

Nonstochastic Multi-Armed Bandit Approach to Stochastic Discrete Optimization

Hyeong Soo Chang, Jiaqiao Hu, Michael C. Fu, and Steven I. Marcus

Abstract

We present a sampling-based algorithm for solving stochastic discrete optimization problems based on Auer *et al.*'s Exp3 algorithm for “nonstochastic multi-armed bandit problems.” The algorithm solves the sample average approximation (SAA) of the original problem by iteratively updating and sampling from a probability distribution over the search space. We show that as the number of samples goes to infinity, the value returned by the algorithm converges to the optimal objective-function value and the probability distribution to a distribution that concentrates only on the set of best solutions of the original problem. We then extend the Exp3-based algorithm to solving finite-horizon Markov decision processes (MDPs), where the underlying MDP is approximated by a recursive SAA problem. We show that the estimate of the “recursive” sample-average-maximum computed by the extended algorithm at a given state approaches the optimal value of the state as the sample size per state per stage goes to infinity. The recursive Exp3-based algorithm for MDPs is then further extended for finite-horizon two-person zero-sum Markov games (MGs), providing a finite-iteration bound to the equilibrium value of the induced SAA game problem and asymptotic convergence to the equilibrium value of the original

H.S. Chang is with the Department of Computer Science and Engineering at Sogang University, Seoul 121-742, Korea, and can be reached by e-mail at hschang@sogang.ac.kr.

J. Hu is with the Department of Applied Mathematics & Statistics, SUNY, Stony Brook, and can be reached by email at jqhu@ams.sunysb.edu.

M.C. Fu is with the Robert H. Smith School of Business and Institute for Systems Research at the University of Maryland, College Park, USA, and can be reached by email at mfu@rhsmith.umd.edu.

S.I. Marcus is with the Department of Electrical and Computer Engineering and Institute for Systems Research at the University of Maryland, College Park, USA, and can be reached by e-mail at marcus@umd.edu.

This work was supported in part by the National Science Foundation under Grant DMI-0323220, in part by the Air Force Office of Scientific Research under Grant FA95500410210, and in part by the Department of Defense.

Preliminary portions of this paper appeared in the *Proceedings of the 45th and 46th IEEE Conferences on Decision and Control*, 2006 and 2007.

game. The time and space complexities of the extended algorithms for MDPs and MGs are independent of their state spaces.

Index Terms

Stochastic optimization, Markov decision process, Markov game, sample average approximation, sampling

I. INTRODUCTION

Consider a stochastic discrete optimization problem Ψ of $\max_{s \in S} E_\omega[F(s, \omega)]$, where S is a finite subset of $\mathbb{R}^n$, ω is a random (data) vector supported on a set $\Omega \subset \mathbb{R}^d$, $F : S \times \Omega \rightarrow \mathbb{R}^+$, and the expectation is taken with respect to a probability distribution $\mathcal{P}$ of the random vector ω . We assume that the expectations over $\mathcal{P}$ for all $s \in S$ are well-defined and F is bounded such that $F(s, \omega) \in [0, 1]$ for any $s \in S$ and $\omega \in \Omega$.

Because it is usually not possible to find a closed form expression for $E_\omega[F(s, \omega)]$, solving the optimization problem Ψ exactly is difficult in general. Assume that $F(s, w)$ can be evaluated explicitly for any $s \in S$ and $w \in \Omega$ and that by sampling from $\mathcal{P}$, samples $w_1, w_2, \dots$ of independent realizations of w can be generated. In the sequel of this paper, we consider the method of discrete *sample average approximation* (SAA) [19]: a random sample sequence $\{w_1, \dots, w_T\}$ of size T is generated, and a problem Ψ_T of obtaining a solution in S that achieves the *sample-average-maximum* of $\max_{s \in S} T^{-1} \sum_{k=1}^T F(s, w_k)$ is solved. The solution to Ψ_T can then be taken as an approximate solution to an optimal solution in $S_\Psi^* := \arg \max_{s \in S} E_\omega[F(s, \omega)]$.

Kleywegt *et al.* [19] provide the value of $T(\alpha, \epsilon)$, $\alpha \in (0, 1)$, $\epsilon > 0$ such that a sample size $T \geq T(\alpha, \epsilon)$ guarantees that any $\epsilon/2$ -optimal solution of the discrete SAA problem Ψ_T is an ϵ -optimal solution of the problem Ψ with probability at least $1 - \alpha$ under mild regularity conditions. Moreover, the probability that the optimal solution of the SAA problem Ψ_T is in the set of “near-optimal” solutions of the problem Ψ converges to one exponentially fast in T . However, as noted in [19], no general (approximation) algorithm exists for actually solving the deterministic SAA problem, i.e., obtaining $(\arg) \max_{s \in S} T^{-1} \sum_{k=1}^T F(s, w_k)$, other than an exhaustive search. Furthermore, the computational complexity for solving the SAA problem often increases exponentially in the sample size T , and algorithms tailored to specific SAA problems often need to be devised [19]. Ahmed and Shaprio presented a general branch-and-bound based

SAA algorithm [2] for a class of two-stage stochastic programs with integer recourse when the first stage variables are *continuous*, corresponding to S being continuous here. The algorithm successively partitions its search space and uses some lower and upper bound information to eliminate parts of the search space, avoiding complete enumeration. The computation for these bounds still involves solving some optimization problems, which generally require further approximations.

In this paper, we first present a general *convergent* sampling-based algorithm for the stochastic optimization problem Ψ based on the Exp3 algorithm developed by Auer *et al.* [4] for solving “nonstochastic (or adversarial) multi-armed bandit problems.” The input to the algorithm is a T -length sequence of data samples $\{w_k, k = 1, \dots, T\}$ of w , where w_k ’s are i.i.d. samples from $\mathcal{P}$. The Exp3-based algorithm is invoked for solving the SAA problem Ψ_T induced by the sequence of samples $\{w_1, \dots, w_T\}$. At each iteration $k \geq 1$ of the algorithm, a *single* solution $s_k \in S$ is sampled from a probability distribution $P(k)$ over the solution space S and the sampled solution s_k is evaluated with a single data sample w_k , obtaining $F(s_k, w_k)$. The probability distribution $P(k)$ is updated from $F(s_k, w_k)$, yielding $P(k+1)$. This process is repeated at iteration $k+1$. We show that with a proper tuning of a control parameter in the algorithm as a function of T , as $T \rightarrow \infty$, the expected performance of the Exp3-based algorithm converges to the value of $\max_{s \in S} E_\omega[F(s, \omega)]$ and the sequence of the distributions $\{P(T)\}$ converges to a distribution concentrated only on the solutions in S_Ψ^* .

As applications of this idea of solving *non-sequential* stochastic optimization problems in a nonstochastic bandit-setting, we consider finite-horizon sequential decision making problems with uncertainties formulated by Markov decision processes (MDPs) [22] and two-person zero-sum Markov games (MGs) [27] [12] [3]. We provide Exp3-based “recursive” sampling algorithms for MDPs and MGs that can be used for addressing the issue of *curse of dimensionality* from large state spaces.

Based on the observation that the sequential structure of MDPs allows us to formulate “recursive SAA problems,” we recursively extend the Exp3 algorithm, yielding an algorithm, called RExp3MDP (Recursive Exp3 for MDPs), for solving MDPs. The worst-case running-time complexity of RExp3MDP is $O((|A|T)^H)$, where $|A|$ is the action space size, H is the horizon length, and T is the number of samples per state per stage. The space-complexity is $O(|A|T^H)$ because for each sampled state, a probability distribution over A needs to be maintained. The

complexities are independent of the state space size $|X|$ but exponentially dependent on the horizon size H . Similar to the non-sequential stochastic optimization case, the algorithm is invoked for solving a deterministic recursive SAA problem. We show that the estimate of the recursive sample-average-maximum by RExp3MDP for an induced recursive SAA problem at an initial state approaches the optimal value of the state for the original MDP problem as the sample size per state per stage goes to infinity.

We then extend the Exp3-based algorithm proposed for MDPs into an algorithm, called RExp3MG (Recursive Exp3 for MGs), for finite horizon two-person zero-sum MGs by formulating a “recursive SAA game.” We show that a *finite-iteration* error bound on the difference between the expected performance of RExp3MG and the “recursive sample-average equilibrium value” is given by $O\left(H\sqrt{|A_{\max}|\ln|A_{\max}|/T}\right)$, where $|A_{\max}|$ is the larger of the two players’ action space sizes, H is the horizon length, and T is the total number of action-pair samples that are used per state per stage. As in MDPs, the expected performance of the algorithm converges to the equilibrium value of the original game as the number of samples T goes to infinity. The worst-case running-time complexity of the algorithm is $O((|A_{\max}|JT)^H)$ and the space-complexity is $O(|A_{\max}|(JT)^H)$, where J is the total number of samples that are used per action-pair sample to estimate the recursive sample-average equilibrium value. Similar to the MDP case, the complexities are still independent of the state space size $|X|$ but exponentially dependent on the horizon size H .

To the best of our knowledge, this is the *first* work applying the theory of the (nonstochastic) multi-armed bandit problem to derive a provably convergent *adaptive sampling-based* algorithm for solving general finite-horizon MGs.

It should be noted that characterizing a general class of stochastic optimization problems to which the nonstochastic multi-armed bandit approach is better than other (simulation-based) approaches in terms of the “speed” of convergence is difficult. The works here should be understood as *another theoretical framework* for stochastic optimization problems motivated from the non-existence of a general algorithm for discrete SAA problems.

This paper is organized as follows. We start by presenting the Exp3-based algorithm for non-sequential stochastic optimization problems with a convergence analysis in Section II. In Section III, we provide RExp3MDP for MDPs and analyze its performance. RExp3MG for two-person zero-sum MGs is presented and analyzed in Section IV. We conclude the paper in

Section V.

II. THE EXP3 ALGORITHM FOR STOCHASTIC OPTIMIZATION

Let $F^* = \max_{s \in S} E_\omega[F(s, \omega)]$. Again, to estimate F^* of Ψ by SAA, we generate T independent random samples $\omega_k, k = 1, \dots, T$ of ω according to the distribution $\mathcal{P}$ of ω and then obtain the sample-average-maximum of $\Psi_T, F_{\max}^T$, given by

$$F_{\max}^T = \max_{s \in S} \frac{1}{T} \sum_{k=1}^T F(s, \omega_k). \quad (1)$$

Note that (1) has a positive bias, i.e., $E_{\omega_1, \dots, \omega_T}[F_{\max}^T] \geq F^*$. However, as $T \rightarrow \infty, F_{\max}^T \rightarrow F^*$ w. p. 1 [19].

The SAA problem in (1) can be cast into a nonstochastic or adversarial bandit problem (see [4] for a detailed discussion), where an adversary generates an arbitrary and unknown sequence of bandit-rewards, chosen from a bounded real interval. The adversary, rather than a well-behaved stochastic process as in the classical multi-armed bandit [25], has complete control over the bandit-rewards. In what follows, we use some terms differently from those in [4] in order to be consistent with our problem setup. Each solution s (corresponding to an arm of the bandit) is denoted by an integer in $\{1, \dots, |S|\}$. A sequence $\{r(1), r(2), \dots\}$ of bandit-reward vectors is assigned by an adversary, where the bandit-reward vector $r(k) = (r_1(k), \dots, r_{|S|}(k))$, and $r_s(k) \in [0, 1]$ denotes the bandit-reward obtained by a bandit player if solution s is chosen at iteration k .

The bandit-reward assignment sequence is given as follows: At step 1, we obtain a sample ω_1 of w and set $r_s(1) = F(s, \omega_1), 1 \leq s \leq |S|$. That is, the same random vector ω_1 is used for assigning the bandit-reward of taking each solution $s = 1, \dots, |S|$. We do the same thing for the step 2, 3, ..., and so forth. The sequence of samples $\{\omega_k, k = 1, 2, \dots\}$ results in a deterministic bandit-reward assignment sequence $\{r(k), k = 1, 2, \dots\}$ and this sequence is *pre-determined* before a player begins solving the bandit problem.

At iteration $k \geq 1$, a player algorithm $\mathcal{A}$ generates a probability distribution $P(k) = \langle p_1(k), \dots, p_{|S|}(k) \rangle$ over S and selects a solution s_k with probability $p_{s_k}(k)$ independently of the past selected solutions $s_1, \dots, s_{k-1}$. Given a player algorithm $\mathcal{A}$ and a bandit-reward sequence $\{r(k), k = 1, \dots, T\}$, we define the *expected regret* of $\mathcal{A}$ for the best solution over T -iterations

by

$$\max_{s \in S} \frac{1}{T} \sum_{k=1}^T r_s(k) - E_{\mathcal{P}_{1:T}} \left[\frac{1}{T} \sum_{k=1}^T r_{s_k}(k) \right],$$

where the expectation $E_{\mathcal{P}_{1:T}}$ is taken with respect to the joint distribution $\mathcal{P}_{1:T}$ over the set of all possible solution sequences of length T obtained by the product of the distributions $P(1), \dots, P(T)$. The goal of the nonstochastic bandit problem is to design a player algorithm $\mathcal{A}$ such that the expected regret approaches zero as $T \rightarrow \infty$. Notice that $\max_{s \in S} \frac{1}{T} \sum_{k=1}^T r_s(k)$ is equal to the sample-average-maximum value of $F_{\max}^T$ for the *deterministic* SAA problems induced from the sequence of sampled random vectors $\{\omega_1, \dots, \omega_T\}$ used for assigning the bandit-reward sequence $\{r(1), \dots, r(T)\}$.

Auer *et al.* [4] provide a player algorithm, called Exp3 (which stands for “exponential-weight algorithm for exploration and exploitation”), for solving nonstochastic multi-armed bandit problems. A finite-iteration upper bound on the expected regret of Exp3 that holds for *any* bandit-reward assignment sequence is analyzed. On the other hand, the lower bound analysis in [4, Theorem 5.1] is given only for a specific bandit-reward sequence but for any player algorithm $\mathcal{A}$.

We first provide a slightly modified high-level pseudocode of Exp3 in Figure 1. The input to the algorithm is a T -length sequence of data samples $\{w_k, k = 1, \dots, T\}$ of w for a selected $T > 0$, where w_k ’s are i.i.d. samples from $\mathcal{P}$ and the outputs are the estimate $\hat{F}^T$ of the sample-average-maximum $F_{\max}^T$ for the SAA problem defined in terms of the input data-sample sequence $\{w_k, k = 1, \dots, T\}$ and the probability distribution $P(T)$ for an estimate of the best solution in S_{Ψ}^* .

At each iteration $k = 1, \dots, T$, Exp3 draws a solution $s_k \in S$ according to the distribution $P(k) = \langle p_1(k), \dots, p_{|S|}(k) \rangle$ independently of $s_1, \dots, s_{k-1}$ and $F(s_k, w_k)$ is obtained, where in fact w_k can be drawn from $\mathcal{P}$ at the iteration instead of being drawn in advance. $\hat{F}^T$ is updated based on only $F(s_k, w_k)$ and $\mu_s(k+1)$ is updated only for $s = s_k$, making $\mu(k)$ be incrementally updated only from $\mu_{s_k}(k+1)$. This implies that $P(k)$ is incrementally updated to $P(k+1)$, e.g., for s_k ,

$$p_{s_k}(k+1) = \frac{p_{s_k} - \frac{\gamma}{|S|}}{1 + \left(p_{s_k} - \frac{\gamma}{|S|} \right) \frac{\mu_{s_k}(k+1) - \mu_{s_k}(k)}{(1-\gamma)\mu_{s_k}(k)}} + \frac{\gamma}{|S|}.$$

The distribution $P(k)$ is a weighted sum (by the control parameter $\gamma \in (0, 1)$) of the uniform

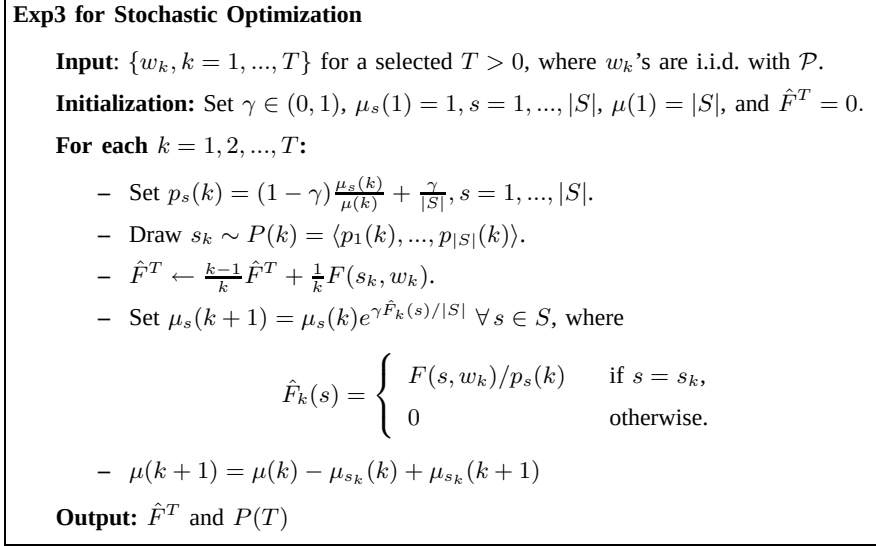


Fig. 1. Exp3 algorithm for stochastic optimization

distribution and a distribution which assigns to each solution $s \in S$ a probability mass which is exponential in the estimated cumulative bandit-reward $\sum_{k' \in \{k'' | s_{k''} = s, k'' = 1, \dots, k-1\}} \hat{F}_{k'}(s)$ for that solution s . This ensures that the algorithm tries out all $|S|$ solutions and obtains good estimates of the bandit-rewards for each $s \in S$ [4]. For the randomly selected solution s_k , Exp3 sets the estimated bandit-reward $\hat{F}_k(s_k)$ to $F(s_k, w_k)/p_{s_k}(k)$. This division compensates the bandit-reward of solutions that are unlikely to be chosen and the choice of estimated bandit-rewards makes their expectations equal to the actual bandit-rewards for each solution, i.e., $E_{P(k)}[\hat{F}_k(s) | s_1, \dots, s_{k-1}] = F(s, w_k), s \in S$, where the conditional expectation, denoted by $E_{P(k)}$, is taken only with respect to $P(k)$ given the past solution choices $s_1, \dots, s_{k-1}$.

As the Exp3 algorithm is randomized, the algorithm induces a joint probability distribution $\mathcal{P}_{1:T}^{\text{Exp3}}$ over the set of all possible sequences of solutions $s_k, k = 1, \dots, T$, obtained by the product of $P(1), \dots, P(T)$. In this section, the expectation $E[\cdot]$, denoted by $E_{\mathcal{P}_{1:T}^{\text{Exp3}}}$, will be taken with respect to this distribution. To achieve convergence of the performance made by Exp3, we need a proper “annealing” of the exploration-and-exploitation control parameter γ as a function of T . We set γ in Figure 1 to $\gamma(T)$ for a given T and make the value of $\gamma(T)$ be gradually decreased to zero as T increases while $\gamma(T)T$ approaches infinity. An example of such sequence is $\gamma(T) = T^{-1/2}$.

Theorem 2.1: Let $\{\gamma(T)\}$ be a decreasing sequence such that $\gamma(T) \in (0, 1)$ for all $T > 1$,

$\lim_{T \rightarrow \infty} \gamma(T) = 0$, and $\lim_{T \rightarrow \infty} \gamma(T)T = \infty$. Suppose that γ is set to $\gamma(T)$ in Figure 1 for a given $T > 1$. Then for any input data-sample sequence $\{w_k, k = 1, \dots, T\}$,

1. as $T \rightarrow \infty$, $E_{\mathcal{P}_{1:T}^{\text{Exp3}}}[\hat{F}^T] \rightarrow F^*$ w.p. 1.
2. as $T \rightarrow \infty$, $\{P(T)\}$ converges to $p^* \in \{p \in R^{|S|} \mid \sum_{s \in S} p(s) = 1, p(s) = 0 \forall s \notin S^*\}$ w.p. 1.

Proof: By Theorem 3.1 in [4], for any $T > 0$,

$$F_{\max}^T - E_{\mathcal{P}_{1:T}^{\text{Exp3}}} \left[\frac{1}{T} \sum_{k=1}^T F(s_k, w_k) \right] \leq (e-1)\gamma(T) + \frac{|S| \ln |S|}{\gamma(T)T},$$

which implies that $F^* \leq E_{\mathcal{P}_{1:T}^{\text{Exp3}}}[\hat{F}^T]$ as $T \rightarrow \infty$ because $F_{\max}^T$ converges to F^* by the law of large numbers [19] and $\gamma(T) \rightarrow 0$ and $\gamma(T)T \rightarrow \infty$ as $T \rightarrow \infty$. We show below that $F^* \geq E_{\mathcal{P}_{1:T}^{\text{Exp3}}}[\hat{F}^T]$ as $T \rightarrow \infty$ based on the fact that the sequence $\{P(T)\}$ converges to a stationary distribution as $T \rightarrow \infty$.

For any $s \in S$ and $T > 0$,

$$\frac{p_s(T) - \gamma(T)/|S|}{p_s(T+1) - \gamma(T)/|S|} = \frac{\frac{(1-\gamma(T))\mu_s(T)}{\mu(T)}}{\frac{(1-\gamma(T))\mu_s(T+1)}{\mu(T+1)}} = \frac{\mu(T+1)}{\mu(T)} \cdot \frac{1}{\exp(\gamma(T)\hat{F}_T(s)/|S|)} \quad (2)$$

$$\leq \frac{1}{\exp(\gamma(T)\hat{F}_T(s)/|S|)} \left(1 + \frac{\frac{\gamma(T)}{|S|}}{1 - \gamma(T)} F(s_T, w_T) + \frac{(e-2)\frac{\gamma(T)^2}{|S|^2}}{1 - \gamma(T)} \sum_{s=1}^{|S|} \hat{F}_T(s) \right) \quad (3)$$

by using inequality (8) in [4]

$$\leq 1 + \frac{\frac{\gamma(T)}{|S|}}{1 - \gamma(T)} + \frac{(e-2)\frac{\gamma(T)^2}{|S|^2}}{1 - \gamma(T)} \cdot \frac{|S|^2}{\gamma(T)} \quad (4)$$

because $0 \leq \hat{F}_T(s) \leq |S|/\gamma(T)$

$$= 1 + \frac{\gamma(T)}{1 - \gamma(T)} (|S|^{-1} + e - 2). \quad (5)$$

This implies that $\liminf_{T \rightarrow \infty} \frac{p_s(T)}{p_s(T+1)} \leq \limsup_{T \rightarrow \infty} \frac{p_s(T)}{p_s(T+1)} \leq 1$ for all $s \in S$. Define the set $\xi := \{s \in S : \liminf_{T \rightarrow \infty} \frac{p_s(T)}{p_s(T+1)} < 1\}$, and assume that $\xi \neq \emptyset$. For any $s' \in \xi$, let $\liminf_{T \rightarrow \infty} \frac{p_{s'}(T)}{p_{s'}(T+1)} = \alpha(s') < 1$. Thus, there exists a subsequence $\{\frac{p_{s'}(T_r)}{p_{s'}(T_r+1)}\}$ such that $\lim_{r \rightarrow \infty} \{\frac{p_{s'}(T_r)}{p_{s'}(T_r+1)}\} = \alpha(s') < 1$ and $\limsup_{r \rightarrow \infty} \{\frac{p_s(T_r)}{p_s(T_r+1)}\} \leq 1$ for all $s \in S$. It follows that as $r \rightarrow \infty$,

$$1 = \sum_{s \in S} p_s(T_r) = \sum_{s \in S \setminus \{s'\}} p_s(T_r) + p_{s'}(T_r) < \sum_{s \in S \setminus \{s'\}} p_s(T_r + 1) + p_{s'}(T_r + 1) = 1,$$

which is a contradiction, since both $p_s(T_r)$ and $p_s(T_r + 1)$ are probability distribution functions. Consequently, we must have that $\lim_{T \rightarrow \infty} \frac{p_s(T)}{p_s(T+1)} = 1$ for all $s \in S$. This implies that for any $n \geq 1$, as $T \rightarrow \infty$,

$$\frac{p_s(T)}{p_s(T+n)} = \frac{p_s(T)}{p_s(T+1)} \times \frac{p_s(T+1)}{p_s(T+2)} \times \dots \times \frac{p_s(T+n-1)}{p_s(T+n)} \rightarrow 1,$$

i.e.,

$$\lim_{T \rightarrow \infty} \sum_{s \in S} |p_s(T) - p_s(T+n)| = 0.$$

Therefore, for every $\epsilon > 0$, there exists $T' < \infty$ such that $\sum_{s \in S} |p_s(k) - p_s(k+n)| \leq \epsilon$ for all $k > T'$ and any integer $n \geq 1$. Then, for $T > T'$, we have (a.s.)

$$\begin{aligned} F_{\max}^T - E_{\mathcal{P}_{1:T}^{\text{Exp3}}} \left[\frac{1}{T} \sum_{k=1}^T F(s_k, w_k) \right] &= \max_{s \in S} \frac{1}{T} \sum_{k=1}^T F(s, w_k) - \frac{1}{T} \sum_{k=1}^T \sum_{s=1}^{|S|} p_s(k) F(s, w_k) \\ &= \max_{s \in S} \frac{1}{T} \sum_{k=1}^T F(s, w_k) - \frac{1}{T} \left(\sum_{k=1}^{T'} \sum_{s=1}^{|S|} p_s(k) F(s, w_k) + \sum_{k=T'+1}^T \sum_{s=1}^{|S|} p_s(k) F(s, w_k) \right) \end{aligned} \quad (6)$$

$$\begin{aligned} &\geq \max_{s \in S} \frac{1}{T} \sum_{k=1}^T F(s, w_k) - \frac{1}{T} \sum_{k=1}^{T'} \sum_{s=1}^{|S|} p_s(k) F(s, w_k) - \frac{1}{T} \sum_{k=1}^T \sum_{s=1}^{|S|} p_s(T'+1) F(s, w_k) \\ &\quad - \frac{1}{T} \sum_{k=T'+1}^T \sum_{s=1}^{|S|} |p_s(T'+1) - p_s(k)| F(s, w_k) \end{aligned} \quad (7)$$

$$\geq -\frac{1}{T} \sum_{k=1}^{T'} \sum_{s=1}^{|S|} p_s(k) F(s, w_k) - \frac{1}{T} \sum_{k=T'+1}^T \epsilon \quad (8)$$

$$\text{because } \max_{s \in S} \frac{1}{T} \sum_{k=1}^T F(s, w_k) - \frac{1}{T} \sum_{k=1}^T \sum_{s=1}^{|S|} p_s(T'+1) F(s, w_k) \geq 0,$$

which implies that $F^* \geq E_{\mathcal{P}_{1:T}^{\text{Exp3}}}[\hat{F}^T]$ as $T \rightarrow \infty$ because the first term in (8) vanishes to zero and ϵ in the second term can be chosen arbitrarily close to zero. This concludes the proof of the first part of the theorem.

For the second part, let the converged distribution for the sequence $\{P(T)\}$ be p^* . The convergence is obtained from the arguments given in the proof of the first part. We first show

that $F^* = \sum_{s \in S} p^*(s) E_w[F(s, w)]$. For the input data-sample sequence $\{w_k, k = 1, \dots, T\}$,

$$\begin{aligned}
& \max_{s \in S} \frac{1}{T} \sum_{k=1}^T F(s, w_k) - \frac{1}{T} \sum_{k=1}^T \sum_{s=1}^{|S|} p^*(s) F(s, w_k) \\
& \leq \max_{s \in S} \frac{1}{T} \sum_{k=1}^T F(s, w_k) - \frac{1}{T} \sum_{k=1}^T \sum_{s=1}^{|S|} p_s(k) F(s, w_k) + \frac{1}{T} \sum_{k=1}^T \sum_{s=1}^{|S|} |p_s(k) - p^*(s)| F(s, w_k) \quad (9) \\
& \leq (e - 1) \gamma(T) + \frac{|S| \ln |S|}{\gamma(T) T} + \frac{1}{T} \sum_{k=1}^T \sum_{s=1}^{|S|} |p_s(k) - p^*(s)| F(s, w_k). \quad (10)
\end{aligned}$$

Letting $T \rightarrow \infty$ at the both sides of the above inequality, we have that $F^* \leq \sum_{s \in S} p^*(s) E_w[F(s, w)]$. Furthermore, it is obvious that $F^* \geq \sum_{s \in S} p^*(s) E_w[F(s, w)]$. Then the second part of the theorem that $p^* \in \{p \in R^{|S|} \mid \sum_{s \in S} p(s) = 1, p(s) = 0 \forall s \notin S_\Psi\}$ as $T \rightarrow \infty$ follows directly from a proof obtained in a straightforward manner by assuming there exists $s' \in S$ such that $p^*(s') \neq 0$ and $E_w[F(s', w)] < F^*$, leading to a contradiction. We skip the details. ■

There exist some works on developing regret-based algorithms for *deterministic* optimization problems in a nonstochastic multi-armed bandit setting. Flaxman *et al.* [13] studied a general “on-line” deterministic convex optimization problem. At each time, an adversary generates a convex function unknown to the player and the player chooses a solution and observes only the function value evaluated at the solution. Based on an approximation of the gradient with a single sample, they provide an algorithm for solving the optimization problem with an analysis of the regret bound. Kleinberg [18] provides a different algorithm for the same problem and shows a similar regret bound which is proportional to a polynomial (square root) in the number of iterations. Hazan *et al.* [14] recently improved the regret bound, presenting some algorithms that achieve a regret proportional to the logarithm of the number of iterations.

Even though the idea of updating probability distributions over the solution space from a single sample-response from the execution of a sampled solution in Exp3 is similar to the learning automata approach (see, e.g., [23]), the Exp3 algorithm is based on the nonstochastic multi-armed bandit model.

In the following two sections, we apply the idea of solving non-sequential stochastic optimization problems in a nonstochastic bandit-setting to finite-horizon sequential stochastic optimization problems formulated by MDPs and to two-person zero-sum MGs.

III. RECURSIVE EXTENSION OF EXP3 FOR MDPs

A. Background

Consider a discrete-time dynamic system with a finite horizon $H < \infty$: $x_{i+1} = f(x_i, a_i, w_i)$ for $i = 0, 1, 2, \dots, H-1$, where x_i is the state of the system at stage i taking its value from an infinite state set X , a_i is the control to be chosen from a *finite* action set A at stage i , $\{w_i, i = 0, 1, \dots, H-1\}$ is a sequence of independent random variables, each uniformly distributed on $[0,1]$, representing the uncertainty in the system, and $f : X \times A \times [0,1] \rightarrow X$ is the next-state transition function.

Let Π be the set of all possible nonstationary Markovian policies $\pi = \{\pi_i, i = 0, \dots, H-1\}$, where $\pi_i : X \rightarrow A$. Defining the optimal reward-to-go value function for state x at stage i by

$$V_i^*(x) = \sup_{\pi \in \Pi} E_w \left[\sum_{t=i}^{H-1} R(x_t, \pi_t(x_t), w_t) \middle| x_i = x \right],$$

where $x \in X, w = (w_i, w_{i+1}, \dots, w_{H-1}), w_j \sim U(0,1), j = i, \dots, H-1$, with a bounded nonnegative reward function $R : X \times A \times [0,1] \rightarrow \mathbb{R}^+$, $V_H^*(x) = 0$ for all $x \in X$, and $x_t = f(x_{t-1}, \pi_{t-1}(x_{t-1}), w_{t-1})$. We wish to compute $V_0^*(x_0)$ and obtain an optimal policy $\pi^* \in \Pi$ that achieves $V_0^*(x_0), x_0 \in X$. For ease of exposition, we assume that R is bounded with $R_{\max} := \sup_{x \in X, a \in A, w \in [0,1]} R(x, a, w) \leq 1/H$, every action in A is admissible at every state, and the same random number is associated with the reward and transition functions.

It is well known (see, e.g., [22]) that V_i^* can be written recursively as follows: For all $x \in X$ and $i = 0, \dots, H-1$,

$$\begin{aligned} V_i^*(x) &= \max_{a \in A} Q_i^*(x, a), \text{ where} \\ Q_i^*(x, a) &= E_w[R(x, a, w) + V_{i+1}^*(f(x, a, w))], \end{aligned} \quad (11)$$

where $w \sim U(0,1)$ and $V_H^*(x) = 0, x \in X$.

Consider the SAA problem of obtaining $V_{T_0, \max}^0(x_0)$ (defined in (12)) under the assumption that the V_1^* -function is known. To estimate $V_0^*(x_0)$ for a given $x_0 \in X$, we apply Exp3: We sample T_0 independent random numbers $w_k^0 \sim U(0,1), k = 1, \dots, T_0$, and then obtain the sample-average-maximum, $V_{T_0, \max}^0(x_0)$, given by

$$V_{T_0, \max}^0(x_0) = \max_{a \in A} \frac{1}{T_0} \sum_{k=1}^{T_0} (R(x_0, a, w_k^0) + V_1^*(f(x_0, a, w_k^0))). \quad (12)$$

As $T_0 \rightarrow \infty$, $V_{T_0, \max}^0(x_0) \rightarrow V_0^*(x_0)$ w. p. 1 and has a positive bias such that $E_{w_1^0, \dots, w_{T_0}^0} [V_{T_0, \max}^0(x_0)] \geq V_0^*(x_0)$, $x_0 \in X$. As in Section II, the SAA problem can be cast into a nonstochastic bandit problem. Each action a is denoted by an integer in $\{1, \dots, |A|\}$. A sequence $\{r^0(1), r^0(2), \dots\}$ of bandit-reward vectors is assigned, where the bandit-reward vector $r^0(k) = (r_1^0(k), \dots, r_{|A|}^0(k))$, and $r_a^0(k) := R(x_0, a, w_k^0) + V_1^*(f(x_0, a, w_k^0)) \in [0, 1]$ denotes the bandit-reward obtained if action a is chosen at iteration k . Note that the same sequence of random numbers $W^0 = \{\omega_k^0, k = 1, 2, \dots\}$ is used to assign the bandit-reward sequence $\{r^0(k), k = 1, 2, \dots\}$.

We now relax the assumption that we know the exact V_1^* -values at all states. We then recursively estimate V_1^* -function values, yielding the following recursive equations: for $i = 0, \dots, H - 1$,

$$V_{T_i, \max}^i(x_i) = \max_{a \in A} \frac{1}{T_i} \sum_{k=1}^{T_i} \left(R(x_i, a, w_k^i) + V_{T_{i+1}, \max}^{i+1}(f(x_i, a, w_k^i)) \right), x_i \in X, \quad (13)$$

where $W^i = \{w_k^i, k = 1, \dots, T_i\}$ is a sequence of independently sampled random numbers from $[0, 1]$ for the estimate of the optimal reward-to-go value at stage i and $V_{T_H, \max}^H(x) = 0, x \in X$. $V_{T_{i+1}, \max}^{i+1}$ -values are computed only over T_i sampled states. Note that here we use *the same random number sequence* at each stage. That is, the same T_i -length random number sequence W^i is used to compute $V_{T_i, \max}^i(x)$ at stage i for any state x that is sampled from some state at stage $i - 1$. This results in a “*recursive sample average approximation*” for solving MDPs. Note also that the value of $V_{T_i, \max}^i(x_i)$ is an upper bound on the following recursively estimated sample-average approximate value $\hat{V}_{T_i}^*(x_i)$ given by

$$\hat{V}_{T_i}^*(x_i) = \frac{1}{T_i} \sum_{k=1}^{T_i} \left(R(x_i, \pi_i^*(x_i), w_k^i) + \hat{V}_{T_{i+1}}^*(f(x_i, \pi_i^*(x_i), w_k^i)) \right),$$

where the same random number sequences $W^i, \dots, W^{H-1}$ in (13) are used.

Extending the result of [19], we have that

$$\lim_{T_0 \rightarrow \infty} \lim_{T_1 \rightarrow \infty} \dots \lim_{T_{H-1} \rightarrow \infty} V_{T_0, \max}^0(x_0) = V_0^*(x_0), \text{ w.p. 1, } x_0 \in X.$$

With larger values of the T_i 's, we have more accurate estimates of $V_0^*(x_0)$. We then consider a “*recursive nonstochastic bandit problem*” to estimate $V_{T_0, \max}^0(x_0)$ and provide a player algorithm, called “Recursive Exp3 for MDPs,” (RExp3MDP) for the problem in Section III-B. The performance analysis is given in Section III-C.

B. Recursive Exp3 for MDPs

As in the Exp3 algorithm for stochastic optimization in Section II, RExp3MDP runs with pre-sampled random number sequences that will be used for creating a “sampled forward-tree”. We first select $T_i > 0, i = 0, \dots, H - 1$, and generate a sequence of T_i random numbers $W^i = \{w_k^i, k = 1, \dots, T_i\}$ for each $i = 0, \dots, H - 1$, where w_k^i 's are all independently sampled from $U(0, 1)$. We then construct a mapping $g_{T_i} : \{1, \dots, T_i\} \rightarrow [0, 1]$ for $i = 0, \dots, H - 1$ such that $g_{T_i}(k) = w_k^i$ for $k \in \{1, \dots, T_i\}$. That is, the mapping g_{T_i} provides a sampled random number to be used by RExp3MDP for a pair of stage i and iteration number at the stage i . Obtaining the mapping $g_{T_i}, i = 0, \dots, H - 1$, is a preprocessing task for running RExp3MDP. We refer to the mapping $g_{T_i}, i = 0, \dots, H - 1$, as “adversary mappings” because these mappings correspond to an adversary who generates arbitrary sequences of bandit-rewards.

Rewriting (13) with $g_{T_i}, i = 1, \dots, H - 1$, as

$$V_{T_i, \max}^i(x_i) = \max_{a \in A} \frac{1}{T_i} \sum_{k=1}^{T_i} \left(R(x_i, a, g_{T_i}(k)) + V_{T_{i+1}, \max}^{i+1}(f(x_i, a, g_{T_i}(k))) \right), \quad (14)$$

we wish to estimate $V_{T_0, \max}^0(x_0)$ based on the following recursive equation: for $x_i \in X, i = 0, \dots, H - 1$,

$$\hat{V}_{T_i}^i(x_i) = \frac{1}{T_i} \sum_{k=1}^{T_i} \left(R(x_i, a_k^i(x_i), g_{T_i}(k)) + \hat{V}_{T_{i+1}}^{i+1}(f(x_i, a_k^i(x_i), g_{T_i}(k))) \right), \quad (15)$$

where $\hat{V}_{T_H}^H(x_H) = 0, x_H \in X$, and $a_k^i(x_i)$ denotes a randomly selected action with respect to a probability distribution $P_{x_i}^i(k)$ generated by RExp3MDP for x_i at iteration k in stage i , i.e., $a_k^i(x_i)$ is a random variable that takes the value in $\{1, \dots, |A|\}$ with respect to $P_{x_i}^i(k)$ over A . RExp3MDP estimates $V_{T_i, \max}^i(x_i), i = 0, \dots, H - 1$, in (14) by $\hat{V}_{T_i}^i(x_i)$ in (15). By setting the given initial state x_0 to the root of a tree and setting all sampled states to the nodes of the tree where a directed arc from a node x to y is associated whenever $y = f(x, a, w)$ for some a and w , the sampling process by RExp3MDP based on (15) constructs a sampled forward-tree.

In the nonstochastic bandit-setting, at state x_i , a nonstochastic bandit problem with the bandit-reward sequence $\{r^i(k), k = 1, \dots, T_i\}$, where $r_a^i(k) := R(x_i, a, g_{T_i}(k)) + \hat{V}_{T_{i+1}}^{i+1}(f(x_i, a, g_{T_i}(k))), a \in A$, is being solved here by RExp3MDP to minimize the expected regret relative to

$$\max_{a \in A} \frac{1}{T_i} \sum_{k=1}^{T_i} \left(R(x_i, a, g_{T_i}(k)) + \hat{V}_{T_{i+1}}^{i+1}(f(x_i, a, g_{T_i}(k))) \right), \quad (16)$$

and the average value

$$\frac{1}{T_i} \sum_{k=1}^{T_i} \left(R(x_i, a_k^i(x_i), g_{T_i}(k)) + \hat{V}_{T_{i+1}}^{i+1}(f(x_i, a_k^i(x_i), g_{T_i}(k))) \right) = \hat{V}_{T_i}^i(x_i) \quad (17)$$

is in turn used as an estimate of $V_{T_i, \max}^i(x_i)$ in (14). If $r_a^i(k)$ is replaced with $R(x_i, a, g_{T_i}(k)) + V_{T_{i+1}, \max}^{i+1}(f(x_i, a, g_{T_i}(k)))$, then we have a (non-recursive version of) “one-level” nonstochastic bandit problem treated in Section II.

A high-level description of RExp3MDP to estimate $V_{T_0, \max}^0(x_0)$ for a given state x_0 is given in Figure 2. The inputs to RExp3MDP are a state $x \in X$, stage i , and the pre-computed adversary mapping g_{T_i} , and the output of RExp3MDP is $\hat{V}_{T_i}^i(x)$, an estimate of $V_{T_i, \max}^i(x)$. When we encounter $\hat{V}_{T_{i'}}^{i'}(y)$ for a state $y \in X$ and stage i' in the **For each** portion of the RExp3MDP algorithm, we call RExp3MDP recursively. The initial call to RExp3MDP is done with $i = 0$, g_{T_0} , and the initial state x_0 , and every sampling (for action selection) is done independently of the previous samplings. At each iteration $k = 1, \dots, T_i, i \neq H$, RExp3MDP draws an action $a_k^i(x)$ according to the distribution $P_x^i(k) = \langle p_{x,1}^i(k), \dots, p_{x,|A|}^i(k) \rangle$. As in Exp3, the distribution $P_x^i(k)$ is a weighted sum of the uniform distribution and a distribution which assigns to each action $a \in A$ a probability mass which is exponential in the estimated cumulative value of $\sum_{k' \in \{k'' | a_{k''}^i(x) = a, k'' = 1, \dots, k-1\}} \hat{Q}_{k'}^i(x, a)$ for that action a .

The running time-complexity of the RExp3MDP algorithm is $O((|A|T)^H)$ with $T = \max_i T_i$, and the space-complexity is $O(|A|T^H)$ because for each sampled state in the sampled forward-tree, a probability distribution over A needs to be maintained. The complexities are independent of the state space size $|X|$ but exponentially dependent on the horizon size H .

C. Analysis of RExp3MDP

As the RExp3MDP algorithm is randomized, the algorithm induces a probability distribution over the set of all possible sequences of randomly chosen actions. In this section, the expectation $E[\cdot]$ will be taken with respect to this distribution unless stated otherwise.

Theorem 3.1: Let $\{\gamma(T_i)\}$ be a decreasing sequence such that $\gamma(T_i) \in (0, 1)$ for all $T_i > 1$, $\lim_{T_i \rightarrow \infty} \gamma(T_i) = 0$, and $\lim_{T_i \rightarrow \infty} \gamma(T_i)T_i = \infty$, $i = 0, \dots, H-1$. Suppose that γ_i is set to $\gamma(T_i)$ in Figure 2 for a given $T_i > 1$. Then for all $x \in X$,

$$\lim_{T_0 \rightarrow \infty} \lim_{T_1 \rightarrow \infty} \cdots \lim_{T_{H-1} \rightarrow \infty} E[\hat{V}_{T_0}^0(x)] = V_0^*(x).$$

Recursive Exp3 for MDPs: RExp3MDP

Input: state $x \in X$, stage i , adversary mapping g_{T_i} . (For $i = H$, $\hat{V}_{T_H}^H(x) = 0$.)

Initialization: Set $\gamma_i \in (0, 1)$, $\mu_a(1) = 1$, $a = 1, \dots, |A|$, and $\hat{V}_{T_i}^i(x) = 0$.

For each $k = 1, 2, \dots, T_i$:

– Obtain $w_k^i = g_{T_i}(k)$.

– Set

$$p_{x,a}^i(k) = (1 - \gamma_i) \frac{\mu_a(k)}{\sum_{a'=1}^{|A|} \mu_{a'}(k)} + \frac{\gamma_i}{|A|}, a = 1, \dots, |A|.$$

– Draw $a_k^i(x) \sim P_x^i(k) = \langle p_{x,1}^i(k), \dots, p_{x,|A|}^i(k) \rangle$.

– Receive the bandit-reward of

$$Q_k^i(x, a_k^i(x)) := R(x, a_k^i(x), w_k^i) + \hat{V}_{T_{i+1}}^{i+1}(f(x, a_k^i(x), w_k^i)).$$

– $\hat{V}_{T_i}^i(x) \leftarrow \frac{k-1}{k} \hat{V}_{T_i}^i(x) + \frac{1}{k} Q_k^i(x, a_k^i(x))$.

– For $a = 1, \dots, |A|$, set

$$\hat{Q}_k^i(x, a) = \begin{cases} Q_k^i(x, a)/p_{x,a}^i(k), & \text{if } a = a_k^i(x) \\ 0 & \text{otherwise.} \end{cases}$$

$$\mu_a(k+1) = \mu_a(k) e^{\gamma_i \hat{Q}_k^i(x, a)/|A|}.$$

Output: $\hat{V}_{T_i}^i(x)$

Fig. 2. Pseudocode for RExp3MDP algorithm

Proof: The proof is simple. From the recursive equation of $\hat{V}_{T_{H-1}}^{H-1}$ in RExp3MDP,

$$\hat{V}_{T_{H-1}}^{H-1}(x) = \frac{1}{T_{H-1}} \sum_{k=1}^{T_{H-1}} \left(R(x, a_k^{H-1}(x), g_{T_{H-1}}(k)) + \hat{V}_{T_H}^H(f(x, a_k^{H-1}(x), g_{T_{H-1}}(k))) \right), x \in X,$$

because $\hat{V}_{T_H}^H(x) = V_H^*(x) = 0$, $x \in X$, we have that

$$E[\hat{V}_{T_{H-1}}^{H-1}(x)] \xrightarrow{T_{H-1} \rightarrow \infty} V_{H-1}^*(x), x \in X,$$

by Theorem 2.1. This in turn leads to $E[\hat{V}_{T_{H-2}}^{H-2}(x)] \rightarrow V_{H-2}^*(x)$ as $T_{H-2} \rightarrow \infty$ for arbitrary $x \in X$ from

$$\hat{V}_{T_{H-2}}^{H-2}(x) = \frac{1}{T_{H-2}} \sum_{k=1}^{T_{H-2}} \left(R(x, a_k^{H-2}(x), g_{T_{H-2}}(k)) + \hat{V}_{T_{H-1}}^{H-1}(f(x, a_k^{H-2}(x), g_{T_{H-2}}(k))) \right).$$

By an inductive argument, we have that

$$\lim_{T_0 \rightarrow \infty} \lim_{T_1 \rightarrow \infty} \dots \lim_{T_{H-1} \rightarrow \infty} E[\hat{V}_{T_0}^0(x)] = V_0^*(x) \text{ for all } x \in X,$$

which concludes the proof. ■

We remark that the result for the finite-iteration bound of Exp3 in Theorem 3.1 in [4] can be extended to RExp3MDP by an inductive argument: For any $g_{T_i}, i = 0, \dots, H - 1$, and for all $x \in X$,

$$V_{T_0, \max}^0(x) - E[\hat{V}_{T_0}^0(x)] \leq \sum_{i=0}^{H-1} \left((e - 1)\gamma_i + \frac{|A| \ln |A|}{\gamma_i T_i} \right).$$

We skip the details.

D. Related work

Recent theoretical progress (e.g., [8]–[10]) on the development of (simulation-based) algorithms has led to practically viable approaches to solving large MDPs. Algorithms closely related to RExp3MDP are “adaptive multi-stage sampling” (AMS) algorithm [8], which is based on the theory of the *classical* multi-armed bandit problems, and “recursive automata sampling algorithm” (RASA) [11], which is based on the theory of the learning automata.

The classical multi-armed bandit problem, originally studied by Robbins [25], models the trade-off between exploration and exploitation based on certain statistical assumptions on rewards obtained by the player. The UCB1 algorithm in [5] uses the “upper confidence bound” as a criterion for selecting between exploration and exploitation. AMS is based on the UCB1 algorithm, where the expected bias of AMS relative to $V_0^*(x_0), x_0 \in X$, converges to zero.

RASA extends in a recursive manner the learning automata Pursuit algorithm of Rajaraman and Sastry [23] designed for solving non-sequential stochastic optimization problems. RASA returns an estimate of both an optimal action from a given state and the corresponding optimal value. Based on the finite-iteration analysis of the Pursuit algorithm, the following probability bounds as a function of the number of samples are derived for a given initial state: (i) a lower bound on the probability that RASA will sample the optimal action; and (ii) an upper bound on the probability that the deviation between the true optimal value and the RASA estimate exceeds a given error.

Like AMS and RASA, RExp3MDP is in the framework of adaptive multi-stage sampling, and the expected value of the estimate returned by RExp3MDP converges to the optimal value of the original MDP problem in the limit. RExp3MDP from the nonstochastic bandit-setting is a counterpart algorithm to AMS and RASA. Considering efficiency and complexity, RExp3MDP

does not give any more than AMS and RASA for solving MDPs. However, making use of the common random numbers is first explored here for RExp3MDP; for the sampling complexity, RExp3MDP requires $O(TH)$ sampled random numbers whereas RASA does $O(T^H)$ sampled random numbers.

IV. TWO-PERSON ZERO-SUM MARKOV GAMES

A. Backgrounds

We first describe a simulation model for finite horizon two-person zero-sum MGs (for a substantial discussion on MGs, see, e.g., [12] [27] [3]).

Let X denote an infinite state set, and A and B finite sets of actions for the maximizer and the minimizer, respectively. For simplicity, we assume that every action in the respective set A and B is admissible, and that $|A| = |B|$. Define *mixed action* sets Δ and Θ such that Δ and Θ are the respective sets of all possible probability distributions over A and B . We denote $\delta(a)$ as the probability of taking action $a \in A$ by the mixed action $\delta \in \Delta$ and similarly for $\theta \in \Theta$.

The players play underlying actions simultaneously at each state, with complete knowledge of the state of the system but without knowing each other's current action. Once the actions $a \in A$ and $b \in B$ are taken at state $x \in X$, the state x transitions to a state $y = f(x, a, b, w) \in X$, $w \sim U(0, 1)$, according to the next state function $f : X \times A \times B \times [0, 1] \rightarrow X$, and the maximizer obtains the nonnegative real-valued reward of $R(x, a, b, w) \in \mathbb{R}^+$ (the minimizer obtains the negative of the reward). We assume that $\sup_{x \in X, a \in A, b \in B, w \in [0, 1]} R(x, a, b, w) \leq 1/H$ where the horizon length $H < \infty$.

Define a (nonstationary) policy $\pi = \{\pi_0, \dots, \pi_{H-1}\}$ as a sequence of mappings where each mapping $\pi_t : X \rightarrow \Delta$, $t = 0, \dots, H-1$, prescribes a mixed action to take at each state for the maximizer and let Π be the set of all such policies. We similarly define a policy ϕ and the set Φ for the minimizer.

Given a policy pair $\pi \in \Pi$ and $\phi \in \Phi$ and an initial state x at stage $i = 0, \dots, H-1$, we define *the value of the game* played with π and ϕ by the maximizer and the minimizer as

$$V_i(\pi, \phi)(x) := E \left\{ \sum_{t=i}^{H-1} \left(\sum_{a \in A} \sum_{b \in B} R(x_t, a, b, w_t) \pi_t(x_t)(a) \phi_t(x_t)(b) \right) \middle| x_i = x \right\},$$

where x_t is a random variable denoting the state at time t following the policies π and ϕ . We assume that $V_H(\pi, \phi)(x) = 0$, $x \in X$, for all $\pi \in \Pi$ and $\phi \in \Phi$.

The maximizer (the minimizer) wishes to find a policy $\pi \in \Pi$ ($\phi \in \Phi$) which maximizes (minimizes) the value of the game for stage 0. It is known (see, e.g., [27] [3]) that there exists an *optimal equilibrium policy pair* $\pi^* \in \Pi$ and $\phi^* \in \Phi$ such that for all $\pi \in \Pi$, $\phi \in \Phi$, and $x_0 \in X$,

$$V_0(\pi, \phi^*)(x_0) \leq V_0(\pi^*, \phi^*)(x_0) \leq V_0(\pi^*, \phi)(x_0).$$

We will refer to the value $V_0(\pi^*, \phi^*)(x_0)$ as the *equilibrium value of the game* associated with state x_0 and to π^* and ϕ^* as the *equilibrium policies* for the maximizer and the minimizer, respectively. By a slight abuse of notation, and for consistency with Section III, we write $V_i(\pi^*, \phi^*)$, $i = 0, \dots, H$ as V_i^* . Our goal is to approximate $V_0^*(x_0)$, $x_0 \in X$.

As in MDPs, the following recursive equilibrium equations hold (see, e.g., [27] [3]): for $i = 0, 1, \dots, H - 1$, $x \in X$,

$$V_i^*(x) = \sup_{\delta \in \Delta} \inf_{\theta \in \Theta} \left\{ \sum_{a \in A} \sum_{b \in B} \delta(a) \theta(b) Q_i^*(x, a, b) \right\} \quad (18)$$

$$= \inf_{\theta \in \Theta} \sup_{\delta \in \Delta} \left\{ \sum_{a \in A} \sum_{b \in B} \delta(a) \theta(b) Q_i^*(x, a, b) \right\}, \text{ where} \quad (19)$$

$$Q_i^*(x, a, b) = E_w[R(x, a, b, w) + V_{i+1}^*(f(x, a, b, w))], \quad (20)$$

where $w \sim U(0, 1)$ and $V_H^*(x) = 0$.

Let $J_i > 0$ be the number of random observations allocated to each action pair a and b at stage i . We start by sampling independently $\sum_{i=0}^{H-1} J_i$ random numbers from $U(0, 1)$ and constructing mappings g_{J_i} , $i = 0, \dots, H - 1$, such that $g_{J_i}(j)$ is the j th sampled random number ($j = 1, \dots, J_i$) for stage i . We then estimate V_i^* -function values based on the following recursive equations: for $i = 0, \dots, H - 1$, $x_i \in X$,

$$V_{J_i, \text{eq}}^i(x_i) = \sup_{\delta \in \Delta} \inf_{\theta \in \Theta} \frac{1}{J_i} \sum_{j=1}^{J_i} \left(\sum_{a \in A} \sum_{b \in B} \delta(a) \theta(b) Q_{J_i, \text{eq}}^i(x_i, a, b, g_{J_i}(j)) \right) \quad (21)$$

$$= \inf_{\theta \in \Theta} \sup_{\delta \in \Delta} \frac{1}{J_i} \sum_{j=1}^{J_i} \left(\sum_{a \in A} \sum_{b \in B} \delta(a) \theta(b) Q_{J_i, \text{eq}}^i(x_i, a, b, g_{J_i}(j)) \right), \quad (22)$$

where

$$Q_{J_i, \text{eq}}^i(x_i, a, b, g_{J_i}(j)) = R(x_i, a, b, g_{J_i}(j)) + V_{J_{i+1}, \text{eq}}^{i+1}(f(x_i, a, b, g_{J_i}(j))),$$

and $V_{J_H, \text{eq}}^H(x) := 0$, $x \in X$. We now have a “recursive sample average approximation game.” We refer to $V_{J_i, \text{eq}}^i(x)$ as the *sample-average equilibrium value* associated with $x \in X$ at stage

i. Note that the supremum and infimum operators are interchangeable in (21) and (22) because we can view the game at stage i as a one-shot bimatrix game defined by an $|A| \times |B|$ -matrix, where the row player chooses a row a in A and the column player chooses a column b in B and the row player then gains the quantity $J_i^{-1} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x_i, a, b, g_{J_i}(j))$.

By an inductive argument based on (21) and (22), it is easy to show that

$$\lim_{J_0 \rightarrow \infty} \lim_{J_1 \rightarrow \infty} \cdots \lim_{J_{H-1} \rightarrow \infty} V_{J_0, \text{eq}}^0(x_0) = V_0^*(x_0) \quad w.p.1, \quad x_0 \in X,$$

as in MDPs, i.e., with larger values of J_i 's, we have more accurate estimates of $V_0^*(x_0)$.

B. Recursive Exp3 for MGs

“Recursive Exp3 for MGs” (RExp3MG) is a recursive extension of the Exp3 algorithm to solving MGs in an *adaptive* adversarial bandit framework [4].

Consider a repeated bimatrix zero-sum game defined by an $|A| \times |B|$ -matrix, where the row player chooses a row a in A and the column player chooses a column b in B . At the k th round (iteration) of the game, if the row player plays a row a_k and the column player plays a column b_k , the row player gains the quantity $J_i^{-1} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x_i, a_k, b_k, g_{J_i}(j))$. For this game, we can consider applying the Exp3 algorithm for the maximizer by viewing the bandit-reward $r_{a_k}(k)$, received by the maximizer at iteration k , as $J_i^{-1} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x_i, a_k, b_k, g_{J_i}(j))$ incurred to the maximizer at the k th round of the game. However, unlike the adversarial or nonstochastic bandit setting we considered in Section II, the bandit-reward $J_i^{-1} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x_i, a_k, b_k, g_{J_i}(j))$ depends on the randomized choices of the maximizer and the minimizer, which in turn, are functions of their realized payoffs. In other words, the bandit-reward function of the maximizer for the round k is chosen by an adaptive adversary who knows the maximizer's playing strategy and the outcome of the maximizer's random draws up to iteration $k - 1$.

Auer *et al.* remarked (without formal proofs) that all of their results for the nonstochastic or adversarial bandit setting still hold for the adaptive adversarial bandit setting (see the remarks in [4, p.72]). In particular, they showed in Theorem 9.3 [4] that for a given repeated bimatrix game, if the row player uses their Exp3.1 algorithm, then the row player's expected payoff per round converges to the optimal maximin payoff. The proof is based on an extended result of Theorem 4.1 [4] for the nonadaptive adversary case into the one for the adaptive adversary case. By observing that the result for the maximizer holds for *any* (randomized) playing strategy

by the minimizer, the minimizer can symmetrically employ the Exp3.1 algorithm such that the minimizer's expected payoff per round also converges to the optimal minimax payoff. In other words, Auer *et al.* showed that *if both players play according to the Exp3.1 algorithm, then the expected payoff per round converges to the equilibrium value of the game* (see, also Section 7.3 in [6] for a related discussion). Extending the result into our setting of the repeated bimatrix game, this fact corresponds to

$$\lim_{J_i \rightarrow \infty} \frac{1}{J_i} \sum_{j=1}^{J_i} \left(\lim_{T_i \rightarrow \infty} \frac{1}{T_i} \sum_{k=1}^{T_i} E[Q_{J_i, \text{eq}}^i(x_i, a_k, b_k, g_{J_i}(j))] \right) = V_i^*(x),$$

where we assume that V_{i+1}^* -function is known and replaces $V_{J_{i+1}, \text{eq}}^{i+1}$, and a_k and b_k are independently chosen by the Exp3 algorithm played by the maximizer and the minimizer, respectively. The expectation is taken over the joint probability distribution over the set of all possible sequences of action pairs $(a_k, b_k), k = 1, \dots, T_i$, obtained by the product of the distributions over $A \times B$ at $k = 1, \dots, T_i$ where at each k , the distribution is given as the product of the distributions over A and B generated by the Exp3 algorithm for the maximizer and the minimizer, respectively.

RExp3MG is based on this observation and its analysis extends Theorem 9.3 in [4] within our simulation model framework. Note that RExp3MG employs Exp3 instead of Exp3.1 due to its simplicity; it can be easily modified to be used with Exp3.1.

A high-level description of RExp3MG to estimate $V_{J_0, \text{eq}}^0(x_0)$ in (21) for a given state x_0 is given in Figure 3. The inputs to RExp3MG are a state $x \in X$, adversary mapping g_{J_i} , and stage i , and the output of RExp3MG is $\hat{V}_{T_i}^i(x)$, an estimate of $V_{J_i, \text{eq}}^i(x)$. When we encounter $\hat{V}_{T_{i'}}^{i'}(y)$ for a state $y \in X$ and stage i' in the **For each** portion of the RExp3MG algorithm, we call RExp3MG recursively. For each action pair sampled at stage i , J_i recursive calls need to be made. The initial call to RExp3MG is done with $i = 0$, the initial state x_0 , and g_{J_0} , and every sampling is done independently of the previous samplings. The running time-complexity is $O((|A|JT)^H)$ with $T = \max_i T_i$ and $J = \max_i J_i$ and the space-complexity is $O(|A|(JT)^H)$. The complexities are independent of the state space size $|X|$ but exponentially dependent on the horizon size H .

At each iteration $k = 1, \dots, T_i$, for stage $i \neq H$, RExp3MG draws an action $a_k^i(x) \in A$ according to the distribution $\alpha_x^i(k) = \langle \alpha_{x,1}^i(k), \dots, \alpha_{x,|A|}^i(k) \rangle$ for the maximizer, and $b_k^i(x) \in B$ according to $\beta_x^i(k) = \langle \beta_{x,1}^i(k), \dots, \beta_{x,|B|}^i(k) \rangle$ for the minimizer. For the randomly selected action

pair of $a_k^i(x)$ and $b_k^i(x)$, the maximizer receives the bandit-reward of

$$Q_k^i(x, a_k^i(x), b_k^i(x)) := \frac{1}{J_i} \sum_{j=1}^{J_i} \left(R(x, a_k^i(x), b_k^i(x), g_{J_i}(j)) + \hat{V}_{T_{i+1}}^{i+1}(f(x, a_k^i(x), b_k^i(x), g_{J_i}(j))) \right),$$

and RExp3MG sets the estimated bandit-reward

$$\hat{Q}_k^i(x, a_k^i(x)) = \frac{Q_k^i(x, a_k^i(x), b_k^i(x))}{\alpha_{x, a_k^i(x)}^i(k)}$$

for the maximizer and

$$\tilde{Q}_k^i(x, b_k^i(x)) = -\frac{Q_k^i(x, a_k^i(x), b_k^i(x))}{\beta_{x, b_k^i(x)}^i(k)}$$

for the minimizer, respectively. These estimated bandit-rewards are then used for updating the mixed actions $\alpha_x^i(k)$ and $\beta_x^i(k)$ for the maximizer and the minimizer, respectively.

In fact, the estimated bandit-reward $\hat{Q}_k^i(x, a_k^i(x))$ for the maximizer corresponds to the estimated bandit-reward $\hat{F}_k(s_k)$ with $s_k = a_k^i(x)$ in Figure 1 if we consider the one-level approximation game obtained by replacing the bandit-reward of $Q_k^i(x, a_k^i(x), b_k^i(x))$ with $J_i^{-1} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a_k^i(x), b_k^i(x), g_{J_i}(j))$. In this case, Exp3 in Figure 1 is invoked for solving an adaptive adversarial multi-armed bandit problem. Then the expected performance of Exp3 for the adaptive adversarial problem corresponds to the expected performance of Exp3 in Figure 1 for the nonadaptive adversarial problem defined with the bandit-reward assignment of $r_a(k) := E_{\beta_x^i(k)}[J_i^{-1} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a, b_k^i(x), g_{J_i}(j))]$, $a \in A$, at iteration k . By then applying the result of [4, Corollary 3.2] for the nonadaptive adversarial problem, we have the following performance bound for the adaptive adversarial problem by Exp 3 (with a proper setting of γ):

$$\begin{aligned} & \max_{a \in A} \frac{1}{T_i} \sum_{k=1}^{T_i} E_{\beta_x^i(k)} \left[\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a, b_k^i(x), g_{J_i}(j)) \right] \\ & - E_{\alpha_{1:T_i}^{\text{Exp3}}} \left[\frac{1}{T_i} \sum_{k=1}^{T_i} E_{\beta_x^i(k)} \left[\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a_k^i(x), b_k^i(x), g_{J_i}(j)) \right] \right] \leq O(\sqrt{|A| \ln |A| / T_i}), \quad (23) \end{aligned}$$

where $\alpha_{1:T}^{\text{Exp3}}$ is the joint probability distribution, similar to $\mathcal{P}_{1:T}^{\text{Exp3}}$, over the set of all possible sequences of actions $a_k^i(x)$, $k = 1, \dots, T_i$, selected by the employed Exp3 algorithm for the adaptive adversarial problem. This result will be used for analyzing the expected performance of RExp3MG.

RExp3 for Markov Games (RExp3MG)

Input: stage $i < H$, state $x \in X$, adversary mapping g_{J_i} . (For $i = H$, $\hat{V}_{T_H}^H(x) = 0$.)

Initialization: Set $\gamma_i \in (0, 1)$, $\mu_a(1) = 1$, $a = 1, \dots, |A|$,

$v_b(1) = 1$, $b = 1, \dots, |B|$, and $\hat{V}_{T_i}^i(x) = 0$.

For each $k = 1, 2, \dots, T_i$:

- Set $\alpha_{x,a}^i(k) = (1 - \gamma_i) \frac{\mu_a(k)}{\sum_{a'=1}^{|A|} \mu_{a'}(k)} + \frac{\gamma_i}{|A|}$, $a = 1, \dots, |A|$.
- Set $\beta_{x,b}^i(k) = (1 - \gamma_i) \frac{v_b(k)}{\sum_{b'=1}^{|B|} v_{b'}(k)} + \frac{\gamma_i}{|B|}$, $b = 1, \dots, |B|$.
- Draw $a_k^i(x) \sim \alpha_x^i(k) = \langle \alpha_{x,1}^i(k), \dots, \alpha_{x,|A|}^i(k) \rangle$ for the maximizer.
- Draw $b_k^i(x) \sim \beta_x^i(k) = \langle \beta_{x,1}^i(k), \dots, \beta_{x,|B|}^i(k) \rangle$ for the minimizer.
- Receive the bandit-reward of

$$Q_k^i(x, a_k^i(x), b_k^i(x)) := \frac{1}{J_i} \sum_{j=1}^{J_i} \left(R(x, a_k^i(x), b_k^i(x), g_{J_i}(j)) + \hat{V}_{T_{i+1}}^{i+1}(f(x, a_k^i(x), b_k^i(x), g_{J_i}(j))) \right). \quad (24)$$

- $\hat{V}_{T_i}^i(x) \leftarrow \frac{k-1}{k} \hat{V}_{T_i}^i(x) + \frac{1}{k} Q_k^i(x, a_k^i(x), b_k^i(x))$.
- **Maximizer:** For $a = 1, \dots, |A|$, set

$$\hat{Q}_k^i(x, a) = \begin{cases} Q_k^i(x, a, b_k^i(x)) / \alpha_{x,a}^i(k), & \text{if } a = a_k^i(x) \\ 0 & \text{otherwise.} \end{cases}$$

$$\mu_a(k+1) = \mu_a(k) e^{\gamma_i \hat{Q}_k^i(x, a) / |A|}.$$

- **Minimizer:** For $b = 1, \dots, |B|$, set

$$\tilde{Q}_k^i(x, b) = \begin{cases} -Q_k^i(x, a_k^i(x), b) / \beta_{x,b}^i(k), & \text{if } b = b_k^i(x) \\ 0 & \text{otherwise.} \end{cases}$$

$$v_b(k+1) = v_b(k) e^{\gamma_i \tilde{Q}_k^i(x, b) / |B|}.$$

Output: $\hat{V}_{T_i}^i(x)$

Fig. 3. Pseudocode for RExp3MG algorithm

C. Analysis of RExp3MG

The lemma below provides a finite-iteration bound to the sample-average equilibrium value associated with the recursive SAA game induced from $J_i, i = 0, \dots, H - 1$, samples on the expected value of RExp3MG's estimate output for a given stage under the assumption that the sample-average equilibrium value of the next stage at each state is known. The result is an extension of Theorem 9.3 in [4], which is based on Auer *et al.*'s Exp3.1 algorithm instead of Exp3. We provide the proof for the sake of completeness.

Let A_x^q and B_x^q be the sequence of the randomly selected actions (or random variables) by the maximizer and the minimizer, respectively, at state x and at stage q .

Lemma 4.1: Assume that a non-recursive version of RExp3MG in Figure 3 is run with the bandit-reward in (24) replaced by $J_i^{-1} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x_i, a_k^i(x), b_k^i(x), g_{J_i}(j))$ and $\gamma_i = \sqrt{\frac{|A| \ln |A|}{(e-1)T_i}}$, $T_i > \frac{|A| \ln |A|}{e-1}$ for all $i = 0, \dots, H-1$. Then for any g_{J_i} , for all $x \in X$, and $i = 0, \dots, H-1$,

$$\left| \frac{1}{T_i} E_{A_x^i, B_x^i} \left[\sum_{k=1}^{T_i} \left(\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a_k^i(x), b_k^i(x), g_{J_i}(j)) \right) \right] - V_{J_i, \text{eq}}^i(x) \right| \leq O \left(\sqrt{|A| \ln |A| / T_i} \right).$$

Proof: Fix any $i \neq H$ and $x \in X$. Let $\epsilon(T_i) = O \left(\sqrt{|A| \ln |A| / T_i} \right)$.

$$\frac{1}{T_i} E_{A_x^i, B_x^i} \left[\sum_{k=1}^{T_i} \left(\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a_k^i(x), b_k^i(x), g_{J_i}(j)) \right) \right] \quad (25)$$

$$\geq \max_{a \in A} \frac{1}{T_i} \sum_{k=1}^{T_i} E_{\beta_x^i(k)} \left[\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a, b_k^i(x), g_{J_i}(j)) \right] - \epsilon(T_i) \quad (26)$$

by [4, Corollary 3.2] with the bandit-reward assignment of

$$r_a(k) := E_{\beta_x^i(k)} \left[\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a, b_k^i(x), g_{J_i}(j)) \right] \text{ at } k, a \in A$$

$$\geq \frac{1}{T_i} \sum_{a \in A} \delta_*^i(a) \left(\sum_{k=1}^{T_i} E_{\beta_x^i(k)} \left[\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a, b_k^i(x), g_{J_i}(j)) \right] \right) - \epsilon(T_i) \quad (27)$$

$$= \frac{1}{T_i} \sum_{k=1}^{T_i} \left(\sum_{a \in A} \sum_{b \in B} \delta_*^i(a) \theta_k^i(b) \left(\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a, b, g_{J_i}(j)) \right) \right) - \epsilon(T_i) \quad (28)$$

$$\geq V_{J_i, \text{eq}}^i(x) - \epsilon(T_i), \quad (29)$$

where δ_*^i is a mixed action that satisfies

$$\begin{aligned} V_{J_i, \text{eq}}^i(x) &= \sup_{\delta \in \Delta} \inf_{\theta \in \Theta} \left(\sum_{a \in A} \sum_{b \in B} \delta(a) \theta(b) \left(\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a, b, g_{J_i}(j)) \right) \right) \\ &= \inf_{\theta \in \Theta} \left(\sum_{a \in A} \sum_{b \in B} \delta_*^i(a) \theta(b) \left(\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a, b, g_{J_i}(j)) \right) \right) \end{aligned}$$

and $\{\theta_k^i\}$ is the sequence of probability distributions over B with $\theta_k^i(b_k^i(x)) = 1$ for all $k = 1, \dots, T_i$.

The steps for the minimizer part in RExp3MG work with the negative bandit-reward of the maximizer, symmetrically to the maximizer part. This corresponds to applying Theorem 2.1 (cf., also Corollary 3.2 in [4]) with a loss model (in the adaptive adversarial bandit setting) where the bandit-rewards fall in the range $[-1, 0]$ (see the remark in [4, p.54]). Therefore, similar to the maximizer case, we have that

$$\frac{1}{T_i} E_{A_x^i, B_x^i} \left[\sum_{k=1}^{T_i} \left(\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a_k^i(x), b_k^i(x), g_{J_i}(j)) \right) \right] \quad (30)$$

$$\leq \min_{b \in B} \frac{1}{T_i} \sum_{k=1}^{T_i} E_{\alpha_x^i(k)} \left[\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a_k^i(x), b, g_{J_i}(j)) \right] + \epsilon(T_i) \quad (31)$$

$$\leq \frac{1}{T_i} \sum_{b \in B} \theta_*^i(b) \left(\sum_{k=1}^{T_i} E_{\alpha_x^i(k)} \left[\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a_k^i(x), b, g_{J_i}(j)) \right] \right) + \epsilon(T_i) \quad (32)$$

$$= \frac{1}{T_i} \sum_{k=1}^{T_i} \left(\sum_{a \in A} \sum_{b \in B} \theta_*^i(b) \delta_k^i(a) \left(\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a, b, g_{J_i}(j)) \right) \right) + \epsilon(T_i) \quad (33)$$

$$\leq V_{J_i, \text{eq}}^i(x) + \epsilon(T_i), \quad (34)$$

where θ_*^i is a mixed action that satisfies

$$\begin{aligned} V_{J_i, \text{eq}}^i(x) &= \inf_{\theta \in \Theta} \sup_{\delta \in \Delta} \frac{1}{J_i} \sum_{j=1}^{J_i} \left(\sum_{a \in A} \sum_{b \in B} \delta(a) \theta(b) Q_{J_i, \text{eq}}^i(x_i, a, b, g_{J_i}(j)) \right) \\ &= \sup_{\delta \in \Delta} \frac{1}{J_i} \sum_{j=1}^{J_i} \left(\sum_{a \in A} \sum_{b \in B} \delta(a) \theta_*^i(b) Q_{J_i, \text{eq}}^i(x_i, a, b, g_{J_i}(j)) \right) \end{aligned}$$

and $\{\delta_k^i\}$ is the sequence of probability distributions over A with $\delta_k^i(a_k^i(x)) = 1$ for all $k = 1, \dots, T_i$.

Therefore, we have that

$$\left| \frac{1}{T_i} E_{A_x^i, B_x^i} \left[\sum_{k=1}^{T_i} \left(\frac{1}{J_i} \sum_{j=1}^{J_i} Q_{J_i, \text{eq}}^i(x, a_k^i(x), b_k^i(x), g_{J_i}(j)) \right) \right] - V_{J_i, \text{eq}}^i(x) \right| \leq O \left(\sqrt{|A| \ln |A| / T_i} \right).$$

■

Suppose that RExp3MG is called at stage i for state x . As RExp3MG is randomized, it induces a probability distribution over the set of all possible sequences of action pairs randomly selected according to the action pair sampling mechanism of the algorithm. We use the notation $E_x^i[\cdot]$ to refer to the expectation taken with respect to this distribution.

Theorem 4.1: Suppose that RExp3MG is run with $\gamma_i = \sqrt{\frac{|A| \ln |A|}{(e-1)T_i}}$ and $T_i > \frac{|A| \ln |A|}{e-1}$ for all $i = 0, \dots, H-1$. Then for any $g_{J_i}, i = 0, \dots, H-1$, and for all $x \in X$,

$$\left| V_{J_0, \text{eq}}^0(x) - E_x[\hat{V}_{T_0}^0(x)] \right| \leq \sum_{i=0}^{H-1} O\left(\sqrt{|A| \ln |A| / T_i}\right).$$

Proof: For the value of $\hat{V}_{T_i}^i(x), x \in X, i = 0, 1, \dots, H-2$, at **Output** in RExp3MG, we have that

$$E_x[\hat{V}_{T_i}^i(x)] = \frac{1}{T_i} E_x \left[\sum_{k=1}^{T_i} \left(\frac{1}{J_i} \sum_{j=1}^{J_i} \left(R(x, a_k^i(x), b_k^i(x), g_{J_i}(j)) + \hat{V}_{T_{i+1}}^{i+1}(y_{i,k}^j) \right) \right) \right], \quad (35)$$

$$\begin{aligned} & \text{where } y_{i,k}^j := f(x, a_k^i(x), b_k^i(x), g_{J_i}(j)), k = 1, \dots, T_i \\ &= \frac{1}{T_i} E_x \left[\sum_{k=1}^{T_i} \left(\frac{1}{J_i} \sum_{j=1}^{J_i} \left(R(x, a_k^i(x), b_k^i(x), g_{J_i}(j)) + V_{J_{i+1}, \text{eq}}^{i+1}(y_{i,k}^j) \right) \right) \right] \\ & \quad + \frac{1}{T_i} E_x \left[\sum_{k=1}^{T_i} \left(\frac{1}{J_i} \sum_{j=1}^{J_i} \left(\hat{V}_{T_{i+1}}^{i+1}(y_{i,k}^j) - V_{J_{i+1}, \text{eq}}^{i+1}(y_{i,k}^j) \right) \right) \right] \end{aligned} \quad (36)$$

$$\begin{aligned} &\geq V_{J_i, \text{eq}}^i(x) - O\left(\sqrt{|A| \ln |A| / T_i}\right) \\ & \quad + \frac{1}{T_i} E_x \left[\sum_{k=1}^{T_i} \left(\frac{1}{J_i} \sum_{j=1}^{J_i} \left(\hat{V}_{T_{i+1}}^{i+1}(y_{i,k}^j) - V_{J_{i+1}, \text{eq}}^{i+1}(y_{i,k}^j) \right) \right) \right] \text{ by Lemma 4.1} \end{aligned} \quad (37)$$

$$\geq V_{T_i, \text{eq}}^i(x) - O\left(\sqrt{|A| \ln |A| / T_i}\right) + \min_{y_{i,k}^j} \left(E_{y_{i,k}^j}^{i+1}[\hat{V}_{T_{i+1}}^{i+1}(y_{i,k}^j)] - V_{J_{i+1}, \text{eq}}^{i+1}(y_{i,k}^j) \right). \quad (38)$$

Now for $i = H-1$, because $V_{J_H, \text{eq}}^H(z) = \hat{V}_{T_H}^H(z) = 0, z \in X$,

$$E_x^{H-1}[\hat{V}_{T_{H-1}}^{H-1}(x)] \geq V_{J_{H-1}, \text{eq}}^{H-1}(x) - O\left(\sqrt{|A| \ln |A| / T_{H-1}}\right), x \in X.$$

For $i = H-2$, using the above inequality,

$$\begin{aligned} E_x^{H-2}[\hat{V}_{T_{H-2}}^{H-2}(x)] &\geq V_{J_{H-2}, \text{eq}}^{H-2}(x) - O\left(\sqrt{|A| \ln |A| / T_{H-2}}\right) \\ & \quad + \min_{y_{H-2,k}^j} \left(E_{y_{H-2,k}^j}^{H-1}[\hat{V}_{T_{H-1}}^{H-1}(y_{H-2,k}^j)] - V_{J_{H-1}, \text{eq}}^{H-1}(y_{H-2,k}^j) \right) \end{aligned} \quad (39)$$

$$\geq V_{J_{H-2}, \text{eq}}^{H-2}(x) - O\left(\sqrt{|A| \ln |A| / T_{H-1}}\right) - O\left(\sqrt{|A| \ln |A| / T_{H-2}}\right), x \in X, \quad (40)$$

where $y_{H-2,k}^j = f(x, a_k^{H-1}(x), b_k^{H-1}(x), g_{J_{H-1}}(j))$. Continuing this way, we have that for $x \in X$,

$$E_x^0[\hat{V}_{T_0}^0(x)] \geq V_{J_0, \text{eq}}^0(x) - \sum_{i=0}^{H-1} O\left(\sqrt{|A| \ln |A| / T_i}\right). \quad (41)$$

The upper bound case can be shown similarly. We skip the details. ■

The following result is then immediate:

Theorem 4.2: Suppose that RExp3MG is run with $\gamma_i = \sqrt{\frac{|A|\ln|A|}{(e-1)T_i}}$ and $T_i > \frac{|A|\ln|A|}{e-1}$ for all $i = 0, \dots, H-1$. Then for all $x \in X$,

$$\lim_{J_0 \rightarrow \infty} \lim_{J_1 \rightarrow \infty} \cdots \lim_{J_{H-1} \rightarrow \infty} \left(\lim_{T_0 \rightarrow \infty} \lim_{T_1 \rightarrow \infty} \cdots \lim_{T_{H-1} \rightarrow \infty} E[\hat{V}_{T_0}^0(x)] \right) = V_0^*(x).$$

Kearns *et al.* [17] studied a *nonadaptive* sampling algorithm for a finite horizon two-person zero-sum game and analyzed the sampling complexity for a desired approximation guarantee for the equilibrium value. The algorithm is to create a sampled forward-tree with a fixed width C such that at each sampled state $x \in X$, C next states are sampled for each action pair in $A \times B$ and by using the estimated value of the game at the next states, the value of the game at x is evaluated by using infsup-operators. Note that in RExp3MG, no infsup-valuation is necessary.

There exist some convergent simulation-based algorithms for learning an optimal equilibrium policy pair in *one-stage* ($H = 1$ and $|A| = |B| = 2$ in our setting) two-person zero-sum games based on the theory of learning automata [20] [21]. The algorithms basically update the probability distributions over the action sets of the players from the observed payoffs via simulation, similar to the learning-automata based algorithms for MDPs [23]. Sastry *et al.* [26] consider one-stage games with a more general setup in a multi-player setting.

V. CONCLUDING REMARKS

The solutions presented in this paper to finite horizon problems can be also used for solving infinite horizon problems in the receding/rolling horizon control framework [7] [15].

Interesting future work would be recursively extending the Exp3.P algorithm in [4] for MDPs and analyzing a probability bound that holds uniformly over sampling size as in Theorem 6.3 in [4].

REFERENCES

- [1] R. Agrawal, D. Teneketzis, and V. Anantharam, "Asymptotically efficient adaptive allocation schemes for controlled Markov chains: finite parameter space," *IEEE Trans. on Automatic Control*, vol. 34, pp. 1249–1259, 1989.
- [2] S. Ahmed and A. Shapiro. "The Sample Average Approximation Method for Stochastic Programs with Integer Recourse," *Optimization Online*, <http://www.optimization-online.org>, 2002.
- [3] E. Altman, "Zero-sum Markov games and worst-cast optimal control of queueing systems," *QUESTA*, vol. 21, pp. 415–447, 1995.

- [4] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, “The nonstochastic multiarmed bandit problem,” *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, 2002.
- [5] P. Auer, N. Cesa-Bianchi, and P. Fisher, “Finite-time analysis of the multiarmed bandit problem,” *Machine Learning*, vol. 47, pp. 235–256, 2002.
- [6] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*, Cambridge University Press, 2006.
- [7] H. S. Chang and S. I. Marcus, “Two-person zero-sum Markov games: receding horizon approach,” *IEEE Trans. on Automatic Control*, vol. 48, no. 11, pp. 1951–1961, 2003.
- [8] H. S. Chang, M. C. Fu, J. Hu and S. I. Marcus, “An adaptive sampling algorithm for solving Markov decision processes,” *Operations Research*, vol. 53, no. 1, pp. 126–139, 2005.
- [9] H. S. Chang, M. C. Fu, J. Hu and S. I. Marcus, *Simulation-Based Algorithms for Markov Decision Processes*. Springer-Verlag, 2007.
- [10] H. S. Chang, M. Fu, J. Hu and S. I. Marcus, “An asymptotically efficient simulation-based algorithm for finite horizon stochastic dynamic programming,” *IEEE Trans. on Automatic Control*, vol. 52, no.1, pp. 89–94, 2007.
- [11] H. S. Chang, M. Fu, J. Hu and S. I. Marcus, “Recursive Learning Automata Approach to Markov Decision Processes,” *IEEE Trans. on Automatic Control*, vol. 52, no.7, pp. 1349–1355, 2007.
- [12] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*, Springer-Verlag, 1996.
- [13] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, “Online convex optimization in the bandit setting: gradient descent without a gradient,” in *Proc. of the 16th Annual ACM-SIAM Symposium on Discrete algorithms*, 2005, pp. 385–394.
- [14] E. Hazan, A. Agarwal, and S. Kale, “Logarithmic regret algorithms for online convex optimization,” *Machine Learning*, vol. 69, pp. 169–192, 2007.
- [15] O. Hernández-Lerma and J.B. Lasserre, “Error bounds for rolling horizon policies in discrete-time Markov control processes,” *IEEE Trans. on Automatic Control*, vol.35, pp. 1118–1124, 1990.
- [16] W. Hoeffding, “Probability inequalities for sums of bounded random variables,” *Journal of the American Statistical Association*, vol. 58, pp. 13–30, 1963.
- [17] M. Kearns, Y. Mansour, and S. Singh, “Fast planning in stochastic games,” in *Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence*, 2000, pp. 309–316.
- [18] R. Kleinberg, “Nearly tight bounds for the continuum-armed bandit problem,” in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, eds., MIT Press, Cambridge, MA, 2005, pp. 697–704.
- [19] A. J. Kleywegt, A. Shapiro, and T. Homem-De-Mello, “The sample average approximation method for stochastic discrete optimization,” *SIAM J. Optim.*, vol. 12, no. 2, pp. 479–502, 2001.
- [20] S. Lakshmivarahan and K. S. Narendra, “Learning algorithms for two-person zero-sum stochastic games with incomplete information,” *Mathematics of Operations Research*, vol. 6, pp. 379–386, 1981.
- [21] S. Lakshmivarahan and K. S. Narendra, “Learning algorithms for two-person zero-sum stochastic games with incomplete information: a unified approach,” *SIAM Journal on Control and Optimization*, vol. 20, pp. 541–552, 1982.
- [22] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, 1994.
- [23] K. Rajaraman and P. S. Sastry, “Finite time analysis of the pursuit algorithm for learning automata,” *IEEE Trans. on Systems, Man, and Cybernetics*, Part B, vol. 26, no. 4, pp. 590–598, 1996.
- [24] A. Shapiro, “On complexity of multistage stochastic programs,” *Operations Research Letters*, vol. 34, pp. 1–8, 2006.
- [25] H. Robbins, “Some aspects of the sequential design of experiments,” *Bull. Amer. Math. Soc.*, vol. 55, pp. 527–535, 1952.

- [26] P. S. Sastry, V. V. Phansalkar, and M. A. L. Thathachar, "Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 24, no. 5, pp. 769–777, 1994.
- [27] J. Van Der Wal, *Stochastic Dynamic Programming: successive approximations and nearly optimal strategies for Markov decision processes and Markov games*, Ph.D. Thesis, Eindhoven, 1980.