Anytime Algorithms for Multi-Armed Bandit Problems

Robert Kleinberg*

1 Introduction

How should a decision-maker perform repeated choices so as to optimize the average cost or benefit of those choices in the long run? This question motivates the theory of *online learning*, which encompasses problems such as the well-known *best-expert* [13, 9] and *multi-armed bandit* [10, 1] problems. This paper concerns a new approach to dealing with multi-armed bandit problems in which the decision-maker's strategy set is large (exponential or possibly infinite). Recent theoretical progress on the analysis of algorithms for such problems (e.g. [2, 3, 8, 11, 14]) has led to improved online algorithms for problems in areas such as online routing [2], dynamic pricing mechanisms [4, 5, 12], and analysis of reputation systems in e-commerce and peer-to-peer networks [3].

In a multi-armed bandit problem, the decision-maker must repeatedly choose from a fixed set of alternatives (henceforth called "strategies") in a series of trials such that the costs of the strategies vary from one trial to the next, and only the cost of the chosen strategy is revealed after each trial. The goal is to minimize the average cost of the chosen strategies, and the algorithm's efficacy in meeting this goal is evaluated by comparing its average cost with that of the single *stationary* (i.e. not time-varying) strategy whose average cost is minimum. The difference between these two quantities is called *regret.* One typically studies randomized algorithms whose regret converges to zero over time, and the relevant question is how fast this convergence takes place; the δ-convergence time of the algorithm is the minimum number of trials required for the regret to shrink to $O(\delta)$.

When the set of strategies has a fixed finite size K, it has been known for quite some time that there exist algorithms whose convergence time is $O(K \log K)$ even if the costs of the strategies are determined by an adaptive adversary. (For instance, the Exp3 algorithm of [1] achieves this bound.) This convergence time comes very close to matching the trivial lower bound of $\Omega(K)$. (The algorithm must try each strategy at least once in order to guarantee δ-convergence.) This trivial lower bound seems to eliminate hope of designing rapidly-converging multi-armed bandit algorithms when the size of the strategy set is infinite or exponential in the problem size. However, for many problems (e.g. dynamic pricing and online routing) the set of strategies, though large, is somehow structured and the cost functions are constrained by this structure (e.g. the delay of routing along a path is a linear function of the set of edges traversed by the path). This has recently led to the discovery of rapidly converging multi-armed bandit algorithms for the following special cases:

- The strategy set is a one-parameter interval and the cost functions are Lipschitz-continuous [7, 11].
- The strategy set is a bounded compact subset of $\mathbb{R}^n$ and the cost functions are linear [2, 14].
- The strategy set is a bounded convex subset of $\mathbb{R}^n$ and the cost functions are convex [8, 11].

All of these results place some constraints on the class of cost functions in order to achieve rapid convergence, and indeed (as mentioned above) there are trivial lower bounds showing that some such constraints are necessary. When the strategy set is a K-element set and cost functions are unconstrained, the convergence time must be at least $\Omega(K)$. Similarly, when the strategy set is a convex subset of $\mathbb{R}^n$ and the cost functions are constrained only to be Lipschitz-continuous, the convergence time must be at least $\Omega(2^n)$. To derive the latter lower bound, consider a case in which the interior of the strategy set contains 0 and the cost functions are equal to zero in all but one of the 2^n orthants of $\mathbb{R}^n$, negative in the remaining orthant. Any algorithm requires $\Omega(2^n)$ trials before discovering the orthant which contains negative-cost strategies.

In this paper we introduce an alternative way of overcoming these trivial lower bounds by modifying the definition of convergence time. Our approach is

*MIT CSAIL, 32 Vassar St., Cambridge, MA 02139, and Department of Computer Science, Cornell University, Ithaca, NY 14853. Supported by a Fannie and John Hertz Foundation Fellowship. Email: rdk@csail.mit.edu.

motivated by the following observation: in both of the trivial lower bounds cited above, we can force the algorithm to have exponential convergence time *only by arranging for an exponentially small fraction of the strategy set to be superior to all other strategies.* But perhaps we can still design algorithms which meet the following type of guarantee: if a *polynomially small* fraction of the strategy set achieves an average cost less than y, the algorithm should achieve an average cost less than $y + \delta$ (for arbitrarily small constants $\delta > 0$) in a polynomial number of trials. We call algorithms with this property *anytime bandit algorithms* because they have the property that, if stopped at any time $T > 0$, they satisfy a non-trivial performance guarantee which improves as $T \to \infty$, eventually converging to optimality. For example, Corollary 4.4 provides an example of a bandit algorithm meeting the following performance guarantee: *for all $\varepsilon > 0$, if the algorithm is stopped at any time T and its cost is compared with any strategy x which is not among the best ε-fraction of the strategy set, its expected regret is at most $O\left(\frac{\log^2(T)}{\varepsilon T^{1/3}}\right)$.*

We use the word *anytime* here intentionally to draw a parallel with the use of the term *anytime algorithm* in the artificial intelligence literature (e.g. [6, 15]) to refer to algorithms for optimization problems which generate imprecise answers quickly and proceed to construct progressively better approximate solutions over time, eventually converging to the optimal solution. While such algorithms are widely used in the artificial intelligence community, to date there has not been an adequate theoretical treatment of their properties. In this paper we do not claim to provide a comprehensive theoretical foundation for studying anytime algorithms, but we think it is interesting to observe that the setting of multi-armed bandit problems supplies a context in which one may formulate a rigorous notion of "anytime algorithm" and provide provable performance guarantees and lower bounds.

In addition to the two motivations cited above — the goal of defining rapidly-converging bandit algorithms for large strategy sets with unconstrained cost functions, and the goal of studying a context in which rigorous analysis of anytime algorithms is possible — we have a very specific third motivation for developing anytime bandit algorithms. In recent work by the author and Baruch Awerbuch [3], a collaborative learning algorithm is presented which may be regarded as a rapidly-converging distributed multi-armed bandit algorithm in an environment consisting of a large number of users, an unknown subset of whom are dishonest. The algorithm uses a rather complicated subroutine BBA which may be regarded as a particular implementation of an anytime bandit algorithm. The definition of "anytime bandit algorithm" presented here is tailored to allow any such algorithm to be substituted in place of the BBA algorithm in [3]. This leads to simpler and faster collaborative learning algorithms; in particular, it allows an improvement of the main theorem in [3], substituting a convergence time of $O(\log^3(n))$ in place of $O(\log^{16}(n))$, where n is the number of agents and resources in the collaborative learning environment.

We wish to stress that the construction of anytime bandit algorithm (in Section 4 below) is quite straightforward, exploiting a standard doubling technique which is commonly used in the online learning literature. In our opinion, the novelty of this paper, and its main contribution, lies not in constructing these algorithms but in formulating the definition of "anytime bandit algorithm" and recognizing its usefulness, and also in establishing a surprising and non-trivial lower bound for the convergence time of such algorithms. (See Section 5.)

The rest of this paper is organized as follows. In section 2 we formulate two precise definitions of "anytime bandit algorithm." We also formulate a stronger notion which we call a "perfect anytime bandit algorithm," a straw-man representing the best performance guarantee we could hope for in an anytime bandit algorithm. In section 3 we prove the two definitions of an anytime bandit algorithm are equivalent. In section 4 we present algorithms satisfying either of the equivalent definitions. Finally, in section 5 we prove that no perfect anytime bandit algorithm exists.

2 Definitions

2.1 Basic definitions for multi-armed bandit problems

Definition 2.1. (Multi-armed bandit algorithm) *Suppose given a set $\mathcal{S}$ (whose elements are called "strategies") and a collection of functions Γ (whose elements are called "cost functions") each of which is a mapping from $\mathcal{S}$ to $\mathbb{R}$. A* multi-armed bandit algorithm *for $(\mathcal{S}, \Gamma)$ is a randomized online algorithm specified by a probability space Ω_{alg} and a sequence of functions $X_t : \Omega_{\mathrm{alg}} \times \mathbb{R}^{t-1} \to \mathcal{S}$ for $t = 1, 2, \ldots$. We interpret $X_t(r, y_1, \ldots, y_{t-1}) = x$ to mean that the online algorithm chooses strategy x at time t if its random seed is r and the costs of the strategies observed in trials $1, 2, \ldots, t-1$ are $y_1, \ldots, y_{t-1}$, respectively.*

Throughout this paper, we will assume that Γ is equal to the set $[0,1]^{\mathcal{S}}$ of all functions from $\mathcal{S}$ to the closed interval $[0,1]$.

Note that Definition 2.1 doesn't place any limitations on the computational resources which the algorithm may use in computing the functions X_t. However, all of the algorithms introduced in this paper will be computationally efficient, in that they require only polynomial computation time per trial.

Definition 2.2. (Adversary, oblivious adversary) *An* adversary *for a multi-armed bandit problem with strategy set $\mathcal{S}$ and cost function class Γ is specified by a probability space Ω_{adv} and a sequence of functions $C_t : \Omega_{\text{adv}} \times \mathcal{S}^{t-1} \to \Gamma$ for $t = 1, 2, \ldots$. We interpret $C_t(r', x_1, \ldots, x_{t-1}) = c$ to mean that the adversary chooses cost function c at time t if its random seed is r' and the algorithm has played strategies $x_1, \ldots, x_{t-1}$ in trials $1, \ldots, t-1$, respectively.*

A deterministic oblivious adversary *is an adversary such that each function C_t is a constant function mapping $\Omega_{\text{adv}} \times \mathcal{S}^{t-1}$ to some element $c_t \in \Gamma$. A* randomized oblivious adversary *is an adversary such that for all t, the value of $C_t(r', x_1, \ldots, x_{t-1})$ depends only on r', i.e. C_t is a random variable on Ω_{adv} taking values in Γ.*

Definition 2.3. (Transcript of play) *If* ALG *and* ADV *are an algorithm and adversary for a multi-armed bandit problem with strategy set $\mathcal{S}$ and cost function class Γ, then we may define a probability space $\Omega = \Omega_{\text{alg}} \times \Omega_{\text{adv}}$ and sequences of random variables $(x_t), (c_t), (y_t)$ $(1 \leq t < \infty)$ on Ω representing the strategies, cost functions, and feedback values, respectively, that are selected given the random seeds used by the algorithm and adversary. These random variables are defined recursively according to the formulae:*

$$\begin{aligned} x_t(r, r') &= X_t(r, y_1(r, r'), \ldots, y_{t-1}(r, r')) \\ c_t(r, r') &= C_t(r', x_1(r, r'), \ldots, x_{t-1}(r, r')) \\ y_t(r, r') &= c_t(r, r'). \end{aligned}$$

We refer to the probability space Ω and the random variables $(x_t), (c_t), (y_t)$ $(1 \leq t < \infty)$ collectively as the transcript of play *for* ALG *and* ADV.

Definition 2.4. (Regret) *Suppose given an algorithm* ALG *and adversary* ADV *for the multi-armed bandit problem with strategy set $\mathcal{S}$ and cost function class Γ. For any strategy $x \in \mathcal{S}$ and positive integer T, the* regret *of* ALG *relative to x is defined by:*

$$R(\mathsf{ALG}, \mathsf{ADV}; x, T) = \mathbf{E}\left[\sum_{t=1}^{T} c_t(x_t) - c_t(x)\right]$$

and the normalized regret is defined by

$$\overline{R}(\mathsf{ALG}, \mathsf{ADV}; x, T) = \frac{1}{T} R(\mathsf{ALG}, \mathsf{ADV}; x, T).$$

Suppose now that we are given a set of adversaries $\mathcal{A}$ and a subset $U \subseteq \mathcal{S}$. The normalized U-regret *of* ALG *against $\mathcal{A}$ is defined by:*

$$\overline{R}(\mathsf{ALG}, \mathcal{A}; U, T) = \max_{\mathsf{ADV} \in \mathcal{A}} \max_{x \in U} \overline{R}(\mathsf{ALG}, \mathsf{ADV}; x, T).$$

If U is a singleton set $\{x\}$ or $\mathcal{A}$ is a singleton set $\{\mathsf{ADV}\}$, we will use notations such as $\overline{R}(\mathsf{ALG}, \{\mathsf{ADV}\}; \{x\}, T)$ and $\overline{R}(\mathsf{ALG}, \mathsf{ADV}; x, T)$ interchangeably. If U is the entire strategy set $\mathcal{S}$ we will use the notation $\overline{R}(\mathsf{ALG}, \mathcal{A}; T)$.

2.2 Definitions for anytime bandit algorithms

Definition 2.5. (Anytime bandit algorithm, perfect anytime bandit algorithm, convergence time) *Given a probability space $(\mathcal{S}, \mu)$, an algorithm* ALG *is called an* anytime bandit algorithm for $(\mathcal{S}, \mu)$ *if there exists a function $\tau(\varepsilon, \delta)$, defined for all $\varepsilon, \delta > 0$ and taking values in $\mathbb{N}$, such that for all randomized oblivious adversaries* ADV *there exists a subset $U \subseteq \mathcal{S}$ such that $\mu(\mathcal{S} \setminus U) \leq \varepsilon$ and $\overline{R}(\mathsf{ALG}, \mathsf{ADV}; U, T) < \delta$ for all $T > \tau(\varepsilon, \delta)$. It is a* perfect anytime bandit algorithm *if $\tau(\varepsilon, \delta) \leq (1/\varepsilon)\text{poly}(\log(1/\varepsilon), 1/\delta)$. The function $\tau(\varepsilon, \delta)$ is called the* convergence time *of the algorithm.*

To gain an intuition for Definition 2.5, it is helpful to consider the case in which $\mathcal{S}$ is a finite set of size K and μ is the uniform measure on $\mathcal{S}$. Then the definition states that for all $T > \tau(\varepsilon, \delta)$, there are at most εK strategies $x \in \mathcal{S}$ satisfying $\overline{R}(\mathsf{ALG}, \mathsf{ADV}; x, T) \geq \delta$. (This implies, for instance, that $\tau(1/2K, \delta)$ is an upper bound on the algorithm's δ-convergence time, where "δ-convergence time" is defined as in Section 1.) Generalizing now to an arbitrary measure space $(\mathcal{S}, \mu)$, Definition 2.5 says that ALG is an anytime bandit algorithm for $(\mathcal{S}, \mu)$ if the set of strategies which outperform ALG by more than δ shrinks to have measure zero as $T \to \infty$, and has measure less than ε whenever $T > \tau(\varepsilon, \delta)$.

A useful alternative definition of "anytime bandit algorithm" assumes that $\mathcal{S}$ is a countable set whose elements are arranged in an infinite sequence $x_1, x_2, \ldots$. (Equivalently, we may simply assume that $\mathcal{S} = \mathbb{N}$.) We think of an element's position in this sequence as indicating its "priority" for the algorithm, and the algorithm's objective at time T is to perform nearly as well as all of the highest-priority strategies in the sequence, i.e. those belonging to an initial segment $x_1, x_2, \ldots, x_j$ whose length tends to infinity with T.

Definition 2.6. (anytime bandit algorithm for $\mathbb{N}$) *An algorithm* ALG *with strategy set* $\mathbb{N}$ *is called an* anytime bandit algorithm for $\mathbb{N}$ *if there exists a function* $\tau(j, \delta)$*, defined for all* $j \in \mathbb{N}, \delta > 0$ *and taking values in* $\mathbb{N}$*, such that* $\overline{R}(\mathsf{ALG}, \mathcal{A}_{\mathrm{obl}}; \{1, \ldots, j\}, T) < \delta$ *for all* $T > \tau(j, \delta)$*, where* $\mathcal{A}_{\mathrm{obl}}$ *denotes the class of all oblivious adversaries. It is a* perfect anytime bandit algorithm *if* $\tau(j, \delta) \leq j \,\mathrm{poly}(\log(j), 1/\delta)$*. The function* $\tau(j, \delta)$ *is called the* convergence time *of the algorithm.*

In [1] the authors prove a lower bound of $\Omega(\sqrt{KT})$ for the regret of K-armed bandit algorithms against an oblivious adversary. Observe that this implies a lower bound $\tau(j, \delta) = \Omega(j/\delta^2)$ for the convergence time of anytime bandit algorithms for $\mathbb{N}$; similarly it implies a lower bound $\tau(\varepsilon, \delta) = \Omega(1/\varepsilon\delta^2)$ for the convergence time of anytime bandit algorithms for a probability space $(\mathcal{S}, \mu)$. Hence the definition of "perfect anytime bandit algorithm" ensures that the convergence time of such an algorithm is optimal up to a factor of $\mathrm{poly}(\log(j), 1/\delta)$ or $\mathrm{poly}(\log(1/\varepsilon), 1/\delta)$.

3 Equivalence of the definitions

Theorem 3.1. *The following are equivalent:*

1. *There is an anytime bandit algorithm for* $\mathbb{N}$*.*
2. *For all probability spaces* $(\mathcal{S}, \mu)$*, there is an anytime bandit algorithm for* $(\mathcal{S}, \mu)$*.*

Moreover, the two conclusions remain equivalent with "perfect anytime bandit algorithm" in place of "anytime bandit algorithm".

Proof. **(1) ⇒ (2):** Assume that there is an anytime bandit algorithm $\mathsf{ALG}_{\mathbb{N}}$ for $\mathbb{N}$ with convergence time $\tau(j, \delta)$. Given a probability space $(\mathcal{S}, \mu)$, we implement an anytime bandit algorithm ALG_μ for $(\mathcal{S}, \mu)$ as follows. At initialization time, the algorithm samples an infinite sequence $x_1, x_2, x_3, \ldots$ of elements of $\mathcal{S}$ by drawing independent samples from the distribution μ. Next, ALG_μ simulates algorithm $\mathsf{ALG}_{\mathbb{N}}$, choosing strategy x_j every time $\mathsf{ALG}_{\mathbb{N}}$ chooses a strategy $j \in \mathbb{N}$. (Of course, in an actual implementation of ALG_μ, one need not perform an infinite amount of computation at initialization time. Instead, the samples $x_1, x_2, \ldots$ can be determined by "lazy evaluation": whenever $\mathsf{ALG}_{\mathbb{N}}$ decides to a choose a strategy $j \in \mathbb{N}$ which has not been chosen before, ALG_μ draws a new sample $x_j \in \mathcal{S}$ from distribution μ.)

If $\mathsf{ALG}_{\mathbb{N}}$ has convergence time $\tau(j, \delta)$, we claim that ALG_μ has convergence time

$$\tau^*(\varepsilon, \delta) = \tau\left(\left\lceil \frac{1}{\varepsilon} \log\left(\frac{2}{\delta}\right)\right\rceil, \frac{\delta}{2}\right).$$

To see this, let T be any integer greater than $\tau^*(\varepsilon, \delta)$, and for $\theta \in [0, 1]$ let

$$U_\theta = \left\{ x \in \mathcal{S} \;:\; \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x)\right] > \theta \right\}$$

denote the set of strategies whose average cost exceeds θ. This is a measurable subset of $\mathcal{S}$, so we may define

$$\begin{aligned}
\theta^* &= \inf\{\theta \;:\; \mu(U_\theta) < 1 - \varepsilon\} \\
U &= \bigcap_{\theta < \theta^*} U_\theta \\
V &= U_{\theta^*} = \bigcup_{\theta > \theta^*} U_\theta.
\end{aligned}$$

Note that $V \subseteq U$ and

$$\mu(V) \leq 1 - \varepsilon \leq \mu(U).$$

Now let $j = \lceil (1/\varepsilon) \log(2/\delta) \rceil$, and let $\mathcal{E}$ denote the event that $\{x_1, x_2, \ldots, x_j\}$ is a subset of V. For any $x \in U$,

$$\begin{aligned}
&\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) - c_t(x)\right] \\
&= \Pr(\mathcal{E})\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) - c_t(x) \;\middle\|\; \mathcal{E}\right] \\
&\qquad + (1 - \Pr(\mathcal{E}))\,\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) - c_t(x) \;\middle\|\; \overline{\mathcal{E}}\right] \\
&\leq \Pr(\mathcal{E}) + \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) - c_t(x) \;\middle\|\; \overline{\mathcal{E}}\right]
\end{aligned}$$

(3.1)

We claim each term on the right side of (3.1) is less than $\delta/2$, from which it follows that ALG_μ is an anytime bandit algorithm for $(\mathcal{S}, \mu)$ with convergence time $\tau^*(\varepsilon, \delta)$. The fact that $\Pr(\mathcal{E}) < \delta/2$ rests on a straightforward calculation:

$$\Pr(\mathcal{E}) \leq (1 - \varepsilon)^j < e^{-\varepsilon j} \leq \delta/2.$$

To see that the second term on the right side of (3.1) is at most $\delta/2$, note that by the definition of $\tau(j, \delta/2)$

we have

$$\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) \,\middle\|\, x_1, x_2, \ldots, x_j\right] \\ < \delta/2 + \min_i \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_i) \,\middle\|\, x_1, x_2, \ldots, x_j\right]$$

for any values of $x_1, x_2, \ldots, x_j$. Also, since we are assuming ADV is an oblivious adversary,

$$\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_i) \,\middle\|\, x_1, \ldots, x_j\right] = \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_i)\right].$$

Finally, for any set $\{x_1, \ldots, x_j\}$ which is not a subset of V,

$$\min_i \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_i)\right] \leq \theta^* \leq \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x)\right].$$

Since $\overline{\mathcal{E}}$ denotes the event that $\{x_1, x_2, \ldots, x_j\}$ is not a subset of V and is independent of the random variable $\frac{1}{T}\sum c_t(x)$, we conclude that

$$\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) \,\middle\|\, \overline{\mathcal{E}}\right] < \delta/2 + \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x) \,\middle\|\, \overline{\mathcal{E}}\right],$$

which establishes that the second term on the right side of (3.1) is less than $\delta/2$ as claimed. Finally, note that the inequality $\tau(j, \delta) \leq j\,\mathrm{poly}(\log(j), 1/\delta)$ implies that $\tau^*(\varepsilon, \delta) \leq (1/\varepsilon)\mathrm{poly}(\log(1/\varepsilon), 1/\delta)$, which confirms that (1) implies (2) with "perfect anytime" in place of "anytime."

(2) $\Rightarrow$ (1): Define a probability distribution μ on $\mathbb{N}$ by assigning to each singleton set $\{n\}$ a probability proportional to $(n \log^2 n)^{-1}$. (This is a well-defined probability distribution because $\sum_{n=1}^{\infty}(n \log^2 n)^{-1} = C < \infty$ for some constant C.) Now let ALG be an anytime bandit algorithm for $(\mathbb{N}, \mu)$ with convergence time $\tau(\varepsilon, \delta)$. We claim ALG is also an anytime bandit algorithm for $\mathbb{N}$ with convergence time $\tau^*(j, \delta) = \tau((2Cj \log^2 j)^{-1}, \delta)$. To see this, let $\varepsilon = (2Cj \log^2 j)^{-1}$ and observe that $\mu(\{x\}) > \varepsilon$ for all $x \in \{1, 2, \ldots, j\}$. By the definition of an anytime bandit algorithm for $(\mathbb{N}, \mu)$, $\overline{R}(\mathsf{ALG}, \mathcal{A}; x, T) < \delta$ whenever $\mu(\{x\}) > \varepsilon$ and $T > \tau(\varepsilon, \delta)$. Thus $\overline{R}(\mathsf{ALG}, \mathcal{A}; \{1, 2, \ldots, j\}) < \delta$ for any $T > \tau^*(j, \delta)$, as claimed. Finally, note that the inequality $\tau(\varepsilon, \delta) \leq (1/\varepsilon)\mathrm{poly}(\log(1/\varepsilon), 1/\delta)$ implies $\tau^*(j, \delta) \leq j\,\mathrm{poly}(\log j, 1/\delta)$, which confirms that (2) implies (1) with "perfect anytime" in place of "anytime."

4 Construction of anytime bandit algorithms

In this section we specify an anytime bandit algorithm satisfying Definition 2.6. In fact, the definition may be strengthened by enlarging $\mathcal{A}$ to be the set of all adaptive adversaries for $\mathbb{N}$. The algorithm uses, as a subroutine, the adversarial multi-armed bandit algorithm Exp3 [1]. This algorithm Exp3 achieves regret $O(\sqrt{TK\log(K)})$ with strategy set $\{1, 2, \ldots, K\}$ against an adaptive adversary.

Definition 4.1. ($\mathsf{ABA}(F)$) *For any increasing function $F : \mathbb{N} \to \mathbb{N}$, we define an algorithm $\mathsf{ABA}(F)$ as follows. For each $k \geq 0$, at time $F(k)$ the algorithm initializes an instance of* Exp3 *with strategy set $\{1, 2, \ldots, 2^k\}$. From time $F(k)$ to $F(k+1) - 1$ it uses this instance of* Exp3 *to select strategies in $\mathbb{N}$, and at the end of each trial it feeds the cost of the chosen strategy back to* Exp3.

Theorem 4.1. *Let $\mathcal{A}$ denote the set of all adaptive adversaries for strategy set $\mathbb{N}$ and cost function class $\Gamma = [0, 1]^{\mathbb{N}}$. For any $k > 0$ and any $T < F(k)$, the regret of $\mathsf{ABA}(F)$ satisfies*

$$\begin{aligned} &\overline{R}(\mathsf{ABA}(F), \mathcal{A}; \{1, 2, \ldots, j\}, T) \\ &\quad = O\left(F(\lceil \log_2 j \rceil)/T + \sqrt{k2^k/T}\right). \end{aligned}$$

Proof. For $x \in \{1, 2, \ldots, j\}$ and $\mathsf{ADV} \in \mathcal{A}$, we will prove that

$$\mathbf{E}\left[\sum_{t=1}^{T} c_t(x_t) - c_t(x)\right] \leq F(\lceil \log_2 j \rceil) + O(\sqrt{k2^k T}).$$

To do so, we make use of the fact that for $i \geq \lceil \log_2 j \rceil$, strategy x belongs to the strategy set of the Exp3 subroutine operating from time $t_0 = F(i)$ to time $t_1 - 1 = \min(T, F(i+1) - 1)$. This strategy set has cardinality $K = 2^i$, so the regret bound for Exp3 guarantees that

$$\begin{aligned} \mathbf{E}\left[\sum_{t=t_0}^{t_1 - 1} c_t(x_t) - c_t(x)\right] &= O\left(\sqrt{K\log(K)(t_1 - t_0)}\right) \\ &= O\left(\sqrt{i2^i T}\right) \end{aligned}$$

and therefore

$$\begin{aligned}
&\mathbf{E}\left[\sum_{t=1}^{T} c_t(x_t) - c_t(x)\right] \\
&\quad = \sum_{i=1}^{k-1} \mathbf{E}\left[\sum_{t=F(i)}^{\min(T,F(i+1)-1)} c_t(x_t) - c_t(x)\right] \\
&\quad \leq \sum_{i<\lceil \log_2 j\rceil} \sum_{t=F(i)}^{F(i+1)-1} 1 \\
&\qquad + \sum_{\lceil \log_2 j\rceil \leq i<k} \mathbf{E}\left[\sum_{t=F(i)}^{\min(T,F(i+1)-1)} c_t(x_t) - c_t(x)\right] \\
&\quad \leq F(\lceil \log_2 j\rceil) + \sum_{i=1}^{k-1} O\left(\sqrt{i2^iT}\right) \\
&\quad = F(\lceil \log_2 j\rceil) + O\left(\sqrt{k2^kT}\right).
\end{aligned}$$

Corollary 4.1. *For any $\alpha > 0$, there exists an algorithm* ABA *which is anytime bandit algorithm for* $\mathbb{N}$*, whose regret* $\overline{R} = \overline{R}(\mathsf{ABA}, \mathcal{A}; \{1, 2, \ldots, j\}, T)$ *and convergence time* $\tau = \tau(j, \delta)$ *satisfy*

$$\begin{aligned}
\overline{R} &= O\left(\frac{j^{1+\alpha}}{T} + \sqrt{T^{-\frac{\alpha}{1+\alpha}}\log(T)}\right) \\
\tau &= O\left(\frac{j^{1+\alpha}}{\delta} + \left(\frac{\log(j/\delta)}{\delta^2}\right)^{1+1/\alpha}\right)
\end{aligned}$$

Proof. Let $F(k) = \lceil 2^{(1+\alpha)k}\rceil$, let $\mathsf{ABA} = \mathsf{ABA}(F)$, and apply Theorem 4.1.

Corollary 4.2. *There exists an algorithm* ABA *which is an anytime bandit algorithm for* $\mathbb{N}$*, whose regret* $\overline{R} = \overline{R}(\mathsf{ABA}, \mathcal{A}; \{1, 2, \ldots, j\}, T)$ *and convergence time* $\tau = \tau(j, \delta)$ *satisfy*

$$\begin{aligned}
\overline{R} &= O(j \log^3(j)/T + 1/\log(T)) \\
\tau &= O\left(j\log^3(j)/\delta + 2^{O(1/\delta)}\right).
\end{aligned}$$

Proof. Let $F(k) = k^3 2^k$, let $\mathsf{ABA} = \mathsf{ABA}(F)$, and apply Theorem 4.1.

The next corollary is useful primarily as a construction of an anytime bandit algorithm to be plugged into the collaborative learning algorithm of [3] in place of the complicated and inefficient BBA subroutine.

Corollary 4.3. *There exists an algorithm* ABA *which is an anytime bandit algorithm for* $\mathbb{N}$*, with regret satisfying*

$$\overline{R}(\mathsf{ABA}, \mathcal{A}; \{1, 2, \ldots, j\}, T) \quad = \quad O\left(j\log(T)/T^{1/3}\right).$$

Proof. Setting $\alpha = 2$ in the preceding corollary, we obtain an algorithm whose regret satisfies

$$\overline{R}(\mathsf{ABA}, \mathcal{A}; \{1, 2, \ldots, j\}, T) = O\left(\frac{j^3}{T} + \frac{\log(T)}{T^{1/3}}\right).$$

Trivially, the regret also satisfies $\overline{R}(\mathsf{ABA}, \mathcal{A}; \{1, 2, \ldots, j\}, T) \leq 1$. To prove the corollary, it suffices to prove that for all sufficiently large j, T,

$$j\log(T)/T^{1/3} \geq \min\{1, j^3/T + \log(T)/T^{1/3}\}.$$

Assume, to the contrary, that $j\log(T)/T^{1/3} < 1$ and that $j\log(T)/T^{1/3} < j^3/T + \log(T)/T^{1/3}$. Rearranging terms in the second inequality, we obtain $T^{2/3}\log(T) < \frac{j^3}{j-1}$, while the first inequality implies $\frac{\log^2(T)}{T^{2/3}} < \frac{1}{j^2}$. Multiplying these two together, we obtain $\log^3(T) < \frac{j}{j-1}$, which is not possible for sufficiently large j, T.

Corollary 4.4. *For any probability space $(\mathcal{S}, \mu)$, there exists an algorithm* ABA *which is an anytime bandit algorithm for $(\mathcal{S}, \mu)$, whose regret meets the following guarantee: for every randomized oblivious adversary* ADV*, there is a set $U \subseteq \mathcal{S}$ of measure at least $1 - \varepsilon$, such that*

$$\overline{R}(\mathsf{ABA}, \mathsf{ADV}; U, T) \quad = \quad O\left(\frac{\log^2(T)}{\varepsilon T^{1/3}}\right). \tag{4.2}$$

Proof. The first half of the proof of Theorem 3.1 established a bound on the regret of anytime bandit algorithm algorithms for $(\mathcal{S}, \mu)$ which can be expressed as follows. If $\mathsf{ALG}_{\mathbb{N}}$ is an anytime bandit algorithm for $\mathbb{N}$ whose regret satisfies

$$\overline{R}(\mathsf{ALG}_{\mathbb{N}}, \mathcal{A}; \{1, \ldots, j\}, T) \leq f(j, T) \tag{4.3}$$

for some function f, then there is an anytime bandit algorithm ALG_{μ} for $(\mathcal{S}, \mu)$ whose regret meets the following guarantee for all $\varepsilon > 0$, for all $j > 0$, and for all oblivious adversaries ADV: there exists a set $U \subseteq \mathcal{S}$ of measure at least $1 - \varepsilon$, such that

$$\overline{R}(\mathsf{ALG}_{\mu}, \mathsf{ADV}; U, T) \leq e^{-\varepsilon j} + f(j, T).$$

The corollary follows by applying this fact using the algorithm from Corollary 4.3, which satisfies (4.3) with $f(j, T) = O\left(j\log(T)/T^{1/3}\right)$. One puts $j = \lceil \log(T)/\varepsilon\rceil$ to obtain the bound stated in (4.2).

5 Non-existence of perfect anytime bandit algorithms

In the preceding section we saw that anytime bandit algorithms for $\mathbb{N}$ can achieve convergence time

$O(j^{1+\alpha}\,\mathrm{poly}(1/\delta))$ for arbitrarily small positive constants α, and that they can also achieve convergence time $O(j\,\mathrm{polylog}(j)\,2^{O(1/\delta)})$. Given these positive results, it is natural to wonder whether one can achieve convergence time $O(j\,\mathrm{polylog}(j)\,\mathrm{poly}(1/\delta))$, i.e. whether a perfect anytime bandit algorithm exists. This question is answered negatively by the following theorem.

Theorem 5.1. *Let d be any positive integer. There does not exist an anytime bandit algorithm for $\mathbb{N}$ achieving convergence time $\tau(j,\delta) = O(j\,\log^d(j)\,\delta^{-d})$.*

Proof. Assume, by way of contradiction, that ALG is an algorithm with convergence time $\tau(j,\delta) < Cj\log^d(j)\delta^{-d}$. We will consider the algorithm's regret against an oblivious adversary ADV who supplies an input instance in which all cost functions c_t are equal to a single random cost function c, defined as follows. Let r_k $(1 \le k < \infty)$ be independent random variables, where r_k is uniformly distributed over the set $\{2^{2^{k-1}}+1, 2^{2^{k-1}}+2, \ldots, 2^{2^k}\} \times \{0,1\}$. Let $c(1) = 1/4$, and define $c(j)$ for $j \ge 2$ as follows: let $k = \lceil \log_2(\log_2(j)) \rceil$ and put

$$c(j) = \begin{cases} 2^{-k} & \text{if } r_k = (j,1) \\ 1 & \text{otherwise.} \end{cases}$$

In other words, with probability 1/2 the cost of every element in the set $\{2^{2^{k-1}}+1,\ldots,2^{2^k}\}$ is equal to 1, and with probability 1/2 there is a uniformly distributed random element of this set with cost 2^{-k}, and all others have cost 1.

Presented with this input, the algorithm ALG will select a random sequence of strategies $x_1, x_2, \ldots, x_T$. Let us say that the algorithm performs a *probe* at time t if this is the first time that it samples x_t, i.e. $x_t \notin \{x_1, x_2, \ldots, x_{t-1}\}$. Let q_t be the Bernoulli random variable

$$q_t = \begin{cases} 1 & \text{if } \mathsf{ALG} \text{ performs a probe at time } t \\ 0 & \text{otherwise} \end{cases}$$

and let $Q = \sum_{t=1}^{T} q_t$ denote the random variable which counts the number of probes up to time T. We will frequently need to use the following fact.

Claim 5.1. $\Pr(\min_{1\le t\le T} c(x_t) \le 2^{-k} \parallel Q)$ *is at most* $Q \Big/ \left(2^{2^k} - 2^{2^{k-1}}\right)$.

Proof. For $x > 0$, let $\mathrm{loog}(x) = \lceil \log_2(\log_2(x)) \rceil$ and let $r(x) = 2^{2^{\mathrm{loog}(x)}} - 2^{2^{\mathrm{loog}(x)-1}}$ denote the number of strategies in the set $\{2^{2^{\mathrm{loog}(x)-1}}+1, \ldots, 2^{2^{\mathrm{loog}(x)}}\}$. Let $\tau_1 < \tau_2 < \ldots < \tau_Q$ denote the numbers of the trials in which the algorithm performs its Q probes. For $0 \le s < Q$,

$$\begin{aligned} &\Pr\left(c(x_{\tau_{s+1}}) > 2^{-k} \parallel c(x_{\tau_1}), c(x_{\tau_2}), \ldots, c(x_{\tau_s})\right) \\ &\quad \ge \begin{cases} 1 - \frac{1}{r(x)-s} & \text{if } x > 2^{2^{k-1}} \\ 1 & \text{otherwise} \end{cases} \\ &\quad \ge 1 - \frac{1}{2^{2^k} - 2^{2^{k-1}} - s}. \end{aligned}$$

Hence

$$\begin{aligned} &\Pr\left(\min_{1\le t\le T} c(x_t) > 2^{-k}\right) \\ &\quad \ge \prod_{s=0}^{Q-1} \left(1 - \frac{1}{2^{2^k} - 2^{2^{k-1}} - s}\right) \\ &\quad = 1 - \frac{Q}{2^{2^k} - 2^{2^{k-1}}}, \end{aligned}$$

which establishes the claim.

Resuming the proof of Theorem 5.1, put $T = 2^{2^k + 3dk}$. We distinguish two cases.

Case 1: $\Pr\left(Q > \frac{1}{2}\left(2^{2^k} - 2^{2^{k-1}}\right)\right) < 3/4$.

Case 2: $\Pr\left(Q > \frac{1}{2}\left(2^{2^k} - 2^{2^{k-1}}\right)\right) \ge 3/4$.

In Case 1, let $j = 2^{2^k}$, $\delta = 2^{-k-5}$. For sufficiently large k,

$$\tau(j,\delta) = Cj\log^d(j)\delta^{-d} = 2^{2^k + 2dk + 5d + \log_2(C)} < T.$$

We will prove that

$$\overline{R}(\mathsf{ALG}, \mathsf{ADV}; \{1,2,\ldots,j\}, T) > \delta,$$

thus obtaining a contradiction. Consider the following three events.

$$\begin{aligned} \mathcal{E}_1 &= \left\{Q \le \frac{1}{2}\left(2^{2^k} - 2^{2^{k-1}}\right)\right\} \\ \mathcal{E}_2 &= \left\{\min_{1\le t\le T} c(x_t) \ge 2^{-(k-1)}\right\} \\ \mathcal{E}_3 &= \left\{\min_{1\le x\le 2^{2^k}} c(x) = 2^{-k}\right\} \end{aligned}$$

By assumption, $\Pr(\mathcal{E}_1) > 1/4$. Claim 5.1 establishes that $\Pr(\mathcal{E}_2 \parallel \mathcal{E}_1) \ge 1/2$. Next we argue that $\Pr(\mathcal{E}_3 \parallel \mathcal{E}_1 \wedge \mathcal{E}_2) \ge 1/3$. Let $U = \{2^{2^{k-1}} + 1, \ldots, 2^{2^k}\}$, and let V denote the intersection of U

with $\{x_1, x_2, \ldots, x_T\}$. Conditional on $\mathcal{E}_1$, $|V| \le |U|/2$, and conditional on $\mathcal{E}_2$, r_k is uniformly distributed in the set $U \times \{0,1\} \setminus V \times \{1\}$. Hence the probability is at least 1/3 that $r_k \in (U \setminus V) \times \{1\}$, which implies $\mathcal{E}_3$.

Putting this all together, $\Pr(\mathcal{E}_1 \wedge \mathcal{E}_2 \wedge \mathcal{E}_3) > 1/24$. Assuming $\mathcal{E}_2$ and $\mathcal{E}_3$, there exists a strategy $x \in \{1, 2, \ldots, j\}$ such that $\frac{1}{T}\sum_{t=1}^{T} c(x_t) - c(x) \ge 2^{-k} = 32\delta$. Thus,

$$\begin{aligned}
&\overline{R}(\mathsf{ALG}, \mathsf{ADV}; \{1, 2, \ldots, j\}, T) \\
&\quad = \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c(x_t) - c(x)\right] \\
&\quad \ge 32\delta \Pr(\mathcal{E}_2 \wedge \mathcal{E}_3) > \delta,
\end{aligned}$$

as claimed.

In Case 2, let $j = 2^{2^{k-1}}$ and $\delta = 2^{-2^{k-1}/d}$. Note that $\tau(j, \delta) < T$ provided that k is sufficiently large. Letting $\overline{\mathcal{E}}$ denote the complement of an event $\mathcal{E}$, we have $\Pr(\overline{\mathcal{E}_1}) \ge 3/4$ by assumption, and we have

$$\Pr(\overline{\mathcal{E}_3}) = \Pr\left(r_k \in \{2^{2^{k-1}} + 1, \ldots, 2^{2^k}\} \times \{0\}\right) = \frac{1}{2};$$

hence $\Pr(\overline{\mathcal{E}_1} \wedge \overline{\mathcal{E}_3}) \ge 1/4$. Let

$$\mathcal{E}_4 = \left\{\min_{1 \le t \le T} c(x_t) \ge 2^{-k}\right\}.$$

By Claim 5.1,

$$\Pr(\overline{\mathcal{E}_4}) \;\le\; T \Big/ \left(2^{2^{k+1}} - 2^{2^k}\right) = o(1),$$

so for k sufficiently large

$$\Pr(\overline{\mathcal{E}_1} \wedge \overline{\mathcal{E}_3} \wedge \mathcal{E}_4) > 1/8.$$

Let $x = \arg\min_{1 \le i \le j} c(i)$. Assuming $\overline{\mathcal{E}_3}$ and $\mathcal{E}_4$, we have $c(x_t) \ge c(x)$ for $t = 1, 2, \ldots, T$. Moreover, assuming $\mathcal{E}_4$, there are at least $Q - k - 1$ probes with cost 1, so

$$\frac{1}{T}\sum_{t=1}^{T} c(x_t) - c(x) \ge \frac{1}{T}(Q - k - 1)(1 - c(x)) \ge \frac{Q - k - 1}{2T}.$$

Assuming $\overline{\mathcal{E}_1}$ and assuming k is sufficiently large,

$$\frac{Q - k - 1}{2T} > 2^{-4dk} > 2^{3 - 2^{k-1}/d} = 8\delta,$$

hence

$$\begin{aligned}
&\overline{R}(\mathsf{ALG}, \mathsf{ADV}; \{1, 2, \ldots, j\}, T) \\
&\quad \ge \frac{Q - k - 1}{2T} \Pr(\overline{\mathcal{E}_1} \wedge \overline{\mathcal{E}_3} \wedge \mathcal{E}_4) > \delta,
\end{aligned}$$

contradicting the assumption that ALG is an anytime bandit algorithm with convergence time $\tau(j, \delta)$.

References

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

[2] 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 (STOC), 2004, pp. 45–53.

[3] ———, *Competitive collaborative learning*, in Proceedings of the 18th Annual Conference on Learning Theory (COLT), 2005. To appear.

[4] A. Blum and J. Hartline, *Near-optimal online auctions*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005), 2005, pp. 1156–1163.

[5] A. Blum, V. Kumar, A. Rudra, and F. Wu, *Online learning in online auctions*, Theoretical Computer Science, 324 (2004), pp. 137–146.

[6] M. Boddy, *Anytime problem solving using dynamic programming*, Proceedings of National Conference on Artificial Intelligence (AAAI), (1991).

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

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

[9] Y. Freund and R. E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, Journal of Computer and System Sciences, 55 (1997), pp. 119–139.

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

[11] R. Kleinberg, *Nearly tight bounds for the continuum-armed bandit problem*, in Advances in Neural Information Processing Systems 17, L. K. Saul, Y. Weiss, and L. Bottou, eds., MIT Press, Cambridge, MA, 2005, pp. 697–704.

[12] R. Kleinberg and T. Leighton, *The value of knowing a demand curve: Bounds on regret for on-line posted-price auctions*, in Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FOCS), 2003, pp. 594–605.

[13] N. Littlestone and M. K. Warmuth, *The weighted majority algorithm*, Information and Computation, 108 (1994), pp. 212–260.

[14] 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 (COLT), vol. 3120

of Lecture Notes in Computer Science, Springer Verlag, 2004, pp. 109–123.

[15] S. Zilberstein, *Operational rationality through compilation of anytime algorithms*, 1993.