

# Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary.

Varsha Dani\*      Thomas P. Hayes †

November 12, 2005

## Abstract

We consider “online bandit geometric optimization,” a problem of iterated decision making in a largely unknown and constantly changing environment. The goal is to minimize “regret,” defined as the difference between the actual loss of an online decision-making procedure and that of the best single decision *in hindsight*. “Geometric optimization” refers to a generalization of the well-known multi-armed bandit problem, in which the decision space is some bounded subset of  $\mathbb{R}^d$ , the adversary is restricted to linear loss functions, and regret bounds should depend on the dimensionality  $d$ , rather than the total number of possible decisions. “Bandit” refers to the setting in which the algorithm is only told its loss on each round, rather than the entire loss function.

McMahan and Blum [10] presented the best known algorithm in this setting, and proved that its expected additive regret is  $O(\text{poly}(d)T^{3/4})$ . We simplify and improve their analysis of this algorithm to obtain regret  $O(\text{poly}(d)T^{2/3})$ .

We also prove that, for a large class of full-information online optimization problems, the optimal regret against an adaptive adversary is the same as against a non-adaptive adversary.

## 1 Introduction

Every morning, Alice drives to work, choosing one of several alternative routes, and carefully recording the time taken. At the end of the year, she reads a newspaper report giving the average transit time along each possible route. Under the somewhat pessimistic assumption that the “traffic gods” are “out to get her”, how can Alice minimize her “regret”, defined as the difference between her average transit time, and the best average time from the report? (Note that, under this definition, Alice’s regret will not change if all the transit times increase by an hour! In this sense, “regret” measures Alice’s later dissatisfaction with her choices, rather than their outcomes.)

We study a natural generalization of this problem, known as “online geometric optimization,” or “online linear optimization.” Here, in each of  $T$  rounds, Alice chooses a vector  $\mathbf{x}^t$

---

\*University of Chicago, Department of Computer Science, email: [varsha@cs.uchicago.edu](mailto:varsha@cs.uchicago.edu)

†University of California at Berkeley, Division of Computer Science, email: [hayest@cs.berkeley.edu](mailto:hayest@cs.berkeley.edu). Supported by an NSF Postdoctoral Fellowship and by NSF grant CCR-0121555.

from a fixed *decision set*,  $S \subset \mathbb{R}^d$ , while an adversary simultaneously chooses a vector  $\mathbf{c}^t$  from a fixed *cost set*,  $K \subset \mathbb{R}^d$ . Alice’s *loss* for round  $t$  is the dot product  $\mathbf{x}^t \cdot \mathbf{c}^t$ . Alice’s goal is to minimize regret, defined as

$$\sum_{t=1}^T \mathbf{x}^t \cdot \mathbf{c}^t - \min_{\mathbf{x}^* \in S} \sum_{t=1}^T \mathbf{x}^* \cdot \mathbf{c}^t.$$

More specifically, we will focus on the “bandit” version of this problem, in which at the end of round  $t$ , Alice is only told her loss,  $\mathbf{c}^t \cdot \mathbf{x}^t$ . This is worse for Alice than the “full information” version, in which she is told the entire cost vector  $\mathbf{c}^t$  after round  $t$ . In either case, we assume the adversary is told  $\mathbf{x}^t$  after round  $t$ . Since we seek upper bounds on regret which hold against every adversary, we may assume the adversary’s goal is the reverse of Alice’s; consequently we may view the problem as a two-player zero-sum game, in which the payoff to the adversary equals the regret.

## 1.1 Previous work.

This and several closely related online optimization problems have been considered in the literature, dating at least to the 1950’s (for instance, [11, 6]). We briefly touch on some important connections to and distinctions from some of this earlier work. We emphasize that, in our setting, cost vectors are always chosen adversarially; we will not discuss a large part of the literature, which considers inputs chosen according to a distribution (known or unknown).

One of the first formulations of online optimization was the “multi-armed bandit” problem, in which, in each round, Alice must choose one of  $d$  slot machines to play. Assuming the slot machine payouts are bounded, and are chosen adaptively and adversarially in each round, this corresponds to the special case of online geometric optimization in which the decision set  $S$  is a collection of  $d$  linearly independent vectors. In the full-information setting, the optimal regret bound is  $\Theta(\sqrt{T \log d})$ , as shown by Freund and Schapire [5]. Their algorithm is a slight variant of the “weighted majority” algorithm of Littlestone and Warmuth [9]. This algorithm can also be applied in the geometric optimization context, but unfortunately yields a regret bound of  $O(\sqrt{T \log |S|})$ , where  $|S|$  may be arbitrarily large compared to  $d$ .

The first efficient algorithm for online geometric optimization is due to Hannan [6], and was subsequently rediscovered and clarified by Kalai and Vempala [8]. It achieves a regret bound of  $O(\text{poly}(d)\sqrt{T})$ , which is essentially best possible.

We note here that both of the above results were originally proved under the assumption that the adversary is non-adaptive, but rather chooses the sequence of cost vectors  $\mathbf{c}^1, \dots, \mathbf{c}^T$  in advance. As we shall prove in Section 3, for a broad class of two-player zero-sum games, including the ones above, the value of the game is determined by non-adaptive adversaries, so the extension of these bounds to adaptive adversaries is automatic. In the bandit setting, this is not the case.

$d$ -armed bandit problem		
Adversary:	Non-adaptive	Adaptive
Full info	$\sqrt{T}$ [9, 5]	$\sqrt{T}$
Bandit	$\sqrt{T}$ [1, 2]	$T^{2/3}$ [1] $\sqrt{T \log T}$ [4]

online geometric optimization problem		
Adversary:	Non-adaptive	Adaptive
Full info	$\sqrt{T}$ [6, 8]	$\sqrt{T}$
Bandit	$T^{2/3}$ [3]	$T^{3/4}$ [10] now $T^{2/3}$ .

Table 1: Best known regret bounds for online optimization, with citations. The polynomial dependencies on  $d$  and  $M$  are omitted. The present work improves the  $O(T^{3/4})$  bound for bandit geometric optimization against an adaptive adversary to  $O(T^{2/3})$ . The best known lower bound is  $\Omega(\sqrt{T})$  for all versions.

In the bandit setting, Auer *et al.* [1, 2] presented a modified version of the weighted majority algorithm, for which the expected regret is  $O(\text{poly}(d)\sqrt{T})$  against a non-adaptive adversary, and  $O(\text{poly}(d)T^{2/3})$  in general. The basic principle of this algorithm is “explore and exploit.” (We will discuss this idea in more detail in the next section.) Auer *et al.* also proved that their  $O(\sqrt{T})$  upper bound holds for an adaptive adversary, but under a weaker version of regret: namely, the difference between Alice’s loss and the loss of a single “champion” decision vector, secretly chosen by the adversary in round 0, and which need not be the optimal decision vector in hindsight. Very recently, Dani and Hayes [4] improved this algorithm, using a dynamic tradeoff between exploring and exploiting, achieving expected regret  $O(\sqrt{T \log T})$ .

Awerbuch and Kleinberg [3] presented a rather similar “explore and exploit” algorithm for the bandit version of online geometric optimization, which uses any algorithm for the full-information version of the game as a black box. In particular, with Kalai and Vempala’s algorithm as the black box, they proved  $O(\text{poly}(d)T^{2/3})$  regret against any non-adaptive adversary. They also presented a related algorithm, specific to the “drive to work” problem, which achieves the same upper bound against an adaptive adversary.

McMahan and Blum [10] presented another rather similar “explore and exploit” algorithm, also using a decision-maker for the full-information game as a black box, and proved it has  $O(\text{poly}(d)T^{3/4})$  regret against an adaptive adversary.

## 1.2 Main results.

Assume that the set of attainable costs,  $S \cdot K = \{\mathbf{x} \cdot \mathbf{c} \mid \mathbf{x} \in S, \mathbf{c} \in K\}$ , has diameter  $M$ .

**Theorem 1.1.** *Against any adaptive adversary, the McMahan-Blum algorithm, using Kalai and Vempala’s algorithm as a black box, has expected regret at most  $15dMT^{2/3}$ .*

The upper bound in Theorem 1.1 has the same dependence on  $T$  as the best known bounds

on regret in the special case of non-adaptive adversaries (see Awerbuch and Kleinberg [3]). (Improving this to  $O(\sqrt{T})$ , at least for non-adaptive adversaries, seems a tantalizing open question.)

We would also like to call attention to our Theorem 3.1, which we will state and prove in Section 3. Although its primary purpose here is as a tool for proving Theorem 1.1, it may be useful in other contexts. For instance, it immediately implies that, in the full-information versions of the  $d$ -armed bandit and online geometric optimization problems (or, for instance, the convex optimization problem of Zinkevich [13]) there is no difference between the power of adaptive and non-adaptive adversaries against an optimal randomized algorithm.

### 1.3 Organization of the paper.

In the next section we formulate the problem formally, describe the algorithm, and reduce the main theorem to proving a sequence of inequalities (numbered (1), (2), and (3) at the end of Section 2). Inequality (1) was proved by McMahan and Blum [10, Theorem 3]; for completeness, we include the proof in the Appendix. In section 3, we prove that, for a broad class of problems, adaptive adversaries are no more powerful than oblivious adversaries. In section 4, we apply this result to prove inequality (2). In section 5, a second moment argument is used to prove inequality (3). In section 6 we show our analysis is tight, by giving an  $\Omega(T^{2/3})$  lower bound for a class of “explore and exploit” algorithms.

## 2 Problem Formulation

Our basic setting is a two-player zero-sum game, played by an algorithm,  $\mathcal{A}$ , against an adversary,  $\mathcal{V}$ . The game takes place in a sequence of  $T$  rounds.<sup>1</sup> In round  $i$ , the algorithm selects a decision,  $\mathbf{x}^i \in S \subset \mathbb{R}^d$ , and, simultaneously, the adversary sets a cost vector  $\mathbf{c}^i \in K \subset \mathbb{R}^d$ . The adversary is told  $\mathbf{x}^i$ . In the full-information version of the game, the algorithm is told  $\mathbf{c}^i$ , whereas in the bandit version of the game, the algorithm is only told the dot product  $\mathbf{x}^i \cdot \mathbf{c}^i$ . The payoff to the adversary (*i. e.*, the loss to the algorithm) is  $\sum_{i=1}^T \mathbf{x}^i \cdot \mathbf{c}^i$ .

We assume the decision space,  $S$ , and the cost space,  $K$ , are fixed compact subsets of  $\mathbb{R}^d$ . Suppose  $S \cdot K = \{x \cdot c : x \in S, c \in K\} \subset [0, M]$ ; this set is the set of possible incurred costs in one round of the game.

We assume that  $S$  is of full rank, and moreover that  $S$  contains the standard basis  $\mathbf{e}_1, \dots, \mathbf{e}_d$ . In our analysis, we will further assume that  $S \subseteq [-2, 2]^d$  (or, put another way, that  $\mathbf{e}_1, \dots, \mathbf{e}_d$  is a “nearly barycentric” spanner for  $S$ ).

The simplifying assumptions of the previous paragraph are made without loss of generality. That a nearly barycentric spanner exists and can be found efficiently is proven in [3]. Once found, we can re-coordinatize to make this spanner the standard basis for  $\mathbb{R}^d$ , without

---

<sup>1</sup>Following precedent, we will assume that  $T$  is known to the algorithm in advance. If not, the standard “doubling trick” can be used, in which the algorithm maintains a putative value of  $T$ , doubling it and restarting whenever the number of rounds exceeds  $T$ . This only affects the regret bound by a constant factor.

changing the regret. (We can view the set  $S$  as a fixed subset of an abstract real vector space  $V$ , and the adversary as choosing a linear functional  $f^t: V \rightarrow \mathbb{R}$  in each round, subject to the constraint that  $f^t(S) \subset [0, M]$ . From this perspective, there is no natural coordinatization of  $V$ , so we may take any we like.)

In the box below, we present the algorithm of McMahan and Blum [10], which we shall denote MB. For concreteness, we will assume throughout that the “follow the perturbed leader” algorithm of Kalai and Vempala, denoted by KV, is used as a subroutine.

There are three parameters: the number of rounds,  $T$ , the exploration rate  $\gamma$ , and a sensitivity parameter  $\varepsilon$ , whose inverse is the maximum random noise added by the Kalai-Vempala algorithm, which we shall denote  $\text{KV}_\varepsilon$ . At each round, the algorithm flips a biased coin to decide whether to “explore” or “exploit.” With probability  $\gamma$ , it explores, in which case its decision  $\mathbf{x}^t$  is a randomly chosen basis element  $\mathbf{e}_j$ . Otherwise, it exploits, in which case it plays according to the advice of  $\text{KV}_\varepsilon$ , using “estimated cost history”  $(\hat{\mathbf{c}}^1, \dots, \hat{\mathbf{c}}^{t-1})$

Note that  $\hat{\mathbf{c}}^t$  is an unbiased estimator for  $\mathbf{c}^t$ , in the following strong sense:

$$\mathbf{E}(\hat{\mathbf{c}}^t \mid \mathbf{x}^1, \dots, \mathbf{x}^{t-1}, \mathbf{c}^1, \dots, \mathbf{c}^{t-1}) = \mathbf{c}^t.$$

**Algorithm 2.1:** MB( $T, \gamma, \varepsilon$ )

**for**  $t := 1$  **to**  $T$

$$\chi^t := \begin{cases} 1 & \text{with probability } \gamma \\ 0 & \text{otherwise} \end{cases}$$

Sample  $j$  uniformly randomly from  $\{1, 2, \dots, d\}$

$$\hat{\mathbf{x}}^t := \text{KV}_\varepsilon(\hat{\mathbf{c}}^1, \dots, \hat{\mathbf{c}}^{t-1})$$

$$\mathbf{x}^t := \begin{cases} \mathbf{e}_j & \text{if } \chi^t = 1 \\ \hat{\mathbf{x}}^t & \text{if } \chi^t = 0 \end{cases}$$

Observe and pay  $\ell^t := \mathbf{c}^t \cdot \mathbf{x}^t$

$$\hat{\mathbf{c}}^t := \begin{cases} \frac{d}{\gamma} \ell^t \mathbf{e}_j & \text{if } \chi^t = 1 \\ \mathbf{0} & \text{if } \chi^t = 0 \end{cases}$$

We will use the following notions of loss and additive regret.

**Definition 2.1.** Let  $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^T)$  be a sequence of decisions and  $\mathbf{c} = (\mathbf{c}^1, \dots, \mathbf{c}^T)$  a

sequence of cost vectors (both elements of  $(\mathbb{R}^d)^T$ ). We denote

$$\begin{aligned}\text{loss}(\mathbf{x}, \mathbf{c}) &= \sum_{t=1}^T \mathbf{x}^t \cdot \mathbf{c}^t \\ \text{best}(\mathbf{c}) &= \arg \max_{\mathbf{x}^* \in S} \sum_{t=1}^T \mathbf{x}^* \cdot \mathbf{c}^t \\ \text{opt}(\mathbf{c}) &= \max_{\mathbf{x}^* \in S} \sum_{t=1}^T \mathbf{x}^* \cdot \mathbf{c}^t = \text{best}(\mathbf{c}) \cdot \sum_{t=1}^T \mathbf{c}^t.\end{aligned}$$

The *regret* is defined as

$$\text{regret}(\mathbf{x}, \mathbf{c}) = \text{loss}(\mathbf{x}, \mathbf{c}) - \text{opt}(\mathbf{c}).$$

Fix an adaptive adversary,  $\mathcal{V}$ , and play the algorithm  $\text{MB}(\gamma, \epsilon)$  against it, where  $\epsilon = \frac{1}{M} \sqrt{\gamma/T}$ . Let  $\mathbf{x}$  be the sequence of decisions made by the algorithm,  $\mathbf{c}$  the sequence of costs chosen by the adversary,  $\hat{\mathbf{x}}$  the sequence of recommendations of  $\text{KV}_\epsilon$ , and  $\hat{\mathbf{c}}$  the sequence of approximate costs generated by MB; these are four correlated random variables. Following McMahan and Blum, we split our analysis into three inequalities. These are the same as their inequalities (2)-(4), except with smaller error terms in two of the three.

$$\mathbf{E}(\text{loss}(\mathbf{x}, \mathbf{c})) \leq \mathbf{E}(\text{loss}(\hat{\mathbf{x}}, \hat{\mathbf{c}})) + \gamma MT. \quad (1)$$

$$\mathbf{E}(\text{loss}(\hat{\mathbf{x}}, \hat{\mathbf{c}})) \leq \mathbf{E}(\text{opt}(\hat{\mathbf{c}})) + 2M(4d+1)\sqrt{T/\gamma} \quad (2)$$

$$\mathbf{E}(\text{opt}(\hat{\mathbf{c}})) \leq \mathbf{E}(\text{opt}(\mathbf{c})) + 4d^{3/2}M\sqrt{T/\gamma} \quad (3)$$

Inequality (1) was proved in [10, Theorem 3]; we include the proof in the Appendix. Proofs of (2) and (3) are given in sections 4 and 5, respectively. After setting  $\gamma = dT^{-1/3}$  (which is roughly optimal) summing these inequalities proves Theorem 1.1.  $\square$

### 3 Adaptive and non-adaptive adversaries

In this section we will prove that, for a large class of complete-information online optimization games, adaptive adversaries are no more powerful than non-adaptive adversaries.

For this theorem, we will generalize the class of games considered in three ways.

1. Since the result has nothing to do with the geometric structure of the decision space  $S$ , we think of  $S$  as an arbitrary set, and the adversary's moves as arbitrary real-valued functions  $f^t: S \rightarrow \mathbb{R}$ .
2. For our application, we will need to consider games where the entire sequence  $\mathbf{f} = (f^1, \dots, f^T)$ , of cost functions chosen by the adversary, is constrained to some set  $\mathcal{F} \subset (\mathbb{R}^S)^T$ , which need not be of the form  $K^T$ .

3. For the proof of the theorem, it will be convenient to generalize the notion of regret. Let  $g: \mathcal{F} \rightarrow \mathbb{R}$ . We define the “ $g$  regret” of decision sequence  $\mathbf{x} = (x^1, \dots, x^T)$  against cost function sequence  $\mathbf{f} = (f^1, \dots, f^T)$  as

$$\text{regret}_g(\mathbf{x}, \mathbf{f}) = \sum_{t=1}^T f^t(x^t) - g(\mathbf{f}).$$

**Definition 3.1.** We say algorithm  $\mathcal{A}$  is *forgetful* if, for every  $t \geq 1$ , for every  $s \in S$ ,

$$\begin{aligned} \Pr(x^t = s \mid x^1, \dots, x^{t-1}, f^1, \dots, f^{t-1}) \\ = \Pr(x^t = s \mid f^1, \dots, f^{t-1}). \end{aligned}$$

In words, the distribution of  $x^t$  depends only on the cost functions chosen so far, not on previous decisions made by the algorithm.

**Notation 3.1.** Suppose  $\mathcal{A}$  is an algorithm, and  $\mathcal{V}$  is an adaptive adversary. We will use  $\text{regret}(\mathcal{A}, \mathcal{V})$  as a convenient shorthand for the random variable  $\text{regret}(\mathbf{x}, \mathbf{f})$ , where  $\mathbf{x}$  and  $\mathbf{f}$  are the sequences of decisions and cost functions from a game of  $\mathcal{A}$  against  $\mathcal{V}$ . For a fixed sequence of cost functions,  $\mathbf{f}$ , we will use  $\text{regret}(\mathcal{A}, \mathbf{f})$  as a shorthand for  $\text{regret}(\mathcal{A}, \mathcal{V}_{\mathbf{f}})$ , where  $\mathcal{V}_{\mathbf{f}}$  is the oblivious adversary who always plays the sequence  $\mathbf{f}$ .

We now present the main theorem of this section.

**Theorem 3.1.** *Consider a complete-information online optimization game, with  $\mathcal{F}$  as the set of feasible sequences of cost functions. Let  $\mathcal{A}$  be any forgetful algorithm, and  $\mathcal{V}$  any adaptive adversary. Then*

$$\mathbf{E}(\text{regret}(\mathcal{A}, \mathcal{V})) \leq \max_{\mathbf{f} \in \mathcal{F}} \mathbf{E}(\text{regret}(\mathcal{A}, \mathbf{f})). \quad (4)$$

*Moreover, there is a forgetful algorithm which minimizes the expected worst-case regret over all algorithms. It follows that*

$$\min_{\mathcal{A}} \max_{\mathcal{V}} \mathbf{E}(\text{regret}(\mathcal{A}, \mathcal{V})) = \min_{\mathcal{A}} \max_{\mathbf{f} \in \mathcal{F}} \mathbf{E}(\text{regret}(\mathcal{A}, \mathbf{f})), \quad (5)$$

*where  $\mathcal{A}$  and  $\mathcal{V}$  range over all possible algorithms and adversaries.*

**Remark 3.1.** Note that, in this setting, the adversary never needs to randomize his play, since a randomized adversary is just a convex combination of deterministic adversaries (see, e.g., [1]). However, against randomized algorithms, the adversary can profit greatly by varying its play (in a deterministic way) based on the choices made by the algorithm in previous rounds. An extreme example is when the (far from forgetful) algorithm chooses  $x^1 \in S$  uniformly at random, then sets  $x^t = x^1$  for all  $t \in \{2, \dots, T\}$ .

*Proof of Theorem 3.1.* We will prove (4) with “ $g$  regret” in place of regret, where  $g$  is arbitrary. The result follows from the special case when  $g(\mathbf{f}) = \text{opt}(\mathbf{f}) = \min_{x \in \mathcal{S}} \sum_{t=1}^T f^t(x)$ .

The proof is by induction on  $T$ . The result is clear for  $T = 1$ ; the adversary never gets any information to adapt to.

Suppose the result holds for  $T - 1$ . By Remark 3.1, there is no reason for the adversary to randomize on his own; we may assume without loss of generality that, for all  $t \geq 1$ ,  $f^t$  is uniquely determined by  $x^1, \dots, x^{t-1}$ . In particular,  $f^1$  is fixed. Next, we condition on the value of  $x^1$ .

$$\begin{aligned} \mathbf{E}(\text{regret}_g(\mathbf{x}, \mathbf{f})) &= \mathbf{E}(\mathbf{E}(\text{regret}_g(\mathbf{x}, \mathbf{f}) \mid x^1)) \\ &= \mathbf{E}(f^1(x^1) + \text{regret}_{\tilde{g}}(\mathbf{x}^{2:T}, \mathbf{f}^{2:T})), \end{aligned}$$

where  $\tilde{g}(f^2, \dots, f^T) = g(f^1, f^2, \dots, f^T)$ . Since  $\mathcal{A}$  is forgetful and  $f^1$  is fixed, the algorithm will play the same way on steps  $2, \dots, T$ , irrespective of what happens in round 1. Hence, by inductive hypothesis, there is a non-adaptive adversary maximizing  $\text{regret}_{\tilde{g}}$  against  $\mathcal{A}$ 's play on rounds  $2, \dots, T$ . This concludes the proof of (4).

To see (5), we show that, for every (algorithm, adversary) pair  $(\mathcal{A}, \mathcal{V})$ , there exists a forgetful algorithm  $\mathcal{B} = \mathcal{B}(\mathcal{A}, \mathcal{V})$  such that  $\text{regret}(\mathcal{B}, \mathcal{V})$  has the same distribution as  $\text{regret}(\mathcal{A}, \mathcal{V})$ . To construct  $\mathcal{B}$ , we use a “history revision” approach, similar to that used in [7] to convert “weak” martingales into martingales. At round  $t$ , when the forgetful algorithm only remembers the cost history  $f^1, \dots, f^{t-1}$ ,  $\mathcal{B}$  samples a plausible decision history  $\tilde{x}^1, \dots, \tilde{x}^{t-1}$  with probability

$$p = \Pr(\mathcal{A} \text{ plays } (\tilde{x}^1, \dots, \tilde{x}^{t-1}) \mid \mathcal{V} \text{ plays } (f^1, \dots, f^{t-1})).$$

$\mathcal{B}$  then chooses  $x^t$  by simulating the next step of  $\mathcal{A}$ , conditioned on the sampled history. (The foregoing assumes that  $\Pr(\mathcal{V} \text{ plays } (f^1, \dots, f^{t-1})) \neq 0$ ; when this is not the case,  $\mathcal{B}$  plays in an arbitrary forgetful manner.) To see that  $\text{regret}(\mathcal{B}, \mathcal{V})$  and  $\text{regret}(\mathcal{A}, \mathcal{V})$  have the same distribution is an easy induction argument.  $\square$

## 4 Tinted Glasses

In this section, we prove inequality (2). The key insight behind our improvement is the observation that, if an observer watching the interaction between the Kalai-Vempala-Hannan algorithm and  $\hat{\mathbf{c}}$  were to put on tinted glasses which filtered out all the “exploit” rounds, when  $\hat{\mathbf{c}}^t = 0$ , the observed interaction would still look exactly like the Kalai-Vempala algorithm, only playing a shorter game against an adversary who wastes fewer rounds. Since Kalai and Vempala’s regret bound is a function of the length of the game, this observation leads to an improved regret bound.

This “tinted glasses” claim holds only in the context of a non-adaptive adversary; an adaptive adversary might indirectly detect the presence or absence of such glasses, and make the two outcomes very different. We now state this claim more formally.

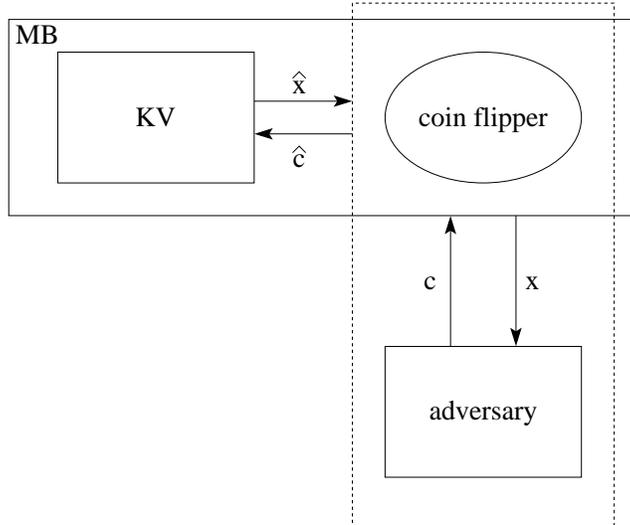


Figure 1: Structural illustration of McMahan and Blum’s algorithm. The key insight for our proof of inequality (2) is that, when analyzing the regret incurred by KV, the “adversary,” represented by the dotted rectangle, includes the impartial coin flipper from the MB algorithm.

**Notation 4.1.** For any fixed sequence  $\mathbf{z} = (z^1, \dots, z^T)$  of cost vectors, let  $\text{nonzero}(\mathbf{z})$  denote the subsequence of non-zero cost vectors ( $z^t \mid 1 \leq t \leq T, z^t \neq \mathbf{0}$ ).

**Observation 4.1.** For all  $\varepsilon > 0$ , and fixed sequence of cost vectors  $\mathbf{z} = (z^1, \dots, z^T) \in (\mathbb{R}^d)^T$ ,  $\text{regret}(\text{KV}_\varepsilon, \mathbf{z})$  and  $\text{regret}(\text{KV}_\varepsilon, \text{nonzero}(\mathbf{z}))$  have the same distribution.

*Proof.* Clearly,  $\text{opt}(\text{nonzero}(\mathbf{z})) = \text{opt}(\mathbf{z})$ . Since rounds when  $\text{KV}_\varepsilon$  encounters a zero cost vector do not affect its behavior on subsequent rounds (the distribution of  $\mathbf{x}^t$  depends only on  $\sum_{i=1}^{t-1} \mathbf{z}^i$ ), and since the direct contribution of such rounds to  $\text{loss}(\text{KV}_\varepsilon, \mathbf{z})$  is also zero, it follows that  $\text{loss}(\text{KV}_\varepsilon, \mathbf{z})$  and  $\text{loss}(\text{KV}_\varepsilon, \text{nonzero}(\mathbf{z}))$  have the same distribution.  $\square \quad \square$

The proof of inequality (2) is somewhat subtle, requiring a careful application of Theorem 3.1 from the previous section, after conditioning on the number of explore steps.

*Proof of (2).* Let  $\ell$  denote the (random) number of explore steps in  $1, \dots, T$ . By Fubini’s theorem,

$$\mathbf{E}(\text{regret}(\text{KV}_\varepsilon, \hat{\mathbf{c}})) = \mathbf{E}(\mathbf{E}(\text{regret}(\text{KV}_\varepsilon, \hat{\mathbf{c}}) \mid \ell)).$$

Condition on the value of  $\ell$ . Now, thinking of the coin flipper as part of the “adversary” opposing  $\text{KV}_\varepsilon$ , we see that this “adversary” is constrained to play sequences of cost vectors with at most  $\ell$  non-zero costs. See Figure 1. Note that allowing the coin flips to be adversarial subject to  $\ell$  can only make the adversary more powerful. Let  $\mathcal{F}_\ell$  denote the set of cost sequences with at most  $\ell$  nonzero costs. Since  $\text{KV}_\varepsilon$  is a forgetful algorithm, Theorem 3.1 implies that there is a fixed cost sequence  $\mathbf{z} \in \mathcal{F}_\ell$  which causes at least as much expected

regret as any adaptive adversary. Together with Observation 4.1, this implies

$$\begin{aligned} \mathbf{E}(\text{regret}(\text{KV}_\varepsilon, \hat{\mathbf{c}}) \mid \ell) &\leq \max_{\mathbf{z} \in \mathcal{F}_\ell} \mathbf{E}(\text{regret}(\text{KV}_\varepsilon, \mathbf{z})) \\ &= \max_{\mathbf{z} \in \mathcal{F}_\ell} \mathbf{E}(\text{regret}(\text{KV}_\varepsilon, \text{nonzero}(\mathbf{z}))). \end{aligned}$$

By the main result of Kalai and Vempala (Theorem 1 of [8]), as extended by McMahan and Blum (Lemma 1 in [10]), applied to the cost sequence  $\text{nonzero}(\hat{\mathbf{c}})$ , we have for  $\mathbf{z} \in \mathcal{F}_\ell$ ,

$$\mathbf{E}(\text{regret}(\text{KV}_\varepsilon, \text{nonzero}(\mathbf{z}))) \leq \varepsilon(4d + 2) \frac{M^2}{\gamma^2} \ell + \frac{4d}{\varepsilon}.$$

Combining all the above, together with the observation that  $\mathbf{E}(\ell) = \gamma T$ , we have

$$\begin{aligned} \mathbf{E}(\text{regret}(\text{KV}_\varepsilon, \hat{\mathbf{c}})) &\leq \mathbf{E}\left(\varepsilon(4d + 2) \frac{M^2}{\gamma^2} \ell + \frac{4d}{\varepsilon}\right) \\ &= \varepsilon(4d + 2) \frac{M^2}{\gamma^2} \gamma T + \frac{4d}{\varepsilon}. \end{aligned}$$

Finally, setting  $\varepsilon = \frac{1}{M} \sqrt{\gamma/T}$ , we conclude

$$\mathbf{E}(\text{regret}(\text{KV}_\varepsilon, \hat{\mathbf{c}})) \leq 2M(4d + 1) \sqrt{T/\gamma}. \quad \square$$

## 5 True versus estimated costs

In this section, we prove inequality (3), which compares the expected optimal loss for the true sequence of costs,  $\mathbf{c}$ , with that for the estimated sequence of costs  $\hat{\mathbf{c}}$ . We use the second moment method.  $\|\cdot\|$  will denote the Euclidean ( $\ell_2$ ) norm on  $\mathbb{R}^d$ .

*Proof of (3).* We first observe that

$$\text{opt}(\hat{\mathbf{c}}) = \text{best}(\hat{\mathbf{c}}) \cdot \sum_{t=1}^T \hat{\mathbf{c}}^t \leq \text{best}(\mathbf{c}) \cdot \sum_{t=1}^T \hat{\mathbf{c}}^t$$

and therefore

$$\begin{aligned} |\text{opt}(\hat{\mathbf{c}}) - \text{opt}(\mathbf{c})| &\leq \left| \text{best}(\mathbf{c}) \cdot \sum_{t=1}^T (\hat{\mathbf{c}}^t - \mathbf{c}^t) \right| \\ &\leq 2\sqrt{d} \left\| \sum_{t=1}^T (\hat{\mathbf{c}}^t - \mathbf{c}^t) \right\|. \end{aligned}$$

For  $1 \leq t \leq T$ , denote  $Y^t = \hat{\mathbf{c}}^t - \mathbf{c}^t$ . Note that  $\mathbf{E}(Y^t | Y^1, \dots, Y^{t-1}) = \mathbf{0}$  since  $\hat{\mathbf{c}}^t$  is an unbiased estimator for  $\mathbf{c}^t$ . It follows by Fubini's theorem that, for any  $s < t$ ,

$$\begin{aligned} \mathbf{E}(Y^s \cdot Y^t) &= \mathbf{E}(\mathbf{E}(Y^s \cdot Y^t | Y^1, \dots, Y^{t-1})) \\ &= \mathbf{E}(Y^s \cdot \mathbf{E}(Y^t | Y^1, \dots, Y^{t-1})) \\ &= \mathbf{0}. \end{aligned}$$

We next find an upper bound on  $\mathbf{E}\left(\left\|\sum_{t=1}^T Y^t\right\|\right)$ . By Jensen's inequality,

$$\begin{aligned} \mathbf{E}\left(\left\|\sum_{t=1}^T Y^t\right\|\right)^2 &\leq \mathbf{E}\left(\left\|\sum_{t=1}^T Y^t\right\|^2\right) \\ &= \mathbf{E}\left(\left(\sum_{t=1}^T Y^t\right) \cdot \left(\sum_{s=1}^T Y^s\right)\right) \\ &= \mathbf{E}\left(\sum_{s,t=1}^T Y^s \cdot Y^t\right) \\ &= \sum_{t=1}^T \mathbf{E}\left(\|Y^t\|^2\right) + 2 \sum_{s=1}^T \sum_{t=s+1}^T \mathbf{E}(Y^s \cdot Y^t) \\ &= \sum_{t=1}^T \mathbf{E}\left(\|Y^t\|^2\right). \end{aligned}$$

Now recall that

$$\hat{\mathbf{c}}^t = \begin{cases} \mathbf{0} & \text{with probability } 1 - \gamma \\ \frac{d}{\gamma} \ell^t \mathbf{e}_j & \text{with probability } \frac{\gamma}{d} \text{ for } j \in \{1, \dots, d\}. \end{cases}$$

By the triangle inequality, it follows that

$$\begin{aligned} \|Y^t\| &\leq \|\hat{\mathbf{c}}^t\| + \|\mathbf{c}^t\| \\ &\leq \begin{cases} M\sqrt{d} & \text{with probability } 1 - \gamma \\ M\left(\frac{d}{\gamma} + \sqrt{d}\right) & \text{otherwise,} \end{cases} \end{aligned}$$

and hence,

$$\mathbf{E}(\|Y^t\|^2) \leq M^2 \left( \frac{d^2}{\gamma} + 2d^{3/2} + d \right) \leq \frac{4d^2 M^2}{\gamma}$$

We conclude that

$$\mathbf{E}(|\text{opt}(\hat{\mathbf{c}}) - \text{opt}(\mathbf{c})|) \leq 4d^{3/2} M \sqrt{T/\gamma}. \quad \square$$

## 6 Lower bounds

The best known lower bound for linear optimization against an adaptive adversary is  $\Omega(\sqrt{T})$ , which is realized by an oblivious adversary whose strategy is just a sequence of independent coin flips.

Here, for completeness, we present an  $\Omega(T^{2/3})$  lower bound for a class of “explore and exploit” algorithms including those of McMahan-Blum [10] and Awerbuch-Kleinberg [3]. This construction is similar to known constructions (see, for example, [1, Theorem 7.1]). Our lower bound uses only oblivious adversaries. It does not apply to the modified experts algorithm of Auer *et. al.* [1]. More recently, a different argument has been found [4] which shows that the  $O(T^{2/3})$  upper bound for the modified experts algorithm is also tight.

For our lower bound to apply, the algorithm must satisfy:

- On each round, it explores with fixed probability  $\gamma$ , otherwise, exploit.
- On explore rounds, the decision is drawn from a fixed probability distribution.
- On exploit rounds, the observation is discarded.

With these constraints on the algorithm, our lower bound applies even in the full information game! In fact, it will be more convenient to assume the full information is available during explore steps.

We assume for concreteness that  $S = \{\mathbf{e}_1, \mathbf{e}_2\}$ , and  $M \geq 1$ , so that  $\mathbf{e}_1$  and  $\mathbf{e}_2$  are feasible cost vectors.

When  $\gamma > T^{-1/3}$ , an oblivious adversary who chooses the same cost vector every round will cause regret  $\Omega(T^{2/3})$  (this much regret will be incurred on explore steps alone).

When  $\gamma \leq T^{-1/3}$ , one of two oblivious adversaries will cause regret  $\Omega(T^{2/3})$ . Both adversaries use a sequence of independent biased coin tosses which come up heads with probability  $\frac{1}{2} + T^{-1/3}$ . The first adversary sets the cost vector to be  $\mathbf{e}_1$  on heads and  $\mathbf{e}_2$  on tails, whereas the second adversary sets  $\mathbf{e}_2$  on heads and  $\mathbf{e}_1$  on tails. If one of these two adversaries is chosen at random, it is clear that the best algorithm is one which always chooses whichever of  $\mathbf{e}_1$  and  $\mathbf{e}_2$  has had less cost so far on explore steps. When this “observed favorite” coincides with the actual favorite, the expected regret from the next round is 0 (assuming it is an exploit round). However, when it does not, the next exploit round produces expected regret  $T^{-1/3}$ . Since the expected total number of explore rounds is  $\gamma T = O(T^{2/3})$ , and the bias of the adversary’s coin is only  $T^{-1/3}$ , this optimal algorithm is expected to make the wrong decision a constant fraction of the time. Hence the total expected regret is  $\Omega(T^{2/3})$ .

## Acknowledgements

We would like to thank Adam Kalai for much encouragement and advice. We are also grateful to Avrim Blum, Sourav Chakraborty and Özgür Sömer for stimulating conversations, and to the anonymous referees for helpful comments and references.

## References

- [1] P. Auer, N. Cesa-Bianchi, Y. Freund, R. Schapire. *Gambling in a rigged casino: The adversarial multi-armed bandit problem*. In: Proceedings of the 36th IEEE Symposium on Foundations of Computer Science (1995) 322–331. There is also an extended version dated June 8, 1998, and a journal version [2].
- [2] P. Auer, N. Cesa-Bianchi, Y. Freund, R. Schapire. *The non-stochastic multi-armed bandit problem*. SIAM Journal on Computing, 32(1):48-77, 2002.
- [3] B. Awerbuch and R. Kleinberg. *Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches*. In: Proceedings of the 36th ACM Symposium on Theory of Computing (2004).
- [4] V. Dani and T. P. Hayes. *How to beat the adversarial multi-armed bandit*. (working title) Manuscript, 2005.
- [5] Y. Freund and R. Schapire. *A decision-theoretic generalization of online learning and an application to boosting*. Journal of Computer and System Sciences, **55(1)** (1997) 119–139.
- [6] J. Hannan. *Approximation to Bayes risk in repeated play*. In: Contributions to the Theory of Games, vol III, 97–139, M. Dresher, A. W. Tucker and P. Wolfe, editors, Princeton University Press (1957).
- [7] T. Hayes. *An Azuma-Hoeffding inequality for vector-valued martingales*. Manuscript (2003).
- [8] A. Kalai and S. Vempala. *Efficient algorithms for on-line optimization*. In: Proceedings of the 16th annual Conference on Learning Theory (2003).
- [9] N. Littlestone and M. Warmuth. *The weighted majority algorithm*. Information and Computation, **108** (1994) 212–261.
- [10] H. B. McMahan and A. Blum. *Online geometric optimization in the bandit setting against an adaptive adversary*. In: Proceedings of the 17th annual Conference on Learning Theory (2004) 109–123.
- [11] H. Robbins. *Some aspects of the sequential design of experiments*. Bulletin of the American Mathematical Society. **55** (1952) 527–535.
- [12] V. Vovk. *Aggregating strategies*. In: Proceedings of the 3rd annual Workshop on Computational Learning Theory (1990) 371–383.
- [13] M. Zinkevich. *Online convex programming and generalized infinitesimal gradient ascent*. In: Proceedings of the 20th International Conference on Machine Learning (2003).

## Appendix

For reference purposes, we include a brief proof of inequality (1), due to McMahan and Blum [10, Theorem 3], but translated into our notation.

*Proof of (1).* For  $1 \leq t \leq T$ , let  $\mathcal{H}^t$  denote the entire history of the algorithm prior to step  $t$ . That is,

$$\mathcal{H}^t = (\mathbf{x}^i, \hat{\mathbf{x}}^i, \chi^i, \mathbf{c}^i : 1 \leq i \leq t - 1).$$

Note that, because the coin flipper operates independently of the black box KV, we have the following conditional independence:

$$\begin{aligned} \mathbf{E}(\hat{\mathbf{x}}^t \cdot \hat{\mathbf{c}}^t \mid \mathcal{H}^t) &= \mathbf{E}(\hat{\mathbf{x}}^t \mid \mathcal{H}^t) \cdot \mathbf{E}(\hat{\mathbf{c}}^t \mid \mathcal{H}^t) \\ &= \mathbf{E}(\hat{\mathbf{x}}^t \mid \mathcal{H}^t) \cdot \mathbf{c}^t \\ &= \mathbf{E}(\hat{\mathbf{x}}^t \cdot \mathbf{c}^t \mid \mathcal{H}^t) \end{aligned}$$

Considering the cases where MB exploits or explores at step  $t$ , we have

$$\begin{aligned} \mathbf{E}(\mathbf{x}^t \cdot \mathbf{c}^t \mid \mathcal{H}^t) &\leq (1 - \gamma)\mathbf{E}(\hat{\mathbf{x}}^t \cdot \mathbf{c}^t \mid \mathcal{H}^t) + \gamma M \\ &= (1 - \gamma)\mathbf{E}(\hat{\mathbf{x}}^t \cdot \hat{\mathbf{c}}^t \mid \mathcal{H}^t) + \gamma M. \end{aligned}$$

Summing this over all  $t$ , and averaging away the conditioning, this becomes

$$\mathbf{E}(\text{loss}(\mathbf{x}, \mathbf{c})) \leq (1 - \gamma)\mathbf{E}(\text{loss}(\hat{\mathbf{x}}, \hat{\mathbf{c}})) + \gamma MT.$$

Since the loss is non-negative, we are done. □