Reinforcement Learning in Partially Observable Multiagent Settings: Monte Carlo Exploring Policies with PAC Bounds

Roi Ceren
THINC Lab, Dept. of
Computer Science
University of Georgia
Athens, GA 30602
rceren@uga.edu

Prashant Doshi
THINC Lab, Dept. of
Computer Science
University of Georgia
Athens, GA 30602
pdoshi@cs.uga.edu

Bikramjit Banerjee
School of Computing
University of S. Mississippi
Hattiesburg, Miss. 39406
bikramjit.banerjee@usm.edu

ABSTRACT

Perkins' Monte Carlo exploring starts for partially observable Markov decision processes (MCES-P) integrates Monte Carlo exploring starts into a local search of policy space to offer a template for reinforcement learning that operates under partial observability of the state. In this paper, we generalize the reinforcement learning under partial observability to the self-interested multiagent setting. We present a new template, MCES-IP, which extends MCES-P by maintaining predictions of the other agent's actions based on dynamic beliefs over models. MCES-IP is instantiated to be approximately locally optimal with some probability by deriving a theoretical bound on the sample size that in part depends on the allowed error from the sampling; we refer to this algorithm as MCESIP+PAC. Our experiments demonstrate that MCESIP+PAC learns policies whose values are comparable or better than those from MCESP+PAC in multiagent domains while utilizing much less samples for each transformation.

1. INTRODUCTION

Action-value based reinforcement learning (RL) algorithms such as Sarsa(λ) [1] and Q-learning [2] represent some of the best performing RL methods in single-agent settings. If the state is partially observable, the agent may simply use its previous observation as the state of a Markov decision process but the learning may not converge [3]. More preferably, the agent conditions its learning on the recent history of observations [4]. Diverging from the traditional action-value that is computed for a state or observation history, Perkins [5] utilizes action-value to evaluate a *policy*. A T-step (deterministic) policy is a mapping from observation histories of length up to T to an action. This less explored avenue has the strong benefit of directly searching a discrete space of policies and does not require the convergence of action-values for each state before the policy is obtained.

Perkins' Monte Carlo exploring starts for POMDP (MCES-P) [5] obtains action-value for a policy from sampled trajectories, and uses it to determine whether a transformed policy in the local neighborhood of the current policy is better. Sutton and Barto [1] introduced a refinement to Monte Carlo Q-learning allowing the agent to start with a random state-action pair, and called it *exploring starts*. Here, MCES-P transforms the policy using a random pair of observation sequence and action to obtain a locally transformed policy. When instantiated with a probably approximately correct (PAC) style of learning [5], MCES-P guarantees ϵ-local optimality with specified probability if all transformations are explored until the resulting policy stops improving.

The simplicity and theoretical guarantees of MCES-P motivate its generalization to partially observable multiagent settings. We generalize to an individual self-interested agent learning its policy in a setting shared with other agents whose actions cannot be perfectly observed. The generalization is suited to both cooperation and noncooperation between agents. In this regard, this paper makes the following contributions:

1. We introduce a novel RL template for partially observable multiagent settings: MCES-IP that aims to learn a policy for an agent acting and planning in a setting shared with other agents.
2. We comprehensively show that MCESIP+PAC which additionally maintains a prediction of likely actions that others performed based on models converges to policies whose values are comparable or better than the baseline MCES-P in stationary and nonstationary environments with significantly less samples needed per transformation.
3. Instantiations MCESP+PAC and MCES-IP both exhibit the PAC guarantee that the policy on termination is ϵ-locally optimal with a probability that depends on ϵ.
4. We present an observation history pruning method which eliminates transforming over rare experiences, trading some of the PAC guarantees for significant decrease in run time.

We now discuss the minimum epistemological requirements for the learning in MCES-IP. Without loss of generality, we assume that the learning agent's observation is decomposed into public and private signals. While the public signal is common to all agents in the environment, the private signal may differ between agents and is perceived by the corresponding agent only. Of course, both public and private signals may influence each agent's action. A partial observation function that gives the distribution over the joint private signals conditioned on the joint actions is known to all agents in the setting; the transition and reward functions of agents are not known. Subsequently, the RL in MCES-IP is not completely model free.

While convergence to a local optimum is a less desirable guarantee, nevertheless we may combine it with a method such as random restarts to almost surely reach the global optimum. However, any theoretical bounds on sample complexity may no longer hold.

2. RELATED WORK

Several approaches exist for optimizing behavior in uncertain environments with or without the presence of other agents. In the single-agent context, partially observable Markov decision processes (POMDP) [6] consider optimal outcomes to strategies given

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016),*
J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.),
May 9–13, 2016, Singapore.

an explicit model of the environment. In multiagent settings, decentralized POMDPs (DEC-POMDP) [7] and interactive POMDPs (I-POMDP) [8] tackle cooperative and self-interested settings, respectively, by considering the effect of other agents and their actions on the state of the game. Solution methods have so far predominantly relied on explicit models of the mechanics of the environment and the opponent.

Early model-free approaches for multiagent problems have been explored in both cooperative and non-cooperative settings. Monte Carlo Q-Alternating (MCQ-Alt) [9] approximates the dynamics of an environment in the presence of another cooperating agent following a fixed policy. After arriving at a locally optimal policy, the agent fixes its own policy and the other agent then learns. Other Q-learning techniques are available for cooperative multiagent settings where communication is not possible [10], and thus the joint action-observation history is unknown. These approaches learn action values based on local observations in the context of agents that are learning concurrently or alternating in action. However, the approach often converges to policies that are considerably suboptimal compared to exact solutions, while better performing heuristics require providing hidden state information prior to learning.

Factored-value partially observable Monte Carlo planning [11] produces policies for agents acting in cooperative multiagent contexts by exploiting actions whose sampled rewards dominate neighboring actions via Q-learning. The approach scales well with the number of agents, observations and states, depending on the method used but requires noise-free, instant communication between agents and does not apply to non-cooperative settings.

A Bayes-Adaptive I-POMDP (BA-IPOMDP) [12] maintains a vector of latent models of environment mechanics and updates its belief over these models online. In contrast to learning policies for all agents, MCES-IP focuses on learning the policy of an individual self-interested agent that shares its environment with cooperative or noncooperative agents. It does not require explicit models of the environment, which are potentially infinite in the case of BA-IPOMDPs. Effectively, BA-IPOMDP casts model-free learning as planning over an infinite space. Along similar lines, Hoang and Low [13] show how a flat Dirichlet Multinomial distribution may be utilized to represent the posterior in interactive Bayes-optimal RL by an agent interacting with other self-interested agents. Differing from our context, the state is assumed to be perfectly observable.

3. BACKGROUND

Perkins' Monte Carlo Exploring Starts for POMDPs (MCES-P) [5] offers a template for online model-free RL in a way that differs substantially from traditional model-free methods such as Q-learning. At its core, it hill climbs the space of neighboring policies, which draws comparisons to policy iteration rather than the value iteration of Q-learning. A random observation sequence, $\vec{o}$, and corresponding action are picked and the latter replaces the previous action at $\vec{o}$ thereby transforming the policy. [1] This is analogous to Sutton's Monte Carlo exploring starts scheme [1] that picked a random state to begin explorations.

To decide whether the transformed policy should be retained instead of the original policy, each policy is simulated to obtain a trajectory of actions, observations and rewards for T time steps: $\tau = \{a^0, r^0, o^1, a^1, r^1, o^2, a^2, r^2, \ldots, o^{T-1}, a^{T-1}, r^{T-1}\}$. Let $R(\tau) = \sum\limits_{t=0}^{T-1} \gamma^t\ r^t$ be the discounted sum of rewards gathered in trajectory τ where $\gamma \in (0,1)$ is the discount factor. Let $R_{pre-\vec{o}}(\tau)$ and $R_{post-\vec{o}}(\tau)$ be the sum of portions of $R(\tau)$ gathered before and after the appearance of $\vec{o}$ in the trajectory, respectively. Notice that the value of a policy may be written as,

$$\begin{aligned} Q_\pi &\approxeq E^{\tau\sim\pi}[R(\tau)] \\ &= E^{\tau\sim\pi}[R_{pre-\vec{o}}(\tau) + R_{post-\vec{o}}(\tau)] \\ &= E^{\tau\sim\pi}[R_{pre-\vec{o}}(\tau)] + E^{\tau\sim\pi}[R_{post-\vec{o}}(\tau)] \end{aligned}$$

Comparing between a policy and its transformation at $\vec{o}$ is facilitated by this decomposition: it allows focus on the difference in the value of the second term obtained for the two policies as the value of the first term remains unchanged because the changed action on observing $\vec{o}$ cannot impact $R_{pre-\vec{o}}(\tau)$.

Subsequently, MCES-P proceeds by randomly picking an observation sequence at which to transform a policy, sufficiently simulating the original and transformed policies and updating their Q-values with new information for comparison. The transformed policy is adopted if its Q-value exceeds that of the original by ϵ both updated across k samples. The algorithm terminates in the absence of policy transformations for some time.

MCESP+PAC Perkins notes a particular instantiation of MCES-P that utilizes Hoeffding's inequality to ensure that a sufficient number k of samples are taken so that a difference of given ϵ is observed with probability $1-\delta$ or more. This instantiation theoretically founded on Greiner's probably approximately locally optimal learning system [14] derives the following value difference threshold between the Q-values of the two policies based on the number of samples, k_m and probability, δ_m at stage m:

$$\epsilon(m,p,q) = \begin{cases} 2T(R_{max} - R_{min})\sqrt{\frac{1}{2p}\ln\frac{2(k_m-1)N}{\delta_m}} & \text{if } p = q < k_m \\ \frac{\epsilon}{2} & \text{if } p = q \geq k_m \\ +\infty & \text{otherwise} \end{cases}$$

Here, p and q are the number of sampled trajectories simulated from the original and transformed policies, respectively; these must be equal for a comparison. Sample complexity k_m is derived as:

$$k_m \leftarrow \left\lceil 2\frac{(2T(R_{max} - R_{min}))^2}{\epsilon^2} \ln\frac{2N}{\delta_m} \right\rceil$$

and

$$\delta_m \leftarrow \frac{6\delta}{m^2\pi^2}$$

where N is the size of the neighborhood of a policy; $R_{max} = \max_{s,a} R(s,a)$ and $R_{min} = \min_{s,a} R(s,a)$. If we allow the policy to change by a single action only, then the size of N is $\mathcal{O}(|A|\frac{|\Omega|^T-1}{|\Omega|-1})$, which is much less than the entire space of policies $\mathcal{O}(|A|^{\frac{|\Omega|^T-1}{|\Omega|-1}})$; notice that the former is no longer exponential in number of observations. The algorithm terminates if the current policy remains better after k_m samples are reached for both the current and transformed policies or if no neighboring policy exceeds it by more than $\epsilon - 2T(R_{max} - R_{min})\sqrt{\frac{1}{2p}\ln\frac{2(k_m-1)N}{\delta_m}}$ when the number of samples p is less than k_m. We note that the method requires prior knowledge of the maximum and minimum rewards across all states and actions, R_{max} and R_{min}, respectively.

An appealing property of MCESP+PAC is that the policy on termination is guaranteed to be ϵ-*locally optimal*. In other words, no neighboring policy exists that exceeds the current policy in value greater than ϵ.

[1] Perkins' MCES-P curbs the policies to actions contingent on a single observation, ie., memory-less policies. Therefore, it initially picks a single observation and action only. We extend the algorithm here to consider a *sequence* of observations for better performance.

4. RL IN MULTIAGENT SETTINGS

In our review of related work as outlined in Section 2, we did not find any algorithm for truly model-free RL in partially observable multiagent settings (technically our setting is a partially observable stochastic game). [2] Subsequently, MCES-P offers an appealing template for generalizing to multiagent settings with the potential to fill this wide gap. While we focus on learning by *self-interested* agents situated in a multiagent setting in this paper, MCES-P is sufficiently generic for extension to learning the joint policies of multiple agents in cooperative settings as well – an investigation that we defer to future work.

Observe that the MCES-P template briefly discussed in Section 3 may be utilized almost as is in a multiagent setting. Therefore, we begin by exploring this approach followed by presenting a new generalization of MCES-P that improves on it significantly.

4.1 MCESP+PAC in Multiagent Settings

As we mentioned previously, the template offered by MCES-P continues to remain viable for a learning agent i when we generalize from single- to multi-agent settings. Toward this end, we reproduce Perkins' algorithm below in Algorithm 1. Let π_i be the initial seed policy and $\pi_i \leftarrow (\vec{o}_i, a_i)$ be this policy but transformed to perform action a_i on observing $\vec{o}_i$; the latter denotes a neighboring policy.

Algorithm 1 MCES-P

```
Require: Q-value table initialized; initial policy $\pi_i$ that is greedy w.r.t. Q-values; learning rate schedule $\alpha$; horizon $T$; and error $\epsilon$
$c_{\vec{o}_i,a_i} \leftarrow 0$ for all $\vec{o}_i$ and $a_i$
$m \leftarrow 0$
repeat
    Pick some observation history $\vec{o}_i$ and action $a_i$
    Modify $\pi_i$ to $\pi_i \leftarrow (\vec{o}_i, a_i)$
    Generate trajectory $\tau$ of length $T$ according to $\pi_i \leftarrow (\vec{o}_i, a_i)$ (this involves simulating the implicit policies of other agents as well)
    $Q_{\pi_i \leftarrow \vec{o}_i,a_i} \leftarrow (1 - \alpha(m, c_{\vec{o}_i,a_i}))\ Q_{\pi_i \leftarrow \vec{o}_i,a_i} + \alpha(m, c_{\vec{o}_i,a_i})\ R_{post-\vec{o}_i}(\tau)$
    $c_{\vec{o}_i,a_i} \leftarrow c_{\vec{o}_i,a_i} + 1$
    if $\max_{a_i'} Q_{\pi_i \leftarrow \vec{o}_i,a_i'} - Q_{\pi_i} > \epsilon(m, c_{\vec{o}_i,a_i}, c_{\vec{o}_i,\pi_i(\vec{o}_i)})$ then
        $\pi_i(\vec{o}_i) \leftarrow a_i'$ where $a_i' \in \arg\max Q_{\pi_i \leftarrow \vec{o}_i,a_i'}$
        $m \leftarrow m + 1$
        for all $\vec{o}_i, a_i$ do
            $c_{\vec{o}_i,a} \leftarrow 0$
until termination
```

In instantiating the above template for PAC bounds, we assume that the other agent is guided by a fixed policy or a fixed distribution over policies (i.e., mixed strategy). This ensures that the sampling distribution is fixed and Hoeffding's inequality continues to apply. We point out that this assumption differentiates our problem from the traditional multiagent RL [15] where all agents are learning simultaneously and therefore the learning problem is not stationary.

Given the above, MCESP+PAC in multiagent settings may face the same four types of errors due to sampling as those faced in single agent settings (see proof of Theorem 1 in Appendix). However, the error due to sampling and the sample complexity differ because of the presence of other agents although the size of the policy neighborhood remains the same. Let $\Lambda(\pi_i, \pi_i')$ be the upper bound on the range of the difference between the action-values of π_i and some other policy π_i'. For simplicity of presentation, we consider one other agent j in the environment. Let $R_{i,max}$ now be defined as, $R_{i,max} \triangleq \max_{s,a_i,a_j} R_i(s, a_i, a_j)$ and analogously for $R_{i,min} \triangleq \min_{s,a_i,a_j} R_i(s, a_i, a_j)$. Then, we get,

$$\begin{aligned}\Lambda(\pi_i, \pi_i') &\triangleq \max_{\tau}\ (Q_{\pi_i} - Q_{\pi_i'}) - \min_{\tau}\ (Q_{\pi_i} - Q_{\pi_i'}) \\ &\leq \sum_{t=0}^{T-1} \{(R_{i,max} - R_{i,min}) - (R_{i,min} - R_{i,max})\} \\ &= \sum_{t=0}^{T-1} 2(R_{i,max} - R_{i,min}) = 2T(R_{i,max} - R_{i,min})\end{aligned} \tag{1}$$

Consequently, the threshold for comparison at stage m becomes:

$$\epsilon(m, p, q) = \begin{cases} \Lambda(\pi_i, \pi_i') \sqrt{\frac{1}{2p} \ln \frac{2(k_m - 1)N}{\delta_m}} & \text{if } p = q < k_m \\ \frac{\epsilon}{2} & \text{if } p = q \geq k_m \\ +\infty & \text{otherwise} \end{cases}$$

and the sample complexity is:

$$k_m = \left\lceil 2 \frac{(\Lambda(\pi_i, \pi_i'))^2}{\epsilon^2} \ln \frac{2N}{\delta_m} \right\rceil$$

while the probability δ_m remains the same as in Section 3. Notice that in so called "neutral settings" where agent i's reward does not depend on j's action (although the state is still impacted by j's action) $R_{i,max}$ and $R_{i,min}$ collapse into R_{max} and R_{min}, respectively causing no change in the PAC bounds from Section 3. However, in competitive settings $R_{i,max}$ is often greater than R_{max} whereas $R_{i,min}$ tends to be smaller than its counterpart due to which the sample complexity is higher or the error is greater for the same number of samples.

MCESP+PAC terminates if there has been no policy change when k_m samples are reached or if no neighboring policy exceeds the current policy in action value by more than $\epsilon - \Lambda(\pi_i, \pi_i') \cdot \sqrt{\frac{1}{2p} \ln \frac{2(k_m - 1)N}{\delta_m}}$ for lesser samples.

Theorem 1 shows that MCESP+PAC in partially observable multiagent settings will terminate and converge to an ϵ-locally optimal policy.

THEOREM 1 (LOCAL OPTIMALITY OF MCESP+PAC). *Instantiation MCESP+PAC incrementally produces a series of policies π_i^1, π_i^2, ..., π_i^m, such that each π_i^{n+1} is a local neighbor of π_i^n and with probability at least $1 - \delta$:*

1. *Each policy π_i^{n+1} has an expected value strictly greater than its predecessor, π_i^n where $1 \leq n \leq m - 1$;*
2. *Final policy π_i^m returned by MCESP+PAC is ϵ-locally optimal such that there is no neighbor of π_i^m given our transformation procedure whose expected value exceeds that of π_i^m by more than ϵ.*

Moreover, MCESP+PAC will terminate with probability 1 if N is finite.

The full proof of this theorem is given in the Appendix. It presents four types of errors that are possible due to sampling. As part of the proof, the expressions for $\epsilon(m, p, q)$, k_m and δ_m are also established.

4.2 MCES-IP Template for RL

While Theorem 1 is appealing, a limitation of MCESP+PAC is that the required sample size is very large. For example, for an ϵ of 0.05 and probability δ = 0.1 the required sample size k_m is 320,200 for the small two-agent competitive Tiger problem. Can we significantly reduce this alarming sample complexity and if so what is the trade off?

[2] Note that MCQ-Alt [9] learns an estimate of the model from samples as an intermediate step and is therefore quasi model based.

A key insight may allow us to mitigate the sample complexity: *If we can predict the other agent's actions, then we may simply optimize in that specific context.* Toward this, we partially relax the model-free characteristic of the RL to obtain savings in samples. We may divide agent i's observations into those that are public and those that are private. The former type are public signals that are shared between agents while the latter are signals privately observed by an agent. This division is without loss of generality because the set of public observations may be empty if agents have private observations only and the set of private observations is empty if only public observations are obtained.

Next, let the private monitoring at time t convey information to the agent about the other agent's action at $t-1$ albeit noisily while the public signal provides information about the common state of the system. We argue that this specificity is often seen in multiagent problems. For example, agents in the multiagent Tiger problem hear growls that inform about the location of the tiger and each agent may also hear a creak from the left or right, or do not hear a creak indicating the door, if any, that was opened by the other agent.

As a final step, we depart from the completely model-free setting of MCES-P and let a joint private observation function that models the information content of private observations only be common knowledge. The agents are not aware of any other model parameters.

Given the setup above, agents in the MCES for interactive POMDP (MCES-IP) template start with a prior distribution over possible models of the other agent and update the prior as they receive private observations. As the agent acts and observes, a sequence of beliefs are obtained and the agent utilizes both sequences: observation sequence and belief sequence to decide on an action. We show the MCES-IP template in Algorithm 2.

Algorithm 2 MCES-IP

```
Require: Q-value table initialized; initial policy, $\pi_i$, that is greedy w.r.t.
    Q-values; prior on set of models $M_j$; learning rate schedule $\alpha$; horizon
    $T$; and error $\epsilon$
1:  $c^{\vec{a}_j}_{\vec{o}_i,a_i} \leftarrow 0$
2:  $m \leftarrow 0$
3:  repeat
4:      Pick some observation history, $\vec{o}_i$, and $a_i$
5:      Modify $\pi_i$ to $\pi_i \leftarrow (\vec{o}_i, a_i)$
6:      Generate trajectory $\tau$ of length $T$ according to $\pi_i \leftarrow (\vec{o}_i, a_i)$
7:      Generate belief sequence $\vec{b}_i$ based on $\tau$
8:      Obtain most probable action sequence $\vec{a}_j$ from $\vec{b}_i$
9:      $Q^{\vec{a}_j}_{\pi_i \leftarrow \vec{o}_i,a_i} \leftarrow (1-\alpha(m, c^{\vec{a}_j}_{\vec{o}_i,a_i})) \cdot Q^{\vec{a}_j}_{\pi_i \leftarrow \vec{o}_i,a_i} + \alpha(m, c^{\vec{a}_j}_{\vec{o}_i,a_i}) \cdot$
        $R_{post-\vec{o}_i}(\tau)$
10:     $c^{\vec{a}_j}_{\vec{o}_i,a_i} \leftarrow c^{\vec{a}_j}_{\vec{o}_i,a_i} + 1$
11:     if $\max_{a'_i} Q^{\vec{a}_j}_{\pi_i \leftarrow \vec{o}_i,a'_i} - Q^{\vec{a}_j}_{\pi_i} > \epsilon^{\vec{a}_j}(m, c^{\vec{a}_j}_{\vec{o}_i,a_i}, c^{\vec{a}_j}_{\vec{o}_i,\pi_i(\vec{o}_i)})$ then
12:         $\pi_i(\vec{o}_i) \leftarrow a'_i$ where $a'_i \in \arg\max_{a'_i} Q^{\vec{a}_j}_{\pi_i \leftarrow \vec{o}_i,a'_i}$
13:         $m \leftarrow m + 1$
14:         for all $\vec{o}_i, a_i, \vec{a}_j$ do
15:             $c^{\vec{a}_j}_{\vec{o}_i,a_i} \leftarrow 0$
16: until termination
```

MCES-IP generally follows the procedure of MCES-P. It builds on the latter by additionally generating a belief sequence $\vec{b}_i$ using the actions and observations in a trajectory (line 7); each belief is a distribution over a pre-defined set of models of the other agent. As the space of possible belief sequences is continuous, MCES-IP picks the most-probable model from each belief and the corresponding predicted action, to obtain a corresponding action sequence $\vec{a}_j$ (line 8). Q-values and update counts are now indexed using this action sequence (lines 9-10). In this way, action values of polices are specific to predicted actions of the other agent, which allow a more informed probabilistic hill climbing in multiagent settings.

Belief sequence, $\vec{b}_i$, is generated as follows. A trajectory τ_i is $\{a_i^0, r_i^0, \langle o^1, \omega_i^1\rangle, a_i^1, r_i^1, \langle o^2, \omega_i^2\rangle, a_i^2, r_i^2, \ldots, \langle o^{T-1}, \omega_i^{T-1}\rangle, a_i^{T-1}, r_i^{T-1}\}$. Notice that each observation in the trajectory is composed of public and private signals; denote $\vec{o}_i \triangleq \langle \vec{o}, \vec{\omega}_i \rangle$. Let M_j^t be a discrete set of j's models at time step t, where $m_j^t \in M_j^t$ is: $m_j^t \triangleq \langle h_j^t, \pi_j \rangle$, h_j^t is j's action-observation history of length t which when given as input to j's policy π_j produces the predicted action at time t. Agent i's belief b_i is a distribution over M_j updated based on i's action and observation as given below:

$$b'_i(m_j^{t+1}|a_i^t, o^{t+1}, \omega_i^{t+1}, b_i) = \sum_{m_j^t \in M_j^t} b_i(m_j^t) \sum_{a_j^t \in A_j} Pr(a_j^t|m_j^t)$$
$$\times\, O_i(\omega_i^{t+1}|a_i^t, a_j^t)\, \delta_K(h_j^{t+1}, \text{APPEND}(h_j^t, a_j^t, o^{t+1})) \qquad (2)$$

As a part of updating its belief, i must first update its models of j and in particular, the action-observation history contained in each model using the predicted action a_j^t and public observation o^{t+1}; this is performed by APPEND. Kronecker delta function, δ_K, is 1 if an updated model matches the one in m_j^{t+1} otherwise it is 0. Private observation function O_i is the marginal of the joint, and it allows using the private signal to weight predicted actions and by backward inference the models that generated the actions. This likelihood is then propagated forward to the updated model, m_j^{t+1}. As such, Eq. 2 is a sophisticated Bayesian belief update.

A sequence of beliefs $\vec{b}_i$ is then generated by updating the uniform prior with the action-observation pairs in a trajectory using Eq. 2; thus the length of this sequence is T. We may pick the most probable model from each belief in the sequence, $\arg\max_{m_j} b_i(m_j)$, and get the model-predicted action, $\arg\max_{a_j} Pr(a_j|m_j)$, to obtain the action sequence $\vec{a}_j$.

MCESIP+PAC We instantiate Algorithm 2 to obtain MCES-IP with PAC bounds, which we denote as MCESIP+PAC. For this, we assume that the error is due to sampling trajectories only and that the monitoring is perfect, ie., private signals perfectly reveal j's action. We discuss the effect on the bounds due to observation noise later in this section. For a given error ϵ and probability δ let,

$$\epsilon^{\vec{a}_j}(m,p,q) = \begin{cases} \Lambda^{\vec{a}_j}(\pi_i, \pi'_i)\sqrt{\frac{1}{2p}\ln\frac{2(k_m-1)N}{\delta_m}} & \text{if } p = q < k_m \\ \frac{\epsilon}{2} & \text{if } p = q \geq k_m \\ +\infty & \text{otherwise} \end{cases}$$

where

$$k_m = \left\lceil 2\frac{(\Lambda^{\vec{a}_j}(\pi_i, \pi'_i))^2}{\epsilon^2} \ln\frac{2N}{\delta_m} \right\rceil$$

and

$$\delta_m = \frac{6\delta}{m^2\pi^2}$$

Here, $\Lambda^{\vec{a}_j}(\pi_i, \pi'_i)$ is an upper bound on the range of the difference in action-values between two policies given j's action sequence is $\vec{a}_j$. Let $R^{a_j}_{i,max} = \max_{s,a_i} R_i(s, a_i, a_j)$ and analogously for $R^{a_j}_{i,min}$; these specific values are assumed to be known. Then, we get:

$$\begin{aligned}\Lambda^{\vec{a}_j}(\pi_i, \pi'_i) &= \max_{\tau}\ \left(Q^{\vec{a}_j}_{\pi_i} - Q^{\vec{a}_j}_{\pi'_i}\right) - \min_{\tau}\ \left(Q^{\vec{a}_j}_{\pi_i} - Q^{\vec{a}_j}_{\pi'_i}\right) \\ &\leq \sum_{t\in T}\left(R^{a_j^t}_{i,max} - R^{a_j^t}_{i,min}\right) - \left(R^{a_j^t}_{i,min} - R^{a_j^t}_{i,max}\right) \\ &= \sum_{t\in T} 2\left(R^{a_j^t}_{i,max} - R^{a_j^t}_{i,min}\right) \qquad (3)\end{aligned}$$

We note the following key proposition that indicates the benefit of predicting the other agent's actions on sample complexity albeit at the expense of the belief update run time.

PROPOSITION 2 (REDUCED SAMPLE COMPLEXITY). *For any predicted action sequence, $\vec{a}_j$,*

$$\Lambda^{\vec{a}_j}(\pi_i, \pi_i') \leq \Lambda(\pi_i, \pi_i')$$

where Eqs. 1 and 3 define $\Lambda(\pi_i, \pi_i')$ and $\Lambda^{\vec{a}_j}(\pi_i, \pi_i')$, respectively. The proof of this proposition is straightforward and is given in the Appendix. Subsequently, Proposition 2 entails that the sample size bound k_m for MCESIP+PAC is also less than or equal to the corresponding sample size bound for MCESP+PAC. The effect is significant because k_m grows quadratically with Λ. On the other hand, the Q-values table expands significantly with up to $\frac{|\Omega|^T-1}{|\Omega|-1}$ values for each i's policy or its transformation. As such, the reduced sample bound of MCESIP+PAC must be less by a factor of $\frac{|\Omega|^T-1}{|\Omega|-1}$ in the worst case to be effective. This is often the case as we demonstrate for the two-agent Tiger problem where k_m for MCESIP+PAC is as low as 106,822 that is almost three times less than that for MCESP+PAC.

If private signals provide perfect information about j's actions, then MCESIP+PAC terminates when,

$$Q^{\vec{a}_j}_{\pi_i \leftarrow \vec{o}_i, a_i'} < Q^{\vec{a}_j}_{\pi_i} + \epsilon - \epsilon^{\vec{a}_j}(m, c^{\vec{a}_j}_{\vec{o}_i, a_i}, c^{\vec{a}_j}_{\vec{o}_i, \pi_i(\vec{o}_i)})$$

for all $\vec{o}_i$, $a_i' \neq \pi_i(\vec{o}_i)$, and we have encountered at most $\frac{|\Omega|^T-1}{|\Omega|-1}$ many distinct $\vec{a}_j$ in the trajectories for each $\vec{o}_i$, a_i' pair. Under the same assumption of perfect monitoring, for the comparison threshold, sample bound and probability as defined above, the following theorem obtains for MCESIP+PAC.

THEOREM 3 (LOCAL OPTIMALITY UNDER PERFECT MONITORING). *Template MCESIP+PAC under perfect monitoring incrementally produces a series of policies π_i^1, π_i^2, ..., π_i^m, such that each π_i^{q+1} is a local neighbor of π_i^q and with probability at least $1 - \delta$:*

1. *Each policy π_i^{q+1} has an expected value strictly greater than its predecessor, π_i^q where $1 \leq q \leq m-1$;*

2. *Final policy, π_i^m returned by MCESIP+PAC is ϵ-locally optimal such that there is no neighbor of π_i^m given our transformation procedure whose expected value exceeds that of π_i^m by more than ϵ.*

Moreover, MCESIP+PAC terminates with probability 1 if N is finite.

The proof of this theorem proceeds analogously to the proof for Theorem 1, with Λ replaced with $\Lambda^{\vec{a}_j}$.

We can generalize the above results on MCESIP+PAC for the case of *imperfect monitoring* – when the probability of error in estimating $\vec{a}_j$ is known, say δ_e. In this case, the agent may place Q-samples in the wrong $Q^{\vec{a}_j}$ bins (see line 9 of Algorithm 2), leading to non-identically distributed samples in a bin. Fortunately, given δ_e we may generalize Theorem 3 to the case of independent but non-identically distributed samples using a more general form of Hoeffding's inequality. Analogously to Eq. 3, let $\bar{\Lambda}^{\vec{a}_j}$ be an upper bound on the range of differences in action-values for all j's action sequences that are *different* from $\vec{a}_j$. Then for the case of $p = q < k_m$, we redefine $\epsilon^{\vec{a}_j}(m, p, q)$ as,

$$\epsilon^{\vec{a}_j}(m, p, q) = \sqrt{(1-\delta_e)(\Lambda^{\vec{a}_j})^2 + \delta_e(\bar{\Lambda}^{\vec{a}_j})^2}\sqrt{\frac{1}{2p}\ln\frac{2(k_m-1)N}{\delta_m}}$$

where k_m is redefined as

$$k_m = \left\lceil \frac{2((1-\delta_e)(\Lambda^{\vec{a}_j})^2 + \delta_e(\bar{\Lambda}^{\vec{a}_j})^2)}{\epsilon^2} \ln\frac{2N}{\delta_m} \right\rceil$$

Algorithm 2 requires a slight modification for this case. For convenience, let $\zeta^{\vec{a}_j} = \max_{a_i'} Q^{\vec{a}_j}_{\pi_i \leftarrow \vec{o}_i, a_i'} - Q^{\vec{a}_j}_{\pi_i}$. Then, line 11 of Algorithm 2 changes to the following:

$$(1-\delta_e)\zeta^{\vec{a}_j} + \delta_e\bar{\zeta}^{\vec{a}_j'} > (1-\delta_e)\epsilon^{\vec{a}_j} + \delta_e\bar{\epsilon}^{\vec{a}_j'}$$

The implicit assumption above is that when $Q^{\vec{a}_j}$ receives a wrong sample meant for bin $\vec{a}_j'$, the action sequence is equally likely to be any $\vec{a}_j' \neq \vec{a}_j$. Therefore, $\bar{\zeta}^{\vec{a}_j'}$ is the mean of $\zeta^{\vec{a}_j'}$ for all $\vec{a}_j' \neq \vec{a}_j$ seen so far, and analogously $\bar{\epsilon}^{\vec{a}_j'}$ is also the mean. Notice that insisting on the test of line 11 for every $\vec{a}_j$ before the current policy is changed, would be a stronger form of this test, and hence also sufficient. Finally, we note that when $\delta_e = 0$, we recover MCESIP+PAC for perfect monitoring as a special case of this setting.

5. PRUNING POLICY SEARCH SPACE

Theorems 1 and 3 require exploring all local transformations of a policy for establishing local optimality. However, some of these transformations prescribe differing actions in response to observation sequences that are not likely to occur. Yet, we are required to obtain k_m samples of trajectories involving such observation sequences or establish a significant difference in action-values for less numbers of such samples. This contributes significantly to long run times of the algorithms as we noticed in our experiments. Subsequently, we seek ways to remove such rare observation sequences from consideration thereby pruning the policy search space. As these sequences are relatively much less likely they also contribute less to the expected value of a policy, but not considering them nonetheless introduces *regret* that we seek to compute.

Ignoring a different action at some observation sequence $\vec{o}_i$ is regrettable because we are foregoing the possibility of improving the expected value of i's policy. Of course, a less likely observation sequence may not add much to the expected value of the current policy. Nevertheless, let ϕ be a user-defined bound on allowable regret. By avoiding transforming on $\vec{o}_i$, we are foregoing at most the largest post-$\vec{o}_i$ rewards; specifically this regret is upper bounded by $\max_\tau R_{post-\vec{o}_i}(\tau) - \min_\tau R_{post-\vec{o}_i}(\tau)$. Regret on the expected value of the policy is bounded by

$$\begin{aligned} regret_{\vec{o}_i} &\leq Pr(\vec{o}_i; \pi_i, \pi_j)\left(\max_\tau R_{post-\vec{o}_i}(\tau) - \min_\tau R_{post-\vec{o}_i}(\tau)\right) \\ &= Pr(\vec{o}_i; \pi_i, \pi_j)\,(T - len(\vec{o}_i))\,\,(R_{i,max} - R_{i,min}) \end{aligned} \tag{4}$$

Here, $Pr(\vec{o}_i; \pi_i, \pi_j)$ is the likelihood of the observation sequence $\vec{o}_i$ whose computation depends on the actions prescribed by both agents' policies, state transition and observation functions; rewards $R_{i,max}$ and $R_{i,min}$ are as defined previously. We may normalize the regret to obtain a proportion between 0 and 1 as, $re\bar{g}ret_{\vec{o}_i} = \frac{regret_{\vec{o}_i}}{T(R_{i,max} - R_{i,min})}$.

Of course, not knowing π_j and the model parameters implies that we cannot compute $Pr(\vec{o}_i; \pi_i, \pi_j)$ exactly. Hence, as a first step in this paper, we settle for a crude approximation where $Pr(\vec{o}_i; \pi_i, \pi_j)$ is estimated by the fraction of times $\vec{o}_i$ occurs in the k_m or more samples generated so far. This requires keeping a count of each observation sequence encountered in the trajectories.

Let $\mathcal{P}$ be the set of i's observation sequences that will be avoided. If a bound ϕ on the regret is given by the user, we may obtain $\mathcal{P}$ in a straightforward way: Sort the set of all observation sequences of all

Algorithm 3 MCES-IP_Prune

```
Require: Q-value table initialized; initial policy, $\pi_i$, that is greedy w.r.t. Q-values; prior on set of models $M_j$; learning rate schedule $\alpha$; horizon $T$; error $\epsilon$; and regret bound $\phi$
 1: $c^{\vec{a}_j}_{\vec{o}_i,a_i} \leftarrow 0$, $c_{\vec{o}_i} \leftarrow 0$ for all $\vec{o}_i$, $a_i$, and $a_j$
 2: $m \leftarrow 0$
 3: $\mathcal{P} \leftarrow \emptyset$
 4: repeat
 5:     Pick some observation history $\vec{o}_i$ and action $a_i$
 6:     $c_{\vec{o}_i} \leftarrow c_{\vec{o}_i} + 1$
 7:     if $\vec{o}_i \notin \mathcal{P}$ then
 8:         Modify $\pi_i$ to $\pi_i \leftarrow (\vec{o}_i, a_i)$
 9:     else
10:         Go to line 5
11:     Generate trajectory $\tau$ of length $T$ according to $\pi_i \leftarrow (\vec{o}_i, a_i)$
12:     Generate belief sequence $\vec{b}_i$ based on $\tau$
13:     Obtain most probable action sequence $\vec{a}_j$ from $\vec{b}_i$
14:     $Q^{\vec{a}_j}_{\pi_i \leftarrow \vec{o}_i,a_i} \leftarrow (1-\alpha(m, c^{\vec{a}_j}_{\vec{o}_i,a_i})) \cdot Q^{\vec{a}_j}_{\pi_i \leftarrow \vec{o}_i,a_i} + \alpha(m, c^{\vec{a}_j}_{\vec{o}_i,a_i}) \cdot R_{post-\vec{o}_i}(\tau)$
15:     $c^{\vec{a}_j}_{\vec{o}_i,a_i} \leftarrow c^{\vec{a}_j}_{\vec{o}_i,a_i} + 1$
16:     if $\max_{a'_i} Q^{\vec{a}_j}_{\pi_i \leftarrow \vec{o}_i,a'_i} \geq Q^{\vec{a}_j}_{\pi_i} + \epsilon^{\vec{a}_j}(m, c^{\vec{a}_j}_{\vec{o}_i,a_i}, c^{\vec{a}_j}_{\vec{o}_i,\pi_i(\vec{o}_i)})$ then
17:         $\pi_i(\vec{o}_i) \leftarrow a'_i$
18:         $m \leftarrow m + 1$
19:         for all $\vec{o}_i, a_i, \vec{a}_j$ do
20:             $c^{\vec{a}_j}_{\vec{o}_i,a_i} \leftarrow 0$
21:     if $\sum_{\vec{o}_i \in \mathcal{P} \cup \vec{o}_i} re\bar{g}ret_{\vec{o}_i} \leq \phi + \rho(\sum_{\vec{o}_i} c_{\vec{o}_i})$ then
22:         $\mathcal{P} \leftarrow \mathcal{P} \cup \vec{o}_i$
23: until termination
```

lengths by their frequency of occurrence in ascending order. Then, add a sequence into $\mathcal{P}$ beginning with the least frequent and moving up the ordering such that, $\sum_{\vec{o}_i \in \mathcal{P}} re\bar{g}ret_{\vec{o}_i} \leq \phi$. Consequently, a bound that is more loose would allow disregarding more observation sequences to meet it and thereby prune a larger portion of the search space. Alternately, the sorting is not necessary and we may simply pick an observation sequence at random and check if adding it to $\mathcal{P}$ would cause the cumulative normalized regret to exceed ϕ.

The pruning mechanism described previously may be incorporated into both the MCES-P and MCES-IP templates. Algorithm 3 outlines how it may be utilized with MCES-IP. Two additions can be observed. Lines 7-10 make a determination if the current $\vec{o}_i$ is in the set $\mathcal{P}$ of sequences that will not be considered (initialized to the empty set), and if not policy π_i is locally transformed at $\vec{o}_i$. Lines 21-22 add the observation sequence $\vec{o}_i$ into set $\mathcal{P}$ if the cumulative normalized regret due to all observation sequences in $\mathcal{P}$ including $\vec{o}_i$ remains less than or equal to given bound $\phi + \rho(\sum_{\vec{o}_i} c_{\vec{o}_i})$, where

$$\rho(i) = \begin{cases} 0 & i \geq k \\ +\infty & \text{otherwise} \end{cases}$$

Thus, $\mathcal{P}$ remains empty unless a reasonable number of samples are obtained. In the case of MCESIP+PAC instantiation of the template, k could be simply set to the derived sample bound $\frac{k_m}{2}$.

6. EXPERIMENTS

We implement MCESP+PAC and MCESIP+PAC to obtain converged policies for two strategic and noncooperative multiagent domains. We first experiment with a competitive, multiagent version of the Tiger problem [8]. Our second domain is a 3×2 autonomous unmanned aerial vehicle (AUAV) reconnaissance domain [16]. We set up the AUAV domain by beginning every round with the AUAV in the bottom-left sector and the fugitive in the top-right of a 3×2 grid, and all sectors in the leftmost column are considered safe. Each agent has 3 actions: the AUAV may move left, right, or up, and the fugitive may move left, right, or down. Both agents receive 1 of 4 noisy public observations, representing whether both agents are East or West of each other, North or South of each other, in the same sector, or none of these. The subject agent, which is the AUAV, additionally receives 1 of 3 noisy private observations each correlated with an action the opponent takes. A third domain is the money laundering (ML) problem [17] where a blue team seeks to confiscate illicit money that the opponent red team is laundering. The red team can move money from the initial state to a series of placement states (banks and insurance), to layering states (offshore accounts and shell companies), to integration states (casinos and real estate), and to the safe clean pot. The blue team may place a sensor at each of these locations or confiscate the illicit funds. Each agent receives a noisy public observation indicating whether the money and sensor are in the same location, in the same laundering state, or if neither are the case. The blue team also receives a noisy observation of the last action of the red team.

Table 1 summarizes the domain statistics and parameter settings. For all domains and both methods, each agent has a 15% chance of receiving a noisy public and private observation. The opponent in both games follows a single policy (stationary environment) or fixed distribution over multiple policies (nonstationary environment).

Domain	Specifications
Multiagent Tiger	$\epsilon = 0.05$, $\delta = 0.1$, $\phi = 0.15$, $T = 3$, $\lvert\Omega\rvert = 2$, $\lvert A_i\rvert = 3$, $\lvert A_j\rvert = 3$, $\lvert\Pi_j\rvert = 14$
3×2 AUAV	$\epsilon = 0.1$, $\delta = 0.1$, $\phi = 0.2$, $T = 3$, $\lvert\Omega\rvert = 4$, $\lvert A_i\rvert = 3$, $\lvert A_j\rvert = 3$, $\lvert\Pi_j\rvert = 4$
Money Laundering	$\epsilon = 0.1$, $\delta = 0.15$, $\phi = 0.2$, $T = 3$, $\lvert\Omega\rvert = 4$, $\lvert A_i\rvert = 4$, $\lvert A_j\rvert = 5$, $\lvert\Pi_j\rvert = 8$

Table 1: Parameter configurations for the three problem domains.

We simulate i's policies with opponent j following either a single policy or a mixture of two policies. These policies are picked from a predefined set Π_j. As per the policy space specified in Table 1, 105 games of the multiagent Tiger problem, 9 of the 3×2 AUAV problem, and 13 of ML comprise the data set.

First, we show that MCESP+PAC and MCESIP+PAC demonstrates the PAC guarantee of monotonically increasing successive transformations. Figure 1 illustrates three example runs and the values for MCESP+PAC and MCESIP+PAC given the same opponent. Different runs undergo varying number of transformations with some policies not transforming at all because they are ϵ-locally optimal initially itself. As shown in Fig. 1, each successive transformation results in a higher value and in no case is the final policy lower in value than the initial policy as we should expect. In every case, MCESP+PAC requires more samples to transform than MCESIP+PAC.

Second, Table 2 lists the mean of the theoretical sample bound k_m across the different stages m over all runs and the mean of the effective number of samples over all runs that were utilized by both methods. In validation of our theoretical result, MCESIP+PAC requires remarkably fewer samples to transform per action sequence, around half in multiagent Tiger, about a quarter in 3×2 AUAV, and nearly a tenth in Money Laundering. Additionally, the bound on k_m is also significantly less compared to the bound for MCESP+PAC. Due to stochasticity in the simulations and finite sampling bounds, MCESP+PAC and MCESIP+PAC may deviate in transformation paths. However, in over 80% of the runs, both result in the same converged policy with MCESIP+PAC converging on average under half the number of samples taken by MCESP+PAC.

Finally, observation sequence pruning plays a crucial role in

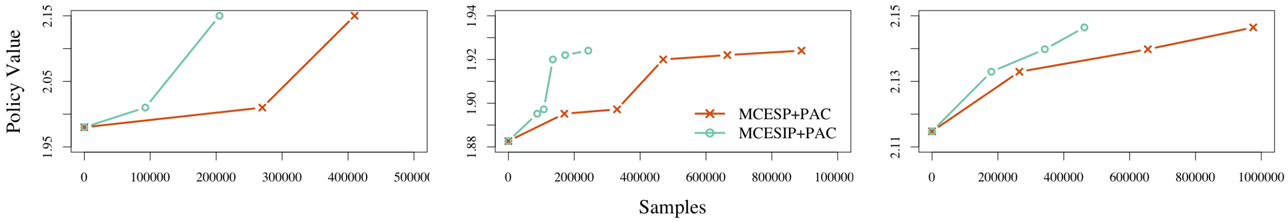

Figure 1: Example policy transformation paths with intermediate values for 3 different opponents in the multiagent Tiger problem. The right most transformation paths are for an opponent with mixed strategy.

Domain	Method	Policy	Mean # of samples per transform	Mean bound on k_m
Tiger	MCESP+PAC	Single	156,328 ± 21,012	265,948 ± 7,909
		Mixed	219,057 ± 9,521	263,565 ± 6,101
	MCESIP+PAC	Single	72,740 ± 5,963	117,590 ± 3,309
		Mixed	117,504 ± 6,678	119,313 ± 1,089
3×2 AUAV	MCESP+PAC	Single	32,397 ± 2,816	91,443 ± 637
		Mixed	42,126 ± 1,689	96,328 ± 118
	MCESIP+PAC	Single	6,437 ± 68	20,397 ± 206
		Mixed	19,499 ± 1,304	22,763 ± 169
ML	MCESP+PAC	Single	20,717 ± 2,418	34,726 ± 617
		Mixed	20,247 ± 4,974	35,612 ± 490
	MCESIP+PAC	Single	1,947 ± 330	24,172 ± 448
		Mixed	3,174 ± 536	24,347 ± 482

Table 2: Mean effective sample size and theoretical bound across the stages for the three problem domains, stratified over method and whether the opponent follows a single policy or a mixed set of policies.

Pruning	Metric	Multiagent Tiger	3×2 AUAV	ML
Without	Neighborhood	128	470	636
	Total k_m	15,893,387	3,704,396	18,911,460
With	Neighborhood	26	32	76
	Total k_m	2,093,328	624,057	2,259,860

Table 3: Neighborhood size and total k_m values for all the domains using their respective parameters in Table 1, with and without pruning for both MCESP+PAC and MCESIP+PAC. Note that the total bound on samples reduces by almost an order of magnitude.

improving the scalability of MCESP+PAC and MCESIP+PAC, dramatically reducing the search space and run time while minimizing the impact on incurred regret. We list the mean of the required total k_m values across all observation sequences for both problem domains in Table 3, both with and without pruning. As the policy search space for both methods is the same, the regret due to pruning the search space does not depend on the method used. Observation sequence pruning benefits both methods equally in reducing the policy search space as we demonstrate in Table 3.

Each neighborhood is calculated with a horizon of 3. For the multiagent Tiger problem, the size of the observation sequence space per round is 6 (2 public × 3 private observations) with 3 possible actions, resulting in a maximum neighborhood of 128. Given a regret bound of 0.15, 34 of 43 distinct observation sequences are eliminated on average, resulting in a neighborhood of 26. For 3×2 AUAV, the observation space is 12 (4 public × 3 private observations) with 3 actions, resulting in a neighborhood of 470. On average, 146 of 157 observation sequences are pruned for a regret bound of 0.2, leaving 32 neighbors. ML's observation space is 9 (3 public × 3 private observations) with 7 actions, resulting in a neighborhood of 636. With a regret bound of 0.2, 80 of 91 sequences are pruned leaving 76 neighbors.

7. CONCLUDING REMARKS

MCES-P offers a template for model-free RL in partially observable settings that differs from traditional Q-learning. It's appealing because it may be instantiated in different ways, resulting in algorithms with differing behavior. In this paper, we generalized MCES-P with PAC bounds to self-interested multiagent settings presenting a model-free RL technique. We exploited the insight that modeling the other agent should catalyze learning leading to a new algorithm, MCESIP+PAC.

MCESIP+PAC dramatically reduces the sample complexity of MCESP+PAC with reductions on sample bounds and empirical sample counts ranging between 50% to 75% less than MCESP+PAC. In nearly every case, MCESIP+PAC is able to achieve the same optima as MCESP+PAC despite the fraction of samples taken. We additionally introduced observation sequence pruning, a methodology that significantly improves run time by reducing the policy search space with bounded regret. We observed over 80% reduction in sample requirement for our domains while introducing a regret between 0.15 and 0.2. Our future work involves extending MCES-P to learning the joint policies in a cooperative setting and analyzing its theoretical properties.

Acknowledgements

This research is supported in part by grants from NSF IIS-0845036 (to PD) and IIS-1526813 (to BB) and a grant from ONR N-00-0-141310870 (to PD). We thank Theodore Perkins for email exchanges that benefited this paper.

APPENDIX

Proof of Theorem 1 MCESP+PAC in the multiagent setting allows for a PAC-style guarantee of ϵ-local optimality. To show that the total error for MCESP+PAC is bounded by the user-defined ϵ with probability $1 - \delta$, we must first define the types of errors that can occur in selecting dominating neighboring policies and terminating when none is found after sampling. In this respect, our proof follows that of Greiner [14].

We define $\mathcal{N}(\pi)$ as the set of neighboring policies of π. A policy is considered a neighbor if it differs from π by only one action for all observation sequences.

1. After seeing p samples (where $p < k$), MCESP+PAC selects some $\pi' \in \mathcal{N}(\pi)$, as π' appears higher value than π, but it is not
2. After seeing p samples (where $p < k$), MCESP+PAC cannot find a $\pi' \in \mathcal{N}(\pi)$ where π' appears higher value than π, but there is one
3. After seeing all k_m samples, MCESP+PAC selects some $\pi' \in \mathcal{N}(\pi)$, as π' appears higher value than π, but it is not
4. After seeing all k_m samples, MCESP+PAC cannot find a $\pi' \in \mathcal{N}(\pi)$ where π' appears higher value than π, but there is one

Recall ϵ and k_m for MCESP+PAC are as follows.

$$\epsilon^*(m,p,q) = \Lambda(\pi_i, \pi_i') \sqrt{\frac{1}{2p} \ln \frac{2(k_m - 1)N}{\delta_m}}$$

$$k_m = \left\lceil 2 \frac{(\Lambda(\pi_i, \pi_i'))^2}{\epsilon^2} \ln \frac{2N}{\delta_m} \right\rceil$$

Also, let $E[\pi]$ be the reward of the policy π, in the context of other agents and the environment, which the Q-value approximates.

$$\begin{aligned}
a_m^n =& Pr\Big[\exists \pi_i' \in \mathcal{N}(\pi_i^m) : (Q^{\pi_i'} - Q^{\pi_i^m}) \geq \epsilon(m, c_i^{\pi_i'}, c_i^{\pi_i^m}) \\
&\quad \text{and } E[\pi_i'] < E[\pi_i^m]\Big] \\
b_m^n =& Pr\Big[\exists \pi_i' \in \mathcal{N}(\pi_i^m) : (Q^{\pi_i'} - Q^{\pi_i^m}) < \epsilon - \epsilon(m, c_i^{\pi_i'}, c_i^{\pi_i^m}) \\
&\quad \text{and } E[\pi_i'] > E[\pi_i^m] + \epsilon\Big] \\
c_m =& Pr\Big[\exists \pi_i' \in \mathcal{N}(\pi_i^m) : (Q^{\pi_i'} - Q^{\pi_i^m}) \geq \frac{\epsilon}{2} \\
&\quad \text{and } E[\pi_i'] < E[\pi_i^m]\Big] \\
d_m =& Pr\Big[\exists \pi_i' \in \mathcal{N}(\pi_i^m) : (Q^{\pi_i'} - Q^{\pi_i^m}) < \frac{\epsilon}{2} \\
&\quad \text{and } E[\pi_i'] > E[\pi_i^m] + \epsilon\Big]
\end{aligned}$$

We represent each of the probabilities above as disjoint sets over neighbors and the agents' policies. For example, $\mathcal{N}_i^<(\pi_i^m) = \{\pi_i \in \mathcal{N}(\pi^m) | E[\pi_i'] < E[\pi_i^m]\}$.

$$\begin{aligned}
a_m^n &= Pr\Big[\bigvee_{\pi_i' \in \mathcal{N}_i^<(\pi_i^m)} (Q^{\pi_i'} - Q^{\pi_i^m}) \geq \epsilon(m, c_i^{\pi_i'}, c_i^{\pi_i^m})\Big] \\
b_m^n &= Pr\Big[\bigvee_{\pi_i' \in \mathcal{N}_i^>(\pi_i^m)} (Q^{\pi_i'} - Q^{\pi_i^m}) < \epsilon - \epsilon(m, c_i^{\pi_i'}, c_i^{\pi_i^m})\Big] \\
c_m &= Pr\Big[\bigvee_{\pi_i' \in \mathcal{N}_i^<(\pi_i^m)} (Q^{\pi_i'} - Q^{\pi_i^m}) \geq \frac{\epsilon}{2}\Big] \\
d_m &= Pr\Big[\bigvee_{\pi_i' \in \mathcal{N}_i^>(\pi_i^m)} (Q^{\pi_i'} - Q^{\pi_i^m}) < \frac{\epsilon}{2}\Big]
\end{aligned}$$

Considering every possible π' in π^m's neighborhood leads to the following summation.

$$\begin{aligned}
a_m^n \leq & \sum_{\pi_i' \in \mathcal{N}_i^<(\pi_i^m)} Pr[(Q^{\pi_i'} - Q^{\pi_i^m}) \geq \epsilon(m, c_i^{\pi_i'}, c_i^{\pi^m}, k_m)] \\
\leq & \sum_{\pi_i' \in \mathcal{N}_i^<(\pi_i^m)} Pr[(Q^{\pi_i'} - Q^{\pi_i^m}) \geq (E[\pi_i'] - E[\pi_i^m]) + \epsilon(m, c_i^{\pi_i'}, c_i^{\pi^m})] \\
\leq & \sum_{\pi_i' \in \mathcal{N}_i^<(\pi_i^m)} exp\left\{-2p\left(\frac{\epsilon(m, c_i^{\pi_i'}, c_i^{\pi_i^m})}{\Lambda(\pi_i', \pi_i^m, \pi_j)}\right)^2\right\} \qquad (5) \\
= & \frac{|\mathcal{N}_i^<(\pi_i^m)|\delta_m}{2(k_m - 1)|\mathcal{N}(\pi_i^m)|}
\end{aligned}$$

Equation 5 follows from Hoeffding's Inequality, where $(Q^{\pi_i'} - Q^{\pi_i^m})$ is the sample average of $E[\pi_i'] - E[\pi_i^m]$. We reduce 5 by utilizing ϵ when $p = q < k_m$. b_m^n follows,

$$\begin{aligned}
b_m^n \leq & \sum_{\pi_i' \in \mathcal{N}_i^>(\pi_i^m)} Pr\left[(Q^{\pi_i'} - Q^{\pi_i^m}) < \epsilon - \ \epsilon(m, c_i^{\pi_i'}, c_i^{\pi^m}, k_m)\right] \\
\leq & \frac{|\mathcal{N}_i^>(\pi_i^m)|\delta_m}{2(k_m - 1)|\mathcal{N}(\pi_i^m)|}
\end{aligned}$$

The other two error types, where $p = k_m$, follow similarly but substitute k_m for n.

$$c_m \leq \sum_{\pi_i' \in \mathcal{N}_i^<(\pi_i^m)} Pr\left[(Q^{\pi_i'} - Q^{\pi_i^m}) \geq \frac{\epsilon}{2}\right] \qquad (6)$$

$$\leq \sum_{\pi_i' \in \mathcal{N}_i^<(\pi_i^m)} exp\left\{-2k_m\left(\frac{\epsilon/2}{\Lambda(\pi_i', \pi_i^m, \pi_j)}\right)^2\right\} = \frac{|\mathcal{N}_i^<(\pi_i^m)|\delta_m}{2|\mathcal{N}(\pi_i^m)|} \qquad (7)$$

where Eq. 7 is obtained by substituting k_m with its derived value in the previous expression and reducing.

$$d_{m,\pi_j} \leq \sum_{\pi_i' \in \mathcal{N}_i^>(\pi_i^m)} Pr\left[(Q^{\pi_i'} - Q^{\pi_i^m}) < \frac{\epsilon}{2}\right] \leq \frac{|\mathcal{N}_i^>(\pi_i^m)|\delta_m}{2|\mathcal{N}(\pi_i^m)|}$$

The sum total of the error is as follows.

$$\begin{aligned}
& \sum_{n=1}^{k_m - 1} [a_m^n + b_m^n] + c_m + d_m \\
&= \sum_{n=1}^{k_m - 1} \left[\frac{|\mathcal{N}_i^<(\pi_i^m) + |\mathcal{N}_i^>(\pi_i^m)|}{|\mathcal{N}(\pi_i^m)|} \frac{\delta_m}{2(k_m - 1)}\right] \\
&\quad + \frac{|\mathcal{N}_i^<(\pi_i^m)| + |\mathcal{N}_i^>(\pi_i^m)|}{\mathcal{N}(\pi_i^m)|} \frac{\delta_m}{2} = \delta_m
\end{aligned}$$

Over all possible transformations, the total error is bounded by δ.

$$\sum_{m=1}^{\infty} \delta_m = \sum_{m=1}^{\infty} \frac{6\delta}{m^2 \pi^2} = \frac{6\delta}{\pi^2} \sum_{m=1}^{\infty} \frac{1}{m^2} = \frac{6\delta}{\pi^2} \frac{\pi^2}{6} = \delta$$

Proof of Proposition 2 Consider the following expanded definition of Prop. 2.

$$\begin{aligned}
& \sum_{t=0}^{T} [\max_s \{R(a_{\pi_i'}^t, a_j^t, s) - R(a_{\pi_i^k}^t, a_j^t, s)\} - \\
& \min_s \{R(a_{\pi_i'}^t, a_j^t, s) - R(a_{\pi_i^k}^t, a_j^t, s)\}] \leq \\
& \sum_{t=0}^{T} [\max_{s,a_j} \{R(a_{\pi_i'}^t, a_j, s) - R(a_{\pi_i^k}^t, a_j, s)\} - \\
& \min_{s,a_j} \{R(a_{\pi_i'}^t, a_j, s) - R(a_{\pi_i^k}^t, a_j, s)\}]
\end{aligned}$$

- Assume $\min_s = \min_{s,a_j}$ for all $t \in T$, resulting in the expression $\sum_{t=0}^{T} \max_s \{R(a_{\pi_i'}^t, a_j^t, s) - R(a_{\pi_i^k}^t, a_j^t, s)\} \leq \sum_{t=0}^{T} \max_{s,a_j} \{R(a_{\pi_i'}^t, a_j, s) - R(a_{\pi_i^k}^t, a_j, s)\}$. It must be the case that the LHS must be at most equivalent to the RHS, as, if the a_j selected in the LHS is the maximal value, the $\max_{s,a_j}$ will select it. If the a_j selected on the LHS is not the maximal value for a given a_i and s, the RHS must then be greater.
- Assume $\max_s = \max_{s,a_j}$ for all $t \in T$, resulting in the expression $\sum_{t=0}^{T} \min_s \{R(a_{\pi_i'}^t, a_j^t, s) - R(a_{\pi_i^k}^t, a_j^t, s)\} \geq \sum_{t=0}^{T} [\min_{s,a_j} \{R(a_{\pi_i'}^t, a_j, s) - R(a_{\pi_i^k}^t, a_j, s)\}]$. Analogously to point 1, if the a_j on the LHS is the minimal value, then the $\min_{s,a_j}$ on the RHS must select it. If it is not, the RHS must be less.
- Following point 1 and 2, since each component of the LHS must be bounded by the equivalent component on the RHS, the LHS must be a smaller range than the RHS. Therefore, $\Lambda^{\vec{a_j}}(\pi_i', \pi_i^m) \leq \Lambda(\pi_i', \pi_i^m)$

REFERENCES

[1] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*. MIT Press, 1998.

[2] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[3] Leemon Baird et al. Residual algorithms: Reinforcement learning with function approximation. In *International Conference on Machine Learning (ICML)*, pages 30–37, 1995.

[4] Steven D Whitehead. Reinforcement learning for the adaptive control of perception and action. Technical report, DTIC Document, 1992.

[5] Theodore J Perkins. Reinforcement learning for POMDPs based on action values and stochastic optimization. In *AAAI Conference on Artificial Intelligence*, pages 199–204, 2002.

[6] Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25, 1995.

[7] Daniel S Bernstein, Shlomo Zilberstein, and Neil Immerman. The complexity of decentralized control of markov decision processes. In *Sixteenth conference on Uncertainty in Artificial Intelligence (UAI)*, pages 32–37. Morgan Kaufmann Publishers Inc., 2000.

[8] Piotr J. Gmytrasiewicz and Prashant Doshi. A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research*, 24:49–79, 2005.

[9] Bikramjit Banerjee, Jeremy Lyle, Landon Kraemer, and Rajesh Yellamraju. Sample bounded distributed reinforcement learning for decentralized POMDPs. In *AAAI Conference on Artificial Intelligence*, 2012.

[10] Landon Kraemer and Bikramjit Banerjee. Reinforcement learning of informed initial policies for decentralized planning. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 9(4):18, 2014.

[11] Christopher Amato and Frans A Oliehoek. Scalable planning and learning for multiagent POMDPs. In *Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[12] Brenda Ng, Kofi Boakye, Carol Meyers, and Andrew Wang. Bayes-adaptive interactive POMDPs. In *AAAI Conference on Artificial Intelligence*, 2012.

[13] Trong Nghia Hoang and Kian Hsiang Low. A general framework for interacting bayes-optimally with self-interested agents using arbitrary parametric model and model prior. In *Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1394–1400, 2013.

[14] Russell Greiner. Palo: A probabilistic hill-climbing algorithm. *Artificial Intelligence*, 84(1):177–208, 1996.

[15] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations*. Cambridge University Press, 2009.

[16] Ekhlas Sonu and Prashant Doshi. Generalized and bounded policy iteration for finitely-nested interactive pomdps: Scaling up. In *11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1039–1048, 2012.

[17] Brenda Ng, Carol Meyers, Kofi Boakye, and John Nitao. Towards applying interactive POMDPs to real-world adversary modeling. In *Innovative Applications in Artificial Intelligence (IAAI)*, pages 1814–1820, 2010.