Solving for Best Responses and Equilibria in Extensive-Form Games with Reinforcement Learning Methods

Amy Greenwald, Jiacui Li, and Eric Sodomka

Amy Greenwald
Microsoft Research
New York, NY 10011
amy@brown.edu

Jiacui Li
Knight Management Center
Stanford University
Stanford CA, 94305-7298
jiacui@stanford.edu

Eric Sodomka
Department of Computer Science
Brown University
Providence, RI 02912
sodomka@gmail.com

April 16, 2015

Abstract

We present a framework to solve for best responses and equilibria in an extensive-form game (EFG) of imperfect information by transforming the game into a set of Markov decision processes (MDPs), and then applying simulation-based reinforcement learning to those MDPs. More specifically, we first transform a turn-taking partially observable Markov game (TT-POMG) into a set (one per player) of partially observable Markov decision processes (POMDPs), and we then transform that set of POMDPs into a corresponding set of Markov decision processes (MDPs). Next, we observe that EFGs are a special case of TT-POMGs, and hence can be transformed as described. Furthermore, because each transformation preserves the strategically-relevant information of the model to which it is applied, an optimal policy in one of the ensuing MDPs corresponds to a best response in the original EFG.

We then go on to prove that our reinforcement learning algorithm finds a near-optimal policy (and therefore a near-best response in the original EFG) in finite time, although the sample complexity is lower bounded by a function with an exponential dependence on the horizon. Nonetheless, we apply this algorithm iteratively to search for equilibria in an EFG. When the iterative procedure converges, the resulting MDP policies comprise an approximate Bayes-Nash equilibrium. Although this procedure is not guaranteed to converge, it frequently did in numerical experiments with sequential auctions.

1 Introduction

Real-world interactions between individuals and organizations are commonly modeled as extensive-form games (EFGs), in which players act in sequence, and payoffs to one player depend on the actions of all players. Under standard game-theoretic assumptions, each player in an EFG is predicted to play her *best response*, which is a strategy that maximizes her payoff, given fixed other-agent strategies. This notion of best response is central to game theory, and forms the basis of Nash Equilibrium (NE), the canonical game-theoretic solution concept: at NE, all players simultaneously play best responses to one another [25]. Best responses also underpin several classic game-theoretic learning schemes, such as *best-reply dynamics* and *fictitious play* [5]. But best responses are difficult to calculate in general; and NE (and many other game-theoretic solutions) are more difficult to find, still [4].

In this paper, we develop a simulation-based reinforcement learning (RL) algorithm that approximates best responses in EFGs. Our approach relies on transforming EFGs into more computationally-friendly representations. We first show that EFGs are special cases of turn-taking partially-observable Markov games (TT-POMGs), so that any EFG can be viewed as a TT-POMG. Next, from the perspective of a single player searching for a best response, a TT-POMG can be seen as a partially-observable Markov decision process (POMDP). We endeavor to solve such a POMDP by further transforming it into a belief-state MDP. Chaining this whole line of reasoning together, we find that to solve for a best response in an EFG, we need only to compute an optimal policy in the corresponding belief-state MDP.

Our RL algorithm is then specifically designed to solve for optimal policies in the belief-state MDPs that represent transformed EFGs. We derive upper and lower sample complexity bounds for this algorithm, showing that it is guaranteed to find a near-optimal policy in finite time, although there is an unavoidable exponential dependence on the horizon. Relative to Monte Carlo tree search methods such as POMCP [30], our model-based approach is more efficient in games where players' types (i.e., their payoffs, essentially) are private and independent, such as many sequential auctions of interest (e.g., [12], [22]).

Our algorithm can be also be applied iteratively to search for equilibria in EFGs, and if it converges, the resulting strategy profile is an approximate weak perfect Bayes-Nash equilibrium. To evaluate the performance of this iterative search procedure, we ran simulation experiments. Specifically, we looked at three sequential auction models that had been analyzed in the literature to see whether our algorithm could faithfully recover known equilibria. Our algorithm not only discovered close approximations of the known equilibria, but it also discovered heretofore unknown ones. A complete description of these experiments can be found in Greenwald *et al.* [8].

The rest of this paper proceeds as follows. Section 2 spells out generic transformations from TT-POMGs to POMDPs, and then on to MDPs. Section 3 then presents EFGs as a special case of TT-POMGs, and proceeds to apply these transformations to EFGs. Section 4 presents our simulation-based reinforcement-learning algorithm and its performance guarantees. This algorithm, when applied to MDPs that are actually transformed EFGs, is our best-response finding algorithm. Section 5 describes an iterative procedure for searching for equilibria, which employs our best-response finding algorithm as a subroutine. Section 6 surveys related literature, and Section 7 concludes. Technical details are relegated to appendices.

2 From TT-POMGs to POMDPs to MDPs

Our goal in this work is to develop an algorithm that "solves" extensive-form games (EFGs), meaning finds equilibria. Our strategy for solving EFGs will be to transform them to MDPs and then apply simulation-based reinforcement learning to the ensuing MDPs.

In this section, we present a generalization of EFGs, namely turn-taking partially-observable Markov decision processes (TT-POMGs). In the next section, we first observe that TT-POMGs generalize EFGs,

and then we transform EFGs represented as TT-POMGs to MDPs, paving the way for our reinforcement learning approach to solving EFGs.

2.1 Markov Models

To begin our story, we review a few definitions of standard models of sequential decision making, in both multi-agent and single agent environments. We begin with turn-taking partially-observable Markov games (TT-POMGs). These games are very much like the more widely-studied simultaneous-move partially-observable Markov games (see, for example, [9]), except that the players take turns. One other very important difference between TT-POMGs and POMGs themselves is that in the latter, actions are usually assumed to be observable; *in our model, however, a player does not observe the actions of the other players*.

For ease of exposition, and so that we can more easily apply our transformations to EFGs, we assume all models include fully-observable terminal states. Furthermore, rewards are associated with states only, not with state-action pairs. Neither of these assumptions is essential.

Definition 1 (Turn-taking partially-observable Markov game)**.** *A* turn-taking partially-observable Markov game Θ *is defined as a tuple* $\langle N, S, A, T, O, Q, r, \zeta \rangle$ *where:*

- $N = \{1, ..., n\}$ *is the set of players.*
- S *is a (finite) set of states. Further, let* $\langle S_1, \ldots, S_n, Z \rangle$ *be a partition over* S*, where* $S_i \subseteq S$ *is the set of states at which it is player* i*'s turn to move, and* $Z \subseteq S$ *is a designated set of terminal states.*

  (We write $\texttt{player}(s)$ *to denote the player whose turn it is at state* s*.)*
- $A = \bigcup_{ij} A_{ij}$*, where* A_{ij} *denotes the (finite) set of actions available to player* i *at state* $j \in S_i$*. At all states* $z \in Z$*, and for all players* i*,* $A_{iz} = \emptyset$ *(i.e., there are no actions available at terminal states).*
- $O = \langle O_1, \ldots, O_n \rangle$ *is a tuple of (finite) observation sets, where* O_i *is the set of observations for player* i*. Each player makes an observation when he enters a state in which it is his turn to play. Upon entering a terminal state (in which it is no player's turn), all players observe that the game is over.*
- $P : \bigcup_i S_i \times A \times S \times \bigcup_i O_i \to [0, 1]$ *is a function that expresses the joint probability over the next state and accompanying observation. In particular,* $P(j, k, j', h)$ *is the joint probability of reaching state* j' *and observing observation* $h \in O_{\texttt{player}(j')}$*, conditional on the player whose turn it is taking action* k *in state* j*. There are no transitions out of terminal states. (***Note:** P *is Markov, by definition.)*

  (We write $\texttt{obs}(s)$ *to denote the observation observed at state* s*.)*

  The following abbreviations will come in handy:

$$T(j, k, j') \equiv \sum_{h \in H} P(j, k, j', h) \tag{1}$$

$$Q_{\texttt{player}(j')}(j', h) \equiv \frac{\sum_{j,k} P(j, k, j', h)}{\sum_{j,k} T(j, k, j')} \tag{2}$$

- $r = \langle r_1, \ldots, r_n \rangle$ *is a tuple of reward functions where* $r_i : S \to \mathbb{R}$ *is player* i*'s reward function. That is,* $r_i(j)$ *denotes player* i*'s reward upon reaching state* $j \in S$*.*
- $\zeta : S \to [0, 1]$ *is a distribution over initial states. Specifically,* $\zeta(j)$ *is the probability that the initial state is* $j \in S$*.*

For all players i and for all observations $h \in O_i$, if there exist j and j' s.t. $Q(j, h) > 0$ and $Q(j', h) > 0$, then there must be a 1-to-1 correspondence between A_{ij} and $A_{ij'}$, and it must be the case that $r_i(j) = r_i(j')$. Otherwise, players would be able to distinguish states j and j' from one another. With this in mind, we let $A_i(h)$ denote the set of actions available to player i given observation h (i.e., $A_i(h) = A_{ij} = A_{ij'}$, for all such j and j'). Likewise, we write $r_i(h)$ to denote player i's rewards given observation h (i.e., $r_i(h) = r_i(j) = r_i(j')$, for all such j and j').

Here is how the play of a game unfolds: An initial game state is drawn from the distribution ζ. The player designated to move at that state does not observe the state; rather, he observes h_1 and then takes action a_1. Next, the game transitions according to T to a new state, where it is another player's turn to move. Again, that player does not observe the state, but rather, he observes h_2 and then takes action a_2. The players accrue rewards along the way, and receive further rewards upon transitioning to a terminal state.

A *history* μ^t in a turn-taking partially-observable Markov game is a sequence $\langle s^1, h^1, a^1, \ldots, s^t, h^t, a^t \rangle$ of t states, observations, and actions, where each observation is observed by exactly one player, and that is the player whose turn it is at the corresponding state. *Notably, actions are not publicly observable*. Hence, each game history μ^t can be restricted to a subsequence μ_i^t of player i's states, observations, and actions.

Generally speaking, a *policy* for a player in a TT-POMG is a mapping from the set of that player's histories to the set of available actions (or a probability distribution over actions). We restrict our attention to a subclass of these general policies. Specifically, we define a policy θ_i for player i as a mapping from a single observation $h \in O_i$ to an action $a \in A_i(h)$ (or a probability distribution over actions $a \in \Delta(A_i(h))$). Such policies are called *memoryless*, because they depend on the most recent observation only.

Let $\langle r_i^1, \ldots, r_i^t \rangle$ denote player i's sequence of rewards through time t. Then player i's expected reward through time t is given by $\mathbb{E}\left[\sum_{l=1}^{t} r_i^l\right]$, where the expectation is taken over all possible state sequences of length t. The probabilities of the various sequences are dictated by the players' (possibly randomized) policies $\boldsymbol{\theta}$ and the game's transition and observation probabilities. We abbreviate this expectation as $R_i^t(\boldsymbol{\theta})$.

Definition 2 (Best response)**.** *Given a profile of other-agent policies $\boldsymbol{\theta}_{-i}$, a* t-best response *for player i in a TT-POMG is a policy θ_i such that $R_i^t(\theta_i, \boldsymbol{\theta}_{-i}) \geq R_i^t(\theta_i', \boldsymbol{\theta}_{-i})$, for all policies θ_i'.*

Definition 3 (Partially-observable Markov decision process)**.** *A partially-observable Markov decision process Ψ is a TT-POMG with only one player, who is called the "agent".*

Definition 4 (Optimal policy)**.** *An* t-optimal policy *in a POMDP is a policy ψ such that $R^t(\psi) \geq R^t(\psi')$, for all policies ψ'.*

Definition 5 (Markov decision process)**.** *A Markov decision process Φ is defined as a tuple $\langle S, A, T, r, \zeta \rangle$, where:*

- *S is a (finite) set of states, with $Z \subseteq S$ a designated set of terminal states.*
- *$A = \bigcup_j A_j$, where A_j denotes the set of possible actions at state $j \in S$. At all states $z \in Z$, $A_z = \emptyset$ (i.e., there are no actions available at terminal states).*
- *$T : S \times A \times S \to [0, 1]$ is a possibly stochastic transition function. More specifically, $T(s, a, s')$ denotes the probability of reaching state $s' \in S \setminus Z$, conditioned on taking action $a \in A_s$ in state $s \in S$. (***Note:** T is Markov, by definition.)*
- *$r : S \to \mathbb{R}$ is a reward function. That is, $r(s)$ denotes the agent's reward upon reaching state $s \in S$.*
- *$\zeta : S \to [0, 1]$ is a distribution over initial states. More specifically, $\zeta(s)$ denotes the initial probability of state $s \in S$.*

Here is how the dynamics of an MDP unfold: Initially, a state is drawn from the distribution ζ. The agent observes this state s_1, and then chooses an action a_1. Next, the game transitions according to T to a new state. Again, that agent observes the state s_2, and then chooses an action a_2. The agent accrues rewards along the way, and receive further rewards upon transitioning to a terminal state.

A *history* μ^t in an MDP is a sequence of t states and actions: e.g., $\langle s^1, a^1, \ldots, s^t, a^t \rangle$. Generally speaking, a *policy* in an MDP is a mapping from the set of histories to the set of actions (or a probability distribution over the set of actions). Still WLOG (see, for example, Puterman [27]), we define a policy ϕ as a mapping from a state $j \in S$ to an action $a \in A_j$ (or a probability distribution over actions $a \in \Delta(A_j)$).

Let $\langle r^1, \ldots, r^t \rangle$ denote the agent's sequence of rewards through time t. Then the expected reward through time t is given by $\mathbb{E}\left[\sum_{l=1}^{t} r^l\right]$, where the expectation is taken over all possible state sequences of length t. The probabilities of the various sequences are dictated by the agent's (possibly randomized) policy and the MDP's transition probabilities. We abbreviate this expectation as $R^t(\phi)$.

Definition 6 (Optimal policy)**.** *An* t-optimal policy *in an MDP is a policy* ϕ *such that* $R^t(\phi) \geq R^t(\phi')$*, for all policies* ϕ'*.*

With these definitions of various Markov models in hand, we are ready to present our transformations. First, we transform TT-POMGs into POMDPs, from the point of view of a single player. Second, we transform POMDPs into MDPs. Taken together, the two transformations yield a procedure for converting TT-POMGs to MDPs.

2.2 TT-POMG to POMDP Transformation

The main idea of our transformation from TT-POMGs to POMDPs is to collapse a game into a single-agent decision process by folding given other-agent strategies into the transition probabilities and the initial probability distribution of the TT-POMG. Our goal is to carry out this transformation so that a best-response in the TT-POMG corresponds exactly to an optimal policy in the POMDP.

Before we present the transformation, we make the following important observation, which follows directly from our assumptions (state and observation probabilities are Markov, and policies are memoryless):

$$\begin{aligned}
& \mathtt{Pr}_{\boldsymbol{\theta}}^{\mathtt{POMG}}[h^y, a^y, s^{y+1} \mid \mu^{y-1}, s^y] && (3)\\
= \; & \mathtt{Pr}_{\boldsymbol{\theta}}^{\mathtt{POMG}}[s^{y+1} \mid \mu^{y-1}, s^y, h^y, a^y] \, \mathtt{Pr}_{\boldsymbol{\theta}}^{\mathtt{POMG}}[a^y \mid \mu^{y-1}, s^y, h^y] \, \mathtt{Pr}_{\boldsymbol{\theta}}^{\mathtt{POMG}}[h^y \mid \mu^{y-1}, s^y] && (4)\\
= \; & T^{\mathtt{POMG}}(s^y, a^y, s^{y+1}) \; (\theta_{\mathtt{player}(s^y)}(h^y))(a^y) \; Q^{\mathtt{POMG}}_{\mathtt{player}(s^y)}(s^y, h^y) && (5)
\end{aligned}$$

Now based on this observation, we derive the probability $\mathtt{Pr}_{\boldsymbol{\theta}}^{\mathtt{POMG}}$ of transitioning from one state, say s^1, to another, say s^{t+1}, assuming players are abiding by policy profile $\boldsymbol{\theta}$:

$$\begin{aligned}
\mathtt{Pr}_{\boldsymbol{\theta}}^{\mathtt{POMG}}[s^1] &= \zeta^{\mathtt{POMG}}(s^1) && (6)\\
\mathtt{Pr}_{\boldsymbol{\theta}}^{\mathtt{POMG}}[s^{t+1} \mid s^1] &= \sum_{\mu^t_{-s_1}} \mathtt{Pr}_{\boldsymbol{\theta}}^{\mathtt{POMG}}[h^1, a^1, \ldots, s^t, h^t, a^t, s^{t+1} \mid s^1] && (7)\\
&= \sum_{\mu^t_{-s_1}} \prod_{y=1}^{t} \mathtt{Pr}_{\boldsymbol{\theta}}^{\mathtt{POMG}}[h^y, a^y, s^{y+1} \mid \mu^{y-1}, s^y] && (8)\\
&= \sum_{\mu^t} \prod_{y=1}^{t} T^{\mathtt{POMG}}(s^y, a^y, s^{y+1}) \; \underbrace{[(\theta_{\mathtt{player}(s^y)}(h^y))(a^y)]}_{\text{probability } \mathtt{player}(s^y) \text{ plays } a^y} \; Q^{\mathtt{POMG}}_{\mathtt{player}(s^y)}(s^y, h^y) && (9)
\end{aligned}$$

In words, this probability is the sum over all possible histories $\mu^t_{-s_1} = \langle h^1, a^1, \ldots, s^t, h^t, a^t \rangle$ of the product of the probability of observing each h^y, the probability (dictated by the players' policies) of playing each a^y, and the probability of transitioning to each s^{y+1}.

In our transformation from a multi-player TT-POMG to a single-agent POMDP, of particular concern are histories in which it is never a given player, say i's, turn to play. In such cases, we denote the corresponding probability as follows: $\Pr^{\text{POMG}}_{\boldsymbol{\theta}_{-i}}[s^{t+1} \mid s^1]$, indicating dependence on only $\boldsymbol{\theta}_{-i}$ not the full policy profile $\boldsymbol{\theta}$. This probability is also calculated using Equation (9), but the sum over histories $\langle h^1, a^1, \ldots, s^t, h^t, a^t \rangle$ only includes those histories in which $s^2, \ldots, s^t \in S^{\text{POMG}}_{-i} \equiv \cup_{i' \neq i} S^{\text{POMG}}_i$. (N.B. $s_1 \in S^{\text{POMG}}_{-i}$.)

Our first transformation is encapsulated by the following definition of iPOMDPs, which are TT-POMGs from the point of view of a select player i.

Definition 7 (iPOMDP)**.** *Given a TT-POMG Θ, and an other-agent profile of memoryless policies $\boldsymbol{\theta}_{-i}$, we define an iPOMDP $\Psi_i(\Theta, \boldsymbol{\theta}_{-i})$, from the point of view of player i, as follows:*

- $S^{iPOMDP} = S_i \cup Z$, *where* $S_i = S_i^{POMG}$ *and* $Z = Z^{POMG}$. *(Observe that* $S^{iPOMDP} = S^{POMG} \setminus S_{-i}^{POMG}$.*)*
- $A^{iPOMDP} = \bigcup_j A_j$, *where* $A_j = A_{ij}^{POMG}$ *is the set of i's actions at state $j \in S_i$.*
- T^{iPOMDP} *defines the probability of transitioning from state $j \in S_i$ to state $j' \in S^{iPOMDP}$ when action $k \in A_j$ is chosen. This probability is calculated as you would expect—it is the sum over all other-player next states s in Θ of the probability of transitioning to s times the probability of transitioning from s to j', as determined by the other-agent policy profile $\boldsymbol{\theta}_{-i}$:*

$$T^{iPOMDP}(j, k, j') = \sum_{s \in S_{-i}^{POMG}} T^{POMG}(j, k, s) \Pr^{POMG}_{\boldsymbol{\theta}_{-i}}[j' \mid s] \tag{10}$$

- $O^{iPOMDP} = O_i^{POMG}$.
- Q^{iPOMDP} *is an observation function defined such that* $Q^{iPOMDP}(j, h) = Q_i^{POMG}(j, h)$, *for all* $j \in S^{iPOMDP}$ *and* $h \in O^{iPOMDP}$.
- r^{iPOMDP} *is a reward function defined such that* $r^{iPOMDP}(j) = r_i^{POMG}(j)$, *for all* $j \in S^{iPOMDP}$.
- *Recall that* $\Pr^{POMG}_{\boldsymbol{\theta}_{-i}}[j \mid s]$ *is the total probability of all histories through the TT-POMG that start at state s, and lead to state $j \in S^{iPOMDP}$, without encountering a state in which it is i's turn to move. So, the total probability of all histories through the TT-POMG that lead to state $j \in S^{iPOMDP}$ without encountering a state in which it is i's turn to move is given by:*

$$\zeta^{iPOMDP}(j) = \zeta^{POMG}(j) + \sum_{s \in S_{-i}^{POMG}} \zeta^{POMG}(s) \Pr^{POMG}_{\boldsymbol{\theta}_{-i}}[j \mid s] \tag{11}$$

We have defined iPOMDPs so that, from the point of view of player i, a TT-POMG Θ and an iPOMDP $\Psi_i(\Theta, \boldsymbol{\theta}_{-i})$ are identical. First, a policy for the agent in $\Psi_i(\Theta, \boldsymbol{\theta}_{-i})$ is a mathematically equivalent object to a policy for player i in Θ. This is because a policy in a POMDP depends on a history—a sequence of observations and actions. Likewise, in a TT-POMG, a policy for a player depends only on that player's observations and actions. In other words, player i observes exactly the same history in the TT-POMG and the corresponding iPOMDP, and hence makes decisions on exactly the same basis in both models. Second, by construction, the probability of all i-histories (i.e., $\langle s_i^1, h_i^1, a_i^1, \ldots s_i^t, h_i^t, a_i^t \rangle$) are equal in both models, so that $R^{i\text{POMDP},t}(\psi) = R^{\text{POMG},t}(\psi)$, for all policies ψ. From these two observations, it follows that a best-response to a memoryless policy profile $\boldsymbol{\theta}_{-i}$ in Θ is an optimal policy in $\Psi_i(\Theta, \boldsymbol{\theta}_{-i})$, and vice versa.

Theorem 1. *An optimal (not necessarily memoryless) policy ψ in an iPOMDP $\Psi_i(\Theta, \boldsymbol{\theta}_{-i})$ is a best-response to an other-agent profile of memoryless policies $\boldsymbol{\theta}_{-i}$ in the original TT-POMG Θ, and vice versa.*

A proof of this theorem appears in Appendix A.

2.3 POMDP to MDP Transformation

The next transformation we present is a variation of one that is commonly used in the literature (e.g., []): from POMDPs to *belief-state* MDPs. Like our first transformation, this transformation preserves optimal policies: i.e., an optimal policy in the ensuing belief-state MDP is an optimal policy in the original POMDP, and vice versa.

Although this transformation applies more generally, we restrict our attention to POMDPs with *perceptual aliasing* [] (i.e., those in which multiple states give rise to a single observation), because these are sufficient to represent EFGs. Specifically, we assume the set of observations O is a partition of the underlying state space. Furthermore, we assume that each state $j \in h$ gives rise precisely to observation $h \in O$. With this specific structure in mind, we describe a belief state not as a probability distribution over the entire state space, but as a conditional probability distribution over a restricted set of states (i.e., an observation). We call these MDPs *observation MDPs*, which we abbreviate OMDPs.

Definition 8 (OMDP)**.** *Given a POMDP Ψ, we construct an OMDP $\Phi(\Psi)$ as follows:*

- *S^{OMDP} is the set of all possible pairs $\langle h, \beta \rangle$ consisting of an observation $h \in O^{POMDP}$ together with beliefs $\beta \in \Delta(h)$ (i.e., a probability distribution over the set of states contained in the observation h).*
- *$A^{OMDP} = \bigcup_{h \in O^{POMDP}} A(h)$, where $A(h)$ denotes the set of actions available to the agent at observation $h \in O^{POMDP}$.*
- *The probability of transitioning from observation h to observation h' upon taking action $k \in A(h)$ is computed in expectation, where the expectation is taken with respect to beliefs β:*

$$T^{POMDP}(h, k, h') = \sum_{j \in h} \beta[j \mid h] \left(\sum_{j' \in h'} T^{POMDP}(j, k, j')\, Q^{POMDP}(j', h') \right) \tag{12}$$

$$= \sum_{j \in h} \beta[j \mid h] \left(\sum_{j' \in h'} T^{POMDP}(j, k, j') \right) \tag{13}$$

Conditioned on h', beliefs transition deterministically. Assuming perceptual aliasing, and then simplifying accordingly yields: for $j' \notin h'$, $\beta'[j' \mid h'] = 0$, while for $j' \in h'$,

$$\beta'[j' \mid h'] = \frac{\beta(k)[j' \mid h]}{\sum_{j'' \in h'} \beta(k)[j'' \mid h]} \tag{14}$$

where for $j' \in S^{POMDP}$,

$$\beta(k)[j' \mid h] = \sum_{j \in h} \beta[j \mid h]\, T^{POMDP}(j, k, j') \tag{15}$$

Thus, $T^{OMDP}(\langle h, \beta \rangle, k, \langle h', \beta' \rangle) = T^{POMDP}(h, k, h')$.

- *The reward at state $\langle h, \beta \rangle$ is computed in expectation, where the expectation is taken with respect to the belief β:*

$$r^{OMDP}(\langle h, \beta \rangle) = \sum_{j \in h} \beta[j \mid h]\, r^{POMDP}(j) \tag{16}$$

- *The initial probability of observation h is given by:*

$$\zeta^{POMDP}(h) = \sum_{j \in h} \zeta^{POMDP}(j) \tag{17}$$

Conditioned on h, the initial beliefs β are deterministic: for $j \in h$,

$$\beta[j \mid h] = \frac{\zeta^{POMDP}(j)}{\zeta^{POMDP}(h)} \tag{18}$$

Thus, $\zeta^{OMDP}(\langle h, \beta \rangle) = \zeta^{POMDP}(h)$.

We have defined OMDPs so that, from the point of view of the agent, a POMDP Ψ and an OMDP $\Phi(\Psi)$ are identical. First, a policy for the agent in Ψ is a mathematically equivalent object to a policy for the agent in $\Phi(\Psi)$. This is because a policy in an OMDP is a function from observations together with beliefs to actions. A (general, not memoryless) policy in a POMDP is a function from histories to actions. But beliefs in an OMDP encode histories in a POMDP! (Beliefs are updated as actions are taken and observations are made.) Hence, the agent makes decisions on exactly the same basis in both models. Second, the probability of all histories (i.e., $\langle\langle h^1, \beta^1 \rangle, a^1, \ldots, \langle h^t, \beta^t \rangle, a^t \rangle$ in the OMDP and $\langle\langle h^1, a^1 \rangle, \langle h^1, a^1, h^2, a^2 \rangle, \ldots, \langle h^1, a^1, \ldots, h^t, a^t \rangle\rangle$ in the POMDP) are equal in both models, so that $R^{\text{OMDP},t}(\phi) = R^{\text{POMDP},t}(\phi)$, for all policies ϕ. From these two observations, it follows that an optimal policy in Ψ is an optimal policy in $\Phi(\Psi)$, and vice versa.

Theorem 2. *An optimal policy ϕ in an OMDP $\Phi(\Psi)$ is an optimal policy in the original POMDP Ψ, and vice versa.*

3 From EFGs to OMDPs

Thus far, we have described how to interpret a TT-POMG Θ from the point of view of player i, together with a strategy profile $\boldsymbol{\theta}_{-i}$ for all players other than i, as a partially-observable Markov decision process (POMDP), $\Psi(\Theta, \boldsymbol{\theta}_{-i})$, such that an optimal policy in $\Psi(\Theta, \boldsymbol{\theta}_{-i})$ is a best-response to $\boldsymbol{\theta}_{-i}$ in Θ. We achieved this result by folding the other-agent strategies into the transition probabilities and the initial probability distribution of the TT-POMG. We also argue that, by taking observations and beliefs together as states, and by defining transition probabilities, rewards, and initial probabilities appropriately, a POMDP can be viewed as an equivalent *OMDP*—an observation MDP—such that an optimal policy in the OMDP is an optimal policy in the POMDP. Combining these observations, we can solve for the best-response in a TT-POMG by solving for an optimal policy in the equivalent MDP.

In this section, we apply these insights to extensive-form games, which we view as a special kind of TT-POMG. After doing so, we obtain the following theorem:

Theorem 3. *If we view a given EFG Γ as a TT-POMG, and then transform Γ and a given strategy profile $\boldsymbol{\theta}_{-i}$ into an OMDP $\Phi(\Psi(\Gamma, \boldsymbol{\theta}_{-i}))$ via the aforementioned transformations, then an optimal policy in $\Phi(\Psi(\Gamma, \boldsymbol{\theta}_{-i}))$ is a best-response to $\boldsymbol{\theta}_{-i}$ in Γ.*

3.1 Extensive-Form Games of Imperfect Information

An extensive-form game (EFG) is a model of dynamic multi-agent interactions. It is most frequently used to represent a non-cooperative sequential decision-making setting.

Technically, an EFG is a game *tree* comprised of *nodes* and *directed edges*, as illustrated in Figure 3.1. Each non-terminal node is a decision point, controlled by exactly one player, who chooses from the available actions, represented by outgoing edges. A *trajectory*, which arises as the game is played, is a sequence of nodes from the root to a leaf (i.e., a terminal node), with *payoffs* specified at the leaves. Consequently, a game trajectory and ensuing payoffs are jointly determined by all the players' actions.

In an EFG of *imperfect information*, each player's decision nodes are partitioned into *information sets*. When choosing actions, players know only their information set, not the precise decision node within that set, and therefore make decisions under uncertainty. In Figure 3.1, nodes in the same information set are encircled with dashed lines; any node that is not so encircled is its own singleton information set.

To model non-strategic uncertainty (i.e., uncertainty due to forces other than the strategic play of other agents), some games include a special player called nature, who chooses actions at *chance nodes* according to a distribution commonly known to all (other) players. In Figure 3.1, chance nodes are labeled 0. Player 1 does not observe nature's action before her first decision, while player 2 does observe player 1's action if nature chooses the right branch, but not if she chooses the left branch.

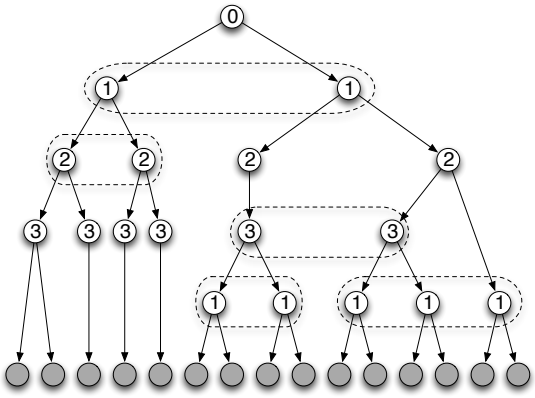


Figure 1: Sample EFG of imperfect information: circles are nodes, arrows are edges, and dashed ovals are non-singleton information sets. Rewards are not specified. Each node is labeled with the identity of its controlling player. Nature is player 0.

With this picture in mind, we now proceed to formally define EFGs of imperfect information. Our definition is based on that of Hart [10].

Definition 9 (Extensive-form game of imperfect information)**.** *An* extensive-form game of imperfect information Γ *is defined as a tuple* $\langle N, T, P, H, u \rangle$ *where:*

- $N = \{0, 1, ..., n\}$ *is the set of players, with a select (non-strategic) player* 0 *called nature.*
- T *is a tree, called the game tree, with nodes* V *(including a select root node), directed edges* E*, and terminal nodes* $Z \subset V$*.*
- P *is a partition of non-terminal nodes* $V \setminus Z$ *into subsets* $P_0, P_1, \ldots, P_n$*, where* P_i *is the set of player* i*'s decision nodes: i.e., those in which it is player* i*'s turn to choose an action. (The nodes in* P_0 *are the chance nodes.)*

We denote by A_{ij} the set of actions available to player i at node j. This set is isomorphic to the set of outgoing edges leaving node $j \in P_i$. (We write $\texttt{action}(e)$ to denote the action associated with edge e.)

- $H = \langle H_0, H_1, \ldots, H_n \rangle$ *is a tuple of partitions, one for each player i. Each H_i is a partition of the corresponding P_i. Each subset of nodes $h \in H_i$ is called an* information set. *Intuitively, a player cannot distinguish among the decision nodes in one of his information sets. However, H_0 is assumed to consist of only singletons, so nature can always distinguish among all her decision nodes.*

  Note: *For any two nodes j and j' in the same information set, there must be a 1-to-1 correspondence between j's and j''s outgoing edges: i.e., exactly the same set of actions must be available at all nodes in an information set. If this were not the case, then players would be able to distinguish among nodes in an information set, but information sets are meant to model precisely those nodes that players* cannot *distinguish from one another. With this in mind, we let $A_i(h)$ denote the set of possible actions at player i's information set $h \in H_i$ (i.e., the set of outgoing edges leaving every node in h).*

  (We write $\texttt{infoset}(j)$ to denote the information set associated with state j.)

- u *is a payoff function. For each terminal node $z \in Z$, $u(z) = \langle u_1(z), \ldots, u_n(z) \rangle$ is an n-dimensional vector of payoffs, where $u_i(z)$ denotes player i's payoff at terminal node z.*

- $\mathcal{P}$ *is a set of probability distributions, one for each node $j \in P_0$ (or equivalently, each information set $h \in H_0$). Each $p \in \mathcal{P}$ is a probability distribution over the outgoing edges (i.e., nature's possible actions): i.e., $p(\texttt{action}(e))$ is the probability of the action in $A_0(h)$ associated with edge e.*

A *pure strategy* for player i is a function s_i on H_i such that $s_i(h) \in A_i(h)$; that is, s_i specifies an action for player i at each information set. Let S_i be the set of all possible pure strategies. A *mixed strategy* is a randomization over pure strategies; that is, a mixed strategy is an element of $\Delta(S_i)$.[1]

A *behavioral strategy* for player i is a function σ_i on H_i such that $\sigma_i(h) \in \Delta(A_i(h))$; that is, σ_i specifies a distribution over actions for player i at each information set. Let Σ_i be the set of player i's possible behavioral strategies.

We assume a game of *perfect recall*, in which agents do not forget information they have previously acquired (e.g., players do not forget their own past actions). By Kuhn's theorem [17], in a game of perfect recall, every mixed strategy has a corresponding payoff-equivalent behavioral strategy. Thus, as is common in the game-theoretic literature, we restrict our attention to behavioral strategies.

A *strategy profile* $\sigma = (\sigma_1, \ldots, \sigma_n)$ dictates a behavioral strategy for each player. Let $U_i(\sigma)$ be player i's *expected* utility when strategy profile σ is played, where the expectation is taken over game trajectories induced by σ (the players' behavioral strategies) and over nature's random actions.

Let $\sigma_{-i} = (\sigma_1, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_n)$ dictate a strategy for every player except player i.

Definition 10 (Best response)**.** *Given an other-agent strategy profile σ_{-i}, a* best response *for player $i \neq 0$ in an EFG Γ of imperfect information is a strategy σ_i such that $U_i(\sigma) \geq U_i(\sigma'_i, \sigma_{-i}), \forall \sigma'_i \in \Sigma_i$.*

Definition 11 (Bayes-Nash equilibrium)**.** *A strategy profile σ is a* Bayes-Nash Equilibrium *(NE) of an EFG Γ of imperfect information if σ consists of only best responses: i.e., $U_i(\sigma) \geq U_i(\sigma'_i, \sigma_{-i})$, for all players $i \neq 0$ and for all $\sigma'_i \in \Sigma_i$. Note that nature, whose actions are specified by $\mathcal{P}$, need not best-respond.*

Having defined EFGs, we will now go on to show that EFGs are a special kind of turn-taking partially-observable Markov game.

[1] Given a discrete set X, the notation $\Delta(X)$ is commonly used to represent the set of all probability distributions over X: i.e., $\Delta(X) = \{\langle p_1, \ldots, p_{|X|} \rangle \mid \sum_{i=1}^{|X|} p_i = 1 \text{ and } p_i \geq 0, \forall i \in \{1, \ldots, |X|\}\}$.

3.2 EFGs as TT-POMGs

In this section, we instantiate our definitionn of TT-POMGs with EFGs, viewing the latter as an instance of the former. Indeed, TT-POMGs generalize EFGs in the following ways:

1. TT-POMGs may distribute rewards at intermediate states, while EFGs distribute payoffs at terminal nodes only. Furthermore, TT-POMG rewards can depend on state-action pairs, while EFG payoffs are typically associated with states alone.

2. EFGs are typically trees of finite horizon, with designated terminal nodes, while TT-POMGs can be more general graphs.

3. TT-POMGs may have more general informational structures. In EFGs (with perfect recall), there is a unique path to each information set; the same cannot be said for TT-POMGs.

To simplify the exposition, we restrict our attention to EFGs in which nature plays just once, and first. In this case, we can easily fold nature's play into the initial probability distribution of the TT-POMG, and eliminate nature as a player in the game. Modulo notational complications, our results extend immediately to EFGs, such as backgammon, in which nature plays intermittently.

Notation: Given a set X, we let $X^* = \{\{x\} \mid x \in X\}$.

Observation 4. *Every EFG Γ can be represented as a TT-POMG Θ by encoding information sets as observations.*

Proof.

- $N^{\text{POMG}} = N^{\text{EFG}} \setminus \{0\}$: i.e., nature is not a player in this game.
- $S^{\text{POMG}} = \langle S_1, \ldots, S_n, Z \rangle$, where $S_i = P_i^{\text{EFG}}$, for all $i \in N^{\text{POMG}}$, and $Z = Z^{\text{EFG}}$.
- A^{POMG} includes all edges in E^{EFG} except those emanating from the root: i.e., $A^{\text{POMG}} = \bigcup_{i \in N^{\text{POMG}}} A_i^{\text{POMG}}$, where $A_i^{\text{POMG}} = \bigcup_{j \in S_i} A_{ij}^{\text{EFG}}$. The edges emanating from the root represent nature's moves, which are folded into the initial probabilities.
- $T^{\text{POMG}}(j, k, j')$ is a deterministic function that specifies whether or not the game transitions to state j' when the player whose turn it is in state j takes action k.
- $O^{\text{POMG}} = \langle O_1, \ldots, O_n \rangle$, where $O_i = H_i^{\text{EFG}} \cup Z^*$. In other words, information sets in the EFG are encoded as observations in the TT-POMG, as are the terminal states.
- $Q^{\text{POMG}} = \langle Q_1, \ldots, Q_n \rangle$, where Q_i is an observation function such that

$$Q_i(j, h) = \begin{cases} 1 & \text{if } j \in h \\ 0 & \text{otherwise} \end{cases}$$

  for all $j \in S_i$ and $h \in O_i$. The players observe termination.

- $r^{\text{POMG}} = \langle r_1, \ldots, r_n \rangle$, where r_i is a reward function such that

$$r_i(j) = \begin{cases} u_i(j) & \text{if } j \in Z \\ 0 & \text{otherwise} \end{cases}$$

  for all $j \in S^{\text{POMG}}$.

- The initial probability $\zeta^{\text{POMG}}(j) = p^{\text{EFG}}(\texttt{action}(\text{root}, j))$, for all $j \in S^{\text{POMG}}$.

□

In the next two sections, we apply the transformations described in Sections 2.2 and 2.3 to the TT-POMG defined in Observation 4, namely to an EFG. The result is an MDP that is strategically equivalent from the point of view of a single player to an EFG (together with an other-agent strategy profile).

3.3 EFGs as iPOMDPs

After transforming an EFG, viewed as a TT-POMG, together with other-agent policies $\boldsymbol{\theta}_{-i}$, this is the iPOMDP that ensues (see Figure 3.3):

- $S^{i\text{POMDP}} = S_i \cup Z$, where $S_i = P_i^{\text{EFG}}$ and $Z = Z^{\text{EFG}}$.
- $A^{i\text{POMDP}}$ includes all edges in E^{EFG} emanating from states at which it is i's turn to play. Let A_j denote the actions available at state $j \in S_i$.
- $T^{i\text{POMDP}}$ defines the probability of transitioning from state $j \in S_i$ to state $j' \in S^{i\text{POMDP}}$ when action $k \in A_j$ is chosen. More specifically, $T^{i\text{POMDP}}(j, k, j')$ is the total probability of all histories (in which i has no moves) to state j', originating at state j with action k:

$$\begin{aligned} T^{i\text{POMDP}}(j,k,j') &= \sum_{s \in S_{-i}^{\text{POMG}}} T^{\text{POMG}}(j,k,s)\, \texttt{Pr}_{\boldsymbol{\theta}_{-i}}^{\text{POMG}}[j' \mid s] \qquad (19) \\ &= \sum_{\mu^t \in H_{-i}^t} \prod_{y=1}^{t} T^{\text{POMG}}(s^y, a^y, s^{y+1}) \underbrace{[(\theta_{\texttt{player}(s^y)}(h^y))(a^y)]}_{\text{probability } \texttt{player}(s^y) \text{ plays } a^y} Q^{\text{POMG}}_{\texttt{player}(s^y)}(s^y, h^y) \end{aligned}$$

- $O^{i\text{POMDP}} = H_i^{\text{EFG}} \cup Z^*$. In other words, information sets in the EFG are encoded as observations in the iPOMDP, as are the terminal states.
- $Q^{i\text{POMDP}}$ is an observation function such that

$$Q^{i\text{POMDP}}(j,h) = \begin{cases} 1 & \text{if } j \in h \\ 0 & \text{otherwise} \end{cases}$$

  for all $j \in S^{i\text{POMDP}}$ and $h \in O^{i\text{POMDP}}$. The agent observes termination.
- $r^{i\text{POMDP}}$ is a reward function such that

$$r^{i\text{POMDP}}(j) = \begin{cases} u_i(j) & \text{if } j \in Z \\ 0 & \text{otherwise} \end{cases}$$

  for all $j \in S^{i\text{POMDP}}$.
- The initial probability of a state $j \in S^{i\text{POMDP}}$ is the total probability of all paths in the TT-POMG that lead to j without encountering a state in which it is i's turn to move:

$$\zeta^{i\text{POMDP}}(j) = p^{\text{EFG}}(\texttt{action}(\text{root}, j)) + \sum_{s \in S_{-i}^{\text{POMG}}} p^{\text{EFG}}(\texttt{action}(\text{root}, s))\, \texttt{Pr}_{\boldsymbol{\theta}_{-i}}^{\text{POMG}}[j \mid s] \qquad (21)$$

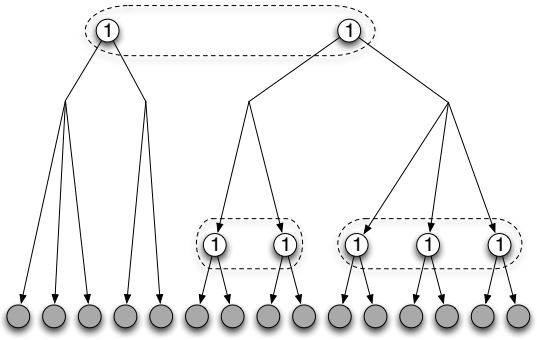


Figure 2: An EFG viewed as a iPOMDP.

3.4 EFGs as OMDPs

After transforming an EFG, viewed as a iPOMDP, this is the OMDP that ensues (see Figure 3.4):

- S^{OMDP} is the set of all possible pairs $\langle h, \beta \rangle$ consisting of an information set or terminal state $h \in H_i^{\text{EFG}} \cup (Z^{\text{EFG}})^*$, together with a belief $\beta \in \Delta(h)$ (i.e., a probability distribution over the set of states contained in h).
- A^{OMDP} includes all edges in E^{EFG} emanating from information sets at which it is i's turn to play. Let $A(h)$ denote the actions available at information set $h \in H_i^{\text{EFG}}$.
- The probability of transitioning from information set h to information set h' upon taking action $k \in A(h)$ is computed in expectation, where the expectation is taken with respect to beliefs β:

$$T^{i\text{POMDP}}(h, k, h') = \sum_{j \in h} \beta[j \mid h] \left(\sum_{j' \in h'} T^{i\text{POMDP}}(j, k, j') \right) \tag{22}$$

$$= \sum_{j \in h} \beta[j \mid h] \left(\sum_{j' \in h'} \sum_{s \in S_{-i}^{\text{POMG}}} T^{\text{POMG}}(j, k, s) \, \Pr_{\boldsymbol{\theta}_{-i}}^{\text{POMG}}[j' \mid s] \right) \tag{23}$$

Conditioned on h', beliefs transition deterministically. Assuming perceptual aliasing, and then simplifying accordingly yields: for $j' \notin h'$, $\beta'[j' \mid h'] = 0$, while for $j' \in h'$,

$$\beta'[j' \mid h'] = \frac{\beta(k)[j' \mid h]}{\sum_{j'' \in h'} \beta(k)[j'' \mid h]} \tag{24}$$

where for all $s^{t+1} \in S^{i\text{POMDP}}$,

$$\beta(k)[s^{t+1} \mid h] = \sum_{j \in h} \beta[j \mid h] \, T^{i\text{POMDP}}(j, k, s^{t+1}) \tag{25}$$

$$= \sum_{j \in h} \beta[j \mid h] \left(\sum_{s \in S_{-i}^{\text{POMG}}} T^{\text{POMG}}(j, k, s) \, \Pr_{\boldsymbol{\theta}_{-i}}^{\text{POMG}}[s^{t+1} \mid s] \right) \tag{26}$$

Thus, $T^{\text{OMDP}}(\langle h, \beta \rangle, k, \langle h', \beta' \rangle) = T^{i\text{POMDP}}(h, k, h')$.

- The reward at state $\langle h, \beta \rangle$ is computed in expectation, where the expectation is taken with respect to beliefs β:

$$r^{\text{OMDP}}(\langle h, \beta \rangle) = \sum_{j \in h} \beta[j \mid h]\, r^{i\text{POMDP}}(j) \tag{27}$$

- The initial probability of observation h is given by:

$$\begin{aligned} \zeta^{i\text{POMDP}}(h) &= \sum_{j \in h} \zeta^{i\text{POMDP}}(j) && (28) \\ &= \sum_{j \in h} p^{\text{EFG}}(\texttt{action}(\text{root}, j)) + \sum_{s \in S_{-i}^{\text{POMG}}} p^{\text{EFG}}(\texttt{action}(\text{root}, s))\, \Pr_{\boldsymbol{\theta}_{-i}}^{\text{POMG}}[j \mid s] && (29) \end{aligned}$$

  Conditioned on h, the initial beliefs β are deterministic: for $j \in h$,

$$\beta[j \mid h] = \frac{\zeta^{i\text{POMDP}}(j)}{\zeta^{i\text{POMDP}}(h)} \tag{30}$$

  Thus, $\zeta^{\text{OMDP}}(\langle h, \beta \rangle) = \zeta^{i\text{POMDP}}(h)$.

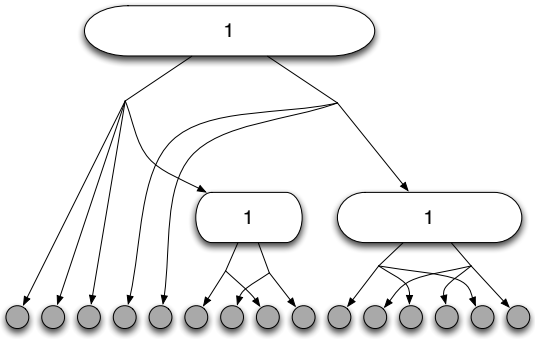


Figure 3: An EFG viewed as an OMDP.

3.5 Beliefs

Generally speaking, OMDP beliefs can be arbitrary. An agent need not base its beliefs on reality if it prefers not to. However, only so-called *consistent* beliefs (i.e., those which coincide with reality) are relevant when searching for an optimal strategy in a OMDP, or an equilibrium strategy in an EFG. Consistent beliefs can be calculated via Bayesian updating, as in Equation (24).

However, it can happen that $\beta[h']$ (i.e., the denominator in Equation (24)) evaluates to 0. In this case, it is not possible to compute β', so transition probabilities out of state $\langle h, \beta \rangle$ in the OMDP are left unspecified. We will encounter this issue in Section 5, when we search for equilibria in EFGs. For now, we simply note that our OMDPs have not yet been fully specified.

Setting aside the problem of zero-probability events, there is another problem with computing beliefs, and that is their sheer complexity. In even relatively small EFGs, the Bayesian updating that is required is often intractable.[2] Consequently, we proceed to leverage reinforcement learning methods to solve (fully-specified) OMDPs. Our algorithm and its analysis, however, do not apply only to OMDPs; they apply to any tree-shaped MDP in which simulations must originate at the root.

[2]We commented when working with sequential auctions: "All existing theoretical work on Bayesian sequential auctions with multi-unit demand is confined to two-round cases, due to the increased complexity of additional rounds." [8]

4 RL Approach to Solving OMDPs

Recall that our goal is to develop a best-response finding algorithm for EFGs. Having reduced EFGs to OMDPs, our present subgoal, then, is to develop an algorithm that solves for an optimal policy in an OMDP. We take a reinforcement learning (RL) approach, and assume a black box capable of simulating the MDP. Our algorithm is applicable to any tree-shaped MDP in which simulations must originate at the root.

At a high level, our approach is entirely straightforward: first, learn an approximation of the MDP via simulation, and then, solve for an optimal policy in the approximate MDP, which we call the *learned* policy. We do not specify a means of solving the learned MDP; this can be accomplished via dynamic programming, for example [3]. Hence, our RL method primarily concerns the learning of the MDP via simulation.

4.1 Sampling Algorithm

The simulation of player i's OMDP is best understood as repeatedly replaying the EFG. Concretely, to learn player i's MDP while the other agents are playing $\boldsymbol{\sigma}_{-i}$:

- Repeatedly play the EFG, with:
  - nature playing its prescribed strategy
  - other-agents playing their prescribed strategies, $\boldsymbol{\sigma}_{-i}$
  - player i exploring all actions (e.g., with uniform probability)
- After collecting histories of many game trajectories, approximate i's OMDP as follows:
  - the transition probabilities $T(h, k, h')$, by the empirical frequency of transitioning to information set h' when player i chooses action k at information set h
  - the expected rewards $r(j)$, by the average of all rewards received at terminal state j

This algorithm is restated using our mathematical notation in Algorithm 1.

Our approach can be compared with model-free reinforcement learning methods that learn optimal policies directly, without first learning an explicit model of the MDP (e.g. UCT [15]). We opt for a model-based approach so that we can more easily exploit an independence structure that is typical of EFGs. In particular, when nature moves first and chooses a type θ_i for each player i independently, her move factors the game into independent strategic components, one per player.[3] Specifically, from player i's perspective, nature's move factors into $\boldsymbol{\theta} = (\theta_i, \theta_{-i})$, where θ_i influences only player i's rewards, and likewise for θ_{-i}. Therefore, in player i's MDP, the transition probabilities do not depend on θ_i, and the rewards do not depend on θ_{-i}. Our approach efficiently exploits this independence structure, while a typical model-free method would be forced to learn a different policy for all other-agent type profiles $\boldsymbol{\theta}_{-i}$.

4.2 Sample Complexity

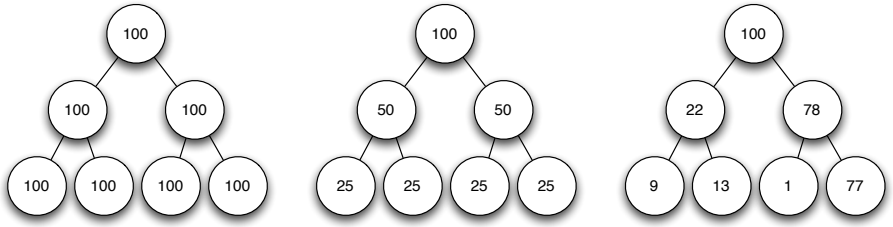
Properties inherited from an EFG pose important challenges to learning an OMDP effectively. First, because EFGs are trees, the number of OMDP states is exponential in the horizon. Since the player remembers her own past moves (perfect recall), even without observing her opponents' or nature actions, the number of information sets after H actions is at least A^H, where A is the number of available actions.

[3] For example, consider a one-shot first-price auction with *independent private values* [16]. At the start of the game, nature draws a vector of independent valuations, one per player: $\theta_i \sim F_i$. Each player i then bids according to strategy $\sigma_i : \theta_i \to \text{BidSet}$. The highest bidder wins, paying her bid, and getting a payoff of θ_i less her bid, while the others get 0.

Algorithm 1: Sampling Algorithm

Input: Γ, $\boldsymbol{\sigma}_{-i}$, σ_i, L
Output: an estimate of $T^{\texttt{OMDP}}$ and $r^{\texttt{OMDP}}$
  $T(h, k, h') \leftarrow 0$, for all h, k, h'
  $r(j) \leftarrow 0$, for all j
  for $l \leftarrow 1$ to L **do**
    $j \leftarrow$ sample from $\zeta^{\texttt{OMDP}}(\cdot)$
    while $j \notin Z$ **do**
      $h \leftarrow \texttt{infoset}(j)$
      $k \leftarrow$ sample from $\sigma_i(h)$
      $j' \leftarrow$ sample from $T^{\texttt{OMDP}}(j, k, \cdot)$
      $h' \leftarrow \texttt{infoset}(j')$
      Increment $T(h, k, h')$
      $j \leftarrow j'$
    end while
    Increment $r(j)$
  end for
  Normalize T
  Average r
  return T
  return r

Second, each simulation consists of a single uninterrupted trajectory from the root to some leaf, so the nodes deep in the tree can only be sampled very infrequently and their dynamics poorly learned, even after numerous simulations. Even among nodes at the same level in the tree, the distribution can be very uneven. Kearns *et al.* [13] designed a sparse sampling algorithm (SS) to efficiently plan in large MDPs with exponential state spaces, but their algorithm only works when it is possible to resample actions at any point along a trajectory. That way, the simulator can make the distribution of sample points look like that in Figure 4(a), instead of what it typically looks like in our case, namely Figure 4(c).



(a) (b) (c)

Figure 4: The number of times states were sampled in a hypothetical OMDP. (a): The case assumed achievable in the sparse sampling algorithm of Kearns *et al.* [13]. (b): The best case that can possibly occur in learning OMDPs. (c): A typical black box simulation of an OMDP. The states deep in the tree can be poorly learned even if the total number of simulations is large.

These two challenges—that the state space is exponential, and that all sample trajectories must originate at the root—more or less ensure that there will always be some states whose transitions are poorly learned, rendering it difficult to derive optimality guarantees for learned policies. Nonetheless, Theorem 5 states that our simple RL algorithm is guaranteed to learn a near-optimal policy with high probability in finite time.

Theorem 5. *Consider a tree-shaped MDP with finite horizon H, branching factor S, at most A actions per state, and total rewards in the range $[0, 1]$. Further, suppose transitions are unknown, but that we are given a black box that simulates the MDP. Assuming undiscounted rewards, there exists an exploration strategy for learning the MDP's transitions such that, for any $\epsilon, \delta > 0$, the following sample size guarantees that the learned policy (i.e., the optimal policy in the learned MDP) is ϵ-optimal with probability at least $1 - \delta$:*

$$N = \mathcal{O}\left(\left(\frac{2ASH}{\epsilon}\right)^{2H}\left(\log\frac{1}{\delta} + H\log S\right)\right) \tag{31}$$

This bound can perhaps be improved by considering selective sampling schemes, such as UCT [15], and by better proof techniques. However this bound is not "too unsharp" either. We also prove a lower sample complexity bound which says it is impossible to get rid of an exponential dependence on the horizon H:

Theorem 6 (Lower Bound)**.** *For any sampling algorithm, even adaptive, there exists an OMDP such that the algorithm must generate at least $\Omega\left(\frac{A^H}{\epsilon}(1 - H\epsilon^{1/H})(1 - \delta)\right)$ samples to guarantee that the learned policy be ϵ-optimal with probability at least $1 - \delta$.*

The proofs of both theorems are constructive. For the upper bound, we examine convergence of the learned MDP policy under a *balanced wandering* exploration strategy: at every state, choose the least sampled action, and break ties uniformly randomly. For the lower bound, we explicitly construct an OMDP that is difficult to learn. The full proofs are relegated to Appendices B and C.

5 An Iterative Search for Equilibrium

While our RL approach only computes one player's best response, we can apply it iteratively, to each player in turn, to search for an equilibrium in an EFG. This idea is motivated by the method of best-reply dynamics, a commonly studied in game theory [5]. Specifically, we propose the following iterative procedure:

1. Initialize with an arbitrary strategy profile $\boldsymbol{\sigma}^{(0)} = (\sigma_1^{(0)}, \sigma_2^{(0)}, \ldots, \sigma_n^{(0)})$
2. For each iteration $l = 1, \ldots, L$ and for each player $i = 1, \ldots, n$
   (a) Run Algorithm 1 to learn player i's OMDP on input strategy profile $\boldsymbol{\sigma}_{-i}^{(l-1)}$
   (b) Update strategy $\sigma_i^{(l)}$ to be an optimal policy in the learned OMDP: i.e., a best response to $\boldsymbol{\sigma}_{-i}^{(l-1)}$
3. Terminate if $\boldsymbol{\sigma}^{(l)}$ is the same as, or very similar to, $\boldsymbol{\sigma}^{(l-1)}$

It is desirable to initialize $\boldsymbol{\sigma}^{(0)}$ to be a totally-mixed (behavioral) strategy profile in which players choose all actions with positive probability, because doing so makes it more likely that all parts of the EFG will be explored. But regardless of initialization, after even one round of updates, we cannot be guaranteed that all information sets will be reached with positive probability.

Definition 12. *Information sets that cannot be reached (i.e., with positive probability) are called* off-path*, because they are guaranteed to be off the path of play. Information sets that can be reached with positive probability are called* on-path*. (These definitions rely on a specified strategy profile.)*

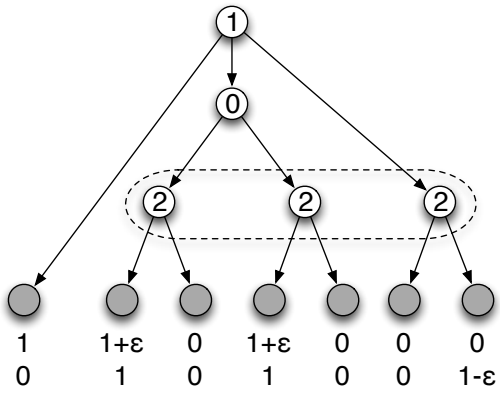


Figure 5: An example EFG. The top and bottom rows of numbers below the terminal nodes are the payoffs for P1 and P2, respectively. Nature chooses among her branches with equal probability.

Consider the example in Figure 5, which involves two players, P1 and P2. Suppose beliefs are initialized so that the two leftmost states in P2's information set occur with total probability significantly less than 0.5, while the rightmost state occurs with probability significantly greater than 0.5. Suppose further that P2 updates first. Given the aforementioned beliefs, his best response is to play R. Now it is P1's turn to update; his best response to P2's move is L. The baton is passed back to P2 to update, but when P1 plays L, there is zero probability on P2's sole information set.

There is nothing for P2 to do when his sole information set is off-path, so let's allow P2 to invent some arbitrary beliefs. If he chooses uniform, then his best response is L. P1's best response when P2 plays L is M. And now, P2's information set is no longer off-path. The two leftmost states each occur with probability 0.5. P2's best response to these beliefs remains L, and the dynamics have converged. Furthermore, they have converged to a Bayes-Nash equilibrium: P1 is best-responding to P2's choice of L; likewise, P2 is best-responding to P1's choice of M.

At this point, we would be remiss not to fully specify a means of constructing transition probabilities in an OMDP. One possibility is to proceed as described above: to simply allow players to invent arbitrary beliefs about the states in off-path information sets. This possibility has the drawback that it is unlikely to lead to convergence, particularly if his choice of beliefs differs from iteration to iteration.

Another possibility is to initialize all actions uniformly (so that there are no off-path information sets), and then to insist that if ever in our search for an equilibrium, the dynamics lead to a situation in which an information set is off-path, then the beliefs about the states in that information set should be inherited from the previous iteration. Returning to our example, if, when its sole information set is off-path, P2 instead operates under the assumption that its beliefs are unchanged, then P1 chooses to play L again. At this point, the dynamics have converged, once again to a Bayes-Nash equilibrium (P1 is best-responding to P2's choice of R; likewise, P2 is best-responding to P1's choice of L), albeit, a less-compelling one.

At this point, it should be clear why we deferred fully specifying OMDPs until now. We simply wanted to integrate their full specification into our equilibrium search algorithm, which we do presently:

1. Initialize with an arbitrary, yet totally mixed, strategy profile $\boldsymbol{\sigma}^{(0)} = (\sigma_1^{(0)}, \sigma_2^{(0)}, \ldots, \sigma_n^{(0)})$.
2. For each iteration $l = 1, \ldots, L$ and for each player $i = 1, \ldots, n$
   (a) Run Algorithm 1 to learn player i's OMDP on input strategy profile $\boldsymbol{\sigma}_{-i}^{(l-1)}$.

(b) *If any information sets are off-path, define beliefs at the states in those information sets to be whatever the corresponding beliefs were during the previous iteration.*

(c) Update strategy $\sigma_i^{(l)}$ to be an optimal policy in the learned OMDP: i.e., a best response to $\boldsymbol{\sigma}_{-i}^{(l-1)}$.

3. Terminate if $\boldsymbol{\sigma}^{(l)}$ is the same as, or very similar to, $\boldsymbol{\sigma}^{(l-1)}$.

Theorem 7. *If our equilibrium search algorithm terminates, it finds an approximate Bayes-Nash equilibrium.*

By definition, if our iterative search algorithm terminates, then $\boldsymbol{\sigma}^{(l)} = \boldsymbol{\sigma}^{(l-1)}$. In this case, $\sigma_i^{(l)}$ is an approximate best response (because of the nature of our best-response finding algorithm) to $\boldsymbol{\sigma}_{-i}^{(l-1)} = \boldsymbol{\sigma}_{-i}^{(l)}$. Since all players are near-best responding to one another's strategies, their play is an approximate Bayes-Nash equilibrium.

Another source of error in the approximation could arise if play does not converge precisely, but if instead $\boldsymbol{\sigma}^{(l)}$ is very similar to $\boldsymbol{\sigma}^{(l-1)}$. Indeed, in practice our iterative search procedure requires a user-specified measure of dissimilarity between $\boldsymbol{\sigma}^{(l)}$ and $\boldsymbol{\sigma}^{(l-1)}$. One such dissimilarity measure is the proportion of information sets at which the two strategy profiles prescribe different actions. This measure is simple to calculate, but it does not take into account the fact that some information sets are more likely to be encountered in equilibrium than others, so should perhaps be weighted more heavily. (Indeed some information sets may be off-path entirely!)

A more sophisticated measure would estimate player i's *expected* loss in payoffs, $\hat{U}$, for playing $\sigma_i^{(l-1)}$ rather than $\sigma_i^{(l)}$ when other agents play $\boldsymbol{\sigma}_{-i}^{(l-1)}$: i.e., $\left|\hat{U}(\sigma_i^{(l)}, \boldsymbol{\sigma}_{-i}^{(l-1)}) - \hat{U}(\boldsymbol{\sigma}^{(l-1)})\right|$. Taking the maximum of this measure across all players yields the ϵ-*factor* that makes $\boldsymbol{\sigma}^{(l-1)}$ an ϵ-BNE. For more details about how to effectively implement our iterative search procedure, including information about approximating and interpreting ϵ-factors, we refer readers to [8].

6 Related Work

The problem of finding a Nash equilibrium or even an approximate Nash equilibrium in a normal-form game is PPAD-complete Daskalakis et al. [4]. Regardless, various approaches have been taken to search for such equilibria. Black-box game solvers such as GAMBIT [21] provide easy off-the-shelf solutions for finding equilibria in games, but they are limited by the size of games in which they can tractably solve.

More specialized algorithms have been proposed for solving games with some particular structure. Rabinovich et al. [28] generalize fictitious play to incomplete information games with finite actions; applying this idea, they compute approximate equilibria in environments with utilities expressible as linear functions over a one-dimensional type space. Reeves and Wellman [29] describe an algorithm for computing a best response in two-player games of incomplete information with infinite action spaces, using a two-player one-shot auction as a motivating example. Ganzfried and Sandholm [6] study the problem of computing approximate equilibria in the context of poker, a stochastic game of incomplete information.

Some propose solving games after restricting the strategy space for computational tractability. Armantier et al. [1] study the problem of approximating equilibrium in Bayesian games by searching for constrained strategic equilibrium (CSE). A strategy profile is a CSE if no agent is better off by deviating to a different strategy within the constrained strategy set, typically represented by some parametrization of the full strategy space. Mostafa and Lesser [23] describe an anytime algorithm for approximating equilibria in general incomplete information games; the algorithm reduces the size of the game tree by collapsing nodes with "obvious" decisions and guides the search using local and global measures of profile stability.

Lee et al. [18] represent multi-stage games as *iterated semi net form-games*, and then use reinforcement learning techniques to solve those games. Their experiments are for two-player games with perfect information about opponent actions, but their approach is not conceptually limited to such models.

In the auction domain, the iterative method *self-confirming price prediction* employed in Osepayshvili et al. [26] is very similar to the best response dynamics we implemented, as is the approach taken in Naroditskiy and Greenwald [24].

In terms of applying reinforcement learning methods to games, many variants of Q-learning [19][11][7] have been used to approximate game-theoretic equilibrium. While we consider EFGs, others have applied their methods to stochastic games. Sample complexity bounds for reinforcement learning have been studied by many; a summary appears in Strehl et al. [31].

7 Conclusion

In this paper, we purport to make three main contributions. First, we clarify the theoretical relationships between four game-theoretic and decision-theoretic representations (EFG, TT-POMG, POMDP, and MDP), and provide general recipes from transforming one to another. Second, we design an algorithm that can solve for a near-optimal policy in finite time in tree-shaped MDPs that do not admit resampling part-way through a trajectory, and hence near-best responses in an EFG. Finally, we use our best-response finding algorithm to iteratively search for equilibria, and argue that our iterative search procedure, when it converges, finds approximate Bayes-Nash equilibria.

Although our iterative search procedure is not guaranteed to converge, we found that it often converges to near-equilibrium in experiments with sequential auctions [8]. One limitation, however, is that we can only search for pure-strategy equilibria, not mixed-strategy equilibria. This is an inherent difficulty in applying reinforcement learning methods to games, because there is no principled way to break ties between equally attractive actions, and exploring the space of all possible mixed strategies is computationally prohibitive. The good news is that pure-strategy equilibria exist in many games with general properties (Proposition 8.D.3 in [20]), while mixed equilibria have frequently been regarded as unnatural [2].

A Proof of Theorem 1

In this appendix, we show that, from the point of view of player i, a TT-POMG Θ and an iPOMDP $\Psi_i(\Theta, \boldsymbol{\theta}_{-i})$ are identical. That is, the probability of all histories are equal in both models. From this, it follows that a best-response to a memoryless policy profile $\boldsymbol{\theta}_{-i}$ in Θ is an optimal policy in $\Psi_i(\Theta, \boldsymbol{\theta}_{-i})$, and vice versa.

Lemma 1. *Given a TT-POMG Θ together with an other-agent memoryless policy profile $\boldsymbol{\theta}_{-i}$ and a corresponding iPOMDP $\Psi(\Theta, \boldsymbol{\theta}_{-i})$ with policy ψ, the probability of a history $\langle \mu^{t-1}, s^t \rangle$ through Ψ is equal to the probability of that same history (denoted $\langle \mu_i^{t-1}, s_i^t \rangle$) through Θ: i.e., $\Pr_{\psi}^{i\mathit{POMDP}}[\mu^{t-1}, s^t] = \Pr_{\psi, \boldsymbol{\theta}_{-i}}^{\mathit{POMG}}[\mu_i^{t-1}, s_i^t]$, for all $t = 1, \ldots$.*

Proof. The proof is by induction on t.

Basis: In the base case, when $t = 1$, the history μ^{t-1} through Ψ is of length 0. Correspondingly, the only histories of interest in Θ are those leading to state s^1 in which it is never i's turn to move. Consequently, $\Pr_{\psi}^{i\mathrm{POMDP}}[s^1] = \zeta^{i\mathrm{POMDP}}(s^1) = \sum_{s \in S_{-i}^{\mathrm{POMG}}} \zeta^{\mathrm{POMG}}(s) \Pr_{\boldsymbol{\theta}_{-i}}^{\mathrm{POMG}}[s^1 \mid s] = \Pr_{\psi, \boldsymbol{\theta}_{-i}}^{\mathrm{POMG}}[s^1]$.

Step: Assume the induction hypothesis, namely, $\Pr_{\psi}^{i\mathrm{POMDP}}[\mu^{t-1}, s^t] = \Pr_{\psi, \boldsymbol{\theta}_{-i}}^{\mathrm{POMG}}[\mu_i^{t-1}, s_i^t]$, for a history of length $t-1$ through Ψ. We must show the same for a history of length t.

First, note how the dynamics of Ψ and Θ play out:

$$\begin{aligned}
\Pr_{\psi}^{i\text{POMDP}}[h^t, a^t, s^{t+1} \mid \mu^{t-1}, s^t] &= \Pr_{\psi}^{i\text{POMDP}}[s^{t+1} \mid \mu^{t-1}, s^t, h^t, a^t] \Pr_{\psi}^{i\text{POMDP}}[a^t \mid \mu^{t-1}, s^t, h^t] \Pr_{\psi}^{i\text{POMDP}}[h^t \mid \mu^{t-1}, s^t] && (32)\\
&= T^{i\text{POMDP}}(s^t, a^t, s^{t+1})\,(\psi(\mu^{t-1}, s^t, h^t))(a^t)\, Q^{i\text{POMDP}}(s^t, h^t) && (33)\\
&= T^{\text{POMG}}(s_i^t, a_i^t, s_i^{t+1})\,(\psi(\mu^{t-1}, s^t, h_i^t))(a_i^t)\, Q_i^{\text{POMG}}(s_i^t, h_i^t) && (34)\\
&= \Pr_{\psi,\boldsymbol{\theta}_{-i}}^{\text{POMG}}[s_i^{t+1} \mid \mu_i^{t-1}, s_i^t, h_i^t, a_i^t] \Pr_{\psi,\boldsymbol{\theta}_{-i}}^{\text{POMG}}[a_i^t \mid \mu_i^{t-1}, s_i^t, h_i^t] \Pr_{\psi,\boldsymbol{\theta}_{-i}}^{\text{POMG}}[h_i^t \mid \mu_i^{t-1}, s_i^t] && (35)\\
&= \Pr_{\psi,\boldsymbol{\theta}_{-i}}^{\text{POMG}}[h_i^t, a_i^t, s_i^{t+1} \mid \mu_i^{t-1}, s_i^t] && (36)
\end{aligned}$$

Equation (33) follows immediately via the iPOMDP construction. Equation 34 follows from the fact that Θ is Markov in both transition and observation probabilities. The claim now follows via the induction hypothesis (Equation (39)):

$$\begin{aligned}
\Pr_{\psi}^{i\text{POMDP}}[\mu^t, s^{t+1}] &= \Pr_{\psi}^{i\text{POMDP}}[\mu^{t-1}, s^t, h^t, a^t, s^{t+1}] && (37)\\
&= \Pr_{\psi}^{i\text{POMDP}}[h^t, a^t, s^{t+1} \mid \mu^{t-1}, s^t] \Pr_{\psi}^{i\text{POMDP}}[\mu^{t-1}, s^t] && (38)\\
&= \Pr_{\psi,\boldsymbol{\theta}_{-i}}^{\text{POMG}}[h_i^t, a_i^t, s_i^{t+1} \mid \mu_i^{t-1}, s_i^t] \Pr_{\psi,\boldsymbol{\theta}_{-i}}^{\text{POMG}}[\mu_i^{t-1}, s_i^t] && (39)\\
&= \Pr_{\psi,\boldsymbol{\theta}_{-i}}^{\text{POMG}}[\mu_i^{t-1}, s_i^t, h_i^t, a_i^t, s_i^{t+1}] && (40)\\
&= \Pr_{\psi,\boldsymbol{\theta}_{-i}}^{\text{POMG}}[\mu_i^t, s_i^{t+1}] && (41)
\end{aligned}$$

□

Theorem 8. *An optimal (not necessarily memoryless) policy ψ in an iPOMDP $\Psi_i(\Theta, \boldsymbol{\theta}_{-i})$ is a best-response to an other-agent profile of memoryless policies $\boldsymbol{\theta}_{-i}$ in the original TT-POMG Θ, and vice versa.*

Proof. We claim that for all times t and for all policies ψ, $R^{i\text{POMDP},t}(\psi) = R^{\text{POMG},t}(\psi, \boldsymbol{\theta}_{-i})$. Therefore, since ψ is an optimal policy in $\Psi_i(\Theta, \boldsymbol{\theta}_{-i})$, it follows that ψ is a best-response to $\boldsymbol{\theta}_{-i}$ in Θ.

This claim follows from Lemma 1, which states that, under any policy ψ, the probability of a history through Ψ is equal to the probability of that same history through Θ, together with the fact that player i's rewards in Ψ are by definition the rewards in Θ.

$$\begin{aligned}
R^{i\text{POMDP},t}(\psi) &= \mathbb{E}_{\langle \mu^{t-1}, s^t \rangle \sim \Pr_{\psi}^{i\text{POMDP}}} \left[\sum_{l=1}^{t} r^{i\text{POMDP}}(s^l) \right] && (42)\\
&= \mathbb{E}_{\langle \mu_i^{t-1}, s_i^t \rangle \sim \Pr_{\psi,\boldsymbol{\theta}_{-i}}^{\text{POMG}}} \left[\sum_{l=1}^{t} r_i^{\text{POMG}}(s_i^l) \right] && (43)\\
&= R_i^{\text{POMG},t}(\psi, \boldsymbol{\theta}_{-i}) && (44)
\end{aligned}$$

□

B Proof of Theorem 5

In our proof, we assume the simulation process follows a *balanced wandering* policy π_0: during each iteration, one of the actions sampled least often so far is chosen, with ties broken uniformly. A desirable feature of balanced wandering[4] is that if a state is reached mA times, then all actions at that state are guaranteed to have been chosen m times.

[4]Desirable in terms of facilitating the proof. In our simulation experiments, we observe that adaptive sampling methods frequently produce better MDP approximations, and hence policies with higher rewards, under the same computational budget.

Note: To simplify exposition, we prove the theorem for undiscounted MDPs with deterministic rewards, assuming the maximum reward along each path bounded by $[0, 1]$, but the result can be extended to handle stochastic and arbitrarily bounded non-negative rewards.

B.1 Proof Plan

Recall that our proof pertains only to tree-shaped MDPs of finite horizon. We use s^h to denote a generic MDP state at depth h, and $p^h = (s^1, s^2, ..., .s^h)$ to denote a path of length h. Then $r(s)$ from (resp., $r(p)$) denotes the reward upon reaching state s (resp., following path p). We let $P_h^\pi(s^h)$ (resp., $P_h^\pi(p^h)$) denote the probability of reaching state s^h (resp., following path p^h) under policy π, omitting the subscript π when the choice of policy is clear. Estimators appear with hats over their true counterparts: e.g., $\hat{T}(s^{h-1}, a, s^h)$, $\hat{P}_h(p^h)$.

Because of the MDP's tree structure, each state uniquely defines a path, so we can equivalently associate rewards with paths, and then define the expected rewards $R(\pi)$ earned by policy π as path-probability weighted rewards. Furthermore, inspired by Kearns and Singh [14], we divide up the set of paths through the MDP into a set of *rare* paths, and its complement set of *regular* paths:

$$R(\pi) = \sum_{p^H} P_H^\pi(p^H) r(p^H) \tag{45}$$

$$= \sum_{p^H \text{ rare}} P_H^\pi(p^H) r(p^H) + \sum_{p^H \text{ regular}} P_H^\pi(p^H) r(p^H) \tag{46}$$

$$\stackrel{\text{def}}{=} R^{\text{rare}}(\pi) + R^{\text{regular}}(\pi) \tag{47}$$

Denote the optimal policy by π^* and our learned policy by $\hat{\pi}^*$. Then the loss in rewards experienced by our simulation algorithm is:

$$R(\pi^*) - R(\hat{\pi}^*) = \underbrace{(R^{\text{rare}}(\pi^*) - R^{\text{rare}}(\hat{\pi}^*))}_{\texttt{Err}_{\text{rare}}} + \underbrace{(R^{\text{regular}}(\pi^*) - R^{\text{regular}}(\hat{\pi}^*))}_{\texttt{Err}_{\text{regular}}} \tag{48}$$

Our proof strategy targets rare and regular paths separately:

- Rare paths will be visited insufficiently often in the simulation process, so transition probabilities along these paths are not guaranteed to be learned well. But $\texttt{Err}_{\text{rare}}$ is still small precisely because these paths will be visited rarely, so choosing suboptimal actions along them has limited impact on total rewards.
- $\texttt{Err}_{\text{regular}}$ will also be small (with high probability), because the transitions along regular paths will be sampled sufficiently often and thus well learned, so our learned policy will fare well in the regular MDP.
  - Lemmas 3, 4, and 5 show that with sufficient sampling, all regular path probabilities will be estimated with low error.
  - Lemma 6 shows that when path probabilities are well estimated, $\texttt{Err}_{\text{regular}}$ is small.

B.2 Rare MDP

Theorem 9 shows that $\texttt{Err}_{\text{rare}}$ is small if we choose the parameter β to be small.

Definition 13 (β-rare paths)**.** *We call a state s^h β-*rare *if the probability of transitioning to s^h from its parent state s^{h-1} is bounded above by a small constant $\beta > 0$ under any policy: $\frac{1}{A}\sum_a T(s^{h-1}, a, s^h) < \beta$. Likewise, we call a path β-*rare *if any of its states are β-rare.*

Given an MDP and a policy π, let $P^{\text{rare}}(\pi) = \sum_{p^H \text{ rare}} P_H^\pi(p^H)$. That is, $P^{\text{rare}}(\pi)$ is the probability of traversing a rare path in the MDP under policy π. Then the probability of traversing a rare path (regardless of policy) is bounded above by $\max_\pi P^{\text{rare}}(\pi)$. The following lemma upper bounds this latter probability.

Lemma 2. *The probability of traversing a path that is β-rare will be small if β is small:*

$$\max_\pi P^{\textit{rare}}(\pi) \leq \beta ASH \tag{49}$$

Proof. Define $\mathcal{E}_h$ as the event of transitioning to a β-rare state on the h^{th} step under an arbitrary policy π. Then:

$$\begin{aligned}
P^\pi(\mathcal{E}_h) &= \sum_{s^h \text{ rare}} T(s^{h-1}, \pi(s^{h-1}), s^h) && (50)\\
&\leq \sum_{s^h \text{ rare}} \max_a T(s^{h-1}, a, s^h) && (51)\\
&\leq \sum_{s^h \text{ rare}} \sum_a T(s^{h-1}, a, s^h) && (52)\\
&\overset{\text{Definition of } \beta\text{-rare}}{\leq} \sum_{s^h \text{ rare}} \beta A && (53)\\
&\overset{\text{Branching factor} = S}{\leq} \beta AS && (54)
\end{aligned}$$

Now, to traverse a β-rare path under π requires visiting at least one β-rare state:

$$\begin{aligned}
P^{\text{rare}}(\pi) &= P^\pi(\cup_{h=1}^H \mathcal{E}_h) && (55)\\
&\overset{\text{Union bound}}{\leq} \sum_{h=1}^H P^\pi(\mathcal{E}_h) && (56)\\
&\overset{\text{Equation (54)}}{\leq} \sum_{h=1}^H \beta AS && (57)\\
&= \beta ASH && (58)
\end{aligned}$$

Finally, since π was arbitrary, the result follows. □

Theorem 9. *The error along rare paths is small if β is small. In particular, $\texttt{Err}_{\textit{rare}} \leq \beta ASH$.*

Proof.

$$\texttt{Err}_{\text{rare}} = R^{\text{rare}}(\pi^*) - R^{\text{rare}}(\hat{\pi}^*) \tag{59}$$

$$\overset{\text{Rewards}\geq 0}{\leq} R^{\text{rare}}(\pi^*) \tag{60}$$

$$= \max_{\pi} R^{\text{rare}}(\pi) \tag{61}$$

$$= \max_{\pi} \sum_{p^H \text{ rare}} P_H^{\pi}(p^H) r(p^H) \tag{62}$$

$$\overset{\text{Rewards}\leq 1}{\leq} \max_{\pi} \sum_{p^H \text{ rare}} P_H^{\pi}(p^H) \tag{63}$$

$$\overset{\text{Lemma 2}}{=} \beta ASH \tag{64}$$

□

B.3 Regular MDP

A state that is *not* β-rare is called *regular*. Likewise, a *regular* path is one that visits no β-rare states. The following lemma gives a lower bound on how frequently a regular path will be sampled.

Lemma 3. *Assuming N simulations and balanced wandering, all regular paths will be sampled at least $\frac{\beta^H N}{2}$ times with probability at least $1 - S^H \exp\left(-\frac{\beta^{2H} N}{2}\right)$.*

Proof. Under balanced wandering, the probability of transitioning to a regular state s^h from its parent state s^{h-1} given action a is $\frac{1}{A} T(s^{h-1}, a, s^h)$. Therefore, the total probability of transitioning to a regular state s^h from its parent s^{h-1} (summing up over all actions) is at least β, because s^h is regular. It follows that $P(p^H) \geq \beta^H$ for regular path p^H, because regular paths visit no β-rare states.

Denote by $N(p^H)$ the number of times p^H is sampled during N simulations. Then a direct application of the Chernoff bound gives:

$$P\left(N(p^H) < \frac{\beta^H N}{2}\right) < \exp\left(-\frac{\beta^{2H} N}{2}\right) \tag{65}$$

Now since there are at most S^H paths:

$$P\left(\exists \text{ a regular path } \bar{p}^H \text{ s.t. } N(\bar{p}^H) < \frac{\beta^H N}{2}\right) = P\left(\bigcup_{p^H} \left\{N(p^H) < \frac{\beta^H N}{2}\right\}\right) \tag{66}$$

$$\leq \sum_{p^H} P\left(N(p^H) < \frac{\beta^H N}{2}\right) \tag{67}$$

$$\leq \sum_{p^H} \exp\left(-\frac{\beta^{2H} N}{2}\right) \tag{68}$$

$$\leq S^H \exp\left(-\frac{\beta^{2H} N}{2}\right) \tag{69}$$

Therefore, $P\left(\forall \text{ regular paths } \bar{p}^H, N(\bar{p}^H) \geq \frac{\beta^H N}{2}\right) \geq 1 - S^H \exp\left(-\frac{\beta^{2H} N}{2}\right)$. □

Once again, Lemma 3 guarantees that regular paths will be sampled sufficiently often. But then, under balanced wandering, all state-action pairs along regular paths will also be sampled sufficiently often. The next lemma, Lemma 4, shows that, as a consequence, the error in learned transition probabilities will be small. We calculate errors in terms of L1-norms:

$$||\hat{T}(s,a,\cdot) - T(s,a,\cdot)||_1 \stackrel{\text{def}}{=} \sum_{s'} |\hat{T}(s,a,s') - T(s,a,s')| \tag{70}$$

$$||\hat{P}_h(\cdot) - P_h(\cdot)||_1 \stackrel{\text{def}}{=} \sum_{p^h} |\hat{P}_h(p^h) - P_h(p^h)| \tag{71}$$

The following lemma, which lower bounds the number of times $N(s,a)$ that state action pair (s,a) should be sampled, is due to Weissman, *et al.* [32]:

Lemma 4. *For each state-action pair (s,a), to ensure that $P(||\hat{T}(s,a,\cdot) - T(s,a,\cdot)||_1 \leq \epsilon_0) \geq 1 - \delta_0$, (s,a) must be sampled at least N_0 times: i.e.,*

$$N(s,a) \geq N_0 \stackrel{def}{=} \frac{2}{\epsilon_0^2} \ln\left(\frac{2^S}{\delta_0}\right) \tag{72}$$

Next, we bound the sampling errors in estimating regular path probabilities by summing up the sampling errors in estimating one-step transition probabilities.

Lemma 5. *If the state-action pairs along all regular paths obey $||\hat{T}(s,a,\cdot) - T(s,a,\cdot)||_1 \leq \epsilon_0$ as in Lemma 4, then under any policy π, it follows that $||\hat{P}_H^\pi(\cdot) - P_H^\pi(\cdot)||_1 \leq H\epsilon_0$.*

Proof. Recall that $P_h(\cdot)$ is the probability distribution over length-h paths (under π). The proof is by induction on the path length h.

Basis: $||\hat{P}_1(\cdot) - P_1(\cdot)||_1 = ||\hat{T}(s^0,a,\cdot) - T(s^0,a,\cdot)||_1$, which, by assumption, is bounded above by ϵ_0.

Step: Assume the induction hypothesis: $||\hat{P}_{H-1}(\cdot) - P_{H-1}(\cdot)||_1 \leq (H-1)\epsilon_0$. Then:

$$||\hat{P}_H(\cdot) - P_H(\cdot)||_1 = \sum_{p^{H-1}} \sum_{s^H} \left|\hat{P}_H(p^{h-1} \cup s^H) - P_H(p^{H-1} \cup s^H)\right| \tag{73}$$

$$= \sum_{p^{H-1}} \sum_{s^H} \left|\hat{P}_{H-1}(p^{H-1})\hat{T}(s^{H-1},a,s^H) - P_{H-1}(p^{H-1})T(s^{H-1},a,s^H)\right| \tag{74}$$

$$\stackrel{\text{Simplify notation: } p \stackrel{\text{def}}{=} p^{H-1}, s \stackrel{\text{def}}{=} s^H, P(\cdot) \stackrel{\text{def}}{=} P_H(\cdot), T(\cdot) \stackrel{\text{def}}{=} T(s,a,\cdot)}{=} \sum_p \sum_s \left|\hat{P}(p)\hat{T}(s) - P(p)T(s)\right| \tag{75}$$

$$= \sum_p \sum_s \left|\hat{P}(p)\hat{T}(s) - \hat{P}(p)T(s) + \hat{P}(p)T(s) - P(p)T(s)\right| \tag{76}$$

$$\leq \sum_p \sum_s \left(\left|\hat{P}(p)\hat{T}(s) - \hat{P}(p)T(s)\right| + \left|\hat{P}(p)T(s) - P(p)T(s)\right|\right) \tag{77}$$

$$= \sum_p \sum_s \left(\hat{P}(p)\left|\hat{T}(s) - T(s)\right| + T(s)\left|\hat{P}(p) - P(p)\right|\right) \tag{78}$$

$$= \sum_p \hat{P}(p) \sum_s \left|\hat{T}(s) - T(s)\right| + \sum_s T(s) \sum_p \left|\hat{P}(p) - P(p)\right| \tag{79}$$

$$\leq (1)\epsilon_0 + (1)(H-1)\epsilon_0 \tag{80}$$

$$= H\epsilon_0 \tag{81}$$

□

If all regular path probabilities have low estimation errors, this leads to low estimation error in rewards. More specifically, the following *simulation lemma*[5] states that an estimate of a policy's rewards cannot differ by more than $H\epsilon_0$ from the true value of the policy.

Lemma 6. *If the estimation errors of length-H path probabilities are at most $H\epsilon_0$, as in Lemma 5, then the estimated expected reward of any policy π is bounded above by $H\epsilon_0$.*

Proof.

$$|\hat{R}(\pi) - R(\pi)| = \left| \sum_{p^H} \hat{P}_H(p^H) r(p^H) - \sum_{p^H} P_H(p^H) r(p^H) \right| \tag{82}$$

$$\leq \sum_{p^H} \left| \hat{P}_H(p^H) - P_H(p^H) \right| r(p^H) \tag{83}$$

$$\overset{\text{Rewards bounded above by 1}}{\leq} \sum_{p^H} \left| \hat{P}_H(p^H) - P_H(p^H) \right| \tag{84}$$

$$= ||\hat{P}_H(\cdot) - P_H(\cdot)||_1 \tag{85}$$

$$\leq H\epsilon_0 \tag{86}$$

Equation (86) follows from Lemma 5. □

Theorem 10. *The learned policy will be within $2H\epsilon_0$ of the true optimal.*

Proof. Denote the true optimal policy as π^* and the learned policy as $\hat{\pi}^*$. If they are not the same policy, it must be that $\hat{\pi}^*$ is estimated to have higher rewards than π^* in the learned MDP. Therefore, we can infer the following lower bound on $\hat{\pi}^*$'s reward in the true MDP:

$$R(\hat{\pi}^*) \geq \hat{R}(\hat{\pi}^*) - H\epsilon_0 \tag{87}$$

$$\geq \hat{R}(\pi^*) - H\epsilon_0 \tag{88}$$

$$= R(\pi^*) - 2H\epsilon_0 \tag{89}$$

Equations (87) and (89) follow from our simulation lemma, Lemma 6. Equation (88) follows from the assumption that $\hat{\pi}^*$ is an optimal policy in the learned MDP. □

B.4 Choosing Parameters

Let's sum up the story so far. The total reward contribution of rare-paths is bounded (Lemma 2), so Err_{rare} is small. With high probability, all states-action pairs in the regular MDP will be sampled sufficiently often (Lemma 3), so all regular paths will be learned well (Lemmas 4 and 5), hence the learned policy will be near-optimal in the regular MDP (Theorem 10).

All that is left to do is express the free parameters $\beta, \epsilon_0, N_0, \delta_0$ as functions of input variables $A, S, H, \delta, \epsilon$, and then to solve the sample size N.

[5]There are several versions of the simulation lemma, the earliest of which appears in Kearns and Singh [14].

1. **Reward loss in rare MDP**: We choose $\beta = \frac{\epsilon}{2ASH}$, so that the maximum loss in rewards from choosing suboptimal actions on rare paths is bounded by $\epsilon/2$ according to Lemma 2. That is, in the worst case, and in the rare MDP, our learned policy achieves 0 reward while the optimal policy achieves $\epsilon/2$. This result is deterministic, meaning it is independent of N.

   All of the following steps are for the regular MDP.

2. **Reward loss in regular MDP**: We choose $\epsilon_0 = \frac{\epsilon}{4H}$, so that, according to Theorem 10, the maximum reward loss from regular paths is also bounded by $\epsilon/2$ with high probability. Therefore, total reward loss (in both the rare and the regular MDP) is bounded by ϵ with high probability.

3. In the regular MDP, bound $P($**sampled sufficiently**$)$: Lemma 3 states that all regular paths, and therefore all state-action pairs along regular paths, will be sampled $N_0 = \frac{\beta^H N}{2}$ times (i.e., "sufficiently") with probability at least $1 - S^H \exp\left(-\frac{\beta^{2H} N}{2}\right)$.

4. In the regular MDP, bound $P($$\exists$ **a transition that is not well learned** $\mid$ **it is sampled sufficiently**$)$: A state-action pair (s, a) is sampled sufficiently if $N(s, a) \geq N_0$. The transition probabilities of a particular state-action pair (s, a) are "not well learned" if $||\hat{T}(s, a, \cdot) - T(s, a, \cdot)||_1 > \epsilon_0$. By Lemma 4, $P\left((s, a) \text{ well learned} \mid \text{it is sampled sufficiently}\right) \geq 1 - \delta_0$. So:

$$P\left((s, a) \text{ not well learned} \mid \text{it is sampled sufficiently}\right) \leq \delta_0 \tag{90}$$

$$\overset{\text{Inverting Equation (72)}}{=} 2^S \exp\left(-\frac{\epsilon_0^2}{2} N_0\right) \tag{91}$$

   Applying a union bound, and noting that there are at most $AS + AS^2 + \ldots + AS^H \leq AHS^H$ state-action pairs yields:

$$P\left(\exists\ (s, a) \text{ not well learned} \mid \text{all } (s, a) \text{ are sampled sufficiently}\right) \tag{92}$$

$$= P\left(\bigcup_{(s,a)} \{(s, a) \text{ not well learned }\} \mid \text{all } (s, a) \text{ are sampled sufficiently}\right) \tag{93}$$

$$\leq \sum_{(s,a)} P\left((s, a) \text{ not well learned} \mid (s, a) \text{ is sampled sufficiently}\right) \tag{94}$$

$$\leq AHS^H \delta_0 \tag{95}$$

5. **Putting it all together**: Recall that the loss in rewards obtained in the rare MDP is no greater than $\epsilon/2$ *deterministically*, so we need only bound the probability of failing to achieve $\epsilon/2$-optimality in

the regular MDP. Putting everything together yields:

$$\delta \stackrel{\text{def}}{=} P(\text{learned policy not } \epsilon\text{-optimal}) \tag{96}$$
$$= P(\text{learned policy not } \epsilon/2\text{-optimal in the regular MDP}) \tag{97}$$
$$= P(\text{failed to learn the regular MDP well}) \tag{98}$$
$$= P(\text{didn't sample sufficiently}) + P(\text{sampled sufficiently})P(\exists \text{ a transition that is not well learned} \mid \text{it is sampled sufficiently}) \tag{99}$$
$$\leq P(\text{didn't sample sufficiently}) + P(\exists \text{ a transition that is not well learned} \mid \text{it is sampled sufficiently}) \tag{100}$$
$$\leq S^H \exp\left(-\frac{\beta^{2H}}{2}N\right) + AHS^H\delta_0 \tag{101}$$
$$= S^H\left[\exp\left(-\frac{\beta^{2H}}{2}N\right) + AH\delta_0\right] \tag{102}$$
$$\stackrel{\text{Plugging in } \delta_0}{=} S^H\left[\exp\left(-\frac{\beta^{2H}}{2}N\right) + AH2^S\exp\left(-\frac{\epsilon_0^2}{2}N_0\right)\right] \tag{103}$$
$$\stackrel{\text{Plugging in } \epsilon_0}{=} S^H\left[\exp\left(-\frac{\beta^{2H}}{2}N\right) + AH2^S\exp\left(-\frac{\epsilon^2}{32H^2}N_0\right)\right] \tag{104}$$
$$\stackrel{\text{Plugging in } N_0}{=} S^H\left[\exp\left(-\frac{\beta^{2H}}{2}N\right) + AH2^S\exp\left(-\frac{\epsilon^2\beta^H}{64AH^2}N\right)\right] \tag{105}$$
$$\stackrel{\text{Plugging in } \beta}{=} S^H\left[\exp\left(-\underbrace{\frac{\epsilon^{2H}}{2(2ASH)^{2H}}}_{\gamma_1}N\right) + AH2^S\exp\left(-\underbrace{\frac{\epsilon^{H+2}}{64AH^2(2ASH)^H}}_{\gamma_2}N\right)\right] \tag{106}$$

The two terms in Equation (106) are both exponentially decaying in N, but for small enough values of ϵ, and when $H > 2$, $\gamma_1 < \gamma_2$. This means that the second term will vanish much faster than the first, asymptotically, so the first will dominate. Therefore, when N is large, Equation (106) simplifies to:

$$\delta \leq 2S^H e^{-\gamma_1 N} \tag{107}$$
$$= 2S^H \exp\left[-\frac{1}{2}\left(\frac{\epsilon}{2ASH}\right)^{2H}N\right] \tag{108}$$

Rearranging yields:

$$N \leq 2\left(\frac{2ASH}{\epsilon}\right)^{2H}\ln\left(\frac{2S^H}{\delta}\right) \tag{109}$$
$$= 2\left(\frac{2ASH}{\epsilon}\right)^{2H}\left(\ln 2 + H\ln S + \ln\frac{1}{\delta}\right) \tag{110}$$

Ignoring constants and simplifying expressions proves the theorem.

C Proof of Theorem 6

To prove this theorem, we construct a specific MDP which achieves this bound.

Suppose there is only one terminal state with reward 1 while all other states have reward 0. Further suppose that from each state there is only one optimal action that successfully transitions to the desired next-round state with probably $\epsilon^{1/H}$; all other actions will only lead to "wrong" states: i.e., states that can only lead to zero reward.

In the remainder of this proof, we rely on the following terminology:

- Choose the right action: choosing the only optimal action, call it a^*, at a state.
- Transition successfully: choosing a^* *and* transitioning to the desired next round state.
- Getting a hit: reaching a rewarding terminal state.

The optimal policy, which chooses the right action at each state, is the only policy that can possibly get a hit, and its expected reward is 1 times the probability of a hit, which is ϵ:

$$P(\text{hit}) = P(\text{transition 1 successful}, \text{transition 2 successful}, ..., \text{transition } H \text{ successful}) \tag{111}$$

$$\overset{\text{Independence of transitions}}{=} P(\text{transition 1 successful}) \cdot P(\text{transition 2 successful}) \cdot ... \cdot P(\text{transition } H \text{ successful}) \tag{112}$$

$$= (\epsilon^{1/H})^H = \epsilon \tag{113}$$

All other policies generate an expected reward of 0, so to learn an ϵ-optimal policy in this MDP is equivalent to discovering the optimal policy, which in turn requires continued simulation until a hit occurs. Therefore our task becomes bounding the number of samples N needed until a hit occurs. We will now prove the theorem by showing that even under the optimal exploration policy, the probability of getting a hit during each simulation will be small, and hence N needs to be large.

Lemma 7. *In this specially constructed MDP, balanced wandering (choosing the least-sampled action and breaking ties uniformly at random) is the optimal exploration policy.*

Proof. Balanced wandering is optimal because at any point in the simulation, it maximizes the probability of choosing the right action.

- When all actions have been sampled the same number of times, all actions are equally likely to be the correct one, so breaking ties uniformly at random is optimal.
- When an action a has been sampled less often than any of its alternatives, as long as there has not been a hit yet, a is more likely than its alternatives to be right.

□

Next, we bound the probability of getting a hit during a simulation.

Lemma 8. *Under balanced wandering, the probability of a successful transition is bounded by* $\frac{\epsilon^{1/H}}{A(1-\epsilon^{1/H})}$*, and the probability of getting a hit is bounded by* $\bar{p} = \frac{\epsilon}{A^H(1-H\epsilon^{1/H})}$.

Proof. When all actions at a state have been sampled the same number of times, the probability of a successful transition under balanced wandering is the probability of choosing the right action, $1/A$, times the probability of a successful transition upon the correct choice, $\epsilon^{1/H}$. This yields a success probability of $\frac{\epsilon^{1/H}}{A}$, at each step, and since there are H steps, the probability of a hit is $(\frac{\epsilon^{1/H}}{A})^H = \frac{\epsilon}{A^H}$.

But we can obtain a tighter bound by considering the situation in the midst of repeated simulations, when some actions have been sampled less than others, balanced wandering will choose the right action with probability greater than $1/A$. This is because actions that have been chosen more often and have failed more often are less likely than the others to be the right action.

Denote by $\mathcal{A}_i(m)$ the event that action a_i has been tried m times, but that none of those tries led to a successful transition. The most favorable case for an action, say a_1 WLOG, is the event $\mathcal{A} \stackrel{\text{def}}{=} \mathcal{A}_1(n-1) \cup \bigcup_{i \neq 1} \mathcal{A}_i(n)$. In other words, this event generates the highest probability that a_1 is a^*.[6] Balanced wandering will choose a_1 in this case, and $P(a^* = a_1)$ is indeed greater than $1/A$, but it cannot be too much greater:

$$P(a^* = a_1 \mid \mathcal{A}) = \frac{P(\mathcal{A} \mid a^* = a_1)P(a^* = a_1)}{P(\mathcal{A})} \tag{114}$$

$$= \frac{P(\mathcal{A} \mid a^* = a_1)P(a^* = a_1)}{P(\mathcal{A} \mid a^* = a_1)P(a^* = a_1) + \sum_{i \neq 1} P(\mathcal{A} \mid a^* = a_i)P(a^* = a_i)} \tag{115}$$

$$= \frac{P(\mathcal{A}_1(n-1) \mid a^* = a_1)P(a^* = a_1)}{P(\mathcal{A}_1(n-1) \mid a^* = a_1)P(a^* = a_1) + \sum_{i \neq 1} P(\mathcal{A}_i(n) \mid a^* = a_i)P(a^* = a_i)} \tag{116}$$

$$= \frac{(1-\epsilon^{1/H})^{n-1} \cdot \frac{1}{A}}{(1-\epsilon^{1/H})^{n-1} \cdot \frac{1}{A} + (1-\epsilon^{1/H})^{n} \cdot \frac{A-1}{A}} \tag{117}$$

$$= \frac{1}{1 + (1-\epsilon^{1/H})(A-1)} \tag{118}$$

$$= \frac{1}{A(1-\epsilon^{1/H}) + \epsilon^{1/H}} \tag{119}$$

$$< \frac{1}{A(1-\epsilon^{1/H})} \tag{120}$$

Equation (116) follows because for all $j \neq 1$, $P(\mathcal{A}_j(n) \mid a^* = a_1) = 1$, for all n.

Therefore, the probability of a successful transition is actually bounded by $\frac{1}{A(1-\epsilon^{1/H})} \cdot \epsilon^{1/H} = \frac{\epsilon^{1/H}}{A(1-\epsilon^{1/H})}$. It follows that the probability of getting a hit is bounded by:

$$P(\text{hit}) < \left(\frac{\epsilon^{1/H}}{A(1-\epsilon^{1/H})}\right)^H \tag{121}$$

$$= \frac{\epsilon}{A^H(1-\epsilon^{1/H})^H} \tag{122}$$

$$\leq \frac{\epsilon}{A^H(1-H\epsilon^{1/H})} \tag{123}$$

$$\stackrel{\text{def}}{=} \bar{p} \tag{124}$$

□

[6] That this is the most favorable case (i.e., that this situation yields the highest probability a_1 is the right action) is easy to see by example. Suppose there are three actions, that have all been tried (and failed) the same number of times. Then the probability distribution over the right action is $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Now suppose action 2 is again tried without success; then the situation becomes $(\frac{1}{3} + \frac{\eta}{2}, \frac{1}{3} - \eta, \frac{1}{3} + \frac{\eta}{2})$, where η is a small probability calculated using Bayesian updating. Still more probability will be shifted to action 1 if action 3 is then tried without success. This is the highest probability that action 1 is the right action.

Putting Lemmas 7 and 8 together, we see that to guarantee $P(\text{learned policy is } \epsilon\text{-optimal}) \geq 1 - \delta$ requires that $\delta \geq 1 - P(\text{learned policy is } \epsilon\text{-optimal}) = P(\text{learned policy is } \textit{not } \epsilon\text{-optimal})$. In other words, we require that

$$\begin{aligned}
\delta &\geq P(\text{learned policy is } \textit{not } \epsilon\text{-optimal}) && (125)\\
&= P(\text{no hit in } N \text{ simulations}) && (126)\\
&> (1-\bar{p})^N && (127)\\
&\geq 1 - N\bar{p} && (128)
\end{aligned}$$

But then, it is necessary that

$$\begin{aligned}
N &> \frac{1-\delta}{\bar{p}} && (129)\\
&\overset{\text{plug in } \bar{p} \text{ as in Equation (124)}}{=} \frac{A^H}{\epsilon}\left(1 - H\epsilon^{1/H}\right)(1-\delta) && (130)
\end{aligned}$$

References

[1] Olivier Armantier, Jean-Pierre Florens, and Jean-Francois Richard. Approximation of Nash Equilibria in Bayesian Games. *Journal of Applied Econometrics*, 23(7):965–981, 2008.

[2] Robert J Aumann. What is game theory trying to accomplish? In *Frontiers of Economics, edited by K. Arrow and S. Honkapohja*. Citeseer, 1985.

[3] Richard E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[4] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[5] D. Fudenberg and D.K. Levine. *The theory of learning in games*, volume 2. MIT press, 1998.

[6] Sam Ganzfried and Tuomas Sandholm. Computing Equilibria in Multiplayer Stochastic Games of Imperfect Information. In *Twenty-First International Joint Conference on Artificial Intelligence*, pages 140–146, Pasadena, July 2009.

[7] Amy Greenwald and Keith Hall. Correlated Q-Learning. In *Twentieth International Conference on Machine Learning*, pages 242–249, Washington, DC, August 2003.

[8] Amy Greenwald, Jiacui Li, and Eric Sodomka. Approximating equilibria in sequential auctions with incomplete information and multi-unit demand. In *Advances in Neural Information Processing Systems 25*, pages 2330–2338, 2012.

[9] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI*, pages 709–715, 2004.

[10] Sergiu Hart. Games in extensive and strategic forms. *Handbook of Game Theory with Economic Applications*, 1:19–40, 1992.

[11] Junling Hu and Michael P Wellman. Nash Q-Learning for General-Sum Stochastic Games. *Journal of Machine Learning Research*, 4:1039–1069, 2003.

[12] Brett Katzman. A Two Stage Sequential Auction with Multi-Unit Demands. *Journal of Economic Theory*, 86(1):77–99, 1999.

[13] Michael Kearns, Yishay Mansour, and Andrew Y Ng. A Sparse Sampling Algorithm for Near-Optimal Planning in Large Markov Decision Processes. *Machine Learning*, 49:193–208, 2002.

[14] Michael J. Kearns and Satinder P. Singh. Near-Optimal Reinforcement Learning in Polynomial Time. *Machine Learning*, 49(2):209–232, 2002.

[15] Levente Kocsis and Csaba Szepesvári. Bandit Based Monte-Carlo Planning. In *Seventeenth European Conference on Machine Learning*, pages 282–293, Berlin, September 2006.

[16] Vijay Krishna. *Auction theory*. Academic press, 2009.

[17] Harold W. Kuhn. Extensive Games and the Problem of Information. *Contributions to the Theory of Games*, 2(28):193–216, 1953.

[18] Ritchie Lee, David H. Wolpert, Scott Backhaus, Russell Bent, James Bono, and Brendan Tracey. Modeling Humans as Reinforcement Learners : How to Predict Human Behavior in Multi-Stage Games. In *NIPS-11 Workshop on Decision Making with Multiple Imperfect Decision Makers*, Granada, December 2011.

[19] Michael L. Littman. Friend-or-Foe Q-Learning in General-Sum Games. In *Eighteenth International Conference on Machine Learning*, pages 322–328, Williamstown, June 2001.

[20] Andreu Mas-Collell, Michael Whinston, and Jerry R Green. Microeconomic theory. 1995.

[21] Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. Gambit: Software tools for game theory. Technical report, Version 0.2010.09.01, 2010. URL `http://www.gambit-project.org`.

[22] Flavio M. Menezes and Paulo K. Monteiro. Synergies and Price Trends in Sequential Auctions. *Review of Economic Design*, 8(1):85–98, 2003.

[23] Hala Mostafa and Victor Lesser. Approximately Solving Sequential Games With Incomplete Information. In *AAMAS-08 Workshop on Multi-Agent Sequential Decision Making in Uncertain Multi-Agent Domains*, pages 92–106, Estoril, May 2008.

[24] Victor Naroditskiy and Amy Greenwald. Using Iterated Best-Response to Find Symmetric Bayes-Nash Equilibria in Auctions. In *Twenty-Second National Conference on Artificial Intelligence*, pages 1894–1895, Vancouver, July 2007.

[25] John Nash. Non-Cooperative Games. *Annals of Mathematics*, 54(2):286–295, 1951. ISSN 0003486X. doi: 10.2307/1969529. URL `http://www.jstor.org/stable/1969529`.

[26] Anna Osepayshvili, Michael P. Wellman, Daniel M. Reeves, and Jeffrey K. MacKie-Mason. Self-Confirming Price Prediction for Bidding in Simultaneous Ascending Auctions. In *Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 441–449, Edinburgh, July 2005.

[27] M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.

[28] Zinovi Rabinovich, Victor Naroditskiy, Enrico H Gerding, and Nicholas R Jennings. Computing Pure Bayesian Nash Equilibria in Games with Finite Actions and Continuous Types. *Artificial Intelligence*, 195:106–139, 2013.

[29] Daniel M. Reeves and Michael P. Wellman. Computing Best-Response Strategies in Infinite Games of Incomplete Information. In *Twentieth Conference on Uncertainty in Artificial Intelligence*, pages 470–478, Banff, July 2004.

[30] David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Advances in Neural Information Processing Systems*, pages 2164–2172, 2010.

[31] Alexander L. Strehl, Lihong Li, and Michael L. Littman. Reinforcement Learning in Finite MDPs: PAC Analysis. *Journal of Machine Learning Research*, 10:2413–2444, 2009.

[32] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the l1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep*, 2003.