Gambling in a rigged casino:
The adversarial multi-armed bandit problem*

Peter Auer
Institute for Theoretical Computer Science
University of Technology Graz
A-8010 Graz (Austria)
pauer@igi.tu-graz.ac.at

Nicolò Cesa-Bianchi
Department of Computer Science
Università di Milano
I-20135 Milano (Italy)
cesabian@dsi.unimi.it

Yoav Freund **Robert E. Schapire**
AT&T Labs
180 Park Avenue
Florham Park, NJ 07932-0971
{yoav, schapire}@research.att.com

June 8, 1998

Abstract

In the multi-armed bandit problem, a gambler must decide which arm of K non-identical slot machines to play in a sequence of trials so as to maximize his reward. This classical problem has received much attention because of the simple model it provides of the trade-off between exploration (trying out each arm to find the best one) and exploitation (playing the arm believed to give the best payoff). Past solutions for the bandit problem have almost always relied on assumptions about the statistics of the slot machines.

In this work, we make no statistical assumptions whatsoever about the nature of the process generating the payoffs of the slot machines. We give a solution to the bandit problem in which an adversary, rather than a well-behaved stochastic process, has complete control over the payoffs. In a sequence of T plays, we prove that the expected per-round payoff of our algorithm approaches that of the best arm at the rate $O(T^{-1/2})$, and we give an improved rate of convergence when the best arm has fairly low payoff. We also prove a general matching lower bound on the best possible performance of any algorithm in our setting. In addition, we consider a setting in which the player has a team of "experts" advising him on which arm to play; here, we give a strategy that will guarantee expected payoff close to that of the best expert. Finally, we apply our result to the problem of learning to play an unknown repeated matrix game against an all-powerful adversary.

1 Introduction

In the well studied multi-armed bandit problem, originally proposed by Robbins [16], a gambler must choose which of K slot machines to play. At each time step, he pulls the arm of one of the machines and receives a reward or payoff (possibly zero or negative). The gambler's purpose is to maximize his total reward over a sequence of trials. Since each arm is assumed to have a different distribution of rewards, the goal is to find the arm with the best expected return as early as possible, and then to keep gambling using that arm.

*An early extended abstract of this paper appeared in the proceedings of the *36th Annual Symposium on Foundations of Computer Science*, pages 322–331, 1995. The present draft is a very substantially revised and expanded version which has been submitted for journal publication.

The problem is a classical example of the trade-off between exploration and exploitation. On the one hand, if the gambler plays exclusively on the machine that he thinks is best ("exploitation"), he may fail to discover that one of the other arms actually has a higher average return. On the other hand, if he spends too much time trying out all the machines and gathering statistics ("exploration"), he may fail to play the best arm often enough to get a high total return.

As a more practically motivated example, consider the task of repeatedly choosing a route for transmitting packets between two points in a communication network. Suppose there are K possible routes and the transmission cost is reported back to the sender. Then the problem can be seen as that of selecting a route for each packet so that the total cost of transmitting a large set of packets would not be much larger than the cost incurred by sending all of them on the single best route.

In the past, the bandit problem has almost always been studied with the aid of statistical assumptions on the process generating the rewards for each arm. In the gambling example, for instance, it might be natural to assume that the distribution of rewards for each arm is Gaussian and time-invariant. However, it is likely that the costs associated with each route in the routing example cannot be modeled by a stationary distribution, so a more sophisticated set of statistical assumptions would be required. In general, it may be difficult or impossible to determine the right statistical assumptions for a given domain, and some domains may be inherently adversarial in nature so that no such assumptions are appropriate.

In this paper, we present a variant of the bandit problem in which *no* statistical assumptions are made about the generation of rewards. In our model, the reward associated with each arm is determined at each time step by an adversary with unbounded computational power rather than by some benign stochastic process. We only assume that the rewards are chosen from a bounded range. The performance of any player is measured in terms of *regret*, i.e., the expected difference between the total reward scored by the player and the total reward scored by the best arm.

At first it may seem impossible that the player should stand a chance against such a powerful opponent. Indeed, a *deterministic* player will fare very badly against an adversary who assigns low payoff to the chosen arm and high payoff to all the other arms. However, in this paper we present a very efficient, randomized player algorithm that performs well against any adversary. We prove that the difference between the expected gain of our algorithm and the expected gain of the best out of K arms is at most $O(\sqrt{TK\log K})$, where T is the number of time steps. Note that the average per-time-step regret approaches zero at the rate $O(1/\sqrt{T})$. We also present more refined bounds in which the dependence on T is replaced by the total reward of the best arm (or an assumed upper bound thereof).

Our worst-case bounds may appear weaker than the bounds proved using statistical assumptions, such as those shown by Lai and Robbins [12] of the form $O(\log T)$. However, when comparing our results to those in the statistics literature, it is important to point out an important difference in the asymptotic quantification. In the work of Lai and Robbins the assumption is that the distribution of rewards that is associated with each arm is *fixed* as the total number of iterations T increases to infinity. In contrast, our bounds hold for any finite T, and, by the generality of our model, these bounds are applicable when the payoffs are randomly (or adversarially) chosen in a manner that does depend on T. It is this quantification order, and not the adversarial nature of our framework, which is the cause for the apparent gap. We prove this point by showing that, for *any* algorithm for the K-arm bandit problem and for any T there exists a set of K reward distributions such that the expected regret of the algorithm when playing against such arms for T iterations is lower bounded by $\Omega(\sqrt{KT})$.

We can also show that the per-sequence regret is well behaved. More precisely, we show that our algorithm can guarantee that the actual (rather than expected) difference between its gain and the gain of the best arm on any run is upper bounded by $O(T^{2/3}(K\ln K)^{1/3})$ with high probability. This bound is weaker than the bound on the expected regret. It is not clear whether or not this bound can be improved to have a dependence of $O(\sqrt{T})$ on the number of trials.

A non-stochastic bandit problem was also considered by Gittins [9] and Ishikida and Varaiya [11]. However, their version of the bandit problem is very different from ours: they assume that the player can compute ahead of time exactly what payoffs will be received from each arm, and their problem is thus one of optimization, rather than exploration and exploitation.

Our algorithm is based in part on an algorithm presented by Freund and Schapire [6, 7], which in turn is a variant of Littlestone and Warmuth's [13] weighted majority algorithm, and Vovk's [17] aggregating strategies. In the setting analyzed by Freund and Schapire (which we call here the *full information game*), the player on each trial scores the reward of the chosen arm, but gains access to the rewards associated with *all* of the arms (not just the one that was chosen).

In some situations, picking the same action at all trials might not be the best strategy. For example, in the packet routing problem it might be that no single route is good for the whole duration of the message, but switching between routes from time to time can yield better performance. We give a variant of our algorithm which combines the choices of N strategies (or "experts"), each of which recommends one of the K actions at each iteration. We show that the regret with respect to the best strategy is $O(\sqrt{TK \ln N})$. Note that the dependence on the number of strategies is only logarithmic, and therefore the bound is quite reasonable even when the player is combining a very large number of strategies.

The adversarial bandit problem is closely related to the problem of learning to play an unknown repeated matrix game. In this setting, a player without prior knowledge of the game matrix is playing the game repeatedly against an adversary with complete knowledge of the game and unbounded computational power. It is well known that matrix games have an associated *value* which is the best possible expected payoff when playing the game against an adversary. If the matrix is known, then a randomized strategy that achieves the value of the game can be computed (say, using a linear-programming algorithm) and employed by the player. The case where the matrix is entirely unknown was previously considered by Baños [1] and Megiddo [14], who proposed two different strategies whose per-round payoff converges to the game value. Both of these algorithms are extremely inefficient. For the same problem, we show that by using our algorithm the player achieves an expected per-round payoff in T rounds which efficiently approaches the value of the game at the rate $O(T^{-1/2})$. This convergence is much faster than that achieved by Baños and Megiddo.

Our paper is organized as follows. In Section 2, we give the formal definition of the problem. In Section 3, we describe Freund and Schapire's algorithm for the full information game and state its performance. In Section 4, we describe our basic algorithm for the partial information game and prove the bound on the expected regret. In Section 5, we prove a bound on the regret of our algorithm on "typical" sequences. In Section 6, we show how to adaptively tune the parameters of the algorithm when no prior knowledge (such as the length of the game) is available. In Section 7, we give a lower bound on the regret suffered by any algorithm for the partial information game. In Section 8, we show how to modify the algorithm to use expert advice. Finally, in Section 9, we describe the application of our algorithm to repeated matrix games.

2 Notation and terminology

We formalize the bandit problem as a game between a player choosing actions and an adversary choosing the rewards associated with each action. The game is parameterized by the number K of possible actions, where each action is denoted by an integer i, $1 \leq i \leq K$. We will assume that all the rewards belong to the unit interval $[0, 1]$. The generalization to rewards in $[a, b]$ for arbitrary $a < b$ is straightforward.

The game is played in a sequence of trials $t = 1, 2, \ldots, T$. We distinguish two variants: the partial information game, which captures the adversarial multi-armed bandit problem; and the full information game, which is essentially equivalent to the framework studied by Freund and Schapire [6]. On each trial t of the *full information game*:

1. The adversary selects a vector $\mathbf{x}(t) \in [0,1]^K$ of current rewards. The ith component $x_i(t)$ is interpreted as the reward associated with action i at trial t.

2. Without knowledge of the adversary's choice, the player chooses an action by picking a number $i_t \in \{1, 2, \ldots, K\}$ and scores the corresponding reward $x_{i_t}(t)$.

3. The player observes the entire vector $\mathbf{x}(t)$ of current rewards.

The *partial information game* corresponds to the above description of the full information game but with step 3 replaced by:

3′. The player observes only the reward $x_{i_t}(t)$ for the chosen action i_t.

Let $G_A \doteq \sum_{t=1}^{T} x_{i_t}(t)$ be the total reward of player A choosing actions $i_1, i_2, \ldots, i_T$.

We formally define an adversary as a deterministic[1] function mapping the past history of play $i_1, \ldots, i_{t-1}$ to the current reward vector $\mathbf{x}(t)$. As a special case, we say that an adversary is *oblivious* if it is independent of the player's actions, i.e., if the reward at trial t is a function of t only. All of our results, which are proved for a nonoblivious adversary, hold for an oblivious adversary as well.

As our player algorithms will be randomized, fixing an adversary and a player algorithm defines a probability distribution over the set $\{1, \ldots, K\}^T$ of sequences of T actions. All the probabilities and expectations considered in this paper will be with respect to this distribution. For an oblivious adversary, the rewards are fixed quantities with respect to this distribution, but for a nonoblivious adversary, each reward $x_i(t)$ is a random variable defined on the set $\{1, \ldots, K\}^{t-1}$ of player actions up to trial $t-1$. We will not use explicit notation to represent this dependence, but will refer to it in the text when appropriate.

The measure of the performance of our algorithm is the *regret*, which is the difference between the total reward of the algorithm G_A and the total reward of the best action. We shall mostly be concerned with the expected regret of the algorithm. Formally, we define the expected total reward of algorithm A by

$$\mathbf{E}[G_A] \doteq \mathbf{E}_{i_1,\ldots,i_T}\left[\sum_{t=1}^{T} x_{i_t}(t)\right],$$

the expected total reward of the best action by

$$EG_{\max} \doteq \max_{1 \le j \le K} \mathbf{E}_{i_1,\ldots,i_T}\left[\sum_{t=1}^{T} x_j(t)\right],$$

and the expected regret of algorithm A by $R_A \doteq EG_{\max} - \mathbf{E}[G_A]$. This definition is easiest to interpret for an oblivious adversary since, in this case, $EG_{\max}$ truly measures what could have been gained had the best action been played for the entire sequence. However, for a nonoblivious adversary, the definition of regret is a bit strange: It still compares the total reward of the algorithm to the sum of rewards that were associated with taking some action j on all iterations; however, had action j actually been taken, the rewards chosen by the adversary would have been different than those actually generated since the variable $x_j(t)$ depends on the past history of plays $i_1, \ldots, i_{t-1}$. Although the definition of R_A looks difficult to interpret in this

[1]There is no loss of generality in assuming that the adversary is deterministic. To see this, assume that the adversary maps past histories to *distributions* over the values of $\mathbf{x}(t)$. This defines a stochastic strategy for the adversary for the T step game, which is equivalent to a distribution over all *deterministic* adversarial strategies for the T step game. Assume that A is any player algorithm and that B is the worst-case *stochastic* strategy for the adversary playing against A. The stated equivalence implies that there is a *deterministic* adversarial strategy $\tilde{B}$ against which the gain of A is at most as large as the gain of A against B. (The same argument can easily be made for other measures of performance, such as the regret, which is defined shortly.)

Algorithm Hedge
Parameter: A real number $\eta > 0$.
Initialization: Set $G_i(0) := 0$ for $i = 1, \ldots, K$.

Repeat for $t = 1, 2, \ldots$ until game ends

1. Choose action i_t according to the distribution $\mathbf{p}(t)$, where
$$p_i(t) = \frac{\exp(\eta G_i(t-1))}{\sum_{j=1}^{K} \exp(\eta G_j(t-1))}.$$
2. Receive the reward vector $\mathbf{x}(t)$ and score gain $x_{i_t}(t)$.
3. Set $G_i(t) := G_i(t-1) + x_i(t)$ for $i = 1, \ldots, K$.

Figure 1: Algorithm $\mathbf{Hedge}$ for the full information game.

case, in Section 9 we prove that our bounds on the regret for a nonoblivious adversary can also be used to derive an interesting result in the context of repeated matrix games.

We shall also give a bound that holds with high probability on the actual regret of the algorithm, i.e., on the actual difference between the gain of the algorithm and the gain of the best action:
$$\max_j \sum_{t=1}^{T} x_j(t) - G_A.$$

3 The full information game

In this section, we describe an algorithm, called $\mathbf{Hedge}$, for the full information game which will also be used as a building block in the design of our algorithm for the partial information game. The version of $\mathbf{Hedge}$ presented here is a variant[2] of the algorithm introduced by Freund and Schapire [6] which itself is a direct generalization of Littlestone and Warmuth's Weighted Majority [13] algorithm.

$\mathbf{Hedge}$ is described in Figure 1. The main idea is simply to choose action i at time t with probability proportional to exp $(\eta G_i(t-1))$, where $\alpha > 0$ is a parameter and $G_i(t) = \sum_{t'=1}^{t} x_i(t')$ is the total reward scored by action i up through trial t. Thus, actions yielding high rewards quickly gain a high probability of being chosen.

Since we allow for rewards larger than 1, proving bounds for $\mathbf{Hedge}$ is more complex than for Freund and Schapire's original algorithm. The following is an extension of Freund and Schapire's Theorem 2. Here and throughout this paper, we make use of the function $\Phi_M(x)$ which is defined for $M \neq 0$ to be
$$\Phi_M(x) \doteq \frac{e^{Mx} - 1 - Mx}{M^2}.$$

[2]These modifications enable **Hedge** to handle gains (rewards in $[0, M]$) rather than losses (rewards in $[-1, 0]$). Note that we also allow rewards larger than 1. These changes are necessary to use **Hedge** as a building block in the partial information game.

Theorem 3.1 *For $\eta > 0$, and for any sequence of reward vectors $\mathbf{x}(1), \ldots, \mathbf{x}(T)$ with $x_i(t) \in [0, M]$, $M > 0$, the probability vectors $\mathbf{p}(t)$ computed by* **Hedge** *satisfy*

$$\sum_{t=1}^{T} \sum_{i=1}^{K} p_i(t) x_i(t) \geq \sum_{t=1}^{T} x_j(t) - \frac{\ln K}{\eta} - \frac{\Phi_M(\eta)}{\eta} \sum_{t=1}^{T} \sum_{i=1}^{K} p_i(t) x_i(t)^2$$

for all actions $j = 1, \ldots, K$.

In the special case that $M = 1$, we can replace $\sum_{t=1}^{T} \sum_{i=1}^{K} p_i(t) x_i(t)^2$ with its upper bound $\sum_{t=1}^{T} \sum_{i=1}^{K} p_i(t) x_i(t)$ to get the following lower bound on the gain of **Hedge**.

Corollary 3.2 *For $\eta > 0$, and for any sequence of reward vectors $\mathbf{x}(1), \ldots, \mathbf{x}(T)$ with $x_i(t) \in [0, 1]$, the probability vectors $\mathbf{p}(t)$ computed by* **Hedge** *satisfy*

$$\sum_{t=1}^{T} \mathbf{p}(t) \cdot \mathbf{x}(t) \geq \frac{\eta \max_j \sum_{t=1}^{T} x_j(t) - \ln K}{e^\eta - 1}.$$

Note that $\mathbf{p}(t) \cdot \mathbf{x}(t) = \mathbf{E}_{i_t}[x_{i_t}(t) \mid i_1, \ldots, i_{t-1}]$, so this corollary immediately implies the lower bound:

$$\begin{aligned} \mathbf{E}[G_{\mathbf{Hedge}}] = \mathbf{E}\left[\sum_{t=1}^{T} x_{i_t}(t)\right] &= \mathbf{E}\left[\sum_{t=1}^{T} \mathbf{p}(t) \cdot \mathbf{x}(t)\right] \\ &\geq \frac{\eta \mathbf{E}\left[\max_j \sum_{t=1}^{T} x_j(t)\right] - \ln K}{e^\eta - 1} \\ &\geq \frac{\eta E G_{\max} - \ln K}{e^\eta - 1}. \end{aligned}$$

In particular, it can be shown that if we choose $\eta = \ln(1 + \sqrt{2(\ln K)/T})$ then **Hedge** suffers regret at most $\sqrt{2T \ln K}$ in the full information game, i.e.,

$$\mathbf{E}[G_{\mathbf{Hedge}}] \geq E G_{\max} - \sqrt{2T \ln K}.$$

To prove Theorem 3.1, we will use the following inequality.

Lemma 3.3 *For all $\eta > 0$, for all $M \neq 0$ and for all $x \leq M$:*

$$e^{\eta x} \leq 1 + \eta x + \Phi_M(\eta) x^2.$$

Proof. It suffices to show that the function

$$f(x) \doteq \frac{e^x - 1 - x}{x^2}$$

is nondecreasing since the inequality $f(\eta x) \leq f(\eta M)$ immediately implies the lemma. (We can make f continuous with continuous derivatives by defining $f(0) = 1/2$.) We need to show that the first derivative $f'(x) \geq 0$, for which it is sufficient to show that

$$g(x) \doteq \frac{x^3 f'(x)}{e^x + 1} = x - 2\left(\frac{e^x - 1}{e^x + 1}\right)$$

is nonnegative for positive x and nonpositive for negative x. This can be proved by noting that $g(0) = 0$ and that g's first derivative

$$g'(x) = \left(\frac{e^x - 1}{e^x + 1}\right)^2$$

is obviously nonnegative. $\square$

Proof of Theorem 3.1. Let $W_t = \sum_{i=1}^{K} \exp(\eta G_i(t-1))$. By definition of the algorithm, we find that, for all $1 \le t \le T$,

$$\begin{aligned} \frac{W_{t+1}}{W_t} &= \sum_{i=1}^{K} \frac{\exp(\eta G_i(t-1)) \exp(\eta x_i(t))}{W_t} \\ &= \sum_{i=1}^{K} p_i(t) \exp(\eta x_i(t)) \\ &\le 1 + \eta \sum_{i=1}^{K} p_i(t) x_i(t) + \Phi_M(\eta) \sum_{i=1}^{K} p_i(t) x_i(t)^2 \end{aligned}$$

using Lemma 3.3. Taking logarithms and summing over $t = 1, \ldots, T$ yields

$$\begin{aligned} \ln \frac{W_{T+1}}{W_1} &= \sum_{t=1}^{T} \ln \frac{W_{t+1}}{W_t} \\ &\le \sum_{t=1}^{T} \ln \left(1 + \eta \sum_{i=1}^{K} p_i(t) x_i(t) + \Phi_M(\eta) \sum_{i=1}^{K} p_i(t) x_i(t)^2 \right) \\ &\le \eta \sum_{t=1}^{T} \sum_{i=1}^{K} p_i(t) x_i(t) + \Phi_M(\eta) \sum_{t=1}^{T} \sum_{i=1}^{K} p_i(t) x_i(t)^2 \end{aligned} \tag{1}$$

since $1 + x \le e^x$ for all x. Observing that $W_1 = K$ and, for any j, $W_{T+1} \ge \exp(\eta G_j(T))$, we get

$$\ln \frac{W_{T+1}}{W_1} \ge \eta G_j(T) - \ln K \,. \tag{2}$$

Combining Equations (1) and (2) and rearranging we obtain the statement of the theorem. □

4 The partial information game

In this section, we move to the analysis of the partial information game. We present an algorithm $\mathbf{Exp3}$ that runs the algorithm $\mathbf{Hedge}$ of Section 3 as a subroutine. ($\mathbf{Exp3}$ stands for "Exponential-weight algorithm for Exploration and Exploitation.")

The algorithm is described in Figure 2. On each trial t, $\mathbf{Exp3}$ receives the distribution vector $\mathbf{p}(t)$ from $\mathbf{Hedge}$ and selects an action i_t according to the distribution $\hat{\mathbf{p}}(t)$ which is a mixture of $\mathbf{p}(t)$ and the uniform distribution. Intuitively, mixing in the uniform distribution is done to make sure that the algorithm tries out all K actions and gets good estimates of the rewards for each. Otherwise, the algorithm might miss a good action because the initial rewards it observes for this action are low and large rewards that occur later are not observed because the action is not selected.

After $\mathbf{Exp3}$ receives the reward $x_{i_t}(t)$ associated with the chosen action, it generates a simulated reward vector $\hat{\mathbf{x}}(t)$ for $\mathbf{Hedge}$. As $\mathbf{Hedge}$ requires full information, all components of this vector must be filled in, even for the actions that were not selected. For the chosen action i_t, we set the simulated reward $\hat{x}_{i_t}(t)$ to $x_{i_t}(t)/\hat{p}_{i_t}(t)$. Dividing the actual gain by the probability that the action was chosen compensates the reward of actions that are unlikely to be chosen. The other actions all receive a simulated reward of zero. This choice of simulated rewards guarantees that the expected simulated gain associated with any fixed action j is equal to the actual gain of the action; that is, $\mathbf{E}_{i_t}[\hat{x}_j(t) \mid i_1, \ldots, i_{t-1}] = x_j(t)$.

We now give the first main theorem of this paper, which bounds the regret of algorithm $\mathbf{Exp3}$.

Algorithm Exp3
Parameters: Reals $\eta > 0$ and $\gamma \in (0, 1]$
Initialization: Initialize $\mathbf{Hedge}(\eta)$.

Repeat for $t = 1, 2, \ldots$ until game ends

1. Get the distribution $\mathbf{p}(t)$ from $\mathbf{Hedge}$.
2. Select action i_t to be j with probability $\hat{p}_j(t) = (1-\gamma)p_j(t) + \frac{\gamma}{K}$.
3. Receive reward $x_{i_t}(t) \in [0, 1]$.
4. Feed the simulated reward vector $\hat{\mathbf{x}}(t)$ back to $\mathbf{Hedge}$, where $\hat{x}_j(t) = \begin{cases} \frac{x_{i_t}(t)}{\hat{p}_{i_t}(t)} & \text{if } j = i_t \\ 0 & \text{otherwise.} \end{cases}$

Figure 2: Algorithm $\mathbf{Exp3}$ for the partial information game.

Theorem 4.1 *For $\eta > 0$ and $\gamma \in (0, 1]$, the expected gain of algorithm* **Exp3** *is at least*

$$\mathbf{E}[G_{\mathbf{Exp3}}] \geq EG_{\max} - \left(\gamma + \frac{K\Phi_{K/\gamma}(\eta)}{\eta}\right) EG_{\max} - \frac{1-\gamma}{\eta} \ln K \ .$$

To understand this theorem, it is helpful to consider a simpler bound which can be obtained by an appropriate choice of the parameters γ and η:

Corollary 4.2 *Assume that $g \geq EG_{\max}$ and that algorithm* **Exp3** *is run with input parameters $\eta = \gamma/K$ and*

$$\gamma = \min\left\{1, \sqrt{\frac{K \ln K}{(e-1)g}}\right\}.$$

Then the expected regret of algorithm **Exp3** *is at most*

$$R_{\mathbf{Exp3}} \leq 2\sqrt{e-1}\sqrt{gK \ln K} \leq 2.63\sqrt{gK \ln K}.$$

Proof. If $g \leq (K \ln K)/(e-1)$, then the bound is trivial since the expected regret cannot be more than g. Otherwise, by Theorem 4.1, the expected regret is at most

$$\left(\gamma + \frac{K\Phi_{K/\gamma}(\eta)}{\eta}\right) g + \frac{\ln K}{\eta} = 2\sqrt{e-1}\sqrt{gK \ln K}.$$

□

To apply Corollary 4.2, it is necessary that an upper bound g on $EG_{\max}$ be available for tuning η and γ. For example, if the number of trials T is known in advance then, since no action can have payoff greater than 1 on any trial, we can use $g = T$ as an upper bound. In Section 6, we give a technique that does not require prior knowledge of such an upper bound.

If the rewards $x_i(t)$ are in the range $[a, b]$, $a < b$, then $\mathbf{Exp3}$ can be used after the rewards have been translated and rescaled to the range $[0, 1]$. Applying Corollary 4.2 with $g = T$ gives the bound

$(b-a)2\sqrt{e-1}\sqrt{TK\ln K})$ on the regret. For instance, this is applicable to a standard loss model where the “rewards” fall in the range $[-1, 0]$.

Proof of Theorem 4.1. By the definition of the algorithm, we have that $\hat{x}_i(t) \le 1/\hat{p}_i(t) \le K/\gamma$. Thus we find, by Theorem 3.1, that for all actions $j = 1, \dots, K$

$$\sum_{t=1}^{T}\sum_{i=1}^{K} p_i(t)\hat{x}_i(t) \ge \sum_{t=1}^{T}\hat{x}_j(t) - \frac{\ln K}{\eta} - \frac{\Phi_{K/\gamma}(\eta)}{\eta}\sum_{t=1}^{T}\sum_{i=1}^{K} p_i(t)\hat{x}_i(t)^2 .$$

Since

$$\sum_{i=1}^{K} p_i(t)\hat{x}_i(t) = p_{i_t}(t)\frac{x_{i_t}(t)}{\hat{p}_{i_t}(t)} \le \frac{x_{i_t}(t)}{1-\gamma} \tag{3}$$

and

$$\sum_{i=1}^{K} p_i(t)\hat{x}_i(t)^2 = p_{i_t}(t)\frac{x_{i_t}(t)}{\hat{p}_{i_t}(t)}\hat{x}_{i_t}(t) \le \frac{\hat{x}_{i_t}(t)}{1-\gamma} , \tag{4}$$

we get that for all actions $j = 1, \dots, K$

$$G_{\mathbf{Exp3}} = \sum_{t=1}^{T} x_{i_t}(t) \ge (1-\gamma)\sum_{t=1}^{T}\hat{x}_j(t) - \frac{1-\gamma}{\eta}\ln K - \frac{\Phi_{K/\gamma}(\eta)}{\eta}\sum_{t=1}^{T}\hat{x}_{i_t}(t) . \tag{5}$$

Note that

$$\hat{x}_{i_t}(t) = \sum_{i=1}^{K}\hat{x}_i(t). \tag{6}$$

We next take the expectation of Equation (5) with respect to the distribution of $\langle i_1, \dots, i_T\rangle$. For the expected value of $\hat{x}_j(t)$, we have:

$$\begin{aligned}
\mathbf{E}[\hat{x}_j(t)] &= \mathbf{E}_{i_1,\dots,i_{t-1}}\left[\mathbf{E}_{i_t}[\hat{x}_j(t) \mid i_1, \dots, i_{t-1}]\right] \\
&= \mathbf{E}_{i_1,\dots,i_{t-1}}\left[\hat{p}_j(t)\cdot\frac{x_j(t)}{\hat{p}_j(t)} + (1-\hat{p}_j(t))\cdot 0\right] \\
&= \mathbf{E}[x_j(t)]. \qquad (7)
\end{aligned}$$

Combining Equations (5), (6) and (7), we find that

$$\mathbf{E}[G_{\mathbf{Exp3}}] \ge (1-\gamma)\sum_{t=1}^{T}\mathbf{E}[x_j(t)] - \frac{1-\gamma}{\eta}\ln K - \frac{\Phi_{K/\gamma}(\eta)}{\eta}\sum_{t=1}^{T}\sum_{i=1}^{K}\mathbf{E}[x_i(t)] .$$

Since $\max_j \sum_{t=1}^{T}\mathbf{E}[x_j(t)] = EG_{\max}$ and $\sum_{t=1}^{T}\sum_{i=1}^{K}\mathbf{E}[x_i(t)] \le K\ EG_{\max}$ we obtain the inequality in the statement of the theorem. □

5 A bound on the regret that holds with high probability

In the last section, we showed that algorithm $\mathbf{Exp3}$ with appropriately set parameters can guarantee an *expected* regret of at most $O(\sqrt{gK\ln K})$. In the case that the adversarial strategy is oblivious (i.e., when the rewards associated with each action are chosen without regard to the player's past actions), we compare the expected gain of the player to $EG_{\max}$, which, in this case, is the actual gain of the best action. However, if the adversary is not oblivious, our notion of expected regret can be very weak.

Consider, for example, a rather benign but nonoblivious adversary which assigns reward 0 to all actions on the first round, and then, on all future rounds, assigns reward 1 to action i_1 (i.e., to whichever action was played by the player on the first round), and 0 to all other actions. In this case, assuming the player chooses the first action uniformly at random (as do all algorithms considered in this paper), the expected total gain of any action is $(T-1)/K$. This means that the bound that we get from Corollary 4.2 in this case will guarantee only that the expected gain of the algorithm is not much smaller than $EG_{\max} = (T-1)/K$. This is a very weak guarantee since, in each run, there is one action whose actual gain is $T-1$. On the other hand, $\mathbf{Exp3}$ would clearly perform much better than promised in this simple case. Clearly, we need a bound that relates the player's gain to the actual gain of the best action *in the same run*.

In this section, we prove such a bound for $\mathbf{Exp3}$. Specifically, let us define the random variable

$$G_i = \sum_{t=1}^{T} x_i(t)$$

to be the actual total gain of action i, and let

$$G_{\max} = \max_i G_i$$

be the actual total gain of the best action i. The main result of this section is a proof of a bound which holds with high probability relating the player's actual gain $G_{\mathbf{Exp3}}$ to $G_{\max}$.

We show that the dependence of the difference $G_{\max} - G_{\mathbf{Exp3}}$ as a function of T is $O(T^{2/3})$ with high probability for an appropriate setting of $\mathbf{Exp3}$'s parameters. This dependence is sufficient to show that the average per-trial gain of the algorithm approaches that of the best action as $T \to \infty$. However, the dependence is significantly worse than the $O(\sqrt{T})$ dependence of the bound on the expected regret proved in Theorem 4.1. It is an open question whether the gap between the bounds is real or can be closed by this or some other algorithm.

For notational convenience, let us also define the random variables

$$\hat{G}_i = \sum_{t=1}^{T} \hat{x}_i(t)$$

and

$$\hat{G}_{\max} = \max_i \hat{G}_i.$$

The heart of the proof of the result in this section is an upper bound that holds with high probability on the deviation of $\hat{G}_i$ from G_i for any action i. The main difficulty in proving such a bound is that the gains associated with a single action in different trials are not independent of each other, but may be dependent through the decisions made by the adversary. However, using martingale theory, we can prove the following lemma:

Lemma 5.1 *Let $\lambda > 0$ and $\delta > 0$. Then with probability at least $1-\delta$, for every action i,*

$$\hat{G}_i \geq \left(1 - \frac{K\Phi_1(\lambda)}{\gamma\lambda}\right) G_i - \frac{\ln(K/\delta)}{\lambda}.$$

Proof. Given in Appendix A. □

Using this lemma, we can prove the main result of this section:

Theorem 5.2 *Let $\eta > 0$, $\gamma \in (0,1]$, $\lambda > 0$ and $\delta > 0$. Then with probability at least $1-\delta$, the gain of algorithm* **Exp3** *is at least*

$$G_{\textbf{Exp3}} \geq G_{\max} - \left(\gamma + \frac{K\Phi_{K/\gamma}(\eta)}{\eta} + \frac{K\Phi_1(\lambda)}{\gamma\lambda}\right) G_{\max} - \frac{1-\gamma}{\eta}\ln K - \frac{\ln(K/\delta)}{\lambda}.$$

Proof. Note first that

$$\sum_{t=1}^{T} \hat{x}_{i_t}(t) = \sum_{t=1}^{T}\sum_{i=1}^{K} \hat{x}_i(t) = \sum_{i=1}^{K} \hat{G}_i \leq K\hat{G}_{\max}.$$

Combining with Equation (5) gives

$$\begin{aligned} G_{\textbf{Exp3}} &\geq \max_j \left[(1-\gamma)\hat{G}_j - \frac{1-\gamma}{\eta}\ln K - \frac{K\Phi_{K/\gamma}(\eta)}{\eta}\hat{G}_{\max}\right] \\ &= \left(1-\gamma-\frac{K\Phi_{K/\gamma}(\eta)}{\eta}\right)\hat{G}_{\max} - \frac{1-\gamma}{\eta}\ln K \\ &\geq \left(1-\gamma-\frac{K\Phi_{K/\gamma}(\eta)}{\eta}\right)\hat{G}_{i} - \frac{1-\gamma}{\eta}\ln K \end{aligned}$$

for all i.

Next, we apply Lemma 5.1 which implies that, with probability at least $1-\delta$, for all actions i,

$$\begin{aligned} G_{\textbf{Exp3}} &\geq \left(1-\gamma-\frac{K\Phi_{K/\gamma}(\eta)}{\eta}\right)\left(\left(1-\frac{K\Phi_1(\lambda)}{\gamma\lambda}\right)G_i - \frac{\ln(K/\delta)}{\lambda}\right) - \frac{1-\gamma}{\eta}\ln K \\ &\geq G_i - \left(\gamma + \frac{K\Phi_{K/\gamma}(\eta)}{\eta} + \frac{K\Phi_1(\lambda)}{\gamma\lambda}\right)G_i - \frac{\ln(K/\delta)}{\lambda} - \frac{1-\gamma}{\eta}\ln K. \end{aligned}$$

Choosing i to be the best action gives the result. □

To interpret this result we give the following simple corollary.

Corollary 5.3 *Let $\delta > 0$. Assume that $g \geq G_{\max}$ and that algorithm* **Exp3** *is run with input parameters $\eta = \gamma/K$ and*

$$\gamma = \min\left\{1, \left(\frac{K\ln(K/\delta)}{b^2 g}\right)^{1/3}\right\}$$

where $b = (e-1)/2$. Then with probability at least $1-\delta$ the regret of algorithm **Exp3** *is at most*

$$R_{\textbf{Exp3}} \leq (b^{4/3} + 4b^{1/3})g^{2/3}(K\ln(K/\delta))^{1/3} \leq 4.62\, g^{2/3}(K\ln(K/\delta))^{1/3}.$$

Proof. We assume that $g \geq b^{-2}K\ln(K/\delta)$ since otherwise the bound follows from the trivial fact that the regret is at most g. We apply Theorem 5.2 setting

$$\lambda = \left(\frac{(\ln(K/\delta))^2}{bKg^2}\right)^{1/3}.$$

Given our assumed lower bound on g, we have that $\lambda \leq 1$ which implies that $\Phi_1(\lambda) \leq \lambda^2$. Plugging into the bound in Theorem 5.2, this implies a bound on regret of

$$\frac{b^{2/3}g^{1/3}K\ln K}{(K\ln(K/\delta))^{1/3}} + 4(bg^2K\ln(K/\delta))^{1/3}.$$

The result now follows by upper bounding $K\ln K$ in the first term by $(K\ln(K/\delta))^{2/3}(gb^2)^{1/3}$ using our assumed lower bound on g. □

Algorithm Exp3.1
Initialization: Let $t = 0$, $c = \dfrac{K \ln K}{e-1}$, and $\hat{G}_i(0) = 0$ for $i = 1, \ldots, K$
Repeat for $r = 0, 1, 2, \ldots$ until game ends

1. Let $S_r = t + 1$ and $g_r = c\, 4^r$.
2. Restart $\mathbf{Exp3}$ choosing γ and η as in Corollary 4.2 (with $g = g_r$), namely, $\gamma = \gamma_r = 2^{-r}$ and $\eta = \eta_r = \gamma_r / K$.
3. **While** $\max_i \hat{G}_i(t) \le g_r - K/\gamma_r$ **do**:
   (a) $t := t + 1$
   (b) Let $\hat{\mathbf{p}}(t)$ and i_t be the distribution and random action chosen by $\mathbf{Exp3}$.
   (c) Compute $\hat{\mathbf{x}}(t)$ from $\hat{\mathbf{p}}(t)$ and observed reward $x_{i_t}(t)$ as in Figure 2.
   (d) $\hat{G}_i(t) = \hat{G}_i(t-1) + \hat{x}_i(t)$ for $i = 1, \ldots, K$.
4. Let $T_r = t$

Figure 3: Algorithm $\mathbf{Exp3.1}$ for the partial information game when a bound on $EG_{\max}$ is not known.

As $g = T$ is an upper bound that holds for any sequence, we get that the dependence of the regret of the algorithm on T is $O(T^{2/3})$.

6 Guessing the maximal reward

In Section 4, we showed that algorithm $\mathbf{Exp3}$ yields a regret of $O(\sqrt{gK \ln K})$ whenever an upper bound g on the total expected reward $EG_{\max}$ of the best action is known in advance. In this section, we describe an algorithm $\mathbf{Exp3.1}$ which does not require prior knowledge of a bound on $EG_{\max}$ and whose regret is at most $O(\sqrt{EG_{\max} K \ln K})$. Along the same lines, the bounds of Corollary 5.3 can be achieved without prior knowledge about $G_{\max}$.

Our algorithm $\mathbf{Exp3.1}$, described in Figure 3, proceeds in *epochs*, where each epoch consists of a sequence of trials. We use $r = 0, 1, 2, \ldots$ to index the epochs. On epoch r, the algorithm "guesses" a bound g_r for the total reward of the best action. It then uses this guess to tune the parameters η and γ of $\mathbf{Exp3}$, restarting $\mathbf{Exp3}$ at the beginning of each epoch. As usual, we use t to denote the current time step.[3] $\mathbf{Exp3.1}$ maintains an estimate

$$\hat{G}_i(t) = \sum_{t'=1}^{t} \hat{x}_i(t')$$

of the total reward of each action i. Since $\mathbf{E}[\hat{x}_i(t)] = \mathbf{E}[x_i(t)]$, this estimate will be unbiased in the sense that

$$\mathbf{E}[\hat{G}_i(t)] = \mathbf{E}\left[\sum_{t'=1}^{t} x_i(t')\right]$$

[3]Note that, in general, this t may differ from the "local variable" t used by **Exp3** which we now regard as a subroutine. Throughout this section, we will only use t to refer to the total number of trials as in Figure 3.

for all i and t. Using these estimates, the algorithm detects (approximately) when the actual gain of some action has advanced beyond g_r. When this happens, the algorithm goes on to the next epoch, restarting $\mathbf{Exp3}$ with a larger bound on the maximal gain.

The performance of the algorithm is characterized by the following theorem which is the main result of this section.

Theorem 6.1 *The regret suffered by algorithm* **Exp3.1** *is at most*

$$\begin{aligned} R_{\mathbf{Exp3.1}} &\leq 8\sqrt{e-1}\sqrt{EG_{\max}K\ln K} + 8(e-1)K + 2K\ln K \\ &\leq 10.5\,\sqrt{EG_{\max}K\ln K} + 13.8\,K + 2K\ln K. \end{aligned}$$

The proof of the theorem is divided into two lemmas. The first bounds the regret suffered on each epoch, and the second bounds the total number of epochs.

As usual, we use T to denote the total number of time steps (i.e., the final value of t). We also define the following random variables: Let R be the total number of epochs (i.e., the final value of r). As in the figure, S_r and T_r denote the first and last time steps completed on epoch r (where, for convenience, we define $T_R = T$). Thus, epoch r consists of trials $S_r, S_r+1, \ldots, T_r$. Note that, in degenerate cases, some epochs may be empty in which case $S_r = T_r + 1$. Let $\hat{G}_{\max}(t) = \max_i \hat{G}_i(t)$ and let $\hat{G}_{\max} = \hat{G}_{\max}(T)$.

Lemma 6.2 *For any action j and for every epoch r, the gain of* **Exp3.1** *during epoch r is lower bounded by*

$$\sum_{t=S_r}^{T_r} x_{i_t}(t) \geq \sum_{t=S_r}^{T_r} \hat{x}_j(t) - 2\sqrt{e-1}\sqrt{g_r K\ln K}.$$

Proof. If $S_r > T_r$ (so that no trials occur on epoch r), then the lemma holds trivially since both summations will be equal to zero. Assume then that $S_r \leq T_r$. Let $g = g_r$, $\gamma = \gamma_r$ and $\eta = \eta_r$. We use Equation (5) from the proof of Theorem 4.1:

$$\begin{aligned} \sum_{t=S_r}^{T_r} x_{i_t}(t) &\geq (1-\gamma)\sum_{t=S_r}^{T_r}\hat{x}_j(t) - \frac{1-\gamma}{\eta}\ln K - \frac{\Phi_{K/\gamma}(\eta)}{\eta}\sum_{t=S_r}^{T_r}\hat{x}_{i_t}(t) \\ &= \sum_{t=S_r}^{T_r}\hat{x}_j(t) - \gamma\sum_{t=S_r}^{T_r}\hat{x}_j(t) - \frac{\Phi_{K/\gamma}(\eta)}{\eta}\sum_{t=S_r}^{T_r}\sum_{i=1}^{K}\hat{x}_i(t) - \frac{(1-\gamma)\ln K}{\eta} \\ &\geq \sum_{t=S_r}^{T_r}\hat{x}_j(t) - \gamma\sum_{t=1}^{T_r}\hat{x}_j(t) - \frac{\Phi_{K/\gamma}(\eta)}{\eta}\sum_{i=1}^{K}\sum_{t=1}^{T_r}\hat{x}_i(t) - \frac{(1-\gamma)\ln K}{\eta}. \end{aligned}$$

From the definition of the termination condition and since $S_r \leq T_r$, we know that $\hat{G}_i(T_r - 1) \leq g - K/\gamma$. Since $\hat{x}_i(t) \leq K/\gamma$ (by **Exp3**'s choice of $\hat{\mathbf{p}}(t)$), this implies that $\hat{G}_i(T_r) \leq g$ for all i. Thus,

$$\sum_{t=S_r}^{T_r} x_{i_t}(t) \geq \sum_{t=S_r}^{T_r}\hat{x}_j(t) - g\left(\gamma + \frac{K\Phi_{K/\gamma}(\eta)}{\eta}\right) - \frac{(1-\gamma)\ln K}{\eta}.$$

By our choices for η and γ, we get the statement of the lemma. □

The next lemma gives an implicit upper bound on the number of epochs R.

Lemma 6.3 *The number of epochs R satisfies*

$$2^{R-1} \leq \frac{K}{c} + \sqrt{\frac{\hat{G}_{\max}}{c}} + \frac{1}{2}.$$

Proof. If $R = 0$, then the bound holds trivially. So assume $R \geq 1$. Let $z = 2^{R-1}$. Because epoch $R-1$ was completed, by the termination condition,

$$\hat{G}_{\max} \geq \hat{G}_{\max}(T_{R-1}) > g_{R-1} - \frac{K}{\gamma_{R-1}} = c\, 4^{R-1} - K\, 2^{R-1} = cz^2 - Kz. \tag{8}$$

Suppose the claim of the lemma is false. Then $z > K/c + \sqrt{\hat{G}_{\max}/c}$. Since the function $cx^2 - Kx$ is increasing for $x > K/(2c)$, this implies that

$$cz^2 - Kz > c\left(\frac{K}{c} + \sqrt{\frac{\hat{G}_{\max}}{c}}\right)^2 - K\left(\frac{K}{c} + \sqrt{\frac{\hat{G}_{\max}}{c}}\right) = K\sqrt{\frac{\hat{G}_{\max}}{c}} + \hat{G}_{\max},$$

contradicting Equation 8. □

Proof of Theorem 6.1. Using the lemmas, we have that

$$\begin{aligned}
G_{\mathbf{Exp3.1}} = \sum_{t=1}^{T} x_{i_t}(t) &= \sum_{r=0}^{R} \sum_{t=S_r}^{T_r} x_{i_t}(t) \\
&\geq \max_j \sum_{r=0}^{R} \left(\sum_{t=S_r}^{T_r} \hat{x}_j(t) - 2\sqrt{e-1}\sqrt{g_r K \ln K} \right) \\
&= \max_j \hat{G}_j(T) - 2K \ln K \sum_{r=0}^{R} 2^r \\
&= \hat{G}_{\max} - 2K \ln K (2^{R+1} - 1) \\
&\geq \hat{G}_{\max} + 2K \ln K - 8K \ln K \left(\frac{K}{c} + \sqrt{\frac{\hat{G}_{\max}}{c}} + \frac{1}{2} \right) \\
&= \hat{G}_{\max} - 2K \ln K - 8(e-1)K - 8\sqrt{e-1}\sqrt{\hat{G}_{\max} K \ln K}.
\end{aligned} \tag{9}$$

Here, we used Lemma 6.2 for the first inequality and Lemma 6.3 for the second inequality. The other steps follow from definitions and simple algebra.

Let $f(x) = x - a\sqrt{x} - b$ for $x \geq 0$ where $a = 8\sqrt{e-1}\sqrt{K \ln K}$ and $b = 2K \ln K + 8(e-1)K$. Taking expectations of both sides of Equation (9) gives

$$\mathbf{E}[G_{\mathbf{Exp3.1}}] \geq \mathbf{E}[f(\hat{G}_{\max})]. \tag{10}$$

Since the second derivative of f is positive for $x > 0$, f is convex so that, by Jensen's inequality,

$$\mathbf{E}[f(\hat{G}_{\max})] \geq f(\mathbf{E}[\hat{G}_{\max}]). \tag{11}$$

Note that,

$$\mathbf{E}[\hat{G}_{\max}] = \mathbf{E}\left[\max_j \hat{G}_j(T)\right] \geq \max_j \mathbf{E}[\hat{G}_j(T)] = \max_j \mathbf{E}\left[\sum_{t=1}^{T} x_j(t)\right] = EG_{\max}.$$

The function f is increasing if and only if $x > a^2/4$. Therefore, if $EG_{\max} > a^2/4$ then $f(E[\hat{G}_{\max}]) \geq f(EG_{\max})$. Combined with Equations (10) and (11), this gives that $\mathbf{E}[G_{\mathbf{Exp3.1}}] \geq f(EG_{\max})$ which is equivalent to the statement of the theorem. On the other hand, if $EG_{\max} \leq a^2/4$ then, because f is nonincreasing on $[0, a^2/4]$,

$$f(EG_{\max}) \leq f(0) = -b \leq 0 \leq \mathbf{E}[G_{\mathbf{Exp3.1}}]$$

so the theorem follows trivially in this case as well. □

7 A lower bound

In this section, we state an information-theoretic lower bound on the regret of any player, i.e., a lower bound that holds even if the player has unbounded computational power. More precisely, we show that there exists an adversarial strategy for choosing the rewards such that the expected regret of any player algorithm is $\Omega(\sqrt{TK})$. Observe that this does not match the upper bound for our algorithms $\mathbf{Exp3}$ and $\mathbf{Exp3.1}$ (see Corollary 4.2 and Theorem 6.1); it is an open problem to close this gap.

The adversarial strategy we use in our proof is oblivious to the algorithm; it simply assigns the rewards at random according to some distribution, similar to a standard statistical model for the bandit problem. The choice of distribution depends on the number of actions K and the number of iterations T. This dependence of the distribution on T is the reason that our lower bound does not contradict the upper bounds of the form $O(\log T)$ which appear in the statistics literature [12]. There, the distribution over the rewards is fixed as $T \to \infty$.

For the full information game, matching upper and lower bounds of the form $\Theta\left(\sqrt{T \log K}\right)$ were already known [3, 6]. Our lower bound shows that for the partial information game the dependence on the number of actions increases considerably. Specifically, our lower bound implies that no upper bound is possible of the form $O(T^{\alpha}(\log K)^{\beta})$ where $0 \leq \alpha < 1, \beta > 0$.

Theorem 7.1 *For any number of actions $K \geq 2$ and any number of iterations T, there exists a distribution over the rewards assigned to different actions such that the expected regret of any algorithm is at least*

$$\tfrac{1}{20} \min\{\sqrt{KT}, T\}.$$

The proof is given in Appendix B.

The lower bound on the expected regret implies, of course, that for any algorithm there is a particular choice of rewards that will cause the regret to be larger than this expected value.

8 Combining the advice of many experts

Up to this point, we have considered a bandit problem in which the player's goal is to achieve a payoff close to that of the best single action. In a more general setting, the player may have a set of strategies for choosing the best action. These strategies might select different actions at different iterations. The strategies can be computations performed by the player or they can be external advice given to the player by "experts." We will use the more general term "expert" (borrowed from Cesa-Bianchi et al. [3]) because we place no restrictions on the generation of the advice. The player's goal in this case is to combine the advice of the experts in such a way that its total reward is close to that of the best expert (rather than the best single action).

For example, consider the packet-routing problem. In this case there might be several routing strategies, each based on different assumptions regarding network load distribution and using different data to estimate current load. Each of these strategies might suggest different routes at different times, and each might be better in different situations. In this case, we would like to have an algorithm for combining these strategies which, for each set of packets, performs almost as well as the strategy that was best for that set.

Formally, at each trial t, we assume that the player, prior to choosing an action, is provided with a set of N probability vectors $\boldsymbol{\xi}^j(t) \in [0,1]^K$, $j = 1, \ldots, N$, $\sum_{i=1}^{K} \xi_i^j(t) = 1$. We interpret $\boldsymbol{\xi}^j(t)$ as the advice of expert j on trial t, where the ith component $\xi_i^j(t)$ represents the recommended probability of playing action i. (As a special case, the distribution can be concentrated on a single action, which represents a deterministic recommendation.) If the adversary chooses payoff vector $\mathbf{x}(t)$, then the expected reward for expert j (with

Algorithm Exp4
Parameters: Reals $\eta > 0$ and $\gamma \in [0, 1]$
Initialization: Initialize $\mathbf{Hedge}$ (with K replaced by N)

Repeat for $t = 1, 2, \ldots$ until game ends

1. Get the distribution $\mathbf{q}(t) \in [0, 1]^N$ from $\mathbf{Hedge}$.
2. Get advice vectors $\boldsymbol{\xi}^j(t) \in [0, 1]^K$, and let $\mathbf{p}(t) := \sum_{j=1}^N q_j(t)\boldsymbol{\xi}^j(t)$.
3. Select action i_t to be j with probability $\hat{p}_j(t) = (1-\gamma)p_j(t) + \gamma/K$.
4. Receive reward $x_{i_t}(t) \in [0, 1]$.
5. Compute the simulated reward vector $\hat{\mathbf{x}}(t)$ as $\hat{x}_j(t) = \begin{cases} \dfrac{x_{i_t}(t)}{\hat{p}_{i_t}(t)} & \text{if } j = i_t \\ 0 & \text{otherwise.} \end{cases}$
6. Feed the vector $\hat{\mathbf{y}}(t) \in [0, K/\gamma]^N$ to $\mathbf{Hedge}$ where $\hat{y}_j(t) \doteq \boldsymbol{\xi}^j(t) \cdot \hat{\mathbf{x}}(t)$.

Figure 4: Algorithm $\mathbf{Exp4}$ for using expert advice in the partial information game.

respect to the chosen probability vector $\boldsymbol{\xi}^j(t)$) is simply $\boldsymbol{\xi}^j(t) \cdot \mathbf{x}(t)$. In analogy of $EG_{\max}$, we define

$$E\tilde{G}_{\max} \doteq \max_{1 \le j \le N} \mathbf{E}_{i_1,\ldots,i_T}\left[\sum_{t=1}^T \boldsymbol{\xi}^j(t) \cdot \mathbf{x}(t)\right],$$

so that the regret $\tilde{R}_A \doteq \mathbf{E}[G_A] - E\tilde{G}_{\max}$ measures the expected difference between the player's total reward and the total reward of the best expert.

Our results hold for any finite set of experts. Formally, we regard each $\boldsymbol{\xi}^j(t)$ as a random variable which is an arbitrary function of the random sequence of plays $i_1, \ldots, i_{t-1}$ (just like the adversary's payoff vector $\mathbf{x}(t)$). This definition allows for experts whose advice depends on the entire past history as observed by the player, as well as other side information which may be available.

We could at this point view each expert as a "meta-action" in a higher-level bandit problem with payoff vector defined at trial t as $(\boldsymbol{\xi}^1(t) \cdot \mathbf{x}(t), \ldots, \boldsymbol{\xi}^N(t) \cdot \mathbf{x}(t))$. We could then immediately apply Corollary 4.2 to obtain a bound of $O(\sqrt{gN \log N})$ on the player's regret relative to the best expert (where g is an upper bound on $E\tilde{G}_{\max}$). However, this bound is quite weak if the player is combining many experts (i.e., if N is very large). We show below that the algorithm $\mathbf{Exp3}$ from Section 4 can be modified yielding a regret term of the form $O(\sqrt{gK \log N})$. This bound is very reasonable when the number of actions is small, but the number of experts is quite large (even exponential).

Our algorithm $\mathbf{Exp4}$ is shown in Figure 4, and is only a slightly modified version of $\mathbf{Exp3}$. ($\mathbf{Exp4}$ stands for "Exponential-weight algorithm for Exploration and Exploitation using Expert advice.") As before, we use $\mathbf{Hedge}$ as a subroutine, but we now apply $\mathbf{Hedge}$ to a problem of dimension N rather than K. At trial t, we receive a probability vector $\mathbf{q}(t)$ from $\mathbf{Hedge}$ which represents a distribution over strategies. We compute the vector $\mathbf{p}(t)$ as a weighted average (with respect to $\mathbf{q}(t)$) of the strategy vectors $\boldsymbol{\xi}^j(t)$. The vector $\hat{\mathbf{p}}(t)$ is then computed as before using $\mathbf{p}(t)$, and an action i_t is chosen randomly. We define the vector $\hat{\mathbf{x}}(t) \in [0, K/\gamma]^K$ as before, and we finally feed the vector $\hat{\mathbf{y}}(t) \in [0, K/\gamma]^N$ to $\mathbf{Hedge}$ where

$\hat{y}_j(t) \doteq \boldsymbol{\xi}^j(t) \cdot \hat{\mathbf{x}}(t)$. Let us also define $\mathbf{y}(t) \in [0,1]^N$ to be the vector with components corresponding to the gains of the experts: $y_j(t) \doteq \boldsymbol{\xi}^j(t) \cdot \mathbf{x}(t)$.

The simplest possible expert is one which always assigns uniform weight to all actions so that $\xi_i(t) = 1/K$ on each round t. We call this the *uniform expert*. To prove our results, we need to assume that the uniform expert is included in the family of experts.[4] Clearly, the uniform expert can always be added to any given family of experts at the very small expense of increasing N by one.

Theorem 8.1 *For $\eta > 0$ and $\gamma \in (0,1]$, and for any family of experts which includes the uniform expert, the expected gain of algorithm* **Exp4** *is at least*

$$\mathbf{E}[G_{\mathbf{Exp4}}] \geq E\tilde{G}_{\max} - \left(\gamma + \frac{K\Phi_{K/\gamma}(\eta)}{\eta}\right) E\tilde{G}_{\max} - \frac{1-\gamma}{\eta}\ln N\ .$$

Proof. We prove this theorem along the lines of the proof of Theorem 4.1. By Theorem 3.1, for all experts $j = 1, \ldots, N$,

$$\sum_{t=1}^{T} \mathbf{q}(t) \cdot \hat{\mathbf{y}}(t) \geq \sum_{t=1}^{T} \hat{y}_j(t) - \frac{\ln N}{\eta} - \frac{\Phi_{K/\gamma}(\eta)}{\eta} \sum_{t=1}^{T} \sum_{j=1}^{N} q_j(t)\hat{y}_j(t)^2\ .$$

Now

$$\mathbf{q}(t) \cdot \hat{\mathbf{y}}(t) = \sum_{j=1}^{N} q_j(t)\boldsymbol{\xi}^j(t) \cdot \hat{\mathbf{x}}(t) = \mathbf{p}(t) \cdot \hat{\mathbf{x}}(t) \leq \frac{x_{i_t}(t)}{1-\gamma}$$

by Equation (3). Also, similar to Equation (4),

$$\sum_{j=1}^{N} q_j(t)\hat{y}_j(t)^2 = \sum_{j=1}^{N} q_j(t)(\xi^j_{i_t}(t)\hat{x}_{i_t}(t))^2 \leq \hat{x}_{i_t}(t)^2 \sum_{j=1}^{N} q_j(t)\xi^j_{i_t}(t) = p_{i_t}(t)\hat{x}_{i_t}(t)^2 \leq \frac{\hat{x}_{i_t}(t)}{1-\gamma}.$$

Therefore, using Equation (6), for all experts j,

$$G_{\mathbf{Exp4}} = \sum_{t=1}^{T} x_{i_t}(t) \geq (1-\gamma)\sum_{t=1}^{T} \hat{y}_j(t) - \frac{1-\gamma}{\eta}\ln N - \frac{\Phi_{K/\gamma}(\eta)}{\eta} \sum_{t=1}^{T} \sum_{i=1}^{K} \hat{x}_i(t).$$

As before, we take expectations of both sides of this inequality. Note that

$$\mathbf{E}[\hat{y}_j(t)] = \mathbf{E}\left[\sum_{i=1}^{K} \hat{p}_i(t)\xi^j_i(t)\frac{x_i(t)}{\hat{p}_i(t)}\right] = \mathbf{E}\left[\boldsymbol{\xi}^j(t) \cdot \mathbf{x}(t)\right] = \mathbf{E}[y_j(t)].$$

Further,

$$\frac{1}{K}\mathbf{E}\left[\sum_{t=1}^{T} \sum_{i=1}^{K} \hat{x}_i(t)\right] = \mathbf{E}\left[\sum_{t=1}^{T} \frac{1}{K} \sum_{i=1}^{K} x_i(t)\right] \leq \max_j \mathbf{E}\left[\sum_{t=1}^{T} y_j(t)\right] = E\tilde{G}_{\max}$$

since we have assumed that the uniform expert is included in the family of experts. Combining these facts immediately implies the statement of the theorem. □

Analogous versions of the other main results of this paper can be proved in which occurrences of $\ln K$ are replaced by $\ln N$. For Corollary 4.2, this is immediate using Theorem 8.1, yielding a bound on regret of at most $2\sqrt{e-1}\sqrt{gK\ln N}$. For the analog of Lemma 5.1, we need to prove a bound on the difference between $\sum_t y_j(t)$ and $\sum_t \hat{y}_j(t)$ for each expert j which can be done exactly as before replacing δ/K with δ/N in the proof. The analogs of Theorems 5.2 and 6.1 can be proved as before where we again need to assume that the uniform expert is included in the family of experts. The analog of Corollary 5.3 is straightforward.

[4]In fact, we can use a slightly weaker sufficient condition, namely, that the uniform expert is included in the convex hull of the family of experts, i.e., that there exists nonnegative numbers $\alpha_1, \ldots, \alpha_N$ with $\sum_{j=1}^{N} \alpha_j = 1$ such that, for all t and all i, $\sum_{j=1}^{N} \alpha_j \xi^j_i(t) = 1/K$.

9 Nearly optimal play of an unknown repeated game

The bandit problem considered up to this point is closely related to the problem of playing an unknown repeated game against an adversary of unbounded computational power. In this latter setting, a game is defined by an $n \times m$ matrix M. On each trial t, the player (also called the row player) chooses a row i of the matrix. At the same time, the adversary (or column player) chooses a column j. The player then receives the payoff M_{ij}. In repeated play, the player's goal is to maximize its expected total payoff over a sequence of plays.

Suppose in some trial the player chooses its next move i randomly according to a probability distribution on rows represented by a (column) vector $\mathbf{p} \in [0,1]^n$, and the adversary similarly chooses according to a probability vector $\mathbf{q} \in [0,1]^m$. Then the expected payoff is $\mathbf{p}^T \mathrm{M}\mathbf{q}$. Von Neumann's celebrated minimax theorem states that

$$\max_{\mathbf{p}} \min_{\mathbf{q}} \mathbf{p}^T \mathrm{M}\mathbf{q} = \min_{\mathbf{q}} \max_{\mathbf{p}} \mathbf{p}^T \mathrm{M}\mathbf{q} ,$$

where maximum and minimum are taken over the (compact) set of all distribution vectors $\mathbf{p}$ and $\mathbf{q}$. The quantity v defined by the above equation is called the *value* of the game with matrix M. In words, this says that there exists a mixed (randomized) strategy $\overline{\mathbf{p}}$ for the row player that guarantees expected payoff at least v, regardless of the column player's action. Moreover, this payoff is optimal in the sense that the column player can choose a mixed strategy whose expected payoff is at most v, regardless of the row player's action. Thus, if the player knows the matrix M, it can compute a strategy (for instance, using linear programming) that is certain to bring an expected optimal payoff not smaller than v on each trial.

Suppose now that the game M is entirely unknown to the player. To be precise, assume the player knows only the number of rows of the matrix and a bound on the magnitude of the entries of M. The main result of this section is a proof based on the results in Section 4 showing that the player can play in such a manner that its payoff per trial will rapidly converge to the optimal maximin payoff v. This result holds even when the adversary knows the game M and also knows the (randomized) strategy being used by the player.

The problem of learning to play a repeated game when the player gets to see the whole column of rewards associated with the choice of the adversary corresponds to our full-information game. This problem was studied by Hannan [10], Blackwell [2] and more recently by Foster and Vohra [5], Fudenberg and Levin [8] and Freund and Schapire [7]. The problem of learning to play when the player gets to see only the single element of the matrix associated with his choice and the choice of the adversary corresponds to the partial information game which is our emphasis here. This problem was previously considered by Baños [1] and Megiddo [14]. However, these previously proposed strategies are extremely inefficient. Not only is our strategy simpler and much more efficient, but we also are able to prove much faster rates of convergence.

In fact, the application of our earlier algorithms to this problem is entirely straightforward. The player's actions are now identified with the rows of the matrix and are chosen randomly on each trial according to algorithm $\mathbf{Exp3}$, where we tune α and γ as in Corollary 4.2 with $g = T$, where T is the total number of epochs of play.[5] The payoff vector $\mathbf{x}(t)$ is simply $\mathrm{M}_{\cdot j_t}$, the j_t-th column of M chosen by the adversary on trial t.

Theorem 9.1 *Let* $\mathbf{M}$ *be an unknown game matrix in* $[a,b]^{n\times m}$ *with value* v. *Suppose the player, knowing only* a, b *and* n, *uses the algorithm sketched above against any adversary for* T *trials. Then the player's expected payoff per trial is at least*

$$v - 2(b-a)\sqrt{\frac{(e-1)n\ln n}{T}} .$$

[5] If T is not known in advance, the methods developed in Section 6 can be applied.

Proof. We assume that $[a, b] = [0, 1]$; the extension to the general case is straightforward. By Corollary 4.2, we have

$$\begin{aligned} \mathbf{E}\left[\sum_{t=1}^{T} \mathbf{M}_{i_t j_t}\right] &= \mathbf{E}\left[\sum_{t=1}^{T} x_{i_t}(t)\right] \\ &\geq \max_i \mathbf{E}\left[\sum_{t=1}^{T} x_i(t)\right] - 2\sqrt{(e-1)Tn\ln n}. \end{aligned}$$

Let $\overline{\mathbf{p}}$ be a maxmin strategy for the row player such that

$$v = \max_{\mathbf{p}} \min_{\mathbf{q}} \mathbf{p}^T \mathbf{M} \mathbf{q} = \min_{\mathbf{q}} \overline{\mathbf{p}}^T \mathbf{M} \mathbf{q},$$

and let $\mathbf{q}(t)$ be a distribution vector whose j_t-th component is 1. Then

$$\max_i \mathbf{E}\left[\sum_{t=1}^{T} x_i(t)\right] \geq \sum_{i=1}^{n} \overline{p}_i \mathbf{E}\left[\sum_{t=1}^{T} x_i(t)\right] = \mathbf{E}\left[\sum_{t=1}^{T} \overline{\mathbf{p}} \cdot \mathbf{x}(t)\right] = \mathbf{E}\left[\sum_{t=1}^{T} \overline{\mathbf{p}}^T \mathbf{M} \mathbf{q}(t)\right] \geq vT$$

since $\overline{\mathbf{p}}^T \mathbf{M}\mathbf{q} \geq v$ for all $\mathbf{q}$.

Thus, the player's expected payoff is at least

$$vT - 2\sqrt{(e-1)Tn\ln n}.$$

Dividing by T to get the average per-trial payoff gives the result. □

Note that the theorem is independent of the number of columns of M and, with appropriate assumptions, the theorem can be easily generalized to adversaries with an infinite number of strategies. If the matrix M is very large and all entries are small, then, even if M is known to the player, our algorithm may be an efficient alternative to linear programming.

The generality of the theorem also allows us to handle games in which the outcome for given plays i and j is a random variable (rather than a constant M_{ij}). Finally, as pointed out by Megiddo [14], such a result is valid for non-cooperative, multi-person games; the average per-trial payoff of any player using this strategy will converge rapidly to the maximin payoff of the one-shot game.

Acknowledgments

We express special thanks to Kurt Hornik for his advice and his patience when listening to our ideas and proofs for an earlier draft of this paper. We thank Yuval Peres and Amir Dembo for their help regarding the analysis of martingales. Peter Auer and Nicolò Cesa-Bianchi gratefully acknowledge support of ESPRIT Working Group EP 27150, Neural and Computational Learning II (NeuroCOLT II)

References

[1] Alfredo Baños. On pseudo-games. *The Annals of Mathematical Statistics*, 39(6):1932–1945, 1968.

[2] David Blackwell. Controlled random walks. invited address, Institute of Mathematical Statistics Meeting, Seattle, Washington, 1956.

[3] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the Association for Computing Machinery*, 44(3):427–485, May 1997.

[4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[5] Dean P. Foster and Rakesh V. Vohra. A randomization rule for selecting forecasts. *Operations Research*, 41(4):704–709, July–August 1993.

[6] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[7] Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, (to appear).

[8] Drew Fudenberg and David K. Levine. Consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, 19:1065–1089, 1995.

[9] J. C. Gittins. *Multi-armed Bandit Allocation Indices*. John Wiley & Sons, 1989.

[10] James Hannan. Approximation to Bayes risk in repeated play. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume III, pages 97–139. Princeton University Press, 1957.

[11] T. Ishikida and P. Varaiya. Multi-armed bandit problem revisited. *Journal of Optimization Theory and Applications*, 83(1):113–154, October 1994.

[12] T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

[13] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.

[14] N. Megiddo. On repeated games with incomplete information played by non-Bayesian players. *International Journal of Game Theory*, 9(3):157–167, 1980.

[15] J. Neveu. *Discrete-Parameter Martingales*. North Holland, 1975.

[16] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin American Mathematical Society*, 55:527–535, 1952.

[17] Volodimir G. Vovk. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 371–383, 1990.

A Proof of Lemma 5.1

It suffices to prove that, for any fixed action i

$$\mathbf{P}\left\{\hat{G}_i < \left(1 - \frac{K\Phi_1(\lambda)}{\gamma\lambda}\right) G_i - \frac{\ln(K/\delta)}{\lambda}\right\} \leq \frac{\delta}{K} \tag{12}$$

since then the lemma follows by the union bound. Therefore, let us fix i and simplify notation by dropping i subscripts when clear from context.

Let us define the random variable

$$Z_t = \exp\left(\lambda \sum_{t'=1}^{t} (x(t') - \hat{x}(t')) - \frac{K\Phi_1(\lambda)}{\gamma} \sum_{t'=1}^{t} x(t')\right).$$

The main claim of the proof is that $\mathbf{E}[Z_T] \le 1$. Given this claim, we have by Markov's inequality that

$$\mathbf{P}\{Z_T > K/\delta\} \le \delta/K$$

which, by simple algebra, can be seen to be equivalent to Equation (12).

We prove that $\mathbf{E}[Z_t] \le 1$ for $t = 0, \ldots, T$ by induction on t using a method given by Neveu [15][Lemma VII-2-8]. For $t = 0$, $Z_0 = 1$ trivially. To prove the inductive step for $t > 0$, we have that

$$\mathbf{E}_{i_t}[Z_t \mid i_1, \ldots, i_{t-1}] = Z_{t-1} \ \exp\left(-\frac{K\Phi_1(\lambda)x(t)}{\gamma}\right) \ \mathbf{E}_{i_t}[\exp(\lambda(x(t) - \hat{x}(t))) \mid i_1, \ldots, i_{t-1}]. \tag{13}$$

Now by Lemma 3.3, since $x(t) - \hat{x}(t) \le 1$, we have that

$$\begin{aligned}
\mathbf{E}_{i_t}[\exp(\lambda(x(t) - \hat{x}(t))) \mid i_1, \ldots, i_{t-1}] &\le \mathbf{E}_{i_t}[1 + \lambda(x(t) - \hat{x}(t)) + \Phi_1(\lambda)(x(t) - \hat{x}(t))^2 \mid i_1, \ldots, i_{t-1}] \\
&\le 1 + \frac{K\Phi_1(\lambda)}{\gamma} x(t) \\
&\le \exp\left(\frac{K\Phi_1(\lambda)}{\gamma} x(t)\right).
\end{aligned} \tag{14}$$

The second inequality follows from the fact that $\mathbf{E}_{i_t}[\hat{x}(t) \mid i_1, \ldots, i_{t-1}] = x(t)$ and that

$$\begin{aligned}
\mathbf{E}_{i_t}[(x(t) - \hat{x}(t))^2 \mid i_1, \ldots, i_{t-1}] &= \mathbf{E}_{i_t}[\hat{x}(t)^2 \mid i_1, \ldots, i_{t-1}] - x(t)^2 \\
&\le \mathbf{E}_{i_t}[\hat{x}(t)^2 \mid i_1, \ldots, i_{t-1}] \\
&\le \frac{K}{\gamma}\mathbf{E}_{i_t}[\hat{x}(t) \mid i_1, \ldots, i_{t-1}] = \frac{K}{\gamma} x(t).
\end{aligned}$$

The last line uses the fact that $0 \le \hat{x}(t) \le K/\gamma$.

Combining Equations (13) and (14) gives that

$$\mathbf{E}_{i_t}[Z_t \mid i_1, \ldots, i_{t-1}] \le Z_{t-1}$$

(i.e., that the Z_t's forms a supermartingale), and so

$$\mathbf{E}[Z_t] \le \mathbf{E}[Z_{t-1}] \le 1$$

by inductive hypothesis. This completes the proof. □

B Proof Of Theorem 7.1

We construct the random distribution of rewards as follows. First, before play begins, one action I is chosen uniformly at random to be the "good" action. The T rewards $x_I(t)$ associated with the good action are chosen independently at random to be 1 with probability $1/2 + \epsilon$ and 0 otherwise for some small, fixed constant $\epsilon \in (0, 1/2]$ to be chosen later in the proof. The rewards $x_j(t)$ associated with the other actions $j \ne I$ are chosen independently at random to be 0 or 1 with equal odds. Then the expected reward of the best action is at least $(1/2+\epsilon)T$. The main part of the proof below is a derivation of an upper bound on the expected gain of any algorithm for this distribution of rewards.

We write $\mathbf{P}_*\{\cdot\}$ to denote probability with respect to this random choice of rewards, and we also write $\mathbf{P}_i\{\cdot\}$ to denote probability conditioned on i being the good action: $\mathbf{P}_i\{\cdot\} = \mathbf{P}_*\{\cdot \mid I = i\}$. Finally, we write $\mathbf{P}_{unif}\{\cdot\}$ to denote probability with respect to a uniformly random choice of rewards for *all* actions (including the good action). Analogous expectation notation $\mathbf{E}_*[\cdot]$, $\mathbf{E}_i[\cdot]$ and $\mathbf{E}_{unif}[\cdot]$ will also be used.

Let A be the player strategy. Let $r_t = x_{i_t}(t)$ be a random variable denoting the reward received at time t, and let $\mathbf{r}^t$ denote the sequence of rewards received up through trial t: $\mathbf{r}^t = \langle r_1, \ldots, r_t\rangle$. For shorthand, $\mathbf{r} = \mathbf{r}^T$ is the entire sequence of rewards.

Any randomized playing strategy is equivalent to an a-priori random choice from the set of all deterministic strategies. Thus, because the adversary strategy we have defined is oblivious to the actions of the player, it suffices to prove an upper bound on the expected gain of any *deterministic* straregy (this is not crucial for the proof but simplifies the notation). Therefore, we can formally regard the algorithm A as a fixed function which, at each step t, maps the reward history $\mathbf{r}^{t-1}$ to its next action i_t.

As usual, $G_A = \sum_{t=1}^T r_t$ denotes the total reward of the algorithm, and $G_{\max} = \max_j \sum_{t=1}^T x_j(t)$ is the total reward of the best action. Note that, because we here assume an oblivious strategy, $G_{\max}$ and $EG_{\max}$ are the same.

Let N_i be a random variable denoting the number of times action i is chosen by A. Our first lemma bounds the difference between expectations when measured using $\mathbf{E}_i[\cdot]$ or $\mathbf{E}_{unif}[\cdot]$.

Lemma B.1 *Let $f : \{0,1\}^T \to [0, M]$ be any function defined on reward sequences $\mathbf{r}$. Then for any action i,*

$$\mathbf{E}_i[f(\mathbf{r})] \leq \mathbf{E}_{unif}[f(\mathbf{r})] + \frac{M}{2}\sqrt{-\mathbf{E}_{unif}[N_i]\ln(1-4\epsilon^2)}.$$

Proof. We apply standard methods that can be found, for instance, in Cover and Thomas [4]. For any distributions $\mathbf{P}$ and $\mathbf{Q}$, let

$$\|\mathbf{P}-\mathbf{Q}\|_1 \doteq \sum_{\mathbf{r}\in\{0,1\}^T} |\mathbf{P}\{\mathbf{r}\} - \mathbf{Q}\{\mathbf{r}\}|$$

be the variational distance, and let

$$\mathrm{KL}\left(\mathbf{P} \;\|\; \mathbf{Q}\right) \doteq \sum_{\mathbf{r}\in\{0,1\}^T} \mathbf{P}\{\mathbf{r}\} \lg\left(\frac{\mathbf{P}\{\mathbf{r}\}}{\mathbf{Q}\{\mathbf{r}\}}\right)$$

be the Kullback-Liebler divergence or relative entropy between the two distributions. (We use lg to denote $\log_2$.) We also use the notation

$$\mathrm{KL}\left(\mathbf{P}\{r_t \mid \mathbf{r}^{t-1}\} \;\|\; \mathbf{Q}\{r_t \mid \mathbf{r}^{t-1}\}\right) \doteq \sum_{\mathbf{r}^t\in\{0,1\}^t} \mathbf{P}\{\mathbf{r}^t\} \lg\left(\frac{\mathbf{P}\{r_t \mid \mathbf{r}^{t-1}\}}{\mathbf{Q}\{r_t \mid \mathbf{r}^{t-1}\}}\right)$$

for the conditional relative entropy of r_t given $\mathbf{r}^{t-1}$. Finally, for $p, q \in [0, 1]$, we use

$$\mathrm{KL}\left(p \;\|\; q\right) \doteq p\lg\left(\frac{p}{q}\right) + (1-p)\lg\left(\frac{1-p}{1-q}\right)$$

as shorthand for the relative entropy between two Bernoulli random variables with parameters p and q.

We have that

$$
\begin{aligned}
\mathbf{E}_i\left[f(\mathbf{r})\right] - \mathbf{E}_{\textit{unif}}\left[f(\mathbf{r})\right] &= \sum_{\mathbf{r}} f(\mathbf{r})(\mathbf{P}_i\{\mathbf{r}\} - \mathbf{P}_{\textit{unif}}\{\mathbf{r}\}) \\
&\le \sum_{\mathbf{r}:\mathbf{P}_i\{\mathbf{r}\}\ge\mathbf{P}_{\textit{unif}}\{\mathbf{r}\}} f(\mathbf{r})(\mathbf{P}_i\{\mathbf{r}\} - \mathbf{P}_{\textit{unif}}\{\mathbf{r}\}) \\
&\le M \sum_{\mathbf{r}:\mathbf{P}_i\{\mathbf{r}\}\ge\mathbf{P}_{\textit{unif}}\{\mathbf{r}\}} (\mathbf{P}_i\{\mathbf{r}\} - \mathbf{P}_{\textit{unif}}\{\mathbf{r}\}) \\
&= \frac{M}{2}\|\mathbf{P}_i - \mathbf{P}_{\textit{unif}}\|_1. \qquad (15)
\end{aligned}
$$

Also, Cover and Thomas's Lemma 12.6.1 states that

$$
\|\mathbf{P}_{\textit{unif}} - \mathbf{P}_i\|_1^2 \le (2\ln 2)\mathrm{KL}\left(\mathbf{P}_{\textit{unif}} \;\|\; \mathbf{P}_i\right). \qquad (16)
$$

The "chain rule for relative entropy" (Cover and Thomas's Theorem 2.5.3) gives that

$$
\begin{aligned}
\mathrm{KL}\left(\mathbf{P}_{\textit{unif}} \;\|\; \mathbf{P}_i\right) &= \sum_{t=1}^{T} \mathrm{KL}\left(\mathbf{P}_{\textit{unif}}\{r_t \mid \mathbf{r}^{t-1}\} \;\|\; \mathbf{P}_i\{r_t \mid \mathbf{r}^{t-1}\}\right) \\
&= \sum_{t=1}^{T} \left(\mathbf{P}_{\textit{unif}}\{i_t \ne i\}\ \mathrm{KL}\left(\tfrac{1}{2} \;\|\; \tfrac{1}{2}\right) + \mathbf{P}_{\textit{unif}}\{i_t = i\}\ \mathrm{KL}\left(\tfrac{1}{2} \;\|\; \tfrac{1}{2}+\epsilon\right)\right) \\
&= \sum_{t=1}^{T} \mathbf{P}_{\textit{unif}}\{i_t = i\}\left(-\tfrac{1}{2}\lg(1-4\epsilon^2)\right) \\
&= \mathbf{E}_{\textit{unif}}\left[N_i\right]\left(-\tfrac{1}{2}\lg(1-4\epsilon^2)\right). \qquad (17)
\end{aligned}
$$

The second equality can be seen as follows: Regardless of the past history of rewards $\mathbf{r}^{t-1}$, the conditional probability distribution $\mathbf{P}_{\textit{unif}}\{r_t \mid \mathbf{r}^{t-1}\}$ on the next reward r_t is uniform on $\{0,1\}$. The conditional distribution $\mathbf{P}_i\{r_t \mid \mathbf{r}^{t-1}\}$ is also easily computed: Given $\mathbf{r}^{t-1}$, the next action i_t is fixed by A. If this action is not the good action i, then the conditional distribution is uniform on $\{0,1\}$; otherwise, if $i_t = i$, then r_t is 1 with probability $1/2+\epsilon$ and 0 otherwise.

The lemma now follows by combining Equations (15), (16) and (17). □

We are now ready to prove the theorem. Specifically, we show the following:

Theorem B.2 *For any player strategy A, and for the distribution on rewards described above, the expected regret of algorithm A is lower bounded by:*

$$
\mathbf{E}_*\left[G_{\max} - G_A\right] \ge \epsilon\left(T - \frac{T}{K} - \frac{T}{2}\sqrt{-\frac{T}{K}\ln(1-4\epsilon^2)}\right).
$$

Proof. If action i is chosen to be the good action, then clearly the expected payoff at time t is $1/2+\epsilon$ if $i_t = i$ and $1/2$ if $i_t \ne i$:

$$
\begin{aligned}
\mathbf{E}_i\left[r_t\right] &= \left(\tfrac{1}{2}+\epsilon\right)\mathbf{P}_i\{i_t = i\} + \tfrac{1}{2}\mathbf{P}_i\{i_t \ne i\} \\
&= \tfrac{1}{2} + \epsilon\,\mathbf{P}_i\{i_t = i\}.
\end{aligned}
$$

Thus, the expected gain of algorithm A is

$$\mathbf{E}_i[G_A] = \sum_{t=1}^{T} \mathbf{E}_i[r_t] = \frac{T}{2} + \epsilon\, \mathbf{E}_i[N_i]. \tag{18}$$

Next, we apply Lemma B.1 to N_i, which is a function of the reward sequence $\mathbf{r}$ since the actions of player strategy A are determined by the past rewards. Clearly, $N_i \in [0, T]$. Thus,

$$\mathbf{E}_i[N_i] \leq \mathbf{E}_{\mathit{unif}}[N_i] + \frac{T}{2}\sqrt{-\mathbf{E}_{\mathit{unif}}[N_i]\ln(1-4\epsilon^2)}$$

and so

$$\begin{aligned}\sum_{i=1}^{K} \mathbf{E}_i[N_i] &\leq \sum_{i=1}^{K}\left(\mathbf{E}_{\mathit{unif}}[N_i] + \frac{T}{2}\sqrt{-\mathbf{E}_{\mathit{unif}}[N_i]\ln(1-4\epsilon^2)}\right) \\ &\leq T + \frac{T}{2}\sqrt{-TK\ln(1-4\epsilon^2)}\end{aligned}$$

using the fact that $\sum_{i=1}^{K} \mathbf{E}_{\mathit{unif}}[N_i] = T$, which implies that $\sum_{i=1}^{K}\sqrt{\mathbf{E}_{\mathit{unif}}[N_i]} \leq \sqrt{TK}$. Therefore, combining with Equation (18),

$$\mathbf{E}_*[G_A] = \frac{1}{K}\sum_{i=1}^{K}\mathbf{E}_i[G_A] \leq \frac{T}{2} + \epsilon\left(\frac{T}{K} + \frac{T}{2}\sqrt{-\frac{T}{K}\ln(1-4\epsilon^2)}\right).$$

The expected gain of the best action is at least the expected gain of the good action, so $\mathbf{E}_*[G_{\max}] \geq T(1/2+\epsilon)$. Thus, we get that the regret is lower bounded by the bound given in the statement of the theorem. □

For small ϵ, the bound given in Theorem B.2 is of the order

$$\Theta\left(T\epsilon - T\epsilon^2\sqrt{\frac{T}{K}}\right).$$

Choosing $\epsilon = c\sqrt{K/T}$ for some small constant c, gives a lower bound of $\Omega(\sqrt{KT})$. Specifically, the lower bound given in Theorem 7.1 is obtained from Theorem B.2 by choosing $\epsilon = (1/4)\min\{\sqrt{K/T}, 1\}$ and using the inequality $-\ln(1-x) \leq (4\ln(4/3))x$ for $x \in [0, 1/4]$.