Multi-Armed Bandits in Metric Spaces

Robert Kleinberg[*]
Computer Science Dept.
Cornell University
Ithaca, NY, USA
rdk@cs.cornell.edu

Aleksandrs Slivkins[†]
Microsoft Research
Mountain View, CA, USA
slivkins@microsoft.com

Eli Upfal[‡]
Computer Science Dept.
Brown University
Providence, RI, USA
eli@cs.brown.edu

ABSTRACT

In a multi-armed bandit problem, an online algorithm chooses from a set of strategies in a sequence of n trials so as to maximize the total payoff of the chosen strategies. While the performance of bandit algorithms with a small finite strategy set is quite well understood, bandit problems with large strategy sets are still a topic of very active investigation, motivated by practical applications such as online auctions and web advertisement. The goal of such research is to identify broad and natural classes of strategy sets and payoff functions which enable the design of efficient solutions.

In this work we study a very general setting for the multi-armed bandit problem in which the strategies form a metric space, and the payoff function satisfies a Lipschitz condition with respect to the metric. We refer to this problem as the *Lipschitz MAB problem*. We present a complete solution for the multi-armed problem in this setting. That is, for every metric space (L, X) we define an isometry invariant $\texttt{MaxMinCOV}(X)$ which bounds from below the performance of Lipschitz MAB algorithms for X, and we present an algorithm which comes arbitrarily close to meeting this bound. Furthermore, our technique gives even better results for benign payoff functions.

Categories and Subject Descriptors

F.1.2 [**Theory of Computation**]: Computation by Abstract Devices—*Modes of Computation: Online computation*

; F.2.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*Nonnumerical Algorithms*

General Terms

Algorithms, Theory

1. INTRODUCTION

In a multi-armed bandit problem, an online algorithm must choose from a set of strategies in a sequence of n trials so as to maximize the total payoff of the chosen strategies. These problems are the principal theoretical tool for modeling the exploration/exploitation tradeoffs inherent in sequential decision-making under uncertainty. Studied intensively for the last three decades [7, 8, 13], bandit problems are having an increasingly visible impact on computer science because of their diverse applications including online auctions, adaptive routing, and the theory of learning in games. The performance of a multi-armed bandit algorithm is often evaluated in terms of its *regret*, defined as the gap between the expected payoff of the algorithm and that of an optimal strategy. While the performance of bandit algorithms with a small finite strategy set is quite well understood, bandit problems with exponentially or infinitely large strategy sets are still a topic of very active investigation [1, 3, 4, 5, 6, 9, 10, 11, 12, 14, 15, 16, 17].

Absent any assumptions about the strategies and their payoffs, bandit problems with large strategy sets allow for no nontrivial solutions — any multi-armed bandit algorithm performs as badly, on some inputs, as random guessing. But in most applications it is natural to assume a structured class of payoff functions, which often enables the design of efficient learning algorithms [16]. In this paper, we consider a broad and natural class of problems in which the structure is induced by a metric on the space of strategies. While bandit problems have been studied in a few specific metric spaces (such as a one-dimensional interval) [1, 4, 9, 15, 19], the case of general metric spaces has not been treated before, despite being an extremely natural setting for bandit problems. As a motivating example, consider the problem faced by a website choosing from a database of thousands of banner ads to display to users, with the aim of maximizing the click-through rate of the ads displayed by matching ads to users' characterizations and the web content that they are currently watching. Independently experimenting with each advertisement is infeasible, or at least highly inefficient, since the number of ads is too large. Instead, the advertisements are usually organized into a taxonomy based on metadata (such as the category of product being advertised) which allows a similarity measure to be defined. The website can then attempt to optimize its learning algorithm by generalizing from experiments with one ad to make inferences about the performance of similar ads [19]. Abstractly, we have a bandit problem of the following form: there is a strategy set X, with an unknown payoff function $\mu : X \to [0, 1]$ satisfying a set of predefined constraints of the form $|\mu(u) - \mu(v)| \leq \delta(u, v)$ for some $u, v \in X$ and $\delta(u, v) > 0$. In each period the algorithm chooses a point

[*] Supported in part by NSF awards CCF-0643934 and CCF-0729102.

[†] Parts of this work were done while the author was a postdoc at Brown University.

[‡] Supported in part by NSF awards CCR-0121154 and DMI-0600384, and ONR Award N000140610607.

STOC'08, May 17–20, 2008, Victoria, British Columbia, Canada.

$x \in X$ and observes an independent random sample from a payoff distribution whose expectation is $\mu(x)$.

A moment's thought reveals that this abstract problem can be regarded as a bandit problem in a metric space. Specifically, if $L(u,v)$ is defined to be the infimum, over all finite sequences $u = x_0, x_1, \ldots, x_k = v$ in X, of the quantity $\sum_i \delta(x_i, x_{i+1})$, then L is a metric[1] and the constraints $|\mu(u) - \mu(v)| < \delta(u,v)$ may be summarized by stating that μ is a Lipschitz function (of Lipschitz constant 1) on the metric space (L, X). We refer to this problem as the *Lipschitz MAB problem* on (L, X), and we refer to the ordered triple (L, X, μ) as an *instance* of the Lipschitz MAB problem.[2]

Prior work. While our work is the first to treat the Lipschitz MAB problem in general metric spaces, special cases of the problem are implicit in prior work on the continuum-armed bandit problem [1, 4, 9, 15] — which corresponds to the space $[0,1]$ under the metric $L_d(x,y) = |x-y|^{1/d}$, $d \geq 1$ — and the experimental work on "bandits for taxonomies" [19], which corresponds to the case in which (L, X) is a tree metric. Before describing our results in greater detail, it is helpful to put them in context by recounting the nearly optimal bounds for the one-dimensional continuum-armed bandit problem, a problem first formulated by R. Agrawal in 1995 [1] and recently solved (up to logarithmic factors) by various authors [4, 9, 15]. In the following theorem and throughout this paper, the *regret* of a multi-armed bandit algorithm $\mathcal{A}$ running on an instance (L, X, μ) is defined to be the function $R_{\mathcal{A}}(t)$ which measures the difference between its expected payoff at time t and the quantity $t \sup_{x \in X} \mu(x)$. The latter quantity is the expected payoff of always playing a strategy $x \in \operatorname{argmax} \mu(x)$ if such strategy exists.

Theorem 1.1 ([4, 9, 15])**.** *For any $d \geq 1$, consider the Lipschitz MAB problem on $(L_d, [0,1])$. There is an algorithm $\mathcal{A}$ whose regret on any instance μ satisfies $R_{\mathcal{A}}(t) = \tilde{O}(t^\gamma)$ for every t, where $\gamma = \frac{d+1}{d+2}$. No such algorithm exists for any $\gamma < \frac{d+1}{d+2}$.*

In fact, if the time horizon t is known in advance, the upper bound in the theorem can be achieved by an extremely naïve algorithm which simply uses an optimal k-armed bandit algorithm (such as the UCB1 algorithm [2]) to choose strategies from the set $S = \{0, \frac{1}{k}, \frac{2}{k}, \ldots, 1\}$, for a suitable choice of the parameter k. While the regret bound in Theorem 1.1 is essentially optimal for the Lipschitz MAB problem in $(L_d, [0,1])$, it is strikingly odd that it is achieved by such a simple algorithm. In particular, the algorithm approximates the strategy set by a fixed mesh S and does not refine this mesh as it gains information about the location of the optimal strategy. Moreover, the metric contains seemingly useful proximity information, but the algorithm ignores this information after choosing its initial mesh. Is this really the best algorithm?

A closer examination of the lower bound proof raises further reasons for suspicion: it is based on a contrived, highly singular payoff function μ that alternates between being constant on some distance scales and being very steep on other (much smaller) distance scales, to create a multi-scale "needle in haystack" phenomenon which nearly obliterates the usefulness of the proximity information contained in the metric L_d. Can we expect algorithms to do better when the payoff function is more benign? For the Lipschitz MAB problem on $(L_1, [0,1])$, the question was answered affirmatively in [9, 4] for some classes of instances, with algorithms that are tuned to the specific classes.

Our results and techniques. In this paper we consider the Lipschitz MAB problem on arbitrary metric spaces. We are concerned with the following two main questions motivated by the discussion above:

(i) what is the best possible bound on regret for a given metric space?

(ii) how to take advantage of benign payoff functions?

In this paper we give a complete solution to (i), by describing for every metric space X a family of algorithms which come arbitrarily close to achieving the best possible regret bound for X. We also give a very satisfactory answer to (ii); our solution is arbitrarily close to optimal in terms of the zooming dimension defined below. In fact, our algorithm for (i) is an extension of the algorithmic technique used to solve (ii).

Our main technical contribution is a new algorithm, the *zooming algorithm*, that combines the upper confidence bound technique used in earlier bandit algorithms such as UCB1 with a novel *adaptive refinement* step that uses past history to zoom in on regions near the apparent maxima of μ and to explore a denser mesh of strategies in these regions. This algorithm is a key ingredient in our design of an optimal bandit algorithm for every metric space (L, X). Moreover, we show that the zooming algorithm can perform significantly better on benign problem instances. That is, for every instance (L, X, μ) we define a parameter called the *zooming dimension*, and use it to bound the algorithm's performance in a way that is often significantly stronger than the corresponding per-metric bound. Note that the zooming algorithm is *self-tuning*, i.e. it achieves this bound without requiring prior knowledge of the zooming dimension.

To state our theorem on the per-metric optimal solution for (i), we need to sketch a few definitions which arise naturally as one tries to extend the lower bound from [15] to general metric spaces. Let us say that a subset Y in a metric space X has covering dimension d if it can be covered by $O(\delta^{-d})$ sets of diameter δ for all $\delta > 0$. A point $x \in X$ has local covering dimension d if it has an open neighborhood of covering dimension d. The space X has max-min-covering dimension $d = \texttt{MaxMinCOV}(X)$ if it has no subspace whose local covering dimension is uniformly bounded below by a number greater than d.

Theorem 1.2. *Consider the Lipschitz MAB problem on a compact metric space (L, X). If $\gamma > \frac{d+1}{d+2}$, $d = \texttt{MaxMinCOV}(X)$ then there exists a bandit algorithm $\mathcal{A}$ satisfying $R_{\mathcal{A}}(t) = O(t^\gamma)$ for all t. No such algorithm exists if $\gamma < \frac{d+1}{d+2}$.*

In general $\texttt{MaxMinCOV}(X)$ is upper-bounded by the covering dimension of X. For metric spaces which are highly homogeneous (in the sense that any two ϵ-balls are isometric to one another) the two dimensions are equal, and the upper bound in the theorem can be achieved using a generalization of the naïve algorithm described earlier. The difficulty in Theorem 1.2 lies in dealing with inhomogeneities in the metric space.[3] It is important to treat the problem at this level of generality, because

[1] More precisely, it is a pseudometric because some pairs of distinct points $x, y \in X$ may satisfy $L(x,y) = 0$.

[2] When the metric space (L, X) is understood from context, we may also refer to μ as an instance.

[3] To appreciate this issue, it is very instructive to consider a concrete example of a metric space (L, X) where $\texttt{MaxMinCOV}(X)$ is strictly less than the covering dimension, and for this specific example design a bandit algorithm whose regret bounds

some of the most natural applications of the Lipschitz MAB problem, e.g. the web advertising problem described earlier, are based on highly inhomogeneous metric spaces. (That is, in web taxonomies, it is unreasonable to expect different categories at the same level of a topic hierarchy to have the roughly the same number of descendants.)

The algorithm in Theorem 1.2 combines the zooming algorithm described earlier with a delicate transfinite construction over closed subsets consisting of "fat points" whose local covering dimension exceeds a given threshold d. For the lower bound, we craft a new dimensionality notion, the max-min-covering dimension introduced above, which captures the inhomogeneity of a metric space, and we connect this notion with the transfinite construction that underlies the algorithm.

For "benign" input instances we provide a better performance guarantee for the zooming algorithm. The lower bounds in Theorems 1.1 and 1.2 are based on contrived, highly singular, "needle in haystack" instances in which the set of near-optimal strategies is astronomically larger than the set of precisely optimal strategies. Accordingly, we quantify the tractability of a problem instance in terms of the number of near-optimal strategies. We define the *zooming dimension* of an instance (L, X, μ) as the smallest d such that the following covering property holds: for every $\delta > 0$ we require only $O(\delta^{-d})$ sets of diameter $\delta/8$ to cover the set of strategies whose payoff falls short of the maximum by an amount between δ and 2δ.

Theorem 1.3. *If d is the zooming dimension of a Lipschitz MAB instance then at any time t the zooming algorithm suffers regret $\tilde{O}(t^{\gamma})$, $\gamma = \frac{d+1}{d+2}$. Moreover, this is the best possible exponent γ as a function of d.*

The zooming dimension can be significantly smaller than the max-min-covering dimension.[4] Let us illustrate this point with two examples (where for simplicity the max-min-covering dimension is equal to the covering dimension). For the first example, consider a metric space consisting of a high-dimensional part and a low-dimensional part. For concreteness, consider a rooted tree T with two top-level branches T' and T'' which are complete infinite k-ary trees, $k = 2, 10$. Assign edge weights in T that are exponentially decreasing with distance to the root, and let L be the resulting shortest-paths metric on the leaf set X.[5] Then if there is a unique optimal strategy that lies in the low-dimensional part T' then the zooming dimension is upper-bounded by the covering dimension in T', whereas the "global" covering dimension is that in T''. In the second example, let (L, X) be a homogeneous high-dimensional metric, e.g. the Euclidean metric on the unit k-cube, and the payoff function is $\mu(x) = 1 - L(x, S)$ for some subset S. Then the zooming dimension is equal to the covering dimension of S, e.g. it is 0 if S is a finite point set.

Discussion. In stating the theorems above, we have been imprecise about specifying the model of computation. In particular, we have ignored the thorny issue of how to provide an algorithm with an input containing a metric space which may have an infinite number of points. The simplest way to interpret our theorems is to ignore implementation details and interpret "algorithm" to mean an abstract decision rule, i.e. any function mapping a history of past observations $(x_i, r_i) \in X \times [0, 1]$ to a strategy $x \in X$ which is played in the current period. All of our theorems are valid under this interpretation, but they can also be made into precise algorithmic results provided that the algorithm is given appropriate oracle access to the metric space. In most cases, our algorithms require only a *covering oracle* which takes a finite collection of open balls and either declares that they cover X or outputs an uncovered point. We refer to this setting as the standard Lipschitz MAB problem. For example, the zooming algorithm requires only a covering oracle for (L, X), and the algorithm is very efficient, requiring only $O(t \log t)$ operations in total (including oracle queries) to choose its first t strategies. However, the per-metric optimal algorithm in Theorem 1.2 requires a more complicated pair of oracles, and we defer the definition of these oracles to Section 3.

While our definitions and results so far have been tailored for the Lipschitz MAB problem on infinite metrics, some of them can be extended to the finite case as well. In particular, for the zooming algorithm we obtain sharp results (that are meaningful for both finite and infinite metrics) using a more precise, *non-asymptotic* version of the zooming dimension. Extending the notions in Theorem 1.2 to the finite case is an open question.

Preliminaries. Given a metric space, $B(u, r)$ denotes an open ball of radius r. Throughout the paper, he constants in the $O(\cdot)$ notation are absolute unless specified otherwise.

Definition 1.4. In the *Lipschitz MAB problem* on (L, X), there is a strategy set X, a metric space (L, X) of diameter ≤ 1, and an unknown payoff function $\mu : X \to [0, 1]$ such that $|\mu(x) - \mu(y)| \leq L(x, y)$ for all $x, y \in X$. (Call L a *Lipschitz metric* for μ.) In each round the algorithm chooses $x \in X$ and observes an independent random sample from a payoff distribution with support $[0; 1]$ and expectation $\mu(x)$.

The *regret* of a bandit algorithm $\mathcal{A}$ running on a given problem instance is the $R_{\mathcal{A}}(t) = W_{\mathcal{A}}(t) - t\mu^*$, where $W_{\mathcal{A}}(t)$ is the expected payoff of $\mathcal{A}$ at time t and $\mu^* = \sup_{x \in X} \mu(x)$ is the *maximal expected reward*.

The *c-zooming dimension* of the problem instance (L, X, μ) is the smallest d such that for any $r \in (0, 1]$ the set $X_r = \{x \in X : \frac{r}{2} < \mu^* - \mu(x) \leq r\}$ can be covered by $c\, r^{-d}$ sets of diameter at most $r/8$.

Definition 1.5. Fix a metric space on set X. Let $N(r)$ be the smallest number of sets of diameter r required to cover X. The *covering dimension* of X is $\mathtt{COV}(X) = \inf\{\, d : \;\exists c\; \forall r > 0 \quad N(r) \leq c r^{-d} \,\}$. The *c-covering dimension* of X is defined as the infimum of all d such that $N(r) \leq c r^{-d}$ for all $r > 0$.

2. THE ZOOMING ALGORITHM

In this section we present the *zooming algorithm*. Consider the standard Lipschitz MAB problem on (L, X). The algorithm proceeds in phases $i = 1, 2, 3, \ldots$ of 2^i rounds each. Let us focus on a single phase $i_{\mathtt{ph}}$ of the zooming algorithm. For each strategy $v \in X$ and time t, let $n_t(v)$ be the number of times this strategy has been played in this phase before time t, and let $\mu_t(v)$ be the corresponding average reward. Define $\mu_t(v) = 0$ if $n_t(v) = 0$. Note that at time t both quantities are known to the algorithm. Define the *confidence radius* of v as

$$r_t(v) := \sqrt{8\, i_{\mathtt{ph}} \,/\, (2 + n_t(v))}. \tag{1}$$

Let $\mu(v)$ be the expected reward of strategy v. Note that $E[\mu_t(v)] = \mu(v)$. Using Chernoff Bounds, we can bound $|\mu_t(v) - \mu(v)|$ in terms of the confidence radius:

are better than those suggested by the covering dimension. This is further discussed in Section 3.

[4] One can show that in this case the naïve algorithm from Theorem 1.1 performs poorly compared to the zooming algorithm.

[5] Here a *leaf* is defined as an infinite path away from the root.

Definition 2.1. A phase is called *clean* if for each strategy $v \in X$ that has been played at least once during this phase and each time t we have $|\mu_t(v) - \mu(v)| \le r_t(v)$.

Claim 2.2. *Phase i_{ph} is clean with probability at least* $1 - 4^{-i_{\text{ph}}}$.

Throughout the execution of the algorithm, a finite number of strategies are designated *active*. Our algorithm only plays active strategies, among which it chooses a strategy v with the maximal *index*

$$I_t(v) = \mu_t(v) + 2\, r_t(v). \tag{2}$$

Say that strategy v *covers* strategy u at time t if $u \in B(v,\, r_t(v))$. Say that a strategy u is *covered* at time t if at this time it is covered by some active strategy v. Note that the *covering oracle* (as defined in Section 1) can return a strategy which is not covered if such strategy exists, or else inform the algorithm that all strategies are covered. Now we are ready to state the algorithm and the corresponding theorem:

Algorithm 2.3 (Zooming Algorithm)**.** *Each phase i runs for 2^i rounds. In the beginning of the phase no strategies are active. In each round do the following:*

1. *If some strategy is not covered, make it active.*
2. *Play an active strategy with the maximal index (2); break ties arbitrarily.*

Theorem 2.4. *Consider the standard Lipschitz MAB problem. Let $\mathcal{A}$ be Algorithm 2.3. Then $\forall\, C > 0$*

$$R_{\mathcal{A}}(t) \le O(C \log t)^{1/(2+d)} \times t^{1-1/(2+d)} \quad \textit{for all } t, \tag{3}$$

where d is the C-zooming dimension of the problem instance.

In the remainder of this section we prove Theorem 2.4. Note that after step 1 in Algorithm 2.3 all strategies are covered. (Indeed, if some strategy is activated in step 1 then it covers the entire metric.) Let $\mu^* = \sup_{u \in X} \mu(u)$ be the maximal expected reward; note that we do not assume that the supremum is achieved by some strategy. Let $\Delta(v) = \mu^* - \mu(v)$. Let us focus on a given phase i_{ph} of the algorithm.

Lemma 2.5. *If phase i_{ph} is clean then $\Delta(v) \le 4\, r_t(v)$ for any time t and any strategy v. Hence $n_t(v) \le O(i_{\text{ph}})\, \Delta^{-2}(v)$.*

PROOF. Suppose strategy v is played at time t. First we claim that $I_t(v) \ge \mu^*$. Indeed, fix $\epsilon > 0$. By definition of μ^* there exists a strategy v^* such that $\Delta(v^*) < \epsilon$. Let v_t be an active strategy that covers v^*. By the algorithm specification $I_t(v) \ge I_t(v_t)$. Since v is clean at time t, by definition of index we have $I_t(v_t) \ge \mu(v_t) + r_t(v_t)$. By the Lipschitz property we have $\mu(v_t) \ge \mu(v^*) - L(v_t, v^*)$. Since v_t covers v^*, we have $L(v_t, v^*) \le r_t(v_t)$ Putting all these inequalities together, we have $I_t(v) \ge \mu(v^*) \ge \mu^* - \epsilon$. Since this inequality holds for an arbitrary $\epsilon > 0$, we in fact have $I_t(v) \ge \mu^*$. Claim proved.

Furthermore, note that by the definitions of "clean phase" and "index" we have $\mu^* \le I_t(v) \le \mu(v) + 3\, r_t(v)$ and therefore $\Delta(v) \le 3\, r_t(v)$.

Now suppose strategy v is not played at time t. If it has never been played before time t in this phase, then $r_t(v) > 1$ and thus the lemma is trivial. Else, let s be the last time strategy v has been played before time t. Then by definition of the confidence radius $r_t(v) = r_{s+1}(v) \ge \sqrt{2/3}\, r_s(v) \ge \frac{1}{4}\, \Delta(v)$. □

Corollary 2.6. *In a clean phase, for any active strategies u, v*

$$L(u, v) > \tfrac{1}{4} \min(\Delta(u), \Delta(v)).$$

PROOF. Assume u has been activated before v. Let s be the time when v has been activated. Then by the algorithm specification we have $L(u, v) > r_s(u)$. By Lemma 2.5 we have $r_s(u) \ge \frac{1}{4}\Delta(u)$. □

Let d be the the C-zooming dimension. For a given time t in the current phase, let $S(t)$ be the set of all strategies that are active at time t, and let

$$A(i, t) = \{v \in S(t) :\ 2^i \le \Delta^{-1}(v) < 2^{i+1}\}.$$

We claim that $|A(i,t)| \le C\, 2^{id}$. Indeed, set $A(i,t)$ can be covered by $C\, 2^{id}$ sets of diameter at most $2^{-i}/8$; by Corollary 2.6 each of these sets contains at most one strategy from $A(i,t)$.

Claim 2.7. *In a clean phase i_{ph}, for each time t we have*

$$\textstyle\sum_{v \in S(t)} \Delta(v)\, n_t(v) \le O(C\, i_{\text{ph}})^{1-\gamma}\, t^{\gamma}, \tag{4}$$

where $\gamma = \frac{d+1}{d+2}$ and d is the C-zooming dimension.

PROOF. Fix the time horizon t. For a subset $S \subset X$ of strategies, let $R_S = \sum_{v \in S} \Delta(v)\, n_t(v)$. Let us choose $\rho \in (0,1)$ such that $\rho t = (\frac{1}{\rho})^{d+1} (C\, i_{\text{ph}}) = t^{\gamma}\, (C\, i_{\text{ph}})^{1-\gamma}$.

Define B as the set of all strategies $v \in S(t)$ such that $\Delta(v) \le \rho$. Recall that by Lemma 2.5 for each $v \in A(i,t)$ we have $n_t(v) \le O(i_{\text{ph}})\, \Delta^{-2}(v)$. Then

$$\begin{aligned}
R_{A(i,t)} &\le O(i_{\text{ph}}) \textstyle\sum_{v \in A(i,t)} \Delta^{-1}(v) \\
&\le O(2^i\, i_{\text{ph}})\, |A(i,t)| \\
&\le O(C\, i_{\text{ph}})\, 2^{i(d+1)} \\
\sum_{v \in S(t)} \Delta(v)\, n_t(v) &\le R_B + \sum_{i < \log(1/\rho)} R_{A(i,t)} \\
&\le \rho t + O(C\, i_{\text{ph}})\, (\tfrac{1}{\rho})^{d+1} \\
&\le O\left(t^{\gamma}\, (C\, i_{\text{ph}})^{1-\gamma}\right). \qquad \square
\end{aligned}$$

The left-hand side of (4) is essentially the contribution of the current phase to the overall regret. It remains to sum these contributions over all phases.

Proof of Theorem 2.4 Let i_{ph} be the current phase, let t be the time spend in this phase, and let T be the total time since the beginning of phase 1. Let $R_{\text{ph}}(i_{\text{ph}}, t)$ be the left-hand side of (4). Combining Claim 2.2 and Claim 2.7, we have

$$\begin{aligned}
E[R_{\text{ph}}(i_{\text{ph}}, t)] &< O(C\, i_{\text{ph}})^{1-\gamma}\, t^{\gamma}, \\
R_{\mathcal{A}}(T) &= E\left[R_{\text{ph}}(i_{\text{ph}}, t) + \sum_{i=1}^{i_{\text{ph}}-1} R_{\text{ph}}(i, 2^i) \right] \\
&< O(C\, \log T)^{1-\gamma}\, T^{\gamma}.
\end{aligned}$$

3. PER-METRIC OPTIMALITY

In this section we ask, "What is the best possible algorithm for the Lipschitz MAB problem on a given metric space?" We consider the *per-metric performance*, which we define as the worst-case performance of a given algorithm over all possible problem instances on a given metric. As everywhere else in this paper, we focus on minimizing the exponent γ such that $R_{\mathcal{A}}(t) \le t^{\gamma}$ for all sufficiently large t. Equivalently, we can try to minimize the *regret dimension* defined as follows.

Definition 3.1. Consider the Lipschitz MAB problem on a given metric space. For algorithm $\mathcal{A}$ and problem instance $\mathcal{I}$ let $\texttt{DIM}_{\mathcal{I}}(\mathcal{A}) = \inf_d \{\exists t_0\ \forall t \ge t_0 \ \ R_{\mathcal{A}}(t) \le t^{1-1/(d+2)}\}$. The *regret dimension* of $\mathcal{A}$ is $\mathtt{DIM}(\mathcal{A}) = \sup_{\mathcal{I}} \mathtt{DIM}_{\mathcal{I}}(\mathcal{A})$, where the supremum is over all problem instances $\mathcal{I}$.

Recall from Section 1 that if d is the covering dimension of (L, X), then regret dimension d can be achieved by the "naïve algorithm" which divides time into phases of exponentially increasing length, chooses a δ-net of cardinality $K = O(\delta^{-d})$ during each phase (tuning the parameter δ optimally given the phase length), and runs a K-armed bandit algorithm such as UCB1 on the elements of the δ-net. In fact, the covering dimension is the best regret dimension achievable by a naïve algorithm of this sort (we omit the proof), and for highly homogeneous metric spaces (such as those in which all balls of a given radius are isometric to each other) it is the optimal regret dimension of *any* bandit algorithm. We next discuss the proof of this fact.

It is known [3] that a worst-case instance of the K-armed bandit problem consists of $K - 1$ strategies with identical payoff distributions, and one which is slightly better. We refer to this as a "needle-in-haystack" instance. The construction of lower bounds for Lipschitz MAB problems relies on creating a *multi-scale* needle-in-haystack instance in which there are K disjoint open sets, and $K - 1$ of them consist of strategies with identical payoff distributions, but in the remaining open set there are strategies whose payoff is slightly better. Moreover, this special open set contains $K' \gg K$ disjoint subsets, only one of which contains strategies superior to the others, and so on down through infinitely many levels of recursion. To ensure that this construction can be continued indefinitely, one needs to assume a covering property which ensures that *each* of the open sets arising in the construction has sufficiently many disjoint subsets to continue to the next level of recursion.

Definition 3.2. For a metric space (L, X), we say that d is the *min-covering dimension* of X, $d = \mathtt{MinCOV}(X)$, if d is the infimum of $\mathtt{COV}(U)$ over all non-empty open subsets $U \subseteq X$. The *max-min-covering dimension* of X is defined by

$$\mathtt{MaxMinCOV}(X) = \sup\{\mathtt{MinCOV}(Y) \,:\, Y \subseteq X\}.$$

The infimum over open $U \subseteq X$ in the definition of min-covering dimension ensures that every open set which may arise in the needle-in-haystack construction described above will contain $\Omega(\delta^{\varepsilon - d})$ disjoint δ-balls for some sufficiently small δ, ε. Constructing lower bounds for Lipschitz MAB algorithms in a metric space X only requires that X should have *subsets* with large min-covering dimension, which explains the supremum over subsets in the definition of max-min-covering dimension.

We will use the following simple packing lemma.[6]

Lemma 3.3. *If Y is a metric space of covering dimension d, then for any $b < d$ and $r_0 > 0$, there exists $r \in (0, r_0)$ such that Y contains a collection of more than r^{-b} disjoint open balls of radius r.*

Proof. Let $r < r_0$ be a positive number such that every covering of Y requires more than r^{-b} balls of radius $2r$. Such an r exists, because the covering dimension of Y is strictly greater than b. Now let $\mathcal{P} = \{B_1, B_2, \ldots, B_M\}$ be any maximal collection of disjoint r-balls. For every $y \in Y$ there must exist some ball B_i $(1 \leq i \leq M)$ whose center is within distance $2r$ of y, as otherwise $B(y, r)$ would be disjoint from every element of $\mathcal{P}$ contradicting the maximality of that collection. If we enlarge each ball B_i to a ball B_i^+ of radius $2r$, then every $y \in Y$ is contained in one of the balls $\{B_i^+ \,|\, 1 \leq i \leq M\}$, i.e. they form a covering of Y. Hence $M \geq r^{-b}$ as desired. □

Theorem 3.4. *If X is a metric space and d is the max-min-covering dimension of X then $\mathtt{DIM}(\mathcal{A}) \geq d$ for every bandit algorithm $\mathcal{A}$.*

Proof. Given $\gamma < \frac{d+1}{d+2}$, let $a < b < c < d$ be such that $\gamma < \frac{a+1}{a+2}$. Let Y be a subset of X such that $\mathtt{MinCOV}(Y) \geq c$. Using Lemma 3.3 we recursively construct an infinite sequence of sets $\mathcal{P}_0, \mathcal{P}_1, \ldots$ each consisting of finitely many disjoint open balls in X, centered at points of Y. Let $\mathcal{P}_0 = \{X\} = B(y_0, r_{\max})$, where y_0 is an arbitrary point in Y and $r_{\max}$ is a number greater than or equal to the diameter of X. If $i > 0$, for every ball $B \in \mathcal{P}_{i-1}$, let r denote the radius of B and choose a number $0 < r_i(B) < r/4$ such that B contains $n_i(B) = \lceil r_i(B)^{-b} \rceil$ disjoint balls centered at points of Y. Such a collection of disjoint balls exists, by Lemma 3.3. Let $\mathcal{P}_i(B)$ denote this collection of disjoint balls and let $\mathcal{P}_i = \bigcup_{B \in \mathcal{P}_{i-1}} \mathcal{P}_i(B)$.

Now sample a random sequence of balls $B_1, B_2, \ldots$ by picking $B_1 \in \mathcal{P}_1$ uniformly at random, and for $i > 1$ picking $B_i \in \mathcal{P}_i(B_{i-1})$ uniformly at random. Let x_i, r_i be the center and radius of B_i, and let $f_i(x)$ be a Lipschitz function on X defined by

$$f_i(x) = \begin{cases} \min\{r_i - L(x, x_i), r_i/2\} & \text{if } x \in B_i \\ 0 & \text{otherwise} \end{cases}.$$

Let $f_0(x) = 1/3$ for all $x \in X$. The reader may verify that the sum $\mu = \sum_{i=0}^{\infty} f_i$ is a Lipschitz function. Define the payoff distribution for $x \in X$ to be a Bernoulli random variable with expectation $\mu(x)$. We have thus specified a randomized construction of an instance (L, X, μ). We claim that for every algorithm $\mathcal{A}$ and every constant C,

$$\Pr_{\mu, \mathcal{A}} (\forall\, t\ R_{\mathcal{A}}(t) < C t^{\gamma}) = 0. \qquad (5)$$

The proof of this claim is based on a "needle in haystack" lemma (Lemma 3.6 below) which states that for all i, conditional on the sequence $B_1, \ldots, B_{i-1}$, with probability at least $1 - O((r_i(B_i))^{(b-a)/2})$, no more than half of the first $t_i(B_i) = r_i(B_i)^{-a-2}$ strategies picked by $\mathcal{A}$ lie inside B_i. The proof of the lemma is deferred to the end of this section.

Any strategy $x \notin B_i$ satisfies $\mu(x) < \mu(x^*) - r_i/2$, so we may conclude that

$$\Pr\left(R_{\mathcal{A}}(t_i(B_i)) < \tfrac{1}{4}\, r_i(B_i)^{-a-1} \,|\, B_1, \ldots, B_{i-1}\right) \leq O\left((r_i(B_i))^{(b-a)/2}\right). \qquad (6)$$

Denoting $r_i(B_i)$ and $t_i(B_i)$ by r_i, t_i, respectively, we have $\frac{1}{4}\, r_i^{-a-1} = \frac{1}{4}\, t_i^{(a+1)/(a+2)} > C t_i^{\gamma}$ for all sufficiently large i. As i runs through the positive integers, the terms on the right side of (6) are dominated by a geometric progression because $r_i(B_i) \leq 4^{-i}$. By the Borel-Cantelli Lemma, almost surely there are only finitely many i such that the events on the left side of (6) occur. Thus (5) follows. □

Remark. To prove Theorem 3.4 it suffices to show that for every given algorithm there exists a "hard" problem instance. In fact we proved a stronger result (5): essentially, we construct a probability distribution over problem instances which is hard, almost surely, for every given algorithm. This seems to be the best possible bound since, obviously, a single problem instance cannot be hard for every algorithm.

In Section 3.1 we will show that for some metric spaces, there exist algorithms whose regret dimension is smaller than

[6] This is a folklore result; we provide the proof for convenience.

the covering dimension. We develop these ideas further in Section 3.2 and provide an algorithm whose regret dimension is arbitrarily close to optimal.

The needle-in-haystack lemma. We conclude this section by providing a precise formulation and proof of the "needle in haystack" lemma used in the proof of Theorem 3.4. To do this, we need to introduce some notation. Let us fix an abitrary Lipschitz MAB algorithm $\mathcal{A}$. We will assume that $\mathcal{A}$ is deterministic; the corresponding result for randomized algorithms follows by conditioning on the algorithm's random bits (so that its behavior, conditional on these bits, is deterministic), invoking the lemma for deterministic algorithms, and then removing the conditioning by averaging over the distribution of random bits. Note that since our construction uses only $\{0,1\}$-valued payoffs, and the algorithm $\mathcal{A}$ is deterministic, the entire history of play in the first t rounds can be summarized by a binary vector $\sigma \in \{0,1\}^t$, consisting of the payoffs observed by $\mathcal{A}$ in the first t rounds. Thus a payoff function μ determines a probability distribution P_μ on the set $\{0,1\}^t$, i.e. the distribution on t-step histories realized when using algorithm $\mathcal{A}$ on instance μ.

Let B be any ball in the set $\mathcal{P}_{i-1}$, let

$$n = n_i(B), \quad r = r_i(B), \quad t = t_i(B),$$

and let $B^1, B^2, \dots, B^n$ be an enumeration of the balls in $\mathcal{P}(B)$. Choose an arbitrary sequence of balls $B_1 \supseteq B_2 \supseteq \dots \supseteq B_{i-1} = B$ such that $B_1 \in \mathcal{P}_1$ and for all $j > 0$ $B_j \in \mathcal{P}(B_{j-1})$. Similarly, for $k = 1, 2, \dots, n$, choose an arbitrary sequence of balls $B^k = B^k_i \supseteq B^k_{i+1} \supseteq \dots$ such that $B^k_j \in \mathcal{P}(B^k_{j-1})$ for all $j \geq i$. Define functions f_j $(1 \leq j \leq i-1)$ and f^k_j $(j \geq i)$ using the balls B_j, B^k_j, as in the proof of Theorem 3.4. Let $\mu^0 = \sum_{j=0}^{i-1} f_j$ and

$$\mu^k = \mu^0 + \sum_{j=i}^{\infty} f^k_j \quad (\text{for } 1 \leq k \leq n).$$

Note that the instances μ^k $(1 \leq k \leq n)$ are equiprobable under our distribution on input instances μ. The instance μ^0 is not one that could be randomly sampled by our construction, but it is useful as a "reference measure" in the following proof. Note that the functions μ^k have the following properties, by construction.

(a) $1/3 \leq \mu^k(x) \leq 2/3$ for all $x \in X$.
(b) $0 \leq \mu^k(x) - \mu^0(x) \leq r$ for all $x \in X$.
(c) If $x \in X \setminus B^k$, then $\mu^k(x) = \mu^0(x)$.
(d) If $x \in X \setminus B^k$, then there exists some point $x^k \in B^k$ such that $\mu^k(x^k) - \mu^k(x) \geq r/2$.

Each of the payoff functions μ^k $(0 \leq k \leq n)$ gives rise to a probability distribution P_{μ^k} on $\{0,1\}^t$ as described in the preceding section. We will use the shorthand notation P_k instead of P_{μ^k}. We will also use $\mathbf{E}_k$ to denote the expectation of a random variable under distribution P_k. Finally, we let N_k denote the random variable defined on $\{0,1\}^t$ that counts the number of rounds s $(1 \leq s \leq t)$ in which algorithm $\mathcal{A}$ chooses a strategy in B^k given the history σ.

The following lemma is analogous to Lemma A.1 of [3], and its proof is identical to the proof of that lemma.

Lemma 3.5. *Let $f : \{0,1\}^t \to [0, M]$ be any function defined on reward sequences σ. Then for any k,*

$$\mathbf{E}_k[f(\sigma)] \leq \mathbf{E}_0[f(\sigma)] + \tfrac{M}{2}\sqrt{-\ln(1-4r^2)\mathbf{E}_0[N_i]}.$$

Applying Lemma 3.5 with $f = N_k$ and $M = t$, and averaging over k, we may apply exactly the same reasoning as in the proof of Theorem A.2 of [3] to derive the bound

$$\frac{1}{n}\sum_{k=1}^{n} \mathbf{E}_k(N_k) \leq \frac{t}{n} + O\left(tr\sqrt{\frac{t}{n}}\right). \tag{7}$$

Recalling that the actual ball B_k sampled when randomly constructing μ in the proof of Theorem 3.4 is a uniform random sample from $B^1, B^2, \dots, B^n$, we may write N_* to denote the random variable which counts the number of rounds in which the algorithm plays a strategy in B_k and the bound (7) implies

$$\mathbf{E}(N_*) = O\left(\frac{t}{n} + tr\sqrt{\frac{t}{n}}\right)$$

Recalling that $t = r^{-a-2}$ and $n = r^{-b}$, we see that the $O(tr\sqrt{t/n})$ term is the dominant term on the right side, and that it is bounded by $O(tr^{(b-a)/2})$. An application of Markov's inequality now yields:

Lemma 3.6. $\Pr(N_* \geq t/2) = O(r^{(b-a)/2})$.

3.1 Beyond the covering dimension

Thus far, we have seen that every metric space X has a bandit algorithm $\mathcal{A}$ such that $\texttt{DIM}(\mathcal{A}) = \texttt{COV}(X)$ (the naïve algorithm), and we have seen (via the needle-in-haystack construction, Theorem 3.4) that X can never have a bandit algorithm satisfying $\texttt{DIM}(\mathcal{A}) < \texttt{MaxMinCOV}(X)$. When $\texttt{COV}(X) \neq \texttt{MaxMinCOV}(X)$, which of these two bounds is correct, or can they both be wrong? To gain intuition, we will consider two concrete examples. Consider an infinite rooted tree where for each level $i \in \mathbb{N}$ most nodes have out-degree 2, whereas the remaining nodes (called *fat nodes*) have out-degree $x > 2$ so that the total number of nodes is 4^i. In our first example, there is exactly one fat node on every level and the fat nodes form a path (called the *fat leaf*). In our second example, there are exactly 2^i fat nodes on every level i and the fat nodes form a binary tree (called the *fat subtree*). In both examples, we assign a *weight* of 2^{-id} (for some constant $d > 0$) to each level-i node; this weight encodes the diameter of the set of points contained in the corresponding subtree. An infinite rooted tree induces a metric space (L, X) where X is the set of all infinite paths from the root, and for $u, v \in X$ we define $L(u,v)$ to be the weight of the least common ancestor of paths u and v. In both examples, the covering dimension is $2d$, whereas the max-min-covering dimension is only d because the "fat subset" (i.e. the fat leaf or fat subtree) has covering dimension at most d, and every point outside the fat subset has an open neighborhood of covering dimension d. In the next few paragraphs, we sketch some algorithms for dealing with certain metric spaces that have fat subsets, as a means of building intuition leading up to the (rather complicated) optimal algorithm for general metric spaces. The gory details are omitted until we reach the description of the algorithm for general metric spaces.

In both of the metrics described above, the zooming algorithm (Algorithm 2.3) performs poorly when the optimum x^* is located inside the fat subset S, because it is too burdensome to keep covering[7] the profusion of strategies located near x^* as the ball containing x^* shrinks. An improved algorithm, achieving regret exponent d, modifies the zooming algorithm by imposing *quotas* on the number of active strategies that lie outside S. At any given time, some strategies outside S may not be covered; however, it is guaranteed that there exists an optimal

[7] Recall that a strategy u is called *covered* at time t if for some active strategy v we have $L(u,v) \leq r_t(v)$.

strategy which eventually becomes covered and remains covered forever afterward. Intuitively, if some optimal strategy lies in S then imposing a quota on active strategies outside S does not hurt. If no optimal strategy lies in S then all of S gets covered eventually and stays covered thereafter, in which case the uncovered part of the strategy set has low covering dimension and (starting after the time when S becomes permanently covered) no quota is ever exceeded.

This use of quotas extends to the following general setting which abstracts the idea of "fat subsets":

Definition 3.7. Fix a metric space (L, X). A closed subset $S \subset X$ is *d-fat* if $\texttt{COV}(S) \leq d$ and for any open superset U of S we have $\texttt{COV}(X \setminus U) \leq d$. More generally, a *d-fat decomposition* of depth k is a decreasing sequence $X = S_0 \supset \ldots \supset S_k \supset S_{k+1} = \emptyset$ of closed subsets such that $\texttt{COV}(S_k) \leq d$ and $\texttt{COV}(S_i \setminus U) \leq d$ whenever $i \in [k]$ and U is an open superset of S_{i+1}.

Example 3.8. Let (L, X) be the metric space in either of the two "tree with a fat subset" examples. Then the corresponding "fat subset" S is d-fat. For an example of a fat decomposition of depth $k = 2$, consider the product metric $(L^*, X \times X)$ defined by $L^*((x_1, x_2), (y_1, y_2)) = L(x_1, y_1) + L(x_2, y_2)$, with a fat decomposition given by $S_1 = (S \times X) \cup (X \times S)$ and $S_2 = S \times S$.

When X is a metric space with a d^*-fat decomposition $\mathcal{D}$, the algorithm described earlier can be modified to achieve regret $O\left(t^{\gamma}\right)$ for any $\gamma > 1 - 1/(d^* + 2)$, by instituting a separate quota for each subset S_i. The algorithm requires access to a *$\mathcal{D}$-covering oracle* which for a given i and a given finite set of open balls (given by the centers and the radii) either reports that the balls cover S_i, or returns some strategy in S_i which is not covered by the balls. No further knowledge of $\mathcal{D}$ or the metric space is required.

Theorem 3.9. *Consider the Lipschitz MAB problem on a fixed compact metric space with a d^*-fat decomposition $\mathcal{D}$. Then for any $d > d^*$ there is an algorithm $\mathcal{A}_{\mathcal{D}}$ such that $\texttt{DIM}(\mathcal{A}_{\mathcal{D}}) \leq d$.*

Remarks. **(1)** We can relax the compactness assumption in Theorem 3.9: instead, we can assume that the *completion* of the metric space is compact and re-define the sets in the d-fat decomposition as subsets of the completion (possibly disjoint with the strategy set). This corresponds to the "fat leaf" which lies outside the strategy set. Such extension requires some minor modifications; we discuss this further in the full version.

(2) The per-metric guarantee expressed by Theorem 3.9 can be complemented with sharper *per-instance* guarantees. First, for every problem instance $\mathcal{I}$ the per-instance regret dimension $\texttt{DIM}_{\mathcal{I}}(\mathcal{A})$ is upper-bounded by the zooming dimension of $\mathcal{I}$. Second, if the c-covering dimension of X is finite then for some $\gamma < 1$ and all t we have $R_{\mathcal{A}}(t) \leq O(c\, t^{\gamma})$. However, for the ease of exposition in the present version we focus on analyzing the regret dimension.

Our algorithm proceeds in phases $i = 1, 2, 3, \ldots$ of 2^i rounds each. In a given phase, we run a fresh instance of the following *phase algorithm* $\mathcal{A}_{\texttt{ph}}(T, d, \mathcal{D})$ parameterized by the phase length $T = 2^i$, *target dimension* $d > d^*$ and the $\mathcal{D}$-covering oracle. The phase algorithm is a version of a single phase of the zooming algorithm (Algorithm 2.3) with very different rules for activating strategies. As in Algorithm 2.3, the confidence radius and the index are defined by (1) and (2), respectively. At the start of each round some strategies are activated, and then an active strategy with the maximal index is played.

Let us specify the activation rules. Denote $\mathcal{D} = \{S_i\}_{i=0}^{k}$. Initially the algorithm constructs 2^{-j}-nets $\mathcal{N}_j$, $j \in \mathbb{N}$, using the covering oracle. It finds the largest j such that $\mathcal{N} = \mathcal{N}_j$ contains at most $\frac{1}{2}\, T^{d/(d+2)}$ points, and activates all strategies in $\mathcal{N}$. The rest of the active strategies are partitioned into $k+1$ pools $P_i \subset S_i$ such that at each time t each pool P_i satisfies the following *quota Q_i*:

$$|\{u \in P_i : \, r_t(u) \geq \rho\}| \leq C_{\rho}\, \rho^{-d} \tag{8}$$

where $\rho = T^{-1/(d+2)}$ and $C_{\rho} = (64k \, \log \frac{1}{\rho})^{-1}$. In the beginning of each round the following activation routine is performed. If there exists a set S_i such that some strategy in S_i is not covered and *there is room under the corresponding quota* Q_i, pick one such strategy, activate it, and add it to the corresponding pool P_i. Since for a given strategy u the confidence radius $r_t(u)$ is non-increasing in t, the constraint (8) is never violated.

Repeat until there are no such sets S_i left. This completes the description of the algorithm. As was the case in Section 2, the analysis of the unbounded-time-horizon algorithm reduces to proving a lemma about the regret of each phase algorithm.

Lemma 3.10. *Fix a problem instance in the setting of Theorem 3.9. Let $\mathcal{A}_{\texttt{ph}}(T) = \mathcal{A}_{\texttt{ph}}(T, d, \mathcal{D})$. Then*

$$(\exists\, t_{min} < \infty)\ (\forall T \geq t_{min}) \quad R_{\mathcal{A}_{\texttt{ph}}(T)}(T) \leq T^{1-1/(d+2)}. \tag{9}$$

Note that the lemma bounds the regret of $\mathcal{A}_{\texttt{ph}}(T)$ for time T only. Proving Theorem 3.9 is now straightforward:

PROOF OF THEOREM 3.9. Let $\mathcal{A}_{\texttt{ph}}(T)$ be the phase algorithm from Lemma 3.10. Recall that in each phase i in the overall algorithm $\mathcal{A}$ we simply run a fresh instance of algorithm $\mathcal{A}_{\texttt{ph}}(2^i)$ for 2^i steps.

Let t_0 be the t_{min} from (9) rounded up to the nearest end-of-phase time. Let i_0 be the phase starting at time $t_0 + 1$. Note that $R_{\mathcal{A}}(t_0) \leq t_0$. Let R_i be the regret accumulated by $\mathcal{A}$ during phase i. Let $\gamma = \frac{d+1}{d+2}$. Then for any time $t \geq t_0^{1/\gamma}$ in phase i we have

$$R_{\mathcal{A}}(t) \leq t_0 + \sum_{j=i_0}^{i} R_j \leq t_0 + \sum_{j=i_0}^{i} (2^j)^{\gamma} \leq O(t^{\gamma}). \quad \square$$

In the remainder of this section we prove Lemma 3.10. Let us fix a problem instance of the Lipschitz MAB problem on a compact metric space (L, X) with a d^*-fat decomposition $\mathcal{D} = \{S_i\}_{i=0}^{k}$. Fix $d > d^*$ and let $\mathcal{A}_{\texttt{ph}}(T) = \mathcal{A}_{\texttt{ph}}(T, d, \mathcal{D})$ be the phase algorithm. Let μ be the expected reward function and let $\mu^* = \sup_{u \in X} \mu(u)$ be the optimal reward. Let $\Delta(u) = \mu^* - \mu(u)$.

Since L is a Lipschitz metric, it follows that μ is a continuous function on the metric space (L, X). Therefore the supremum μ^* is achieved by some strategy (call such strategies *optimal*). Say that a run of algorithm $\mathcal{A}_{\texttt{ph}}(T)$ is *well-covered* if at every time $t \leq T$ some optimal strategy is covered.

Say that a run of algorithm $\mathcal{A}_{\texttt{ph}}(T)$ is *clean* if the property in Claim 2.2 holds for all times $t \leq T$. Note that a given run is clean with probability at least $1 - T^{-2}$. The following lemma adapts the technique from Lemma 2.5 to the present setting:

Claim 3.11. *Consider a clean run of algorithm $\mathcal{A}_{\texttt{ph}}(T)$.*

(a) If strategies u, v are active at time $t \leq T$ then

$$\Delta(v) - \Delta(u) \leq 4 r_t(v).$$

(b) if the run is well-covered and strategy v is active at time $t \leq T$ then $\Delta(v) \leq 4 r_t(v)$.

The quotas (8) are chosen specifically to make the regret computation in Claim 2.7 work out for a clean and well-covered run of algorithm $\mathcal{A}_{\texttt{ph}}(T)$; we omit the details.

Claim 3.12. *$R_{\mathcal{A}}(T) \leq T^{1-1/(d+2)}$ for any clean well-covered run of algorithm $\mathcal{A} = \mathcal{A}_{\texttt{ph}}(T)$.*

PROOF SKETCH. Let $A_t(\delta)$ be the set of all strategies $u \in X$ such that u is active at time $t \leq T$ and $\delta \leq r_t(u) < 2\delta$. Note that for any such strategy we have $n_t(u) \leq O(\log T)\,\delta^{-2}$ and $\Delta(u) \leq 4r_t(u) < 8\delta$. Write

$$\begin{aligned} R^*(T) &:= \textstyle\sum_{u\in X} \Delta(u)\, n_T(u) \\ &\leq \rho T + \textstyle\sum_{i=0}^{\lceil \log 1/\rho \rceil} \sum_{u \in A_T(2^{-i})} \Delta(u)\, n_T(u), \end{aligned}$$

where $\rho = T^{-1/(d+2)}$ and apply the quotas (8). □

Let S_ℓ be the smallest set in $\mathcal{D}$ which contains some optimal strategy. For simplicity define $S_{k+1} = \emptyset$. Then there is an optimal strategy contained in $S_\ell \setminus S_{\ell+1}$; let u^* be one such strategy. The following claim essentially shows that the irrelevant high-dimensional subset $S_{\ell+1}$ is eventually pruned away.

Claim 3.13. *There exists an open set U containing $S_{\ell+1}$ such that $u^* \notin U$ and U is always covered throughout the first T steps of any clean run of algorithm $\mathcal{A}_{\texttt{ph}}(T)$, provided that T is sufficiently large.*

PROOF. Set $S_{\ell+1}$ is compact since it is a closed subset of a compact metric space. Since function μ is continuous, it assumes a maximum value on $S_{\ell+1}$. By construction, this maximum value is strictly less than μ^*. So there exists $\epsilon > 0$ such that $\Delta(w) > 8\epsilon$ for any $w \in S_{\ell+1}$. Define $U = B(S_{\ell+1}, \epsilon/2)$. Note that $u^* \notin U$ since $8\epsilon < \Delta(w) \leq L(u^*, w)$ for any $w \in S_{\ell+1}$.

Recall that in the beginning of algorithm $\mathcal{A}(T)$ all strategies in some 2^{-j}-net $\mathcal{N}$ are activated. Suppose T is large enough so that $2^{-j} \leq \epsilon$.

Consider a clean run of algorithm $\mathcal{A}_{\texttt{ph}}(T)$. We claim that U is covered at any given time $t \leq T$. Indeed, fix $u \in U$. By definition of U there exists $w \in S_{\ell+1}$ such that $L(u, w) < \epsilon/2$. By definition of $\mathcal{N}$ there exist $v, v^* \in \mathcal{N}$ such that $L(v, w) \leq \epsilon$ and $L(u^*, v^*) \leq \epsilon$.

(a) Note that $\Delta(v^*) = \mu(u^*) - \mu(v^*) \leq L(u^*, v^*) \leq \epsilon$.

(b) Since $L(v, w) \leq \epsilon$ and $\Delta(w) > 8\epsilon$, it follows that $\Delta(v) > 7\epsilon$.

(c) By Claim 3.11 we have $\Delta(v) - \Delta(v^*) \leq 4r_t(v^*)$.

Combining (a-c), it follows that $r_t(v) \geq \frac{3}{2}\,\epsilon \geq L(u, v)$, so v covers u. Claim proved. □

PROOF OF LEMMA 3.10. By Claim 3.12 it suffices to show that if T is sufficiently large then any clean run of algorithm $\mathcal{A}_{\texttt{ph}}(T)$ is well-covered. (Runs that are not clean contribute only $O(1/T)$ to the expected regret of $\mathcal{A}_{\texttt{ph}}(T)$, because the probability that a run is not clean is at most T^{-2} and the regret of such a run is at most T.) Specifically, we will show that u^* is covered at any time $t \leq T$ during a clean run of $\mathcal{A}_{\texttt{ph}}(T)$. It suffices to show that at any time $t \leq T$ there is room under the corresponding quota Q_ℓ in (8).

Let U be the open set from Claim 3.13. Since U is an open neighborhood of $S_{\ell+1}$, by definition of the fat decomposition it follows that $\texttt{COV}(S_\ell \setminus U) \leq d^*$. Define ρ and C_ρ as in (8) and fix $d' \in (d^*, d)$. Then for any sufficiently large T it is the case that (i) $S_\ell \setminus U$ can be covered with $(\frac{1}{\rho})^{d'}$ sets of diameter $< \rho$ and moreover (ii) that $(\frac{1}{\rho})^{d'} \leq \frac{1}{2}\, C_\rho\, \rho^{-d}$.

Fix time $t \leq T$ and let A_t be the set of all strategies u such that u is in the pool P_ℓ at time t and $r_t(u) \geq \rho$. Note that $A_t \subset S_\ell \setminus U$ since U is always covered, and by the specification of $\mathcal{A}_{\texttt{ph}}$ only active uncovered strategies in S_ℓ are added to pool P_ℓ. Moreover, A_t is ρ-separated. (Indeed, let $u, v \in A_t$ and assume u has been activated before v. Then $L(u, v) > r_s(u) \geq r_t(u) \geq \rho$, where s is the time when v was activated.) It follows that $|A_t| \leq \frac{1}{2}\, C_\rho\, \rho^{-d}$, so there is room under the corresponding quota Q_ℓ in (8). □

3.2 The per-metric optimal algorithm

To extend the ideas of the preceding section to arbitrary metric spaces, we must extend Definition 3.7 to *transfinitely infinite* fat decompositions.

Definition 3.14. Fix a metric space (L, X). Let β denote an arbitrary ordinal. A *transfinite d-fat decomposition* of depth β is a transfinite sequence $\{S_\lambda\}_{0 \leq \lambda \leq \beta}$ of closed subsets of X such that:

(a) $S_0 = X$, $S_\beta = \emptyset$, and $S_\nu \supseteq S_\lambda$ whenever $\nu < \lambda$.

(b) if $V \subset X$ is closed, then the set of ordinals $\nu \leq \beta$ such that V intersects S_ν has a maximum element.

(c) for any ordinal $\lambda \leq \beta$ and any open set $U \subset X$ containing $S_{\lambda+1}$ we have $\texttt{COV}(S_\lambda \setminus U) \leq d$.

Note that for a finite depth β the above definition is equivalent to Definition 3.7. In Theorem 3.16 below, we will show how to modify the "quota algorithms" from the previous section to achieve regret dimension d in any metric with a transfinite d^*-fat decomposition for $d^* < d$. This gives an optimal algorithm for every metric space X because of the following surprising relation between the max-min-covering dimension and transfinite fat decompositions.

Proposition 3.15. *For every compact metric space (L, X), the max-min-covering dimension of X is equal to the infimum of all d such that X has a transfinite d-fat decomposition.*

PROOF. If $\emptyset \neq Y \subseteq X$ and $\texttt{MinCOV}(Y) > d$ then, by transfinite induction, $Y \subseteq S_\lambda$ for all λ in any transfinite d-fat decomposition, contradicting the fact that $S_\beta = \emptyset$. Thus, the existence of a transfinite d-fat decomposition of X implies $d \geq \texttt{MaxMinCOV}(X)$. To complete the proof we will construct, given any $d > \texttt{MaxMinCOV}(X)$, a transfinite d-fat decomposition of depth β, where β is any ordinal whose cardinality exceeds that of X. For a metric space Y, define the set of *d-thin points* $\mathrm{TP}(Y, d)$ to be the union of all open sets $U \subseteq Y$ satisfying $\texttt{COV}(U) < d$. Its complement, the set of *d-fat points*, is denoted by $\mathrm{FP}(Y, d)$. Note that it is a closed subset of Y.

For an ordinal $\lambda \leq \beta$, we define a set S_λ using transfinite induction as follows:

1. $S_0 = X$ and $S_{\lambda+1} = \mathrm{FP}(S_\lambda, d)$ for each ordinal λ.
2. If λ is a limit ordinal then $S_\lambda = \bigcap_{\nu<\lambda} S_\nu$.

Note that each S_λ is closed, by transfinite induction. It remains to show that $\mathcal{D} = \{S_\lambda\}_{\lambda \in \mathcal{O}}$ satisfies the properties (a-c) in Definition 3.14. It follows immediately from the construction that $S_0 = X$ and $S_\nu \supseteq S_\lambda$ when $\nu < \lambda$. To prove that $S_\beta = \emptyset$, observe first that the sets $S_\lambda \setminus S_{\lambda+1}$ (for $0 \leq \lambda < \beta$) are disjoint subsets of X, and the number of such sets is greater than the cardinality of X, so at least one of them is empty. This means that $S_\lambda = S_{\lambda+1}$ for some $\lambda < \beta$. If $S_\lambda = \emptyset$ then $S_\beta = \emptyset$ as desired. Otherwise, the relation $\mathrm{FP}(S_\lambda, d) = S_\lambda$ implies that $\texttt{MinCOV}(S_\lambda) \geq d$ contradicting the assumption that $\texttt{MaxMinCOV}(X) < d$. This completes the proof of property (a). To prove property (b), suppose $\{\nu_i \mid i \in \mathcal{I}\}$ is a set of ordinals such that S_{ν_i} intersects V for every i. Let $\nu = \sup\{\nu_i\}$.

Then $S_\nu \cap V = \bigcap_{i\in\mathcal{I}}(S_{\nu_i} \cap V)$, and the latter set is nonempty because X is compact and the closed sets $\{S_{\nu_i} \cap V \,|\, i \in \mathcal{I}\}$ have the finite intersection property. Finally, to prove property (c), note that if U is an open neighborhood of $S_{\lambda+1}$ then the set $T = S_\lambda \setminus U$ is closed (hence compact) and is contained in $\mathrm{TP}(S_\lambda, d)$. Consequently T can be covered by open sets V satisfying $\mathtt{COV}(V) < d$. By compactness of T, this covering has a finite subcover $V_1, \dots, V_m$, and consequently $\mathtt{COV}(T) = \max_{1\le i\le m} \mathtt{COV}(V_i) < d$. □

Theorem 3.16. *Consider the Lipschitz MAB problem on a compact metric space (L, X). For any $d > \mathtt{MaxMinCOV}(X)$ there exists an algorithm $\mathcal{A}_d$ such that $\mathtt{DIM}(\mathcal{A}_d) \le d$.*

Note that Theorem 1.2 follows immediately by combining Theorem 3.16 with Theorem 3.4.

We next describe an algorithm $\mathcal{A}_d$ satisfying Theorem 3.16. The algorithm requires two oracles: a depth oracle $\mathtt{Depth}(\cdot)$ and a $\mathcal{D}$-covering oracle $\mathcal{D}\text{-}\mathtt{Cov}(\cdot)$. For any finite set of open balls $B_0, B_1, \dots, B_n$ (given via the centers and the radii) whose union is denoted by B, $\mathtt{Depth}(B_0, B_1, \dots, B_n)$ returns the maximum ordinal λ such that S_λ intersects the closure $\overline{B}$; such an ordinal exists by Definition 3.14(b).[8] Given a finite set of open balls $B_0, B_1, \dots, B_n$ with union B as above, and an ordinal λ, $\mathcal{D}\text{-}\mathtt{Cov}(\lambda, B_0, B_1, \dots, B_n)$ either reports that B covers S_λ, or it returns a strategy $x \in S_\lambda \setminus B$.

Our algorithm proceeds in phases $i = 1, 2, 3, \dots$ of 2^i rounds each. In any given phase i, there is a "target ordinal" $\lambda(i)$ (defined at the end of the preceding phase), and we run an algorithm during the phase which: (i) activates some nodes initially; (ii) plays a version of the zooming algorithm which only activates strategies in $S_{\lambda(i)}$; (iii) concludes the phase by computing $\lambda(i+1)$. The details are as follows. In a given phase we run a fresh instance of a phase algorithm $\mathcal{A}_{\mathtt{ph}}(T, d, \lambda)$ where $T = 2^i$ and $\lambda = \lambda(i)$ is a *target ordinal* for phase i, defined below when we give the full description of $\mathcal{A}_{\mathtt{ph}}(T, d, \lambda)$. The goal of $\mathcal{A}_{\mathtt{ph}}(T, d, \lambda)$ is to satisfy the per-phase bound

$$R_{\mathcal{A}_{\mathtt{ph}}(T,d,\lambda)}(T) = \widetilde{O}(T^\gamma) \tag{10}$$

for all $T > T_0$, where $\gamma = 1 - 1/(d+2)$ and T_0 is a number which may depend on the instance μ. Then, to derive the bound $R_{\mathcal{A}_d}(t) = \widetilde{O}(t^\gamma)$ for all t we simply sum per-phase bounds over all phases ending before time $2t$.

Initially $\mathcal{A}_{\mathtt{ph}}(T, d, \lambda)$ uses the covering oracle to construct 2^{-j}-nets $\mathcal{N}_j$, $j = 0, 1, 2, \dots$, until it finds the largest j such that $\mathcal{N} = \mathcal{N}_j$ contains at most $\frac{1}{2}\ T^{d/(d+2)} \log(T)$ points. It activates all strategies in $\mathcal{N}$ and sets

$$\varepsilon(i) = \max\{2^{-j}, 32\, T^{-1/(d+2)} \log(T)\}.$$

After this initialization step, for every active strategy v we define the confidence radius

$$r_t(v) := \max\left\{ T^{-1/(d+2)}, \sqrt{\frac{8 \log T}{2 + n_t(v)}} \right\},$$

where $n_t(v)$ is the number of times v has been played by the phase algorithm $\mathcal{A}_{\mathtt{ph}}(T, d, \lambda)$ before time t. Let $B_0, B_1, \dots, B_n$ be an enumeration of the the open balls belonging to the set $\{B(v, r_t(v)) \,|\, v \text{ active at time } t\}$. If $n < \frac{1}{2}\ T^{d/(d+2)} \log(T)$ then we perform the oracle call $\mathcal{D}\text{-}\mathtt{Cov}(\lambda, B_0, \dots, B_n)$, and if it reports that a point $x \in S_\lambda$ is uncovered, we activate x and set $n_t(x) = 0$. The index of an active strategy v is defined as $\mu_t(v) + 4r_t(v)$ — note the slight difference from the index defined in Algorithm 2.3 — and we always play the active strategy with maximum index. To complete the description of the algorithm, it remains to explain how the ordinals $\lambda(i)$ are defined. The definition is recursive, beginning with $\lambda(1) = 0$. At the end of phase i ($i \ge 1$), we let $B_0, B_1, \dots, B_m$ be an enumeration of the open balls in the set $\{B(v, \varepsilon(i)) \,|\, v \text{ active}, r_T(v) < \varepsilon(i)/2\}$. Finally, we set $\lambda(i+1) = \mathtt{Depth}(B_0, B_1, \dots, B_m)$.

PROOF OF THEOREM 3.16. Since we have modified the definition of index, we must prove a variant of Claim 3.11 which asserts that in a clean run of $\mathcal{A}_{\mathtt{ph}}$, if u, v are active at time t then $\Delta(v) - \Delta(u) \le 5r_t(v)$. To prove it, let s be the latest round in $\{1, 2, \dots, t\}$ when v was played. We have $r_t(v) = r_s(v)$, and $\Delta(v) - \Delta(u) = \mu(u) - \mu(v)$, so it remains to prove that

$$\mu(u) - \mu(v) \le 5r_s(v). \tag{11}$$

From the fact that v was played instead of u at time s, together with the fact that both strategies are clean, we have

$$\mu_s(u) + 4r_s(u) \le \mu_s(v) + 4r_s(v) \tag{12}$$
$$\mu(u) - \mu_s(u) \le r_s(u) \tag{13}$$
$$\mu_s(v) - \mu(v) \le r_s(v). \tag{14}$$

We obtain (11) by adding (12)-(14), noting that $r_s(u) > 0$.

Let λ be the maximum ordinal such that S_λ contains an optimal strategy u^*; such an ordinal exists by Definition 3.14(b). We will prove that for sufficiently large i, if the i-th phase is clean, then $\lambda(i) = \lambda$. The set $S_{\lambda+1}$ is compact, and the function μ is continuous, so it assumes a maximum value on $S_{\lambda+1}$ which is, by construction, strictly less than μ^*. Choose $\varepsilon > 0$ such that $\Delta(w) > 5\varepsilon$ for all $w \in S_{\lambda+1}$, and choose $T_0 = 2^{i_0}$ such that $\varepsilon(i_0) \le \varepsilon$. We shall prove that for all $T = 2^i \ge T_0$ and all ordinals ν, a clean run of $\mathcal{A}_{\mathtt{ph}}(T, d, \nu)$ results in setting $\lambda(i+1) = \lambda$. First, let $v^* \in \mathcal{N}$ be such that $L(u^*, v^*) \le \varepsilon(i)$. If v is active and $r_T(v) < \varepsilon(i)/2$ then Claim 3.11 implies that $\Delta(v) - \Delta(v^*) \le \frac{5}{2}\,\varepsilon(i)$ hence $\Delta(v) \le \frac{7}{2}\,\varepsilon(i)$. As $\Delta(w) > 5\varepsilon \ge 5\varepsilon(i)$ for all $w \in S_{\lambda+1}$, it follows that the closure of $B(v, \varepsilon(i))$ does not intersect $S_{\lambda+1}$. This guarantees that $\mathtt{Depth}(B_0, B_1, \dots, B_m)$ returns an ordinal less than or equal to λ. Next we must prove that this ordinal is greater than or equal to λ. Note that the total number of strategies activated by $\mathcal{A}_{\mathtt{ph}}(T, d, \nu)$ is bounded above by $T^{d/(d+2)} \log(T)$. Let A_T denote the set of strategies active at time T and let

$$v^0 = \arg\max_{v \in A_T} n_T(v).$$

By the pigeonhole principle, $n_T(v^0) \ge T^{2/(d+2)}/\log(T)$ and hence $r_T(v^0) < 3T^{-1/(d+2)} \log(T)$. If t denotes the last time at which v^0 was played, then we have

$$\begin{aligned} I_t(v^0) = \mu_t(v^0) + 4r_t(v^0) &\le \mu^* + 5r_t(v^0) \\ &\le \mu^* + 15T^{-1/(d+2)} \log(T) < \mu^* + \varepsilon(i)/2, \end{aligned}$$

provided that the phase is clean and that $T \ge T_0$. Since v^0 had maximum index at time t, we deduce that $I_t(v^*) < \mu^* + \varepsilon(i)/2$ as well. As $L(u^*, v^*) \le \varepsilon(i)$ we have $\mu_t(v^*) \ge \mu^* - \varepsilon(i) - r_t(v^*)$ provided the phase is clean. To finish the proof we observe that

$$\mu^* + \varepsilon(i)/2 > I_t(v^*) \ge \mu^* - \varepsilon(i) + 3r_t(v^*)$$

[8] To avoid the question of how arbitrary ordinals are represented on the oracle's output tape, we can instead say that the oracle outputs a point $u \in S_\lambda$ instead of outputting λ. In this case, the definition of $\mathcal{D}\text{-}\mathtt{Cov}$ should be modified so that its first argument is a point of S_λ rather than λ itself.

which implies $r_t(v^*) < \varepsilon(i)/2$. Since the confidence radius does not increase over time, we have $r_T(v^*) < \varepsilon(i)/2$ so $B(v^*, \varepsilon(i))$ is one of the balls $B_0, B_1, \ldots, B_m$. Since u^* is contained in the closure of this ball, we may conclude that $\texttt{Depth}(B_0, B_1, \ldots, B_m)$ returns the ordinal λ as desired.

Let $U = B(S_{\lambda+1}, \varepsilon(i)/2)$. As in Claim 3.13 it holds that in any clean phase, U is covered throughout the phase by balls centered at points of $\mathcal{N}$. Hence for any pair of consecutive clean phases, in the second phase of the pair our algorithm only calls the covering oracle $\mathcal{D}\text{-}\texttt{Cov}$ with the proper ordinal λ (i.e. the maximum λ such that S_λ contains an optimal strategy) and with a set of balls $B_0, B_1, \ldots, B_n$ that covers U. Also, note that an active strategy v during a run of $\mathcal{A}_{\texttt{ph}}(T, d, \lambda)$ never has a confidence radius $r_t(v)$ less than $\delta = T^{-1/(d+2)}$, so the strategies activated by the covering oracle form a δ-net in the space $S_\lambda \setminus U$. By Definition 3.14(c), a δ-net in $S_\lambda \setminus U$ contains fewer than $O(\delta^{-d})$ points. Hence for sufficiently large T the "quota" of $\frac{1}{2}\, T^{d/(d+2)}$ active strategies is never reached, which implies that every point of S_λ — including u^* — is covered throughout the phase. The upper bound on the regret of $\mathcal{A}_{\texttt{ph}}(T, d, \lambda)$ concludes as in the proof of Theorem 2.4. □

4. EXTENSIONS

Let us briefly discuss several extensions of this work that have been omitted from this extended abstract due to lack of space. The precise theorem statements and proofs will appear in the full version of this paper.

We observe that the proof in Section 2 works under a more abstract definition of confidence radius of strategy u at time t: esentially, it can be any function of t and the history of playing u such that Claim 2.2 holds. The allows our results to be extended in the following three directions. First, we may upgrade the zooming algorithm to satisfy the guarantee in Theorem 2.4 **and** to enjoy the improved guarantee $R_{\mathcal{A}}(t) < \tilde{O}(t^{d/(d+1)})$, if the maximal reward is exactly 1. The key ingredient here is a refined version of the confidence radius which gets much sharper when the sample average is close to 1. Second, we may consider the setting when the reward from playing a given strategy u is the corresponding expected reward $\mu(u)$ plus an independent random sample from a fixed distibution $\mathcal{P}$ known to the algorithm. We obtain improved bounds on regret if $\mathcal{P}$ has a "special region" that can be identified using a small number of samples. For instance, if $\mathcal{P}$ has at least one point mass, the regret is at most $\tilde{O}(t^{1-1/d})$. Third, we may extend our analysis from reward distributions supported on $[0;1]$ to those on unbounded support, assuming a finite absolute third moment. This extension relies on the *Berry-Esseen theorem* [18].

Our techniques also lead to improved bounds for a special case of the standard Lipschitz MAB problem problem where the expected reward function has the appealing *gradient ascent* structure: $\mu(\cdot) = 1 - f(L(\,\cdot\,, S))$, where f is a known non-decreasing function and S is the target subset which is not revealed to the algorithm. The objective is, essentially, to zoom in on S as quickly as possible. For a wide class of *growth-constrained* functions f which includes polynomials, we obtain guarantees in terms of $\texttt{COV}(S)$, the intuition being that $\texttt{COV}(S) \ll \texttt{COV}(X)$. In particular, we obtain poly-logarithmic regret when $f(0) = \texttt{COV}(S) = 0$. We run a version of the zooming algorithm that instead of the original metric space (L, X) uses a modified space $(f(L), X)$. The analysis is based on that in Section 2; the switch from (L, X) to $(f(L), X)$ forces us to revisit the analysis and seek the very minimal assumptions which enable it.

5. REFERENCES

[1] R. Agrawal. The continuum-armed bandit problem. *SIAM J. Control and Optimization*, 33(6):1926–1951, 1995.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002.

[4] P. Auer, R. Ortner, and C. Szepesvári. Improved Rates for the Stochastic Continuum-Armed Bandit Problem. In *20th Conference on Learning Theory (COLT)*, pages 454–468, 2007.

[5] B. Awerbuch and R. Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, February 2008.

[6] J. S. Banks and R. K. Sundaram. Denumerable-armed bandits. *Econometrica*, 60(5):1071–1096, 1992.

[7] D. A. Berry and B. Fristedt. *Bandit problems: sequential allocation of experiments*. Chapman and Hall, 1985.

[8] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[9] E. Cope. Regret and convergence bounds for immediate-reward reinforcement learning with continuous action spaces, 2004. Unpublished manuscript.

[10] V. Dani, T. Hayes, and S. M. Kakade. The Price of Bandit Information for Online Optimization. Preprint, 2007.

[11] V. Dani and T. P. Hayes. Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 937–943, 2006.

[12] A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 385–394, 2005.

[13] J. C. Gittins and D. M. Jones. A dynamic allocation index for the sequential design of experiments. In J. G. et al., editor, *Progress in Statistics*, pages 241–266. North-Holland, 1974.

[14] S. M. Kakade, A. T. Kalai, and K. Ligett. Playing Games with Approximation Algorithms. In *39th ACM Symp. on Theory of Computing (STOC)*, 2007.

[15] R. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *18th Advances in Neural Information Processing Systems (NIPS)*, 2004. Full version appeared as Chapters 4-5 in [16].

[16] R. Kleinberg. *Online Decision Problems with Large Strategy Sets*. PhD thesis, MIT, Boston, MA, 2005.

[17] H. B. McMahan and A. Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *17th Annual Conference on Learning Theory (COLT)*, volume 3120 of *LNCS*, pages 109–123. Springer Verlag, 2004.

[18] K. Neammanee. On the constant in the nonuniform version of the Berry-Esseen theorem. *Intl. J. of Mathematics and Mathematical Sciences*, 2005:12:1951–1967, 2005.

[19] S. Pandey, D. Agarwal, D. Chakrabarti, and V. Josifovski. Bandits for Taxonomies: A Model-based Approach. In *SIAM Intl. Conf. on Data Mining (SDM)*, 2007.