \end{cases}$$

Proof sketch. The algorithm performs a series of phased eliminations for the different objectives. The i^{th} stage of the algorithm optimizes for R_i using the phased elimination algorithm. We adapt the analysis to account for regrets across the different invocations of phased elimination. The formal proof is presented in the appendix. □

The phased algorithm by Auer and Ortner [2010] is constructed to eliminate $\log(n)$ terms from regret bound. It does this by allowing the probability of sub-optimal arms persisting to depend on the sub-optimality value. An arm that is more optimal is allowed to persist with higher probability compared to an arm that is less optimal. However, in our problem where we have multiple objectives, it is hard to assign different probabilities of elimination to different arms. An arm that is more optimal according to one objective can be less optimal according to another objective. We do not know yet if the analysis from Auer and Ortner [2010] can be adapted to eliminate the extra $\log(n)$ terms from the regret upper bounds for our problem.

3.3 Sub-optimality of optimizing linear combination of objectives

Despite the simple form of Algorithm 1, one might ask whether an even simpler linear scalarization of the rewards with a usual single-objective MAB algorithm can achieve the same results that we obtained. In this section, we show that this is not the case. For simplicity, we focus on the two objective case.

Suppose we have two objectives. Consider the class of algorithms $\mathcal{A}_{\text{linear}}$ that optimize an objective that is a linear combination of the regrets of these objectives. For a round $i \in [n]$, let $R^{(1)}_{n,i}, R^{(2)}_{n,i}$ denote the immediate expected regrets accumulated by the algorithm at round i. An algorithm in this class, parameterized by $w \in [0,1]^n$, is an algorithm that minimizes the regret $\mathcal{R}^{(w)}_n = \sum_{i=1}^n R^{(1)}_{n,i} + w_i R^{(2)}_{n,i}$ and obtains $\mathcal{R}^{(w)}_n \in O(\sqrt{n})$.

Theorem 3.10. *For any $r \in \Omega(1)$ such that $r \in o(\sqrt{n})$, any algorithm from $\mathcal{A} \in \mathcal{A}_{linear}$ that has $R^{(1)}_n(\mathcal{A}) \in O(r\sqrt{n})$ has $R^{(2)}_n(\mathcal{A}) \in \Omega\left(\frac{n}{r}\right)$.*

This theorem shows that algorithms in $\mathcal{A}_{\text{linear}}$ make a worse trade-off between the two objectives than what is possible. From Theorem 3.9, we know that we can find an algorithm from the class of phased algorithms that has $R^{(1)}_n \in O(r\sqrt{n})$ and $R^{(2)}_n \in O\left(\frac{n(\log n)^3}{r^2}\right)$. For $r \in \omega((\log n)^3)$, any algorithm from $\mathcal{A}_{\text{linear}}$ with regret of first objective in $O(r\sqrt{n})$, has strictly worse regret in second objective than the algorithm from the class of phased algorithms 1 that has the same regret of $O(r\sqrt{n})$ for the first objective. We describe a sketch of the proof for this theorem below. The full proof is deferred to the appendix.

Proof Sketch. We show that when the weight assigned to the second objective is above a threshold, the scalarized objective does not optimize for the first objective sufficiently. This results in regret of the first objective being worse than $O(r\sqrt{n})$. However, when the weight is below this threshold, the linear objective optimizes for the first objective more than what is necessary for $O(r\sqrt{n})$ regret. As a result, more regret than what is possible is accumulated for the second objective.

□

4 Empirical illustration

We implement algorithms from two classes of algorithms - phased and linear combination. The first class is phased regret minimization algorithms (Algorithms 1). The second class is algorithms that minimize an objective that is a linear combination of all the objectives. To make the comparison more direct, this linear combination objective is minimized through Phased Elimination (Algorithm 2) by Auer and Ortner [2010]. The setup of this implementation is as follows

- Number of rounds: 10^5
- Bandit instances: We run each algorithm on the class of bandit instance class with 2500 instances $\mathcal{B} = \{\nu_{ij} : i, j \in [50]\}$. Each instance has two objectives and two arms. For each arm, the objective distribution is a product distribution of two unit variance Gaussian distributions, whose means are as follows:

$$\nu_{ij} = \begin{array}{c} \\ \text{arm 1} \\ \text{arm 2} \end{array} \begin{array}{c} \begin{array}{cc} \text{Objective 1} & \text{Objective 2} \end{array} \\ \begin{pmatrix} 0.5 & 0.5 \\ \frac{i}{50} & \frac{j}{50} \end{pmatrix} \end{array}$$

- Parameters of algorithms: From the phased class, we implement five algorithms parameterized by $\varepsilon \in \{\varepsilon_1, \ldots, \varepsilon_5\}$ with $\varepsilon_i = \frac{10^{0.5i}}{1000}$. The algorithm from the phased algorithms class that corresponds to parameter ε is the algorithm that optimizes objective one in phase one with parameter ε and optimizes for objective two in the remaining rounds. For the linear combination class, we implement five algorithms parameterized by $\{w_1, w_2, w_3, w_4, w_5\}$ where $w_i = \frac{i}{5}$. The algorithm from the linear combination class that corresponds to w_i is one that optimizes an objective that assigns weight one to objective one and weight w_i to objective two.

We run each algorithm, on each instance ten times and calculate the average regret of the algorithm on that instance, over the ten runs. For each algorithm we find the objective-wise worst case regret over all instances in $\mathcal{B}$.

For both classes of algorithms, we observe that in general as the worst case regret of the first objective (R_1) decreases, the worst case regret of the second objective (R_2) increases. The exception to this is the phased algorithm with parameter

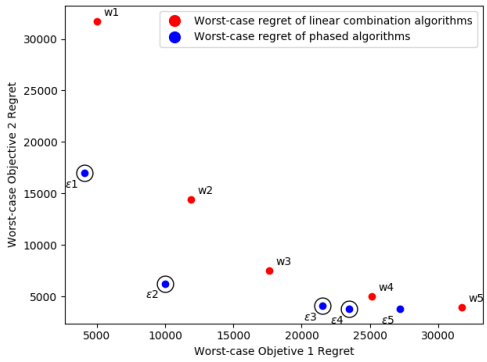

Figure 3: The points indicate worst-case regrets over the set $\mathcal{B}$. The red points indicate algorithms from the linear combination class and the blue points indicate algorithms from the phased class. The circled points are make the Pareto front of the points plotted. The observed standard deviations of the measurements (not shown) are small enough so that we expect the plot to be representative of the mean behavior of the algorithms.

ε_5. Changing the parameter from ε_4 to ε_5 increases the R_1 but R_2 does not decrease. This may be because R_2 has hit the lower bound of $O(\sqrt{n})$ at parameter ε_4 and cannot decrease further.

We find that for each linear combination algorithm, there is a phased algorithm which has lower worst-case regrets on both objectives. This is aligned with the statements of Theorems 3.9 and 3.10.

5 Summary

In conclusion, for the multi-level MAB problem, we show that we cannot have low regrets for all instances, for all objectives. We introduce the notion of the regret Pareto front to analyze the worst-case performance of algorithms in this setting. We provide a lower bound (Theorem 3.6) that characterizes the different necessary trade-offs among the worst-case regrets of objectives. We get the lower bound by reducing the problem of obtaining low worst-case regrets on all objectives to the problem of distinguishing among distributions with low pair-wise KL divergence. We provide an algorithm family for multi-criteria regret minimization and analyze the worst-case regret of members of this family (Theorem 3.9). Combined with the lower bound result, we show that the algorithm initialized with different input parameters results in worst-case regrets that are close to all points on the regret Pareto front – which results in a complete characterization of the Pareto frontier up to a multiplicative factor that depends only logarithmically on the number of rounds. We have also provided a lower bound on the worst-case regrets of algorithms that optimize a linearly scalarized objective. This lower bound shows that the trade-off due to the linear scalarization approach is strictly worse than the best trade-off that is possible to achieve.

Some readers may be wondering about whether the multi-level approach is too limited in that the lower priority objectives are only used for resolving ties between arms that are *exactly* tied as optimal on the primary objective(s). In connection to this, one interpretation of our results is that a learner who is uncertain about the payoffs but wants to keep the regret small on all objectives, necessarily needs to relax the exact ties (in the primary objectives) to treat arms with *nearly identical* payoffs as identical.

In terms of future work, it would be interesting to remove the gap between the lower and upper bounds. It would also be interesting to investigate whether the algorithm design of Tekin and Turğay [2018] could be modified to make the algorithm traverse the Pareto regret frontier. Finally, we believe that the proof techniques we introduced, with appropriate modifications, can be used to achieve similar complete characterization of the regret Pareto frontier in other versions of multi-objective regret minimization.

6 Bibliography

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in Markov decision processes, 2019.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1, 2012.

Shipra Agrawal, Nikhil R Devanur, and Lihong Li. An efficient algorithm for contextual bandits with knapsacks, and an extension to concave objectives. In *Conference on Learning Theory*, pages 4–18, 2016.

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

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 207–216. IEEE, 2013.

Hamsa Bastani and Mohsen Bayati. Online decision-making with high-dimensional covariates. *Forthcoming in Operations Research*, 2015.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, Gilles Stoltz, et al. Kullback–Leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541, 2013.

Cristina Castejón, Giuseppe Carbone, JC García Prada, and M Ceccarelli. A multi-objective optimization of a robotic arm for service tasks. *Strojniski Vestnik/Journal of Mechanical Engineering*, 56(5), 2010.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *COLT*, 2008.

Madalina M Drugan and Ann Nowe. Designing multi-objective multi-armed bandits algorithms: A study. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2013.

Madalina M Drugan and Ann Nowé. Scalarization based Pareto optimal set of arms identification algorithms. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 2690–2697. IEEE, 2014.

Audrey Durand, Charles Bordet, and Christian Gagné. Improving the Pareto UCB1 algorithm on the multi-objective multi-armed bandit. In *Workshop of the 27th Neural Information Processing (NIPS) on Bayesian Optimization*, 2014.

Xiaoguang Huo and Feng Fu. Risk-aware multi-armed bandit problem with application to portfolio selection. *Royal Society open science*, 4(11):171377, 2017.

Anisio Lacerda. Multi-objective ranked bandits for recommender systems. *Neurocomputing*, 246:12–24, 2017.

T. Lattimore and Cs. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2019. draft.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

EJ Solteiro Pires, JA Tenreiro Machado, and Paulo B de Moura Oliveira. Robot trajectory planning using multi-objective genetic algorithm optimization. In *Genetic and Evolutionary Computation Conference*, pages 615–626. Springer, 2004.

Cem Tekin and Eralp Turğay. Multi-objective contextual multi-armed bandit with a dominant objective. *IEEE Transactions on Signal Processing*, 66(14):3799–3813, 2018.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Saba Yahyaa and Bernard Manderick. Thompson sampling for multi-objective multi-armed bandits problem. In *Proceedings*, page 47. Presses universitaires de Louvain, 2015.

Saba Q Yahyaa and Madalina M Drugan. Correlated Gaussian multi-objective multi-armed bandit across arms algorithm. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 593–600. IEEE, 2015.

Saba Q Yahyaa, Madalina M Drugan, and Bernard Manderick. Knowledge gradient for multi-objective multi-armed bandit algorithms. In *ICAART (1)*, pages 74–83, 2014a.

Saba Q Yahyaa, Madalina M Drugan, and Bernard Manderick. The scalarized multi-objective multi-armed bandit problem: An empirical study of its exploration vs. exploitation tradeoff. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 2290–2297. IEEE, 2014b.

Saba Q Yahyaa, Madalina M Drugan, and Bernard Manderick. Thompson sampling in the adaptive linear scalarized multi objective multi armed bandit. In *ICAART (2)*, pages 55–65, 2015.

Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7 (2):117–132, 2003.

Appendix

6.1 LOWER BOUND PROOFS

Minimax lower bound on regret in single objective setting is given by the following theorem which appears as Theorem 15.2 in the book of Lattimore and Szepesvári [2020], where the reader can find the proof.

Theorem 6.1. *Let number of arms be $k > 1$ and number of rounds $n \geq k - 1$. Then, for any algorithm, there is a set of k unit variance Gaussians with means in $[0, 1]$ such that the expected regret of the algorithm with the instance with these distributions as arms is at least $\frac{1}{27}\sqrt{(k-1)n}$*

6.1.1 Proof of Lemma 3.2

In the instance ν_1, the second arm is optimal while in instance ν_2, the first arm is optimal. Both instances have one optimal arm. In both instances, playing a sub-optimal arm t times results in accumulating $\Delta \cdot t$ regret $R_n^{(1)}$. $\mathcal{A}_d$ is the algorithm that first runs $\mathcal{A}$ on instances ν_1 and ν_2. It identifies an arm played at least $\frac{n}{2}$ times as the optimal arm for an instance. If the arm identified as optimal is arm2, then $\mathcal{A}_d$ identifies the bandit instance as ν_1. Otherwise $\mathcal{A}_d$ identifies the bandit instance as ν_1.

It is clear that if the correct arm is identified as optimal for both ν_1 and ν_2, then $\mathcal{A}_d$ has the correct output. Therefore, we can bound the failure probability of $\mathcal{A}_d$ by the probability that A plays sub-optimal arm $\frac{n}{2}$ times for instance ν_0 or ν_j. If this occurs with probability more than $\frac{1}{24}$ for an instance, then for this instance, $R_n^{(1)}$ of $\mathcal{A}$ is at least $\frac{1}{24} \cdot \Delta \cdot \frac{n}{2} = \frac{200r\sqrt{n}}{48} > r\sqrt{n}$. This is a contradiction. Therefore the probability of failure for the algorithm $\mathcal{A}_d$ is less than $\frac{1}{12}$.

Since $\mathcal{A}_d$ is designed to simply run $\mathcal{A}$ on ν_1 and ν_2, the expected number of times each arm of each instance is pulled by $\mathcal{A}_d$ is the same as the expected number of times they are pulled by $\mathcal{A}$. □

6.1.2 Proof of Lemma 3.3

Any algorithm $\mathcal{A}$ along with the reward distributions ν_1, ν_2 induces probability measures $\mathbb{P}_{\nu_1}$ and $\mathbb{P}_{\nu_2}$. Let T_1 be a random variable denoting the number of times $\mathcal{A}$ plays arm 1. By Lemma 15.1 of Lattimore and Szepesvári [2019], the KL-divergence of these probability measures is as follows, for i either one or two:

$$\begin{aligned}\mathrm{KL}(\mathbb{P}_{\nu_1}, \mathbb{P}_{\nu_2}) &= \mathbb{E}_{\nu_i}[T_1] \cdot \mathrm{KL}(\mathcal{N}(1-\Delta, 1), \mathcal{N}(1+\Delta, 1)) \\ &= \mathbb{E}_{\nu_i}[T_1] \cdot 4\Delta^2.\end{aligned}$$

Let E_1 be the event of obtaining rewards based on which the algorithm determines an instance as having reward distributions ν_1. And let E_2 be the complement of E_1. By the Bertagnolle-Huber inequality,

$$\mathbb{P}_{\nu_1}(E_2) + \mathbb{P}_{\nu_2}(E_1) \geq \frac{1}{2}\exp(-\mathrm{KL}(\mathbb{P}_{\nu_1}, \mathbb{P}_{\nu_2})).$$

For a distinguishing algorithm with probability of failure less than δ, $\mathbb{P}_{\nu_1}(E_2) < \delta$ and $\mathbb{P}_{\nu_2}(E_1) < \delta$. Therefore, $\mathbb{E}_{\nu_i}[T_1] \geq \frac{1}{4\Delta^2} \ln\left(\frac{1}{4\delta}\right)$ for $i \in \{1, 2\}$. □

6.1.3 Proof of Theorem 3.5

Here primary objective is one sum-of-rewards and exceedance and the secondary objective is the other. Let $\Delta_s = \frac{1}{4}$ and $\Delta_p = \frac{100r}{\sqrt{n}}$. By the restriction of r in the theorem ($r < \frac{\sqrt{n}}{800}$), $\Delta_p < \frac{1}{8}$. From Lemma 3.4, we know that there are Gaussians F, F_1, F_2 with primary objectives $\frac{3}{4}, \frac{3}{4} - \Delta_p, \frac{3}{4} + \Delta_p$ and secondary objectives $\frac{3}{4}, \frac{3}{4} - \Delta_s, \frac{3}{4} - \Delta_s$ Consider the instances ν_1, ν_2 such that both instances have two arms. The second arm of both instances has the distribution F. The first arm of ν_1 is F_1 and the first arm of ν_2 is F_2.

For each ν_i, consider the instance ν_i' also with two arms. Each arm of ν_i' has a reward distribution that is a product distribution of two unit variance Gaussians. Let F_a be the distribution of an arm in ν_i. The means of Gaussians of the corresponding arm in ν_i' are $p(F_a)$ and $s(F_a)$. It is clear that for any algorithm, the primary regret of ν_i is the same as $R_n^{(1)}$ of ν_i' and secondary regret is the same as $R_n^{(2)}$.

From the proof of Theorem 3.6, we know that any algorithm with $R_n^{(1)} \leq r\sqrt{n}$ on instances ν_1', ν_2' has $R_n^{(2)} \geq \frac{\ln 3}{64 \cdot 10^4} \frac{n}{r^2}$ on ν_1'. □

6.1.4 Proof of Theorem 3.6

Consider the set of bandit instances $B = \{\nu_0, \ldots, \nu_p\}$ with each instance having $p+1$ arms. The joint reward distribution is a product distribution of l unit variance Gaussians. So we can denote a reward distribution of an arm by its mean. For each $1 \leq i < p$ define $\Delta_i = \frac{100\rho_i}{n}$. For the instance ν_0, for an arm a,

$$\mu_a^{(i)}(\nu_0) = \begin{cases} 1 & \text{if } a = p+1 \\ 1 & \text{if } a \leq p \text{ and } i < a \\ 1 - \Delta_a & \text{if } a \leq p \text{ and } i = a \\ 0 & \text{otherwise.} \end{cases}$$

We can denote the reward distributions of the arms of instance ν_0 as

$$\begin{array}{c} \text{arm} \\ 1 \\ 2 \\ 3 \\ \vdots \\ p \\ (p+1) \end{array} \begin{array}{c} \begin{array}{cccccc} \text{Obj } 1 & \text{Obj } 2 & \ldots & \text{Obj } p & \ldots & \text{Obj } l \end{array} \\ \begin{pmatrix} 1-\Delta_1 & 0 & \ldots & 0 & \ldots & 0 \\ 1 & 1-\Delta_2 & \ldots & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \ldots & 1-\Delta_p & \ldots & 0 \\ 1 & 1 & \ldots & 1 & \ldots & 1 \end{pmatrix} \end{array}.$$

An instance ν_q has the same reward distributions as ν_0 except for the q^{th} reward distribution of the q^{th} arm. The mean of this distribution for ν_q is $1 + \Delta_q$ as opposed to $1 - \Delta_q$ which is the corresponding mean for ν_0.

$$\mu_a^{(i)}(\nu_q) = \begin{cases} \mu_a^{(i)}(\nu_0) & \text{if } a \neq q \text{ or } i \neq q \\ 1 + \Delta_q & \text{if } a = q \text{ and } i = q. \end{cases}$$

Let us call any algorithm with $R_n^{(i)} \leq \rho_i$ for all $i \in [p]$ on all bandit instances a constraint-satisfying algorithm. Such an algorithm must in particular satisfy these inequalities on the set of instances B.

We use the following lemmas that follow from proofs of lemmas 3.3, 3.2 to prove the theorem.

Lemma 6.2. *For any $i \in [p]$, from any constraint-satisfying algorithm $\mathcal{A}$, we can derive an algorithm $\mathcal{A}_i$ that distinguishes between bandit instances ν_0 and ν_i with probability $1 - \frac{1}{12}$. The expected number of times each arm of each instance is pulled by $\mathcal{A}_i$ is the same as the expected number of times it is pulled by $\mathcal{A}$.*

Lemma 6.3. *Any algorithm that distinguishes between bandit instances with reward distributions ν_0, ν_i with probability $1 - \delta$, plays arm i at least $\frac{1}{4\Delta^2} \ln\left(\frac{1}{4\delta}\right)$ times on expectation, in either instance ν_0 or instance ν_1.*

From these lemmas, we get that for any constraint-satisfying algorithm, for $i \in [p]$, the expected number of times arm i is pulled for instance ν_0 is at least $N_i \geq \frac{\ln 3}{16 \cdot 10^4} \frac{n}{\rho_i^2}$.

For $j \in [l] \setminus [p]$, value 1 adds to regret j every time an arm $i \neq p+1$ is pulled. Therefore

$$R_j \geq \sum_{i=1}^{j-1} N_i \geq \frac{\ln 3}{16 \cdot 10^4} \sum_{i=1}^{j-1} \frac{n}{\rho_i^2},$$

completing the proof. □

6.2 REGRET ANALYSIS PROOFS

6.2.1 Proof of Theorem 3.9

The algorithm can be summarized as a series of phased eliminations Auer and Ortner [2010] for the different objectives. The i^{th} phase of the algorithm optimizes for $R_n^{(i)}$. Let $a^* \in [k]$ be an arm that is optimal according to the ordering due to all the objectives.

Recall that $m(t) = \left\lceil 4 \cdot 2^{2t} \log\left(2n^2 k\right) \right\rceil$ is the number of times an arm is pulled by round t of a phase of the algorithm for all arms that are not eliminated by round t. Recall also that

$$\mathcal{K}_\varepsilon = \{a \in [k] : \forall i \in [l], \Delta_a^{(i)} \leq \varepsilon_i\}$$
$$p(a) = \begin{cases} \min\{p \in \mathbb{N} : \Delta_a^{(p)} > \varepsilon_p\} & \text{if } a \notin \mathcal{K}_\varepsilon \\ l & \text{otherwise} \end{cases}$$

If at a round t of some phase, if for all arms $a \in \mathcal{A}_t$,

$$\mu_a - \frac{2^{-t}}{2} \leq \hat{\mu}_{a,t} \leq \mu_a + \frac{2^{-t}}{2},$$

then

$$\max_{a \in \mathcal{A}_t} \hat{\mu}_{a,t} \leq \max_{a \in \mathcal{A}_t} \mu_{a,t} + \frac{2^{-t}}{2} \leq \hat{\mu}_{a^*,t} + 2^{-t}.$$

In this case, a^* is not eliminated in that round. So the probability that a^* is eliminated in some round t of some phase is at most

$$kn \exp\left(-\frac{m(t) 2^{-2t}}{4}\right) \leq \frac{1}{2n}.$$

For each arm a, we find an upper bound n_a on the number of times it is pulled. Using this upper bound, we get that $n_a \cdot \max(\Delta_a^{(i)}, 0)$ is an upper bound on the regret for objective i due to arm a. We will separately bound the regret contribution due to arms in the set $\mathcal{K}_\varepsilon$ and the arms not in $\mathcal{K}_\varepsilon$. We start by analyzing the regret due to pulling arms that are not in $\mathcal{K}_\varepsilon$. The number of rounds in a phase p is $T_p = \log_2 \frac{2}{\varepsilon}$. For every $a \in [k] \setminus \mathcal{K}_\varepsilon$, let $t(a) = \min\{t : 2^{-t} \leq \Delta_a^{(p(a))}/2\}$. The probability that $a \in [k] \setminus \mathcal{K}_\varepsilon$ is not eliminated by round $t(a)$ of phase $p(a)$ can be bounded by

$$\begin{aligned} &\Pr\left[a^* \notin A_{t(a)}\right] + \Pr\left[a \in A_{t(a)+1}, a^* \in A_{t(a)}\right] \\ &\leq \frac{1}{2n} + \exp\left(-\frac{m(t(a)) \left(\Delta_a^{(p(a))} - 2^{t(a)}\right)^2}{4}\right) \\ &\leq \frac{1}{n}. \end{aligned}$$

We can bound contribution to regret $R_n^{(i)}$ by arms $[k] \setminus \mathcal{K}_\varepsilon$ by

$$\sum_{a\in[k]\setminus\mathcal{K}_\varepsilon}\left(\frac{1}{n}\cdot n + m(t(a)) + \sum_{j=1}^{p(a)} T_j\right)\bar{\Delta}_a^{(i)} \qquad \text{(where } \bar{\Delta}_a^{(i)} = \max(\Delta_a^{(i)}, 0)\text{)}$$
$$\leq \sum_{a\in[k]\setminus\mathcal{K}_\varepsilon}\left(\bar{\Delta}_a^{(i)} + \frac{64\log(2n^2k)}{\bar{\Delta}_a^{(i)}} + 16\log(2n^2k)\sum_{j=1}^{p(a)}\frac{\bar{\Delta}_a^{(i)}}{\varepsilon_j^2}\right).$$

We use the trivial upper bound of n for the number of times arms in $\mathcal{K}_\varepsilon$ are pulled. We can bound the contribution to $R_n^{(i)}$ by arms $a \in \mathcal{K}_\varepsilon$ by $n \cdot \max_{a\in\mathcal{A}_{l+1}} \bar{\Delta}_a^{(i)}$. Summing the contributions of arms $a \in [k] \setminus \mathcal{K}_\varepsilon$ and of arms $a \in \mathcal{K}_\varepsilon$, we get the required bound. □

6.2.2 Proof of Proposition 3.8

If ε satisfies the three constraints of the theorem, then the regret of the phased algorithm with ε as input satisfies, for each $i \in [l-1]$, $R_n^{(i)}(\mathcal{A}_p(\varepsilon)) \leq 2\varepsilon_i n$. And for the l^{th} objective,

$$\begin{aligned} R_n^{(l)}(\mathcal{A}_p(\varepsilon)) &\leq 2\left(65\sqrt{n}k\log(2n^2k) + 16\log(2n^2k)\sum_{j=1}^{l-1}\frac{1}{\varepsilon_j^2}\right) \\ &\leq \frac{1}{4}C_1 k\log(nk)\left(\frac{1}{27}\sqrt{n} + \frac{\ln 3}{16\cdot 10^4}\sum_{j=1}^{i-1}\frac{1}{4\varepsilon^2}\right), \end{aligned}$$

for a universal constant C.

First we show that when ε satisfies the above conditions, the resulting phased algorithm $\mathcal{A}_p(\varepsilon)$ is $2C_1 k\log(nk)$-Pareto optimal. Suppose for the sake of a contradiction that there is an algorithm $\mathcal{A}$ such that $R_n(\mathcal{A}) \prec R_n(\mathcal{A}_p(\varepsilon))/C_1 k\log(nk)$. For such an algorithm $\mathcal{A}$, the inequalities $R_n^{(i)}(\mathcal{A}) \leq 2\varepsilon$ are satisfied. By the lower bound of theorem 3.6,

$$\begin{aligned} R_n^{(l)}(\mathcal{A}) &\geq \frac{1}{2\cdot 27}\sqrt{n} + \frac{\ln 3}{2\cdot 16\cdot 10^4}\sum_{i=1}^{l-1}\frac{n^2}{R_n^{(i)}(\mathcal{A})^2} \\ &\geq \frac{1}{2\cdot 27}\sqrt{n} + \frac{\ln 3}{2\cdot 16\cdot 10^4}\sum_{i=1}^{l-1}\frac{1}{4\varepsilon^2} \\ &\geq 2\frac{R_n^{(l)}(\mathcal{A}_p(\varepsilon))}{C_1 k\log(nk)} \end{aligned}$$

Since on objective l, $R_n^{(l)}(\mathcal{A})$ is greater than $R_n^{(l)}(\mathcal{A}_p(\varepsilon))/C_1 k\log(nk)$, this contradicts the assumption that $R_n(\mathcal{A}) \prec R_n(\mathcal{A}_p(\varepsilon))/C_1 k\log(nk)$.

Next we show the $Ck\log(nk)$-Pareto optimality completeness of the algorithms resulting from all ε that satisfy the constraints. For any algorithm A, by the lower bound we know that for every $i \in [l]$, $R_n^{(i)}(\mathcal{A}) \geq \max\left(\frac{1}{27}\sqrt{n}, \sum_{j=1}^{i-1}\frac{n^2}{(R_n^{(j)}(\mathcal{A}))^2}\right)$. By choosing $C_2'\varepsilon_i = \frac{R_n^{(i)}(\mathcal{A})}{n}$, for some universal constant C_2, we get $\varepsilon = (\varepsilon_i)_{i=1}^l$ that satisfy the constraints stated in the theorem. Using such ε, we obtain the phased algorithm $\mathcal{A}_p(\varepsilon)$ such that, for every objective $i \in [l]$ $R_n^{(i)}(\mathcal{A}_p(\varepsilon)) \geq \frac{1}{2}C_2 k\log(nk)R_n^{(i)}(\mathcal{A})$ for some universal constant C_3. This means that $R_n(\mathcal{A}_p(\varepsilon))/(C_3 k\log(nk)) \prec R_n(\mathcal{A})$.

This shows that the set of phased algorithms resulting from constraint satisfying ε's as input result in a complete $Ck\log(nk)$-Pareto optimal set for $C = \max(C_1, C_2)$. □

6.3 SUBOPTIMALITY OF OPTIMIZING LINEARLY SCALARIZED OBJECTIVE

6.3.1 Proof of Theorem 3.10

Consider the following set of bandit instance with two arms. Each arm has two objectives. The joint reward distribution is a product distribution of two unit variance Gaussians. So we can denote the distributions by their means. For any $\varepsilon > 0$, consider instances

$$\nu_\varepsilon = \left((0,0), \left(-\frac{rn^{2\varepsilon}}{\sqrt{n}}, 1\right)\right)$$
$$\nu_\varepsilon^{(1)} = \left((0,0), \left(-\frac{n^{2\varepsilon}}{\sqrt{n}}, -\frac{1}{r}\right)\right)$$
$$\nu_\varepsilon^{(2)} = \left((0,0), \left(\frac{n^{2\varepsilon}}{\sqrt{n}}, -\frac{1}{r}\right)\right).$$

We consider two cases on the magnitude of w_i's:

Case 1 : $|i : w_i \geq rn^\varepsilon/\sqrt{n}| \geq \frac{n}{2}$. When $w_t > \frac{rn^\varepsilon}{\sqrt{n}}$, for the instance ν_ε, if arm 1 is pulled instead of arm 2, we accumulate more than the following immediate regret toward $R_n^{(w)}$:

$$\frac{rn^{2\varepsilon}}{\sqrt{n}} - \frac{rn^\varepsilon}{\sqrt{n}} = \frac{rn^{2\varepsilon}}{\sqrt{n}}\left(1 - n^{-\varepsilon}\right) \in \Omega\left(rn^{2\varepsilon}/\sqrt{n}\right).$$

Since A_w has $R_n^{(w)} \in O(\sqrt{n})$, arm 1 cannot be pulled more than half of the rounds where $w_t > \frac{r}{\sqrt{n}}$. Doing so results in $R_n^{(w)} \in \Omega(rn^{1/2+2\varepsilon})$. Therefore in this case arm 2 is pulled at least $\frac{n}{4}$ times. Pulling arm 2 of ν contributes $\frac{rn^{2\varepsilon}}{\sqrt{n}}$ immediate regret to $R_n^{(1)}$. When arm 2 is pulled at least $\frac{n}{4}$ times, $R_n^{(1)}$ accumulated is $\Omega(rn^{\frac{1}{2}+2\varepsilon})$.

Case 2 : $|i : w_i < rn^\varepsilon/\sqrt{n}| \geq \frac{n}{2}$. In a round t when $w_t < \frac{rn^\varepsilon}{\sqrt{n}}$, for $\nu_\varepsilon^{(1)}$, arm 1 is optimal according to $R_n^{(w)}$ and for $\nu_\varepsilon^{(2)}$, arm 2 is optimal. Pulling a sub-optimal arm in such a round results in at least the following immediate regret toward $R_n^{(w)}$:

$$\frac{n^{2\varepsilon}}{\sqrt{n}} - \frac{rn^\varepsilon}{\sqrt{n}} \cdot \frac{1}{r} = \frac{n^{2\varepsilon} - n^\varepsilon}{\sqrt{n}} \in \Omega\left(1/n^{\frac{1}{2}-2\varepsilon}\right).$$

If a sub-optimal arm is played more than half of the rounds where $w_t < \frac{r}{\sqrt{n}}$ with probability greater than $\frac{1}{24}$, then $R_n^{(w)} \in \Omega\left(n^{\frac{1}{2}+\varepsilon}\right)$. Therefore, using the algorithm A_w, we can distinguish between instances ν_ε and ν'_ε with probability $\frac{11}{12}$ by observing an arm that A_w pulls most number of times among the set of rounds $\{i : w_i < rn^\varepsilon/\sqrt{n}\}$.

From lemma 6.3, we get that arm 2 of instance ν_ε is pulled by A_w $\Omega\left(n^{1-4\varepsilon}\right)$ times on expectation. During these pulls, we accumulate $\frac{n^{1-4\varepsilon}}{r}$ $R_n^{(2)}$. Since ε can be made arbitrarily close to zero, the worst case regret $R_n^{(2)}$ approaches $\Omega\left(n/r\right)$ □