JMLR: Workshop and Conference Proceedings vol (2012) 0–26 25th Annual Conference on Learning Theory

Adaptive Regret Minimization in Bounded-Memory Games

author names withheld

Editor: Under Review for COLT 2012

Abstract

Online learning algorithms that minimize regret provide strong guarantees in situations that involve repeatedly making decisions in an uncertain environment, e.g. a driver deciding what route to drive to work every day. While regret minimization has been extensively studied in repeated games, we study regret minimization for a richer class of games called *bounded memory games*. In each round of a two-player bounded memory-m game, both players simultaneously play an action, observe an outcome and receive a reward. The reward may depend on the last m outcomes as well as the actions of the players in the current round. The standard notion of regret for repeated games is no longer suitable because actions and rewards can depend on the history of play. To account for this generality, we introduce the notion of k-*adaptive regret*, which compares the reward obtained by playing actions prescribed by the algorithm against a hypothetical k-*adaptive adversary* with the reward obtained by the best expert in hindsight against the same adversary. Roughly, a hypothetical k-adaptive adversary adapts her strategy to the defender's actions exactly as the real adversary would within each window of k rounds. Our definition is parametrized by a set of experts, which can include both fixed and adaptive defender strategies.

We investigate the inherent complexity of and design algorithms for adaptive regret minimization in bounded memory games of perfect and imperfect information. We prove a hardness result showing that, with imperfect information, any k-adaptive regret minimizing algorithm (with fixed strategies as experts) must be inefficient unless $\mathsf{NP} = \mathsf{RP}$ even when playing against an oblivious adversary. In contrast, for bounded memory games of perfect and imperfect information we present approximate 0-adaptive regret minimization algorithms against an oblivious adversary running in time $n^{O(1)}$.

Keywords: Regret Minimization, Bounded Memory Games, Approximation, Hardness

1. Introduction

Online learning algorithms that minimize regret provide strong guarantees in situations that involve repeatedly making decisions in an uncertain environment. As a concrete example, imagine you are playing rock-paper-scissors against an adversary whose strategy is unknown. A regret minimizing algorithm will guarantee that you will perform as well as the best fixed action (also called an "expert") in hindsight (i.e., rock, paper, or scissor) against any sequence of actions played by the adversary. Indeed, there is a well developed theory for regret minimization in repeated games (see Blum and Mansour (2007) for a survey).

The goal of this paper is to study regret minimization for a richer class of settings. As a motivating example consider a firm that faces a series of different customers or rivals every k rounds (a generalization of the *chain-store game* (Fudenberg and Levine, 2008)). A specific example might be an auctioneer who repeatedly sells goods to different groups of bidders. The auctioneer will want to learn from past experience, even if the bidders are different in every auction (Chakraborty and Stone, 2008). Another motivating involves developing effective auditing strategies in an adversarial environment: Consider a hospital (defender) where a series of different employees or business affiliates (adversary) access patient records for legitimate purposes (e.g., treatment or payment) or inappropriately (e.g., out of curiosity about a family member or for financial gain). The hospital wants to minimize its overall loss by balancing the cost of audits with the risk of externally detected violations.

In these settings, a reasonable strategy for the defender is one that minimizes her regret. Modeling these scenarios as a repeated game of imperfect information is challenging because the games have two additional characteristics that are not captured by a repeated game model: (1) *History-dependent rewards*: The payoff function depends not only on the current outcome but also on previous outcomes. For example, the reputation of a hospital depends on violations detected in the past, not just in the current audit. (2) *History-dependent actions*: Both players may *adapt* their strategies based on history.

We capture this form of history dependence by introducing *bounded memory games*, a subclass of stochastic games (Shapley, 1953). Bounded memory games are an extension of repeated games, in which payoffs may depend on the state of the game. In each round of a two-player bounded-memory-m game, both players simultaneously play an action, observe an outcome and receive a reward. The reward depends only on the outcomes in the last m rounds and the actions of the players in the current round.

In a bounded memory game, the standard notion of regret for a repeated game is not suitable because the adversary may adapt her actions based on the history of play. To account for this generality, we introduce (in Section 3) the notion of *k-adaptive regret*, which compares the reward obtained by playing actions prescribed by the algorithm against a hypothetical *k-adaptive adversary* with the reward obtained by the best expert in hindsight against the same adversary. Roughly, a hypothetical k-adaptive adversary plays exactly the same actions as the real adversary except in the last k rounds where she adapts her strategy to the defender's actions exactly as the real adversary would. When $k = 0$, this definition coincides with the standard definition of an *oblivious adversary* considered in defining regret for repeated games. When $k = \infty$ we get a *fully adaptive adversary*. A k-adaptive adversary is a natural model for the series of different customers in the chainstore game, different bidders in a repeated auction or different employees in a hospital audit. Our definition is parameterized by a set of experts, which can include both fixed and adaptive defender strategies.

An initial unsurprising result is that for the *general class of stochastic games*, there is no k-adaptive regret minimization algorithm, even when $k = 0$. We include this result in the appendix for completeness (Theorem 11).

Next, we investigate the inherent complexity of and design algorithms for adaptive regret minimization in bounded-memory games of perfect and imperfect information. Our results are summarized in Table 1. We prove a hardness result (Section 4; Theorem 2) showing that, with imperfect information, any k-adaptive regret minimizing algorithm (with fixed strategies as experts) must be inefficient unless $\mathsf{NP} = \mathsf{RP}$ even when playing against an oblivious adversary and even when $k = 0$. In fact, the result is even stronger and applies to any γ-approximate k-adaptive regret minimizing algorithm (ensuring that the regret bound converges to γ rather than 0 as the number of rounds $T \to \infty$) for $\gamma < \frac{1}{8n^{\beta}}$ where n is the number of states in the game and $\beta > 0$. Using a slightly stronger complexity-theoretic assumption, we improve this bound to include any value of γ less than $\frac{1}{8\log^2 n}$. Technically, the hardness results are established via reduction from MAX3SAT. In the reduction, each of the exponentially many possible variable assignments corresponds to an expert (fixed strategy); performing as well as the best fixed strategy in the game corresponds to satisfying as many clauses as the best variable assignment. The reduction uses in a critical way the state of the bounded-memory game and the history-dependence of rewards. We also prove that fully adaptive regret minimization algorithms do not exist for bounded-memory games following the impossibility result for stochastic games.

We present an inefficient k-adaptive regret minimizing algorithm by reducing the bounded-memory game to a repeated game. The algorithm is inefficient for bounded-memory games when the number of experts is exponential in the number of states of the game (e.g., if all fixed strategies are experts). However, the algorithm efficiently minimizes k-adaptive regret for repeated games with fixed strategies as experts since such games have only one state. In contrast, for bounded-memory games of perfect information, we present an efficient $n^{O(1/\gamma)}$ time γ-approximate 0-adaptive regret minimization algorithm against an oblivious adversary for any constant $\gamma > 0$ (Section 5;Theorem 5). We also show how this algorithm can be adapted to get an efficient γ-approximate 0-adaptive regret minimization algorithm for bounded-memory games of imperfect information (Section 5;Theorem 6). The main novelty in these algorithms is an implicit weight representation for an exponentially large set of adaptive experts, which includes all fixed strategies.

	Imperfect Information	Perfect Information
Oblivious Regret ($k = 0$)	Hard (Theorem 2) APX (Theorem 6)	APX (Theorem 5)
k-Adaptive Regret ($k \geq 1$)	Hard (Theorem 2)	Hard (Remark 8)
Fully Adaptive Regret ($k = \infty$)	X (Theorem 11)	X (Theorem 11)

Table 1: **Regret Minimization in Bounded Memory Games**
X - no regret minimization algorithm exists
Hard - unless $\mathsf{NP} = \mathsf{RP}$ no regret minimization algorithm is efficiently computable
APX - efficient approximate regret minimization algorithms exist.

Related Work Stochastic games were defined by Shapley (1953). Much of the work on stochastic games has focused on finding and computing equilibria for these games (Shapley, 1953; Mertens and Neyman, 1981).

Regret minimization in stochastic games has not been the subject of much research. Papadimitriou and Tsitsiklis (1999) showed that many natural optimization problems relating to stochastic games are hard. These results don't apply to bounded memory games. Golovin and Krause (2010) recently showed that a simple greedy algorithm can be used when a stochastic optimization problem satisfies a property called adaptive submodularity. In general, bounded memory games do not satisfy this property. Even-Dar et al. (2005) show that regret minimization is possible for a class of stochastic games (Markov Decision Processes) in which the adversary chooses the reward function at each state but does not influence the transitions. They also prove that if the adversary controls the reward function and the transitions, then it is NP-Hard to even approximate the best fixed strategy. Mannor and Shimkin (2003) show that if the adversary completely controls the transition model (a Controlled Markov Process) then it is possible to separate the stochastic game into a series of matrix games and efficiently minimize regret in each matrix game. Bounded-memory games are a different subset of stochastic games where the transitions and rewards are influenced by both players. While our hardness proof shares techniques with Even-Dar et al. (2005), there are significant differences that arise from the bounded-memory nature of the game. We provide a detailed comparison in Section 4.

In a recent paper, Even-Dar et al. (2010) handle a few specific global cost functions related to load balancing. These cost functions depend on history. In their setting, the adversary obliviously plays actions from a joint distribution. In contrast, we consider arbitrary cost functions with bounded dependence on history and adaptive adversaries.

Our efficient regret minimization algorithms represent the weights of the experts implicitly. A related approach is taken by Takimoto and Warmuth (2003) in developing an online shortest path algorithm. In their setting the experts consists of all fixed paths from the source to the destination. In our settings, an additional challenge arises because *experts adapt to adversary actions*. We address this challenge by developing a novel implicit weight representation (see Section 5). Using this implicit weight represent it would have been possible to apply the general framework of Awerbuch and Kleinberg (2008) to achieve approximate regret minimization. However, in the perfect information setting we are able to achieve better regret bounds by simulating the weighted majority algorithm (Littlestone and Warmuth, 1989).

There has been lot of work in regret minimization for repeated games. A closely related work is the regret minimizing audit mechanism of Blocki et al. (2011) that uses a repeated game model for the audit problem. It deals with history-dependent rewards under certain assumptions about the payoff function, but does not consider history-dependent actions. Farias and Megiddo (2006) deal with adaptive adversaries, and not a fixed sequence of adversary actions as is usual in regret minimization for repeated games. They define a general class of adversaries called "flexible" adversaries. A defender playing against a flexible adversary can learn the average expected reward of every expert. Then they present an algorithm that achieves the reward of the best expert asymptotically. Our work differs from theirs in two ways. First, we work with a stochastic game as opposed to a repeated game. Second, our algorithms can handle a sequence of different k-adaptive adversaries instead of learning a single flexible adversary strategy. A single k-adaptive strategy is flexible, but a sequence of k-adaptive adversaries is not.

2. Preliminaries

Stochastic Games Stochastic games are a generalization of repeated games, in which the payoffs depend on the state of play. Formally, a two-player stochastic game between an attacker A and a defender D is given by $(\mathcal{X}_D, \mathcal{X}_A, \Sigma, P, \tau)$, where $\mathcal{X}_A$ and $\mathcal{X}_D$ are the actions spaces for players A and D, respectively, Σ is the state space, $P : \Sigma \times \mathcal{X}_D \times \mathcal{X}_A \rightarrow [0, 1]$ is the payoff function and $\tau : \Sigma \times \mathcal{X}_D \times \mathcal{X}_A \times \{0, 1\}^* \rightarrow \Sigma$ is the randomized transition function linking the different states. Thus, the payoff during round t depends on the current state (denoted σ^t) in addition to the actions of the defender (d^t) and the adversary (a^t).

Bounded-Memory Games Bounded-memory games are a sub-class of stochastic games, in which outcomes and states satisfy certain properties. An outcome of a given round of play is a signal observed by both players (called "public signal" in games Fudenberg and Tirole (1991)). Outcomes depend probabilistically on the actions taken by the players. We use $\mathcal{O}$ to denote the outcome space and $O^t \in \mathcal{O}$ to denote the outcome during round t. We say that a game satisfies *independent outcomes* if O^t is conditionally independent of $\left(O^1, ..., O^{t-1}\right)$ given d^t and a^t. Notice that the defender and the adversary in a game with independent outcomes may still select their actions based on history. However, once those actions have been selected, the outcome is independent of the game history.

A *bounded-memory game with memory* m ($m \in \mathbb{N}$) is a stochastic game with the following properties: (1) The game satisfies independent outcomes, and (2) The states $\Sigma = \mathcal{O}^m$ encode the last m outcomes, i.e., $\sigma^i = \left(O^{i-1}, \ldots, O^{i-m}\right)$. We use $n = |\Sigma|$ to denote the number of states. Note that a repeated game is a bounded-memory-0 game (a bounded-memory game with memory $m = 0$).

A game in which players only observe the outcome O^t after round t but not the actions taken during a round is called an *imperfect information* game. If both players also observe the actions then the game is a *perfect information* game.

The *history* of a game $H = \left(O^1, O^2, \ldots, O^i, \ldots, O^t\right)$, is the sequence of outcomes. We use H_k to denote the k most recent outcomes in the game (i.e., $H_k = \left(O^{t-k+1}; \ldots; O^t\right)$), and $t = |H|$ to denote the total number of rounds played. We use H^i to denote the first i outcomes in a history (i.e., $H^i = \left(O^1, \ldots, O^i\right)$), and $H; H'$ to denote concatenation of histories H and H'.

A *fixed strategy* for the defender in a stochastic game is a function $f : \Sigma \rightarrow \mathcal{X}_D$ mapping each state to a fixed action. F denotes the set of all fixed strategies.

3. Definition of Regret

As discussed earlier, regret minimization in repeated games has received a lot of attention (Blum and Mansour, 2005). Unfortunately, the standard definition of regret in repeated games does not directly apply to stochastic games. In a repeated game, regret is computed by comparing the performance of the defender strategy D with the performance of a fixed strategy f. However, in a stochastic game, the actions of the defender and the adversary in round i influence payoffs in each round for the rest of the game. Thus, it is unclear how to choose a meaningful fixed strategy f as a reference. We solve this conundrum by introducing an adversary-based definition of regret.

3.1. Adversary Model

We define a parameterized class of adversaries called k-adaptive adversaries, where the parameter k denotes the level of adaptiveness of the adversary. Formally, we say that an agent is *k-adaptive* if its strategy $A(H)$ is defined by a function $f : \mathcal{O}^* \times \mathbb{N} \to \mathcal{X}_A$ such that $A(H) = f(H_i, t)$, where $i = t \mod (k+1)$. Recall that H_i is the i most recent outcomes, and $t = |H|$.

As special cases we define an *oblivious adversary* $(k = 0)$ and a *fully adaptive adversary* $(k = \infty)$. Oblivious adversaries essentially play without any memory of the previous outcomes. Fully adaptive adversaries, on the other hand, choose their actions based on the entire outcome history since the start of the game. k-adaptive adversaries lie somewhere in between. At the start of the game, they act as fully adaptive adversaries, playing with the entire outcome history in mind. But, different from fully adaptive adversaries, every k rounds, they "forget" about the entire history of the game and act as if the whole game was starting afresh. As discussed earlier, there are numerous practical instances where k-adaptive adversaries are an appropriate model; for instance, in games in which one player (e.g., a firm) has a much longer length of play than the adversary (e.g., a temporary employee), it may be judicious to model the adversary as k-adaptive. In particular, k-adaptive adversaries are similar to the notion of "patient" players in long-run games discussed by Celentani et al. (1996). Their notion of "fully patient" players correspond to fully adaptive adversaries, "myopic" players correspond to oblivious adversaries, and "not myopic but less patient" players correspond to k-adaptive adversaries.

Another possible adversary definition could be to consider a sliding window of size k as the adversary memory. But, because such an adversary can play actions to remind herself of events in the arbitrary past, her memory is not actually bounded by k, and regret minimization is not possible. See section 8.3 in the appendix for details.

$\mathcal{A}_D^K$ and $\mathcal{A}_A^K$ denote all possible K-adaptive strategies for the defender and adversary, respectively.

3.2. k-Adaptive Regret

Suppose that the defender D and the adversary A have produced history H in a game G lasting T rounds. Let $a^1, ..., a^T$ denote the sequence of actions played by the adversary. In hindsight we can construct a hypothetical k-adaptive adversary A_k as follows:

$$A_k\left(H'\right) = A\left(H^{t-i}; H'_i\right) \ ,$$

where $t = |H'|$ and $i = t \mod (k+1)$. In other words, the hypothetical k-adaptive adversary replicates the plays the real adversary made in the actual game regardless of the strategy of the defender he is playing against, *except* for the last i rounds under consideration where he adapts his strategy to the defender's actions in the same manner the real adversary would. There are two important special cases: (1) Hypothetical Oblivious Adversary (A_0): The hypothetical oblivious adversary plays a fixed sequence of actions always, (2) Hypothetical (Fully) Adaptive Adversary (A_∞): The hypothetical fully adaptive adversary *is* identical to the real adversary.

Abusing notation slightly we write $P(f, A, G, \sigma_0, T)$ to denote the expected payoff the defender would receive over T rounds of G given that the defender plays strategy f, the adversary uses strategy A and the initial state of the bounded-memory game G is σ_0. Similarly, we use $\bar{P}(f, A, G, T)$ to denote the average per-round payoff that the defender would get by playing the strategy f starting from the initial state of G (σ_0), i.e., $\bar{P}(f, A, G, T) = P(f, A, G, \sigma_0, T)/T$.

We use

$$\bar{R}_k\left(D, A, G, T, S\right) = \max_{f \in S} \bar{P}\left(f, A_k, G, T\right) - \bar{P}\left(D, A_k, G, T\right) \ ,$$

to denote the k-*adaptive regret* of the defender strategy D using a fixed set S of experts against an adversary strategy A for T rounds of the game G.

Definition 1 *A defender strategy D using a fixed set S of experts is a* γ-approximate k-adaptive regret minimization algorithm *for the class of games $\mathcal{G}$ if and only if for every adversary strategy A, every $\epsilon > 0$ and every game $G \in \mathcal{G}$ there exists $T' > 0$ such that $\forall T > T'$*

$$\bar{R}_k\left(D, A, G, T, S\right) < \epsilon + \gamma \ .$$

If $\gamma = 0$ then we simply refer to D as a k-regret minimization algorithm. If D runs in time poly $(n, 1/\epsilon)$ we call D efficient.

The k-adaptive regret considers a k-adaptive hypothetical adversary who can adapt within each window of size (at most) $k+1$. Note that the performance of a fixed strategy f against the hypothetical oblivious adversary might be completely different from its performance against the real adversary A. Intuitively, as k increases this measure of regret is more meaningful (as the hypothetical adversary increasingly resembles the real adversary), albeit harder to minimize.

There are two important special cases to consider: $k = 0$ (oblivious regret) and $k = \infty$ (adaptive regret). Observe that if the actual adversary is k-adaptive then the hypothetical adversary A_∞ is same as the hypothetical adversary A_k, and hence $\bar{R}_\infty = \bar{R}_k$. Also, if the actual adversary is oblivious then $\bar{R}_\infty = \bar{R}_0 = \bar{R}_k$. Adaptive regret is the strongest measure of regret.

In this paper $\mathcal{G}$ will typically denote the class of perfect/imperfect information bounded-memory games with memory m. We are interested in expert sets S which contain all of the fixed strategies $F \subseteq S$.

4. Hardness Results

In this section, we show that unless NP = RP no oblivious regret minimization algorithm which uses the fixed strategies F as experts can be efficient in the imperfect information setting. In the appendix (remark 8) we explain how our hardness reduction can be adapted to prove that there is no efficient k-adaptive regret minimization algorithm in the perfect information setting for $k \geq 1$.

Specifically, we consider the subclass of bounded-memory games $\mathcal{G}$ with the following properties: $|\mathcal{O}| = O(1)$, $m = O\left(\log n\right)$, $|\mathcal{X}_A| = O(1)$, $|\mathcal{X}_D| = O(1)$ and imperfect information. Any $G \in \mathcal{G}$ is a game of imperfect information (on round t the defender observes O^t, but not a^t) with $O(n)$ states. Our goal is to prove the following theorem:

Theorem 2 *For any $\beta > 0$ and $\gamma < 1/8n^\beta$ there is no efficient γ-approximate oblivious regret minimization algorithm which uses the fixed strategies F as experts against oblivious adversaries for the class of imperfect information bounded-memory-m games unless* NP = RP.

Given a slightly stronger complexity-theoretic assumption called the randomized exponential time hypothesis (Impagliazzo and Paturi, 2001) we can prove a slightly stronger hardness result. The randomized exponential time hypothesis says that no randomized algorithm running in time $2^{o(n)}$ can solve SAT.

Theorem 3 *Assume that the randomized exponential time hypothesis is true. Then for any $\gamma < 1/\left(8\log^2 n\right)$ there is no efficient γ-approximate oblivious regret minimization algorithm which uses the fixed strategies F as experts against oblivious adversaries for the class of imperfect information bounded-memory-m games.*

The proofs of Theorems 2 and 3 use the fact that it is hard to approximate MAX3SAT within any factor better than $\frac{7}{8}$ (Hastad, 2001). This means that unless $\mathsf{NP} = \mathsf{RP}$ then for every constant $\beta > 0$ and every randomized algorithm S in RP, there exists a MAX3SAT instance ϕ such that the expected number of clauses in ϕ unsatisfied by $S(\phi)$ is $\geq \frac{1}{8} - \beta$ even though there exists an assignment satisfying $(1-\beta)$ fraction of the clauses in ϕ.

We reduce a MAX3SAT formula ϕ with variables $x_1, ..., x_n$ and clauses $C_1, ..., C_\ell$ to a bounded-memory game G described formally below. We provide a high level overview of the game G before describing the details. The main idea is to construct G so that the rewards in G are related to the fraction of clauses of ϕ that are satisfied.

In G, for each variable x there is a state σ_x associated with that variable. The oblivious adversary controls the transitions between variables. This allows the oblivious adversary A_R to partition the game into stages of length n, such that during each stage the adversary causes the game to visit each variable exactly once (each state is associated with a variable). During each stage the adversary picks a clause C at random. In G we have $0, 1 \in \mathcal{X}_D$. Intuitively, the defender chooses assignment $x = 1$ by playing the action 1 while visiting the variable x. The defender receives a reward if and only if he succeeds in satisfying the clause C.

The Game G:
Defender Actions: $\mathcal{X}_D = \{0, 1, 2\}$
Adversary Actions: $\mathcal{X}_A = \{0, 1\} \times \{0, 1, 2, 3\}$
Outcomes and States: Each round i produces two outcomes: observe that these outcomes satisfy the independent outcomes requirement for bounded-memory games. There are $n = 2^{m+1}$ states, where σ^i is the state at round i. Observe that each state encodes the last m outcomes $\tilde{O}$ and the last outcome $\hat{O}^i$. Intuitively, the last m outcomes $\tilde{O}^i$ are used to denote the variable x_i, while $\hat{O}^i$ is 1 if the defender has already received a reward during the current phase.

$$\tilde{O}^i = \boldsymbol{a}^i[1] \quad \text{and} \quad \hat{O}^i = \begin{cases} 1 & \text{if } d^i = 2 \text{ or } d^i = a^i[2]; \\ 0 & \text{otherwise.} \end{cases}$$

$$\sigma^i = \left(\langle \tilde{O}^{i-1}, \ldots, \tilde{O}^{i-m} \rangle, \hat{O}^{i-1}\right) ,$$

The defender actions $0, 1$ correspond to the truth assignments $0, 1$. The defender receives a reward for the correct assignment. The defender is punished if he attempts to obtain a reward in any phase after he has already received a reward in that phase. Once the defender has already received a reward he can play the special action 2 to avoid getting punished. The intuitive meaning of the adversary's actions will be explained in Section 4.

If we ignore the outcome $\hat{O}$ then the states form a De Bruijn graph (Good, 1946) where each node corresponds to a variable of ϕ. Notice that the adversary completely controls the outcomes $\tilde{O}$ with the first component of his action $\boldsymbol{a}[1]$. By playing a De Bruijn sequence $S = s_1 ... s_n$ the adversary can guarantee that we repeatedly take a Hamiltonian cycle over states(for an example see Figure 2 in the appendix).

Rewards:[1]
An intuitive interpretation of the reward function is presented in parallel with the adversary strategy.

$$P\left(\sigma^i, d^i, a^i\right) = \begin{cases} -1 & \text{if } \hat{O}^{i-1} = 1 \text{ and } d^i \neq 2 \text{ and } \boldsymbol{a}^i[2] \neq 3; \\ 1 & \text{if } d^i \neq 2 \text{ and } d^i = \boldsymbol{a}^i[2] \text{ and } \hat{O}^{i-1} = 0; \\ 0 & \text{otherwise.} \end{cases}$$

Adversary Strategy The first component of the adversary's action ($\boldsymbol{a}[1]$) controls the transitions between variables. The adversary will play the action $\boldsymbol{a}^i[2] = 1$ (resp. $\boldsymbol{a}^i[2] = 0$) whenever the corresponding variable assignment $x_i = 1$ (resp. $x_i = 0$) satisfies the clause that the adversary chose for the current phase. If neither variable assignment satisfies the clause (if $x_i \notin C$ and $\bar{x}_i \notin C$) then the adversary plays $\boldsymbol{a}^i[2] = 2$. This ensures that a defender can only be rewarded during a round if he satisfies the clause C, which happens when $d^i = \boldsymbol{a}^i[2] = 0$ or 1.

Notice that whenever $\hat{O} = 1$ there is no way to receive a positive reward. The defender may want the game G to return to a state where $\hat{O} = 0$, but unless the adversary plays the special action $\boldsymbol{a}^i[2] = 3$ he is penalized when this happens. The adversary action $\boldsymbol{a}^i[2] = 3$ is a special 'reset phase' action. By playing $\boldsymbol{a}^i[2] = 3$ once at the end of each phase the adversary can ensure that the maximum payoff the defender receives during any phase is 1. See Figure 1 for a formal description of the adversary strategy.

- **Input:** Random string $R \in \{0,1\}^*$
- **Input:** MAX3SAT instance ϕ, with variables $x_1, \ldots, x_{n-1}$, and clauses $C_1, \ldots, C_\ell$.
- **De Bruijn sequence:** $s_0, ..., s_{n-1}$
- **Round** t**:** Set $i \leftarrow t \mod n$.

1. **Select Clause:** If $i = 0$ then select a clause C uniformly at random from $C_1, ..., C_\ell$ using R.
2. **Select Move:**

$$a^i = \begin{cases} (s_i, 3) & \text{if } i = 0; \\ (s_i, 1) & \text{if } x_i \in C; \\ (s_i, 0) & \text{if } \bar{x}_i \in C; \\ (s_i, 2) & \text{otherwise.} \end{cases}$$

Figure 1: Oblivious Adversary: A_R

Analysis At a high level, our hardness argument proceeds as follows: 1. If there is an assignment that satisfies $(1-\beta)$ fraction of the clauses in ϕ, then there is a fixed strategy that performs well in expectation (see Claim 2). 2. If there a fixed strategy that performs well in expectation, then any γ-approximate oblivious regret minimization algorithm will perform well in expectation (see Claim 3). 3. If an efficiently computable strategy D performs well in expectation, then there is an efficiently computable randomized algorithm S to approximate MAX3SAT. This would imply that $\mathsf{NP} = \mathsf{RP}$. The proofs of theorem 2 and theorem 3 can be found in the appendix.

Our hardness reduction is similar to a result from Even-Dar et al. (2005). They consider regret minimization in a Markov Decision Process where the adversary controls the transition model. Their game is not a bounded-memory game; in particular it does not satisfy our *independent outcomes* condition. The current state in their game can depend on the last n actions. In contrast, we consider bounded-memory games with $m = O(\log n)$, so that the current state only depends on the last m actions. This makes it much more challenging to enforce guarantees such as "the defender can only receive a reward once in each window of n rounds"—a property that is used in the hardness proof. The adversary is oblivious so she will not remember this fact, and the game itself cannot record whether a reward was given $m+1$ rounds ago. We circumvented this problem by designing

1. We use payoffs in the range $[-1, 1]$ for ease of presentation. These payoffs can easily be re-scaled to lie in $[0, 1]$.

a payoff function in which the defender is penalized for allowing the game to "forget" when the last reward was given, thus effectively enforcing the desired property.

5. Regret Minimization Algorithms

In section 5.1 we present a reduction from bounded-memory games to repeated games. This reduction can be used to create a k-adaptive regret minimizing algorithm (see section 8.1 in the appendix). This is significant because there is no k-adaptive regret minimization algorithm for the general class of stochastic games. A consequence of Theorem 2 is that when the expert set includes all fixed strategies F we cannot hope for an efficient algorithm unless $\mathsf{NP} = \mathsf{RP}$. In section 5.2 we present an efficient *approximate* 0-adaptive regret minimization algorithm for bounded-memory games of perfect information. The algorithm uses an implicit weight representation to efficiently sample the experts and update their weights. Finaly, we show how this algorithm can be adapted to obtain an efficient approximate 0-adaptive regret minimization algorithm for bounded-memory games of *imperfect* information.

5.1. Reduction to Repeated Games

All of our regret minimization algorithms work by first reducing the bounded-memory game G to a repeated game $\rho(G, K)$. One round of the repeated game $\rho(G, K)$ corresponds to K rounds of G. Before each round of $\rho(G, K)$ both players commit to an adaptive strategy. In $\rho(G, K)$ the reward that the defender gets for playing a strategy $f \in \mathcal{A}_D^K$ is the reward that the defender would have received for using the strategy f for the next K rounds of the actual game G if the initial state were σ_0: $P(f, g, \rho(G, K)) = P(f, g, G, \sigma_0, K)$.

The rewards in $\rho(G, K)$ may be different from the actual rewards in G because the initial state before each K rounds might not be σ_0. In the appendix we show that this difference is small (see claim 4).

The key idea behind our k-adaptive regret minimization algorithm BW is to reduce the original bounded-memory game to a repeated game $\rho(G, K)$ of imperfect information ($K \equiv 0 \mod k$). In particular we obtain the regret bound in Theorem 4. Details and proofs can be found in the appendix.

Theorem 4 *Let G be any bounded-memory-m game with n states and let A be any adversary strategy. After playing T rounds of G against A, $\mathsf{BW}(G, K)$ achieves regret bound*

$$\bar{R}_k(\mathsf{BW}, A, G, T, S) \quad < \quad \frac{m}{T^{1/4}} + 4\frac{\sqrt{N \log N}}{T^{1/4}} \ ,$$

where $N = |S|$ is the number of experts, A is the adversary strategy and K has been chosen so that $K = T^{1/4}$ and $K \equiv 0 \mod k$.

Intuitively, the $m/T^{1/4} = m/K$ term is due to modeling loss from Claim 4 and the other term comes from the standard regret bound of Auer et al. (1995).

5.2. Efficient Approximate Regret Minimization Algorithms

In this section we present EXBW (Efficient approXimate Bounded Memory Weighted Majority), an efficient algorithm to approximately minimize regret against an oblivious adversary in bounded-memory games with perfect information. The set of experts $\mathcal{E}$ used by our algorithms contains the

fixed strategies F as well as all K-adaptive strategies $\mathcal{A}_D^K$ ($K = m/\gamma$). We prove the following theorem

Theorem 5 *Let G be any bounded-memory-m game of perfect information with n states and let A be any adversary strategy. Playing T rounds of G against A,* EXBW *runs in total time $Tn^{O(1/\gamma)}$ and achieves regret bound*

$$\bar{R}_0\left(\mathsf{EXBW}, A, G, T, \mathcal{E}\right) \leq \gamma + O\left(\frac{m}{\gamma}\sqrt{\frac{\frac{m}{\gamma} n \log\left(N\right)}{T}}\right) ,$$

where K has been set to m/γ and $N = \left|\mathcal{A}_D^K\right| = \left(\left|\mathcal{X}_D\right|\right)^{n^{1/\gamma}}$ is the number of K-adaptive strategies.

In particular, for any constant γ there is an efficient γ-approximate 0-adaptive regret minimization algorithm for bounded-memory games of perfect information. We can adapt this algorithm to get EXBWII (Efficient approXimate Bounded Memory Weighted Majority for Imperfect Information Games), an efficient approximate 0-adaptive regret minimization algorithm for games of imperfect information using a sampling strategy described in the proof of theorem 6.

Theorem 6 *Let G be any bounded-memory-m game of imperfect information with n states and let A be any adversary strategy. There is an algorithm* EXBWII *that runs in total time $Tn^{O(1/\gamma)}$ playing T rounds of G against A, and achieves regret bound*

$$\bar{R}_0\left(\mathsf{EXBWII}, A, G, T, \mathcal{E}\right) \leq 2\gamma + O\left(\frac{mn^{1/\gamma}}{\gamma^2}\sqrt{\frac{\frac{mn^{1/\gamma}}{\gamma} n \log\left(N\right)}{T}}\right) .$$

where K has been set to m/γ and $N = \left|\mathcal{A}_D^K\right| = \left(\left|\mathcal{X}_D\right|\right)^{n^{1/\gamma}}$ is the number of K-adaptive strategies.

The regret bound of Theorem 5 is simply the regret bound achieved by the standard weighted majority algorithm (Littlestone and Warmuth, 1989) plus the modeling loss term from Claim 4. The main challenge is to provide an efficient simulation of the weighted majority algorithm. There are an exponential number of experts so no efficient algorithm can explicitly maintain weights for each of these experts. To simulate the weighted majority algorithm EXBW implicitly maintains the weight of each expert.

To simulate the weighted majority algorithm we must be able to *efficiently sample* from our weighted set of experts (see $\mathbf{Sample}\left(\mathcal{E}\right)$) and efficiently update the weights of each expert in the set after each round of $\rho\left(G, K\right)$ (see update weight stage of EXBW).

Meet the Experts Instead of using F as the set of experts, EXBW uses a larger set of experts $\mathcal{E}$ ($F \subset \mathcal{E}$). Recall that a K-adaptive strategy is a function f mapping the K most recent outcomes H_K to actions. We use a set of K-adaptive strategies $E = \left\{f_\sigma : \sigma \in \Sigma\right\} \subset \mathcal{A}_D^K$ to define an expert E in $\rho\left(G, K\right)$: if the current state of the real bounded-memory game G is σ then E uses the K-adaptive strategy f_σ in the next round of $\rho\left(G, K\right)$ (i.e., the next K rounds of G). $\mathcal{E}$ denotes the set of all such experts.

Maintaining Weights for Experts Implicitly To implicitly maintain the weights of each expert $E \in \mathcal{E}$ we use the concept of a game trace. We say that a game trace $p = \sigma, d^1, O^1, ..., d^{i-1}, O^{i-1}, d^i$ is consistent with an expert E if $f_\sigma\left(O^1, ..., O^{j-1}\right) = d^j$ for each j. We define the set $\mathcal{C}(E)$ to be the set of all such consistent traces of maximum length K and $\mathcal{C} = \bigcup_{E \in \mathcal{E}} \mathcal{C}(E)$ denotes the set of all traces consistent with some expert $E \in \mathcal{E}$. EXBW maintains a weight w_p on each trace $p \in \mathcal{C}$. The weight of an expert E is then defined to be $W_E = \prod_{p \in \mathcal{C}(E)} w_p$.

Given adversary actions $\boldsymbol{a} = a_1, ..., a_K$ and a trace $p = \sigma, d^1, O^1, ..., d^{i-1}, O^{i-1}, d^i$ we define $\mathcal{R}(\boldsymbol{a}, \sigma', p)$.

$$\mathcal{R}\left(\boldsymbol{a}, \sigma', p\right) = \begin{cases} 0 & \text{if } \sigma \neq \sigma'; \\ \prod_{j<i} \Pr\left[O^j \,|\, a^j, d^j\right] & \text{otherwise;} \end{cases}$$

Intuitively, $\mathcal{R}(\boldsymbol{a}, \sigma', p)$ is the probability that each outcome of p would have occurred given the adversary actions were $\boldsymbol{a}$ and the initial state was σ'. We use $\ell(p, \boldsymbol{a}, \sigma')$ to denote the payment that the defender received for playing d^i (the last action in p). Formally $\ell(p, \boldsymbol{a}, \sigma') = P\left(\sigma_p^f, d^i, a^i\right) \mathcal{R}(\boldsymbol{a}, \sigma', p)$, where σ_p^f denotes the state reached following the trace p (after observing outcomes $O^1, ..., O^{i-1}$ starting from σ_0) and d^i is the final defender action in the trace. Notice that in the imperfect information setting the defender could not compute ℓ because he would not observe the adversary's actions $\boldsymbol{a}$.

Updating Weights Efficiently While updating weights EXBW maintains the invariant that $w_p = \beta^{\sum_{j=1}^{T/K} \ell\left(p, \boldsymbol{a}^j, \sigma^{jK}\right)}$, where σ^{jK} is the state of G after jK rounds and $\boldsymbol{a}^t$ is the actions the adversary played during the j'th round of $\rho(G, K)$. The standard weighted majority algorithm maintains the invariant that $W_E = \beta^{\sum_{j=1}^{T/K} P\left(E, \boldsymbol{a}^t, \rho(G,K)\right)}$. In the appendix EXBW also maintains this invariant(see claim 5).

Sampling Experts Efficiently We can also efficiently sample from $\mathcal{E}$ using dynamic programming (see $\textbf{Sample}(\mathcal{E})$). Using the notation $p \sqsubset p'$ for p' extends p we can define $\hat{w}_p$. Intuitively, $\hat{w}_{p;O;d}$ represents the weight of the action d from history $p; O$. Using dynamic programming we can efficiently compute $\hat{w}_p$ for each trace p because there are only $n^{O(1/\gamma)}$ such traces. Using the weights $\hat{w}_p$ we can efficiently sample from $\mathcal{E}$. We use $p; O; d$ to denote a new game trace which contains all of the outcomes/actions in p appended with O and d.

$$\hat{w}_p = \sum_{E: p \in \mathcal{C}(E)} \prod_{p' \in \mathcal{C}(E) \wedge p \sqsubset p'} w_{p'}$$

Algorithm: EXBW (γ, G)

- **Initialize:** $K = m/\gamma$
- **Construct:** $\rho(G, K)$
- **Each Round:**
  1. $\sigma \leftarrow G.CurrentState$
  2. $E \leftarrow \textbf{Sample}(\mathcal{E})$
  3. Play E
  4. Observe adversary actions $\boldsymbol{a} = a^1, ..., a^K$.
  5. **Update Weights:** For each $p \in \mathcal{C}$
     Compute $\ell(p, \boldsymbol{a}, \sigma)$
     Set $w_p \leftarrow w_p \times \beta^{\ell(p, \boldsymbol{a}, \sigma)}$.

Algorithm: $\textbf{Sample}(\mathcal{E})$

- For each trace $p \in \mathcal{C}$ recursively compute $\hat{w}_p$ using the formula:

$$\hat{w}_p = \sum_{O \in \mathcal{O}} \sum_{d \in \mathcal{X}_D} \beta^{\sum_{t=1}^{T} \ell\left(p; O; d, \boldsymbol{a}^t, \sigma^{Kt}\right)} \hat{w}_{p;O;d}\,.$$

- **Build Strategy** E: For each $p \in \mathcal{C}$ and $O \in \mathcal{O}$, randomly select $d \in \mathcal{X}_D$

$$\Pr\left[d \,|\, p,\ O\right] = \frac{\hat{w}_{p;O;d}}{\sum_{d' \in \mathcal{X}_D} \hat{w}_{p;O;d'}}\,.$$

- E play d any time it observes history $p; O$.

In the appendix we prove that $\mathbf{Sample}\left(\mathcal{E}\right)$ outputs each expert E with probability proportional to W_E (see claim 6). Given $\mathbf{Sample}\left(\mathcal{E}\right)$ it is straightforward to simulate the standard weighted majority algorithm. To update weights EXBW simply loops through all traces $p \in \mathcal{C}$ applying the update rule $w_p = w_p \times \beta^{\ell\left(p, \boldsymbol{a}^t, \sigma^{tK}\right)}$, where β is a learning parameter we tune later. The full proof of Theorem 5 can be found in the appendix.

At a high level our algorithm is similar to the online shortest path algorithm developed by Takimoto and Warmuth (2003). In their work, they consider the set of all source-destination paths in a graph as experts. Since there are exponentially many paths they also maintain the weights of the experts implicitly. In their setting, the defender completely controls the chosen path. In contrast, our experts adapt to adversary actions. The challenge was constructing a new implicit weight representation which works for K-adaptive strategies.

Using this implicit weight representation we could have also used the general barycentric spanner approach to online linear optimization developed by Awerbuch and Kleinberg (2008) to design a γ-approximate 0-adaptive regret minimization algorithm running in time $n^{O(1/\gamma)}$. However, we are able to achieve better regret bounds in theorem 5 by simulating the weighted majority algorithm. (Awerbuch and Kleinberg, 2008, Theorem 2.8) achieves average regret bound $O\left(Md^{5/3}/T^{1/3}\right)$, where d is the dimension of the problem space and M is a bound on the cost vectors. By comparison our regret bounds in Theorems 5 and 6 tend to 0 with $1/\sqrt{T}$. In our setting, the dimension of the problem space is $d = O\left(n^{(1/\gamma)}\right)$ (the number of nodes in the decision tree), and $M = K = m/\gamma$ is the upper bound on the cost vector in each round of $\rho\left(G, K\right)$. The average regret bound would be $O\left(\frac{m}{\gamma} n^{5/(3\gamma)}/T^{1/3}\right)$. the regret bound is proportional to $\sqrt{n^{1/\gamma}/T}$. By comparison Theorem 5 has a $\sqrt{n^{1/\gamma}}$ in the numerator.

The standard regret minimization trick for dealing with imperfect information in a repeated game is to break the game up into phases and perform random sampling in each round to estimate the cost of each expert and update weights. The challenge in adapting EXBW is that there are exponentially many experts in $\mathcal{E}$. Our key idea was to estimate $\ell\left(p, \boldsymbol{a}, \sigma\right)$ for each $p \in \mathcal{C}$ so there are only $n^{O(1/\gamma)}$ samples to take in each phase. We can then update the implicit weight representation using the estimated values $\ell\left(p, \boldsymbol{a}, \sigma\right)$.

6. Open Questions

In this paper, we defined a new class of games called bounded-memory games, introduced several new notions of regret, and presented hardness results and algorithms for regret minimization in this subclass of stochastic games. Because both the games and the notions of regret we study in this paper rely on novel definitions, they raise a number of interesting open problems: (1) To what extent can the hardness results of Theorems 2 and 3 be further improved? ($\gamma = 1/\log n$?) Could similar hardness results apply to games with perfect information? (2) Is there an efficient *non-approximate* oblivious regret minimization algorithm for bounded-memory games with perfect information? (3) Is there a γ-approximate oblivious regret minimization algorithm with running time $n^{o(1/\gamma)}$? For example, could one design a γ-approximate oblivious regret minimization algorithm with running time $n^{-\log\gamma}$? (4) For repeated games $(m = 0)$ is there an efficient γ-approximate k-adaptive regret minimization algorithm if we use $\mathcal{A}_D^K$ as our set of experts $(K = \log n)$?

References

P. Auer, N. Cesa-Bianchi, Y. Freund, and R.E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *FOCS*, page 322. Published by the IEEE Computer Society, 1995.

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

J. Blocki, N. Christin, A. Datta, and A. Sinha. Regret minimizing audits: A learning-theoretic basis for privacy protection. In *Computer Security Foundations Symposium, 2011. CSF'11. 24th IEEE*, pages 312–327. IEEE, 2011.

A. Blum and Y. Mansour. From external to internal regret. *Learning Theory*, pages 621–636, 2005.

A. Blum and Y. Mansour. Learning, regret minimization, and equilibria. *Algorithmic Game Theory*, pages 79–102, 2007.

M. Celentani, D. Fudenberg, D.K. Levine, and W. Pesendorfer. Maintaining a reputation against a patient opponent. *Econometrica*, 64:691–704, 1996.

D. Chakraborty and P. Stone. Online multiagent learning against memory bounded adversaries. *Machine Learning and Knowledge Discovery in Databases*, pages 211–226, 2008.

E. Even-Dar, S.M. Kakade, and Y. Mansour. Experts in a Markov decision process. In *Advances in neural information processing systems 17: proceedings of the 2004 conference*, page 401. The MIT Press, 2005.

E. Even-Dar, S. Mannor, and Y. Mansour. Learning with global cost in stochastic environments. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, 2010.

Daniela Pucci De Farias and Nimrod Megiddo. Combining expert advice in reactive environments. *J. ACM*, 53:762–799, September 2006. ISSN 0004-5411.

D. Fudenberg and D. K. Levine. *A Long-run Collaboration on Long-run Games*. World Scientific Publishing Company, 2008.

D. Fudenberg and J. Tirole. *Game theory*. MIT Press, 1991.

Daniel Golovin and Andreas Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization. *CoRR*, abs/1003.3967, 2010.

I. J. Good. Normal recurring decimals. *Journal of the London Mathematical Society*, 1(3):167, 1946. ISSN 0024-6107.

J. Hastad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001.

R. Impagliazzo and R. Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

N. Littlestone and M.K. Warmuth. The weighted majority algorithm. In *Proceedings of FOCS*, pages 256–261, 1989.

S. Mannor and N. Shimkin. The empirical bayes envelope and regret minimization in competitive markov decision processes. *Mathematics of Operations Research*, pages 327–345, 2003.

J.F. Mertens and A. Neyman. Stochastic games. *International Journal of Game Theory*, 10(2): 53–66, 1981.

A. Neyman and S. Sorin. *Stochastic games and applications*. Springer, 2003.

CH Papadimitriou and JN Tsitsiklis. The complexity of optimal queueing network control. inmathematics of operations research, 1999.

L.S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences of the United States of America*, 39(10):1095, 1953.

E. Takimoto and M.K. Warmuth. Path kernels and multiplicative updates. *The Journal of Machine Learning Research*, 4:773–818, 2003.

Jia Yuan Yu and Shie Mannor. (Survey) Online Learning in Stochastic Games and Markov Decision Processes. http://www.cim.mcgill.ca/ jiayuan/survey08.pdf.

7. Hardness Reduction: Proof of Claims

This section contains the proofs of the lemmas and theorems from section 4.

Claim 1 *Fix a polynomial $p(\cdot)$ and let $\alpha = n \cdot E_R\left[\bar{P}\left(D, A_R, G, T\right)\right]$, where $T = p(n)$ and D is any polynomial time computable strategy. There is a polynomial time randomized algorithm S which satisfies α fraction of the clauses from ϕ in expectation.*

Proof Let $p(\circ)$ be given such that $T\left(D\right) \leq p(n)$ and set

$$\alpha = n \times E_R\left[\bar{P}\left(D, A_R, G, T\right)\right] \ .$$

We present S (Algorithm 1) - an algorithm to recover the variable assignment. S runs in time

$$T(S) = O\left(p(n)^2\right) \ .$$

During the simulation we present D with (potentially) false history in each stage, where the defender always thinks he hasn't satisfied the clause C. Let $\mathbf{Y}_j$ be the expected fraction of clauses satisfied in stage j of the simulation. We define the random variable $\mathbf{X}_j$ to be the reward D earns in stage j in the actual game. Observe that the game is structured so that two rewards during the same stage must be separated by a penalty. When the defender receives a reward the outcome $\hat{O}^{t-1}$ is produced. If the defender wishes to avoid an offsetting penalty then he must keep producing the outcome $\hat{O}^{t-1}$ by playing $d^t = 2$, preventing him from receiving an award for the rest of the stage. The maximum payout a defender strategy D can receive during any stage is 1 so $\mathbf{X}_j \in \{0, 1\}$. Because of imperfect information the defender cannot learn any information about the clause the adversary has selected. We have

$$E[\mathbf{X}_j] = \Pr[\mathbf{X}_j = 1] = E[\mathbf{Y}_j] \ .$$

In particular

$$\alpha = \frac{n}{T} \sum_{j=1}^{T/n} E[\mathbf{Y}_j] \ ,$$

so there exists a round j such that $E[\mathbf{Y}_j] \geq \alpha$. Let $\mathbf{Y}$ denote the number of clauses satisfied by S, then

$$\mathbf{Y} = \max_j \mathbf{Y}_j \ ,$$

so we have

$$E[\mathbf{Y}] \geq \alpha \ .$$

■

Claim 2 *Suppose that there is a variable assignment that satisfies $(1-\beta) \cdot \ell$ of the clauses in ϕ. Then there is a fixed strategy f such that $E_R\left[\bar{P}\left(f, A_R, G, n\right)\right] \geq (1-\beta)/n$, where R is used to denote the random coin tosses of the oblivious adversary.*

Proof Let $x_1*, ..., x_{n-1}*$ be the assignment that satisfies at least $(1-\beta)$ fraction of the clauses and let $s_0, ..., s_{n-1}$ be the De Bruijn sequence played by the adversary. x_n is an additional variable that is not in any of the clauses. Then the on round t we have

$$\sigma^t = \left(\langle s_{i-1 \mod n}, ..., s_{i-m \mod n}\rangle, \hat{O}^{t-1}\right) \ ,$$

where $i = t \mod n$ so both these states are associated with the variable x_i. For $0 \leq i < n$ we set

$$f\left(\langle s_{i-1 \mod n}, ..., s_{i-m \mod n}\rangle, 0\right) = x_i * \ .$$

To avoid taking a penalty we set

$$f\left(\langle s_{i-1 \mod n}, ..., s_{i-m \mod n}\rangle, 1\right) = 2 \ ,$$

for $0 < i < n$. For $i = 0$ we set

$$f\left(\langle s_{i-1 \mod n}, ..., s_{i-m \mod n}\rangle, 1\right) = 0 \ ,$$

to produce the outcome $\hat{O}^t = 0$ (recall that the adversary will play $a^t = (s_0, 3)$ whenever $t \equiv 0 \mod n$ so we can avoid the penalty). The fixed strategy f will receive reward 1 in stage j if and only if $x_1*, ..., x_{n-1}*$ satisfies the clause C_j chosen in stage j.

$$E_R\left[\bar{P}\left(f, A_R, G, n\right)\right] \quad \geq \quad \frac{(1-\beta)}{n} \tag{1}$$

■

Claim 3 *Suppose that D is an $\left(\frac{1}{8n} - \frac{3\beta}{n}\right)$-approximate oblivious regret minimization algorithm against the class of oblivious adversaries and there is a variable assignment that satisfies $(1-\beta)$ fraction of the clauses in ϕ. Then for $T = poly(n)$*

$$E_R\left[\bar{P}\left(D, A_R, G, T\right)\right] \geq \frac{7}{8n} + \frac{\beta}{n}\,,$$

where R is used to denote the random coin tosses of the oblivious adversary.

Proof By Claim 2 there is a fixed strategy with

$$E_R\left[\bar{P}\left(D, A_R, G, T\right)\right] \geq \frac{(1-\beta)}{n}\,.$$

Set $\epsilon = \beta/n$, and apply definition 1 to get

$$\bar{P}\left(f, A_R, G, T\right) - \bar{P}\left(D, A_R, G, T\right) \leq \left(\frac{1}{8n} - \frac{3\beta}{n}\right) + \beta/n\,,$$

for any random string R (adversary coin flips). This means that

$$E_R\left[\bar{P}\left(f, A_R, G, T\right)\right] - E_R\left[\bar{P}\left(D, A_R, G, T\right)\right] \leq \left(\frac{1}{8n} - \frac{3\beta}{n}\right) + \frac{\beta}{n}\,.$$

Rearranging terms

$$\begin{aligned} E_R\left[\bar{P}\left(D, A_R, G, T\right)\right] &\geq \frac{(1-\beta)}{n} - \frac{1}{8n} + \frac{2\beta}{n} \\ &= \frac{7}{8n} + \frac{\beta}{n} \end{aligned}$$

■

Before we prove Theorem 2 we will first prove an easier Lemma using these claims. The proof of Lemma 7 can be easily adapted to prove Theorems 2 and 3. Details can be found in the appendix.

Lemma 7 *Unless* $\mathsf{NP} = \mathsf{RP}$, *for $\gamma < 1/8n$ there is no efficient γ-approximate oblivious regret minimization algorithm which uses the fixed strategies F as experts against oblivious adversaries for bounded-memory-m games of imperfect information.*

Proof of Lemma 7. Suppose that D were an efficient γ-approximate oblivious regret minimization algorithm and consider the polynomial time randomized algorithm S. Combining Claim 3 and Claim 1, for every MAX3SAT formula ϕ with $\geq (1-\beta)$ fraction of the clauses satisfiable S satisfies $\geq \frac{7}{8} + \beta$ fraction of the clauses from ϕ in expectation. This would imply that $\mathsf{NP} = \mathsf{RP}$ (Hastad, 2001). □The proof of Theorem 2 is very similar to the proof of Lemma 7.

Reminder of Theorem 2. *For any $\beta > 0$ and $\gamma < 1/8n^{\beta}$ there is no efficient γ-approximate oblivious regret minimization algorithm which uses the fixed strategies F as experts against oblivious adversaries for the class of imperfect information bounded-memory-m games unless* $\mathsf{NP} = \mathsf{RP}$.

Proof of Theorem 2. The key point is that if an algorithm S runs in time $O\left(p(n)\right)$ on instances of size n^{β} for some polynomial $p(n)$ then on instances of size n S runs in time $O\left(p\left(n^{1/\beta}\right)\right)$ which is still polynomial time. Unless $\mathsf{NP} = \mathsf{RP}$ $\forall \epsilon, \beta > 0$ and every algorithm S running in time poly(n), there exists an integer n and a MAX3SAT formula ϕ with n^{β} variables such that

1. There is an assignment satisfying at least $(1-\epsilon)$ of the clauses in ϕ.
2. The expected fraction of clauses in ϕ satisfied by S is $\leq \frac{7}{8}+\epsilon$.

If we reduce from a MAX3SAT instance with n^{β} variables we can construct a game with $O(n)$ states ($n^{1-\beta}$ copies of each variable). One Hamiltonian cycle would now corresponds to $n^{1-\beta}$ phases of the game. This means that the expected average reward of the optimal fixed strategy is at least

$$\max_{f\in F} E_R\left[\bar{P}\left(f, A_R, G, T\right)\right] \geq \frac{n^{1-\beta}\left(1-\epsilon\right)}{n} ,$$

while the expected average reward of an efficient defender strategy D is at most

$$E_R\left[\bar{P}\left(D, A_R, G, T\right)\right] \leq \frac{n^{1-\beta}\left(\frac{7}{8}+\epsilon\right)}{n} .$$

Therefore, the expected average regret is at least

$$\bar{R}_0\left(D, A_R, G, T, F\right) \geq \left(\frac{1}{8}-2\epsilon\right) n^{-\beta} .$$

□

While the proof of Theorem 3 makes use of the randomized exponential time hypothesis the argument is similar to the proof of Theorem 2.

Reminder of Theorem 3. *Assume that the randomized exponential time hypothesis is true. Then for any $\gamma < 1/\left(8\log^2 n\right)$ there is no efficient γ-approximate oblivious regret minimization algorithm which uses the fixed strategies F as experts against oblivious adversaries for the class of imperfect information bounded-memory-m games.*

Proof of Theorem 3. (sketch) Assume that the randomized exponential time hypothesis holds. Then because it is NP-hard to approximate MAX3SAT within any factor better than $\frac{7}{8}$ Hastad (2001) no randomized algorithm which satisfies $\geq \frac{7}{8}+\epsilon$ of the clauses in a MAX3SAT instance in expectation can run in time

$$2^{o(n)} .$$

Now we argue that it is sufficient to reduce from a MAX3SAT instance with $n' = \log^2 n$ variables (instead of n^{β} variables). One Hamiltonian cycle now corresponds to

$$\frac{n}{\log^2 n} ,$$

phases of the game. Our bounded-memory game G has n states then any efficient γ-approximate regret minimization algorithm S must run in time $O\left(n^k\right)$ for some constant k. If the randomized exponential time hypothesis holds then the expected average reward of an efficient defender strategy D is at most

$$E_R\left[\bar{P}\left(D, A_R, G, T\right)\right] \leq \frac{\frac{n}{\log^2 n}\left(\frac{7}{8}+\epsilon\right)}{n} ,$$

since

$$n^c = 2^{k\sqrt{\log^2 n}} = 2^{k\sqrt{n'}} = 2^{o(n')} .$$

However, if the MAX3SAT formula was satisfiable then the expected average reward of the optimal fixed strategy is at least

$$\max_{f \in F} E_R \left[\bar{P} \left(f, A_R, G, T \right) \right] \geq \frac{\frac{n}{\log^2 n} \left(1 - \epsilon \right)}{n} = \frac{1 - \epsilon}{\log^2 n} \ .$$

Therefore, the expected average regret is at least

$$\bar{R}_0 \left(D, A_R, G, T, F \right) \geq \frac{\left(\frac{1}{8} - 2\epsilon \right)}{\log^2 n} \ .$$

Assume for contradiction that $\gamma < \frac{1}{8 \log^2 n}$ then S can be adapted to satisfy $\geq \frac{7}{8} + \epsilon$ of the clauses in MAX3SAT with running time

$$n^c = 2^{k\sqrt{\log^2 n}} = 2^{k\sqrt{n'}} = 2^{o(n')} \ .$$

This contradicts the randomized exponential time hypothesis.

□

Remark 8 how our hardness reduction can be adapted to prove that there is no efficient k-adaptive regret minimization algorithm in the perfect information setting $k \geq 1$.

Remark 8 *In bounded-memory games of perfect information we can replace the oblivious adversary A_R in figure 4 with a 1-adaptive adversary and essentially the same reduction will still work. We only need to make a few small modifications. The states of the game will be modified to store the defenders last action. The adversary again plays a Hamiltonian cycle through the states in each phase. Now the first two states we visit correspond to the variable x_1, the next two visited states will correspond to x_2, etc. If the defender plays actions 1 and 1 (resp. 0 and 0) while visiting the variable x_1 then this corresponds to assigning x_1 to true (resp. false). If the defender plays 1 and 0 (or 0 and 1) which corresponds to no assignment then the adversary strategy will ensure that he cannot receive a reward.*

The 1-adaptive adversary will always play $\boldsymbol{a}^t[2] = 2$ on even rounds ($t = 0 \mod 2$) and on odd rounds the adversary will adaptively select $\boldsymbol{a}^t[2] = d^{t-1}$ if the defender's last action satisfied the chosen clause C, otherwise $\boldsymbol{a}^t[2] = 2$. The defender receives a reward only if (1) he plays a consistent assignment during both rounds (2) the assignment satisfies the chosen clause C and (3) he has not already received a reward during this phase. Now Claim 1 still holds because a defender will always observe the adversary action $\boldsymbol{a}^t[2] = 2$ until he satisfied the clause C.

7.1. Transition Example

By playing a De Bruijn sequence $S = s_1 ... s_n$ the adversary can guarantee that we repeatedly take a Hamiltonian cycle over states. For example, considering 8 states and starting from x_0, the sequence 10111000 corresponds to the Hamiltonian cycle $x_0, x_1, x_2, x_5, x_3, x_7, x_6, x_4$

8. Regret Minimization Algorithms

8.1. Regret Minimization Algorithm with Imperfect Information

We present BW (Bounded Memory Weighted Majority), an algorithm that minimizes k-adaptive regret for bounded-memory games. This result is significant because there is no k-adaptive regret

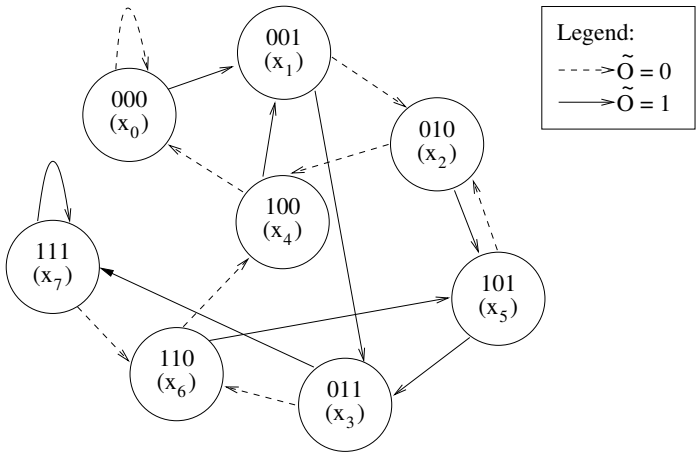


Figure 2: De Bruijn example

minimization algorithm for the general class of stochastic games(see Theorem 11 in the appendix). A consequence of Theorem 2 is that when the expert set includes all fixed strategies F we cannot hope for an efficient algorithm unless $\mathsf{NP} = \mathsf{RP}$. Indeed, our algorithm would not be efficient in this case because it would have to explicitly maintains weights for exponentially many fixed strategies $|F| = |\mathcal{X}_D|^n$.

The key idea behind our k-adaptive regret minimization algorithm BW is to reduce the original bounded-memory game to a repeated game $\rho(G, K)$ of imperfect information ($K \equiv 0 \mod k$). BW uses the Exp3 regret minimization algorithm of Auer et al. (1995) for repeated games of imperfect information. In particular, BW uses the strategies selected by Exp3 in each round of $\rho(G, K)$ to play the next K rounds of G. BW feeds Exp3 the hypothetical losses from $\rho(G, K)$ to update the weights of each expert.

Reminder of Theorem 4. *Let G be any bounded-memory-m game with n states and let A be any adversary strategy. After playing T rounds of G against A,* BW (G, K) *achieves regret bound*

$$\bar{R}_k(\mathsf{BW}, A, G, T, S) \quad < \quad \frac{m}{T^{1/4}} + 4\frac{\sqrt{N \log N}}{T^{1/4}} ,$$

where $N = |S|$ is the number of experts, A is the adversary strategy and K has been chosen so that $K = T^{1/4}$ and $K \equiv 0 \mod k$.

Proof of Theorem 4. (Sketch) The proof of theorem uses standard regret bound for regret minimization algorithms in games of perfect information Auer et al. (1995). After playing T rounds (T/K rounds of $\rho(G, K)$) we have

$$\bar{P}(D, A_k, \rho(G, K), T/K) - \bar{P}(f, A_k, \rho(G, K), T/K) \quad \geq \quad -4\sqrt{\frac{KN \log N}{T/K}} ,$$

for all fixed strategies $f \in F$. Here, N is the number of experts

$$N = |F| = |\mathcal{X}_D|^{|\Sigma|} ,$$

and K also denotes the maximum payout in any round of $\rho(G, K)$.Because K was chosen such that $K \equiv 0 \mod k$ the adversary A_k is always in phase with $\rho(G, K)$ and we can apply Claim 4 to get Theorem 4. □

In particular, BW is a k-adaptive regret minimization algorithm for the class of bounded-memory games in the sense of Definition 1 because $\bar{R}_k \to 0$ as $T \to \infty$.

Remark 9 BW *is inefficient when number of experts $f \in S$ is exponential in n, the number of states in G. For example, if $S = F$ then $|F| = |\mathcal{X}_D|^n$. For small values of n (example: for repeated games $n = 1$) it will still be tractable to run* BW *with $S = F$.*

8.2. Proofs of Claims and Theorems

This section contains the proof of claims and theorems from section 5.

Claim 4 bounds the difference between the hypothetical losses from $\rho(G, K)$ and actual losses in G using the bounded-memory property.

Claim 4 *For any adaptive defender strategy $f \in \mathcal{A}_D^K$ and any adaptive adversary strategy $g \in \mathcal{A}_A^K$ and any state σ of G we have $|P(f, g, G, \sigma, K) - P(f, g, G, \sigma_0, K)| \leq m$.*

Proof of Claim 4. (Sketch) Once the defender selects f and the adversary selects strategy $g \in K - ADAPT_A$, the actions of the adversary and the defender are fixed for the next K rounds of G. Let $d^1, ..., d^K$ (resp. $a^1, ..., a^K$) denote the actions taken by the defender (resp. adversary). Once $R_1, ..., R_K$ (the random coins used by the outcome function) are fixed then the outcomes $O^1, ..., O^K$ are also fixed. Let $\sigma^1, ..., \sigma^K$ states encountered in the actual game and let $\sigma_*^1, ..., \sigma_*^K$ be the states that we would have encountered if we had started at σ_0 as in $\rho(G, K)$. In a bounded-memory property game the state encodes the last m outcomes, but the outcomes do not depend on the starting state so we have

$$\sigma^j = \sigma_*^j \ ,$$

for all $j \geq m$. This means that for $j \geq m$

$$P\left(\sigma^j, d^j, a^j\right) = P\left(\sigma_*^j, d^j, a^j\right) \ .$$

Consequently,

$$\begin{aligned} |P(f, g, \sigma, G) - P(f, g, \sigma_0, G)| &= \left| \sum_{t=1}^{k} P\left(d_t, a_t, \sigma^i\right) - \sum_{t=1}^{k} P\left(d_t, a_t, \sigma_*^i\right) \right| \\ &= \left| \sum_{t=1}^{m-1} P\left(d_t, a_t, \sigma^i\right) - P\left(d_t, a_t, \sigma_*^i\right) \right| \\ &\leq m \ . \end{aligned}$$

□

The standard weighted majority algorithm maintains the invariant that $W_E = \beta^{\sum_{j=1}^{T/K} P\left(E, \boldsymbol{a}^t, \rho(G,K)\right)}$. Claim 5 says that EXBW also maintains this invariant.

Claim 5

$$\prod_{p \in \mathcal{C}(E)} \beta^{\sum_{j=1}^{T/K} \ell\left(p, \boldsymbol{a}^j, \sigma^{jK}\right)} = \beta^{\sum_{j=1}^{T/K} P\left(E, \boldsymbol{a}^j, \rho(G,K)\right)} \ .$$

Proof of Claim 5. First notice that we can write

$$\sum_{j=1}^{T/K} P\left(E, \boldsymbol{a}^j, \rho\left(G, K\right)\right) = \sum_{p \in \mathcal{C}(E)} \sum_{j=1}^{T/K} \ell\left(p, \boldsymbol{a}^j, \sigma^{jK}\right) ,$$

since the overall payoff of an expert E can be expressed as a sum of the individual immediate payoffs after each action.

$$\begin{aligned}
\prod_{p \in \mathcal{C}(E)} \beta^{\sum_{j=1}^{T/K} \ell\left(p, \boldsymbol{a}^j, \sigma^{jK}\right)} &= \beta^{\sum_{p \in \mathcal{C}(E)} \sum_{j=1}^{T/K} \ell\left(p, \boldsymbol{a}^j, \sigma^{jK}\right)} \\
&= \beta^{\sum_{t=1}^{T/K} P\left(E, \boldsymbol{a}^t, \rho(G,K)\right)} .
\end{aligned}$$

□

Claim 6 says that $\mathbf{Sample}\left(\mathcal{E}\right)$ samples from the right distribution.

Claim 6 *For each expert $E \in \mathcal{E}$ Algorithm $\mathbf{Sample}\left(\mathcal{E}\right)$ outputs E with probability*

$$\Pr\left[E\right] \propto W_E .$$

Proof Given a trace $p = p_0; O; d$ let $\mathbf{Chosen}\left(p_0; O\right)$ be the event that the strategy output by Algorithm $\mathbf{Sample}\left(\mathcal{E}\right)$ plays d from given history $p_0; O$.

$$\begin{aligned}
\Pr\left[\text{Output } E\right] &= \prod_{p \in \mathcal{C}, O \in \mathcal{O}} \Pr\left[\mathbf{Chosen}\left(p; O\right) = E\left(p; O\right)\right] \\
&= \prod_{p \in \mathcal{C}, O \in \mathcal{O}, d = E(p,O)} \frac{\hat{w}_{p;O;d}}{\sum_{d' \in \mathcal{X}_D} \hat{w}_{p;O;d'}} \\
&= \prod_{p \in \mathcal{C}, O \in \mathcal{O}, d = E(p,O)} \frac{\sum_{E':(p;O;d) \in \mathcal{C}(E')} \prod_{p' \in \mathcal{C}(E') \wedge p;O;d \sqsubset p'} w_{p'}}{\sum_{d' \in \mathcal{X}_D} \sum_{E':(p;O;d') \in \mathcal{C}(E')} \prod_{p' \in \mathcal{C}(E') \wedge p;O;d' \sqsubset p'} w_{p'}} \\
&= \prod_{p \in \mathcal{C}, O \in \mathcal{O}, d = E(p,O)} \frac{\sum_{E':(p;O;d) \in \mathcal{C}(E')} \prod_{p' \in \mathcal{C}(E') \wedge p;O;d \sqsubset p'} w_{p'}}{\sum_{d' \in \mathcal{X}_D} \sum_{E':(p;O;d') \in \mathcal{C}(E')} \prod_{p' \in \mathcal{C}(E') \wedge p;O;d' \sqsubset p'} w_{p'}} \times \frac{\prod_{p' \sqsubset p} w_{p'}}{\prod_{p' \sqsubset p} w_{p'}} \\
&= \prod_{p \in \mathcal{C}, O \in \mathcal{O}, d = E(p,O)} \frac{\sum_{E':(p;O;d) \in \mathcal{C}(E')} \prod_{p' \in \mathcal{C}(E')} w_{p'}}{\sum_{d' \in \mathcal{X}_D} \sum_{E':(p;O;d') \in \mathcal{C}(E')} \prod_{p' \in \mathcal{C}(E')} w_{p'}} \\
&= \prod_{p \in \mathcal{C}, O \in \mathcal{O}, d = E(p,O)} \frac{\sum_{E':(p;O;d) \in \mathcal{C}(E')} W_{E'}}{\sum_{d' \in \mathcal{X}_D} \sum_{E':(p;O;d') \in \mathcal{C}(E')} W_{E'}} \\
&= \prod_{p \in \mathcal{C}, O \in \mathcal{O}, d = E(p,O)} \frac{\sum_{E':(p;O;d) \in \mathcal{C}(E')} W_{E'}}{\sum_{E' \in \mathcal{E}} W_{E'}} \\
&= \frac{W_E}{\sum_{E' \in \mathcal{E}} W_{E'}} .
\end{aligned}$$

■

Reminder of Theorem 5. *Let G be any bounded-memory-m game of perfect information with n states and let A be any adversary strategy. Playing T rounds of G against A,* EXBW *runs in total time $Tn^{O(1/\gamma)}$ and achieves regret bound*

$$\bar{R}_0\left(\mathsf{EXBW}, A, G, T, \mathcal{E}\right) \leq \gamma + O\left(\frac{m}{\gamma}\sqrt{\frac{\frac{m}{\gamma} n \log\left(N\right)}{T}}\right) ,$$

where K has been set to m/γ and $N = \left|\mathcal{A}_D^K\right| = \left(\left|\mathcal{X}_D\right|\right)^{n^{1/\gamma}}$ is the number of K-adaptive strategies.

Proof of Theorem 5. By Claims 5 and 6 Algorithm EXBW perfectly simulates the weighted majority algorithm Littlestone and Warmuth (1989). Notice that there are N^n experts in $\mathcal{E}$ and we are playing T/K rounds of $\rho\left(G, K\right)$. The maximum payment in round of $\rho\left(G, K\right)$ is $K = m/\gamma$. The regret bound immediately follows from Claim 4 (the $\gamma = m/K$ term) and the standard regret bound from Littlestone and Warmuth (1989) after setting

$$\beta = \min\{\frac{1}{2}, \sqrt{\frac{n \ln\left(N\right)}{T}}\} .$$

The regret bound holds against all experts $E \in \mathcal{E}$ so in particular the regret bound also holds against all fixed experts $f \in F$ since $F \subset \mathcal{E}$.

The running time of EXBW is proportional to the number of traces in $\mathcal{C}$. There are only $n^{O(1/\gamma)}$ total traces in $\mathcal{C}$ so for any constant γ the running time is polynomial. □

Reminder of Theorem 6. *Let G be any bounded-memory-m game with n states and let A be any adversary strategy. After playing T rounds of G against A,* BW (G, K) *achieves regret bound*

$$\bar{R}_k\left(\mathsf{BW}, A, G, T, S\right) \quad < \quad \frac{m}{T^{1/4}} + 4\frac{\sqrt{N \log N}}{T^{1/4}} ,$$

where $N = |S|$ is the number of experts, A is the adversary strategy and K has been chosen so that $K = T^{1/4}$ and $K \equiv 0 \mod k$.

Proof of Theorem 6. (Sketch) We group the rounds of $\rho\left(G, K\right)$ into phases of $\frac{n^{1/\gamma}}{\gamma}$ rounds. Each phase now corresponds to

$$K\frac{n^{1/\gamma}}{\gamma} = \frac{mn^{1/\gamma}}{\gamma^2} ,$$

rounds of $\mathcal{G}$. As before there are N^n experts.

Within a single phase let $\boldsymbol{a}^i$ $\left(i = 1, ..., n^{1/\gamma}/\gamma\right)$ denote the actions of the adversary during round i of that phase. To update our implicit weight representation we would like to compute

$$\sum_i \ell\left(p, \boldsymbol{a}^i, \sigma\right) ,$$

for each $p \in \mathcal{C}$. However, we do not know the adversary actions $\boldsymbol{a}^i$ in each phase. Instead of computing

$$\sum_i \ell\left(p, \boldsymbol{a}^i, \sigma\right) ,$$

we will estimate this quantity. For each

$$\boldsymbol{d} \in \mathcal{X}_D^{\frac{m}{\gamma}} ,$$

we will play the defender actions $\boldsymbol{d}$ in a randomly chosen round of the phase. Let $\boldsymbol{O}$ and $\boldsymbol{\ell} = \left(\ell_1, ..., \ell_{m/\gamma}\right)$ denote the observed outcomes and payoffs in this round and let p^j be the path corresponding to the first j defender actions from $\boldsymbol{d}$ and outcomes from $\boldsymbol{O}$. For each path p^j we set

$$\ell'\left(p^j, \sigma\right) = \frac{n^{1/\gamma}}{\gamma} \ell_j .$$

If the path p never occured during a sampling round of the phase then we set

$$\ell'\left(p^j, \sigma\right) = 0 .$$

For each path $p \in \mathcal{C}$ we have

$$\begin{aligned} E\left[\ell'\left(p, \sigma\right)\right] &= \frac{n^{1/\gamma}}{\gamma} E\left[\ell_i\right] \\ &= \frac{n^{1/\gamma}}{\gamma} \sum_i \frac{\gamma}{n^{1/\gamma}} \ell\left(p, \boldsymbol{a}^i, \sigma\right) \\ &= \sum_i \ell\left(p, \boldsymbol{a}^i, \sigma\right) \end{aligned}$$

where the expectation is taken over the random selection of sampling rounds. Now we can use the estimated losses ℓ' to maintain our implicit weight representation.

The following factors explain why the final regret bound is slightly worse than the bound in the perfect information setting (Theorem 5):

1. We spend at most

$$\left|\mathcal{X}_D^{\frac{m}{\gamma}}\right| \leq n^{1/\gamma} ,$$

rounds of each phase sampling. There are $\frac{n^{1/\gamma}}{\gamma}$ rounds in a phase so the average sampling loss per round is at most

$$\frac{n^{1/\gamma}}{\left(\frac{n^{1/\gamma}}{\gamma}\right)} = \gamma .$$

This is in addition to modeling loss (γ) from claim 4. In the perfect information setting there is no sampling loss just the modeling loss.

2. We are only now only updating weights after each phase. If T is the number of rounds of the bounded-memory game G that we play then we only update weights T' times where

$$T' = \frac{T\gamma^2}{mn^{1/\gamma}} .$$

In the perfect information setting we had $T' = \frac{T\gamma}{m}$.

3. The maximum loss in each phase is now the length of a phase

$$\frac{m}{\gamma}\left(\frac{n^{1/\gamma}}{\gamma}\right),$$

instead of the length of a round m/γ.

□

Remark 10 *Because repeated games are a subset of bounded-memory games,* EXBW *(resp.* EXBWII*) could also be used to minimize oblivious regret in a repeated game of perfect information (resp. imperfect information) using $\mathcal{A}_D^K$ as experts. In this case there is no modeling loss from claim 4 so the guarantee is that we perform as well as the best K-adaptive defender strategy in hindsight. As long as $K = O(\log n)$ the running time of our algorithms will be time polynomial in n.*

8.3. Impossibility of Regret Minimization in Stochastic Games

Stochastic Games Stochastic games are a generalization of repeated games, in which the payoffs depend on the state of play. Formally, a two-player stochastic game between an attacker A and a defender D is given by $(\mathcal{X}_D, \mathcal{X}_A, \Sigma, P, \tau)$, where $\mathcal{X}_A$ and $\mathcal{X}_D$ are the actions spaces for players A and D, respectively, Σ is the state space, $P : \Sigma \times \mathcal{X}_D \times \mathcal{X}_A \to [0,1]$ is the payoff function and $\tau : \Sigma \times \mathcal{X}_D \times \mathcal{X}_A \times \{0,1\}^* \to \Sigma$ is the randomized transition function linking the different states.

Thus, the payoff during round t depends on the current state (denoted σ^t) in addition to the actions of the defender (d^t) and the adversary (a^t). This added flexibility enables us to develop realistic game models for interactions where the rewards depend on game history. The hospital-employee interaction we introduced earlier is one example of such an interaction: an employee committing a given violation for the first time is unlikely to meet the same punishment as an employee committing the same violation for the tenth time.

A *fixed strategy* for the defender in a stochastic game is a function $f : \Sigma \to \mathcal{X}_D$ mapping each state to a fixed action. F denotes the set of all fixed strategies.

In this section we demonstrate that there is no regret minimization algorithm for the general class of stochastic games. More specifically for every notion of regret k (oblivious ($k = 0$), k-adaptive, fully adaptive ($k = \infty$)) there is no k-adaptive minimization algorithm for the class of stochastic games. It suffices to consider 'oblivious regret' against an oblivious adversary (see remark 12). The example in Theorem 11 is fundamentally similar to example IV.1 of Yu and Mannor.

Theorem 11 *There is a stochastic game G such that for any defender strategies D there exists an oblivious adversary A such that*

$$\lim_{T\to\infty} \bar{R}_k(D, A, G, T) > 0 .$$

Proof

In particular, consider the stochastic game G illustrated in Figure 3. The figure shows a game with two players D and A with action sets $\mathcal{X}_D = \{d_1, d_2\}$ and $\mathcal{X}_A = \{a_1, a_2\}$ respectively. The reward function for the defender depends only on his own action as well as the current state σ. Observe that σ_2 is a sink state which the game can never leave. If the game reaches this state then the

defender will be continuously rewarded in every round for the rest of the game. However, the only way to reach σ_2 is if the defender and the adversary play (d_1, a_1) simultaneously in some round t. If the defender fails to play d_1 then he might permanently miss his opportunity to reach σ_2. This suggests that the defender must always play d_1. However, if the adversary never plays a_1 then it is best to use the fixed strategy always play d_2. ■

Notice that for any $A \in \mathcal{A}_A^0$ and any defender strategy D we have

$$\bar{P}\left(D, A, G, T\right) = \bar{P}\left(D, A_0, G, T\right) \quad ,$$

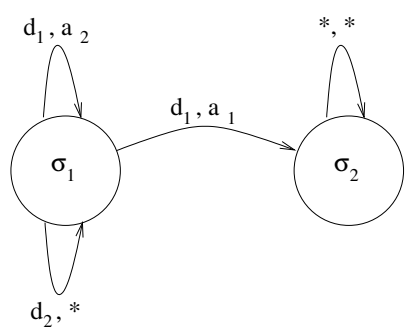
because $A = A_0$. Hence, $\bar{R}_0 = \bar{R}_k$ whenever the adversary is oblivious.

Remark 12 *1. If D can minimize k-adaptive regret against any k-adaptive adversary then D can minimize k-adaptive regret against any oblivious adversary ($k = 0$) because*

$$\mathcal{A}_A^0 \subset \mathcal{A}_A^k \ .$$

2. If D can minimize k-adaptive regret against any k-adaptive adversary then D can minimize k-adaptive regret against any oblivious adversary because $\bar{R}_0 = \bar{R}_k$ whenever the adversary is oblivious.

3. If D is a k-regret minimization algorithm a class of games $\mathcal{G}$ and $\mathcal{G}'$ is a subclass of $\mathcal{G}$ then D is also a k-regret minimization algorithm for the class of games $\mathcal{G}'$.



$P(d_1, \sigma_1) = -1 \quad P(d_1, \sigma_2) = 1$
$P(d_2, \sigma_1) = 0 \quad P(d_2, \sigma_2) = 1$

Figure 3: A counterexample to prove Theorem 11

This example also illustrates why it is impossible to minimize fully adaptive regret against a non-forgetful adversary. In particular a non-forgetful adversary could use the states from 3 to decide whether or not to cooperate. Note that even if the adversary can only see the last m outcomes (sliding window) the adversary could play to remind himself of events arbitrarily long ago. For example, an adversary who wanted to remember whether or not the defender played action d during round 1 might play a special reminder action every m rounds when the latest reminder is about to go out of memory.

Algorithm 1 Assignment Recovery

- **Input:** D
- **Input:** MAX3SAT instance ϕ, with variables

$$x_1, \ldots, x_{n-1} \ ,$$

and clauses

$$C_1, \ldots, C_\ell \ ,$$

- **De Bruin sequence:** $s_0, ..., s_{n-1}$
- **Initialize:** Set $t \leftarrow 0$, $H \leftarrow \emptyset$, $T \leftarrow p(n)$, $\alpha* \leftarrow 0$
- **Round** t**:** Set $i \leftarrow t \mod n$

1. **Check 1:** If $t \geq T$ then return.
2. **Check 2:** If our current assignment $x_1, ..., x_{n-1}$ satisfies y fraction of the clauses where $y > \alpha*$ then set

$$x_i* \leftarrow x_i \ ,$$

and

$$\alpha \leftarrow y \ .$$

3. **Select Clause:** If $i = 0$ then select a new clause C uniformly at random from $C_1, ..., C_\ell$, and set $H' = \emptyset$.
4. **Select Adversary Move:**

$$a^i \leftarrow \begin{cases} (s_i, 3) & \text{if } i = 0; \\ (s_i, 1) & \text{if } x_i \in C; \\ (s_i, 0) & \text{if } \bar{x}_i \in C; \\ (s_i, 2) & \text{otherwise.} \end{cases}$$

5. **Select Defender Move:**

$$d^i \leftarrow D\left(H^{t-i}; H'\right) \ ,$$

6. **Update:** Let O^i be the outcome and set

$$H \leftarrow H + \left(s_i, \hat{O}^i\right) \ ,$$

$$H' \leftarrow H' + (s_i, 0) \, ,$$

$$t \leftarrow t + 1 \ ,$$

$$x_i \leftarrow d^i \ ,$$