Stick-Breaking Policy Learning in DEC-POMDPs

Anonymous Author(s)

Affiliation

Address

email

Abstract

Decentralized POMDPs (DEC-POMDPs) are rich models for multiagent decision-making under uncertainty, but generating high-quality solutions is difficult. Recent work has shown that expectation maximization (EM) is an efficient algorithm for learning finite-state controllers (FSCs) in large decentralized domains. However, the solution quality of current methods is limited by their inability to define an appropriate controller size and the methods often converge to maxima that are far from optimal. In this paper we address this problem by using a variable-size FSC to represent the local policy of each agent. The variable-size FSC is constructed using the stick-breaking prior, leading to a new framework termed a decentralized stick-breaking policy presentation (DEC-SBPR). This approach allows us to study the reinforcement learning (RL) setting, assuming the DEC-POMDP model is not directly available, and yields a variational Bayesian algorithm to infer the variable-size FSC. We demonstrate the performance of DEC-SBPR on benchmark problems, showing that it can scale to large problems while outperforming other state-of-the-art learning methods.

1 Introduction

Decentralized partially observable Markov decision processes (DEC-POMDPs) \cite{16, 2} provide a general and expressive framework for multiagent decision-making problems arising in diverse applications, including robotic soccer \cite{15}, cooperative transportation \cite{21}, extra-planetary exploration \cite{5}, and traffic control \cite{23}. DEC-POMDPs can be viewed as a POMDP controlled by multiple distributed agents. Each agent acts based on local observations, and the joint actions of all agents control the state dynamics and expected reward. Because of the local observation, an individual agent generally does not have enough information to compute the global belief state, a sufficient statistic for decision making in POMDPs. This makes DEC-POMDP policy learning a more difficult problem to solve than for a POMDP \cite{4}.

Because of the lack of centralized belief states in DEC-POMDPs, infinite-horizon DEC-POMDP algorithms have based agent policies on finite-state controllers (FSCs) which map observation histories to actions. While early algorithms employed exhaustive backups \cite{3} or non-linear programming \cite{1} to obtain FSC policies, recent work indicates that expectation-maximization (EM) \cite{8}, a popular algorithm for maximum-likelihood (ML) estimation, can be used as a scalable method for generating controllers for large DEC-POMDPs \cite{11, 18}. In addition, Wu et al. \cite{23} recently showed that EM is also an efficient algorithm for model-free reinforcement learning (RL) in DEC-POMDPs, where each agent learns a FSC based on locally observable trajectories, without knowing the DEC-POMDP model.

An important and yet unanswered question for DEC-POMDP algorithms is how to define an appropriate number of nodes in each FSC. Previous work assumes a fixed FSC size for each agent, but the number of nodes affects both the quality of the policies and the convergence rate. When the number of nodes is too small, the FSC is unable to represent the optimal policy and therefore will quickly...
converge to a sub-optimal policy. By contrast, when the number is set too large, the FSC is overly complex, often yielding slow convergence to a sub-optimal policy that overfits the agent trajectories.

In this paper, we address the model-selection problem in DEC-POMDPs based on a nonparametric Bayesian approach. In particular, we represent the local policy of each agent by a variable-size FSC, which has an unspecified number of nodes. Such a controller is constructed using the stick-breaking (SB) prior \cite{10}. At each node of the controller, the probability distribution of successor nodes, conditional on both the observation and action, is a random draw from the SB prior. This gives rise to a distribution over an infinite number of possible nodes, indexed \( \{1, 2, \cdots, \infty\} \), and yet the probability mass concentrates on only a few nodes with small indices. Since the favored successors of every node all tend to have small indices, this encourages a compact set of nodes to be actively used by the controller. Those nodes that are actually used are determined by the posterior, combining the SB prior and the information from trajectory data. In light of the role played by the SB prior, we name the framework the decentralized stick-breaking policy representation (DEC-SBPR).

In addition to the use of variable-size FSCs, the proposed methods that are further distinguished from previous EM-based algorithms in that we manipulate the DEC-POMDP directly, without transforming it into a mixture of dynamic Bayes nets (DBNs). As a result, the EM algorithm in our case operates on the value function of a DEC-POMDP, instead of the likelihood function of a DBN mixture. Although the proposed DEC-SBPR may be used to find the policy for a given DEC-POMDP model, our focus in this paper is on RL in DEC-POMDPs, assuming the model is not available. Toward this end, we derive a variational Bayesian (VB) algorithm for learning the DEC-SBPR based on the agents’ trajectories (or episodes) of actions, observations, and rewards. The VB algorithm is linear in the number of agents and at most square in the problem size, and is therefore scalable to large application domains. To the best of our knowledge, this is the first application of nonparametric Bayesian methods to the difficult and little-studied problem of model-free RL in DEC-POMDPs.

2 Background

2.1 Decentralized POMDPs

A DEC-POMDP is represented as \( M = \langle \mathcal{N}, \mathcal{A}, \mathcal{S}, \mathcal{O}, \mathcal{T}, \Omega, \mathcal{R}, \gamma \rangle \), where \( \mathcal{N} = \{1, \cdots, N\} \) is a finite set of agents; \( \mathcal{A} = \otimes_n \mathcal{A}_n \) and \( \mathcal{O} = \otimes_n \mathcal{O}_n \) respectively are sets of joint actions and observations, with \( \mathcal{A}_n \) and \( \mathcal{O}_n \) available to agent \( n \). At each step, a joint action \( \vec{a} = (a_1, \cdots, a_N) \in \mathcal{A} \) is selected and a joint observation \( \vec{\sigma} = (o_1, \cdots, o_N) \) is received; \( \mathcal{S} \) is a set of finite world states; \( \mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S} \) is the state transition function with \( \mathcal{T}(s'|s, \vec{a}) \) denoting the probability of transitioning to \( s' \) after taking joint action \( \vec{a} \) in \( s \); \( \mathcal{O} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{O} \) is the observation function with \( \mathcal{O}(\vec{o}|s', \vec{a}) \) the probability of observing \( \vec{o} \) after taking joint action \( \vec{a} \) and arriving in state \( s' \); \( \mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R} \) is the reward function with \( r(s, \vec{a}) \) the immediate reward received after taking joint action \( \vec{a} \) in \( s \); \( \gamma \in (0, 1) \) is a discount factor. A global reward signal is generated for the whole team of agents after joint actions are taken. Due to the lack of access to other agents’ observations, each agent has a local policy \( \pi_n \), defined as a mapping from local observation histories to actions. A joint policy consists of the local policies of all agents. For an infinite-horizon DEC-POMDP with initial state \( s_0 \), the objective is to find a joint policy \( \Pi = \otimes_n \Pi_n \), such that the value function of \( \Pi \) stating from \( s_0 \), \( V^\Pi(s_0) = \mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, \vec{a}_t) | s_0, \Pi \right] \), is maximized.

In a DEC-POMDP, while each agent does not have enough information to compute the global belief state, it may represent a policy as some mapping from histories to actions. A (stochastic) FSC is a compact way to accomplish this. Formally, the stochastic FSC for agent \( n \) is defined as \( \Theta_n = (\mathcal{A}_n, \mathcal{O}_n, Z_n, W_n, \mu_n, \pi_n) \), where \( \mathcal{A}_n \) and \( \mathcal{O}_n \) are the same as defined in the DEC-POMDP; \( Z_n \) is a finite set of nodes, \( W_n \) is a set of Markov transition matrices with \( W_{n,a,o}^z \) denoting the probability of agent \( n \) transiting from \( z \) to \( z' \) when taking action \( a \) in \( o \); \( \mu_n \) is the initial node distribution with \( \mu_n^z \) the probability of agent \( n \) initially being in \( z \); \( \pi_n \) is a set of stochastic policies with \( \pi_n^a_z \) the probability of agent \( n \) taking action \( a \) in \( z \).

For simplicity, we use the following notational conventions. \( Z_n = \{1, 2, \cdots, |Z_n|\} \), where \( |Z_n| \) is the cardinality of \( Z_n \), and \( \mathcal{A}_n \) and \( \mathcal{O}_n \) follow similarly. \( \Theta = \{\Theta_1, \cdots, \Theta_N\} \) is the joint FSC of all agents. A consecutively-indexed variable is abbreviated as the variable with the index range shown in the subscript or superscript; when the index range is obvious from the context, a simple “\( : \)” is used instead. Thus, \( a_{n,0:T} = (a_{n,0}, a_{n,1}, \cdots, a_{n,T}) \), \( W_{n,a,o}^{z_1} = (W_{n,a,o}^{z_1,1}, W_{n,a,o}^{z_1,2}, \cdots, W_{n,a,o}^{z_1,|Z_n|}) \), etc.
Given $h_{n,t} = \{a_{n,t-1}, o_{n,t-1}, \ldots, a_{n,1}, o_{n,1}\}$, a local history of actions and observations up to $t$, agent $n$ chooses its action $a_{n,t}$ according to $p(a_{n,t} | h_{n,t}, \Theta_n) = p(a_{n,t} | h_{n,t}, \Theta_n) / \sum_{a_{n,t}} p(a_{n,t} | h_{n,t}, \Theta_n)$, where $p(a_{n,t} | h_{n,t}, \Theta_n)$ is a marginal distribution of $p(a_{n,t} | h_{n,t}, \Theta_n) = \mu_{n,z_{\theta_n}} \prod_{t=1}^{T_n} \prod_{z_{\theta_n}} W_{h_{n,t}, o_{n,t}, a_{n,t}} \prod_{z_{\theta_n}} \prod_{t=1}^{T_n} \prod_{z_{\theta_n}} \prod_{a_{n,t}} \prod_{o_{n,t}} \prod_{z_{\theta_n}}$, obtained by integrating out the variables $z_{n,0:t}$.

### 2.2 Policy Learning by Direct Value Maximization

Assuming fixed-size FSCs, we can extend the global empirical value function from the single-agent case \[12\]. This serves as the basis for a method to learn fixed-size FSCs.

**Definition 1. (Global empirical value function)** Let $D^{(K)} = \{(a_k^{(1)} r_k^{(1)}, a_k^{(2)} r_k^{(2)}, \ldots, a_k^{(K)} r_k^{(K)}) \}_{k=1, \ldots, K}$ be a set of episodes resulting from $N$ agents who choose actions according to $\Theta = \otimes_{n=1}^{N} \Pi_{n}$, a set of arbitrary stochastic policies with $\mu_{n}(a_{n}) > 0$, $\forall$ action $a_{n}, \forall$ history $h_{n}$. The global empirical value function is defined as $\hat{V}(D^{(K)}; \Theta) = \sum_{k=1}^{K} \sum_{t=0}^{T_k} \sum_{\tau=0}^{n_{\tau}} \prod_{z_{\theta_n}} \sum_{a_{n,t}} \mu_{n}(a_{n,t}) h_{n,t}^{k}, \Theta_{n}$.

The global empirical value function $\hat{V}(D^{(K)}; \Theta)$ is a function of the joint FSC, given only the trajectories collected from the interactions between the agents and the DEC-POMDP model, with the DEC-POMDP model parameters $T$, $\Omega$ and $R$ assumed unknown. The empirical value function offers an objective for learning the decentralized policies, and the objective can be directly maximized by the nested EM algorithm in \[12\], with the following modifications: (i) during the policy evaluation in the outer E-step, the rewards are recomputed based on the information from all agents; (ii) given recomputed rewards (the sum of which amounts to the improved value), the outer M-step consists of $N$ independent applications of the inner EM, each performed for a local agent. Step (ii) can be implemented by parallel or distributed computation. Step (i) collects $\{p(a_{n,t}^{k} | h_{n,t}^{k}, \Theta_{n}) : \forall t, k\}$ from each agent $n$ to update the recomputed rewards $\{\nu_{k}^{n} : \forall t, k\}$; this requires minimal communication of only $T_1 + \cdots + T_K$ numbers per agent, which can be implemented efficiently.

### 3 Bayesian Learning of Fixed-size FSCs

An alternative approach is Bayesian learning, which provides the convenience of encoding expert knowledge and imposing priors for model selection. By measuring the likelihood of $\Theta$ by its empirical value, we obtain $\hat{V}(D^{(K)}; \Theta)$ as a likelihood function, which is combined with the prior $p(\Theta)$ in Bayes’ rule to yield the posterior

$$p(\Theta | D^{(K)}) \propto \hat{V}(D^{(K)}; \Theta) p(\Theta) \hat{V}(D^{(K)})^{-1}$$

where $\hat{V}(D^{(K)}) = \int \hat{V}(D^{(K)}; \Theta) p(\Theta) d\Theta$ is the marginal empirical value of the joint FSC.

To eliminate the sum in $\hat{V}(D^{(K)}; \Theta)$, we consider a variational Bayesian (VB) approximation. Denote by $q(\Theta)$ the variational approximation to $p(\Theta | D^{(K)})$, and by $q_k^{k}(z_{n,t}^{k})$ the approximation to $p(z_{n,t}^{k} | a_{n,t}^{k}, \Theta)$. The variational lower bound to $\ln \hat{V}(D^{(K)})$ (free energy) is given by

$$\text{LB}(q_k^{k}(z_{n,t}^{k})) = \ln \hat{V}(D^{(K)}) - \frac{1}{K} \sum_{k=1}^{K} \sum_{t=0}^{T_k} \sum_{\tau=0}^{n_{\tau}} \text{KL}(q_k^{k}(z_{n,t}^{k}) || p_k^{k}(z_{n,t}^{k} | \Theta)))$$

where KL$(q || p)$ denotes the Kullback-Leibler (KL) divergence between probability measures $q$ and $p$.

The goal is to minimize the KL divergence. Since the first term on the right is a constant, this is equivalent to maximizing the lower bound, leading to the following optimization problem,

$$\max_{q_k^{k}(z_{n,t}^{k})} \ln \text{LB}(q_k^{k}(z_{n,t}^{k})) \quad \text{s.t.} \quad q_k^{k}(z_{n,t}^{k}, \Theta) = \prod_{n=1}^{N} q_k^{k}(z_{n,t}^{k})$$

subject to

$$\sum_{k=1}^{K} \sum_{t=0}^{T_k} \sum_{\tau=0}^{n_{\tau}} q_k^{k}(z_{n,t}^{k}) = 1, \quad \forall n, t, \tau$$

where the constraint in the first line arises both from the mean-field approximation and from the decentralized policy representation, and the last line summarizes the normalization constraints.

### 3.1 Policy Posterior under Dirichlet Priors

The problem in \[3\] has an analytic solution when $p(\Theta)$ is a product of Dirichlet distributions,

$$p(\Theta) = \prod_{n=1}^{N} \text{Dir}((\mu_{n}, \nu_{n}) \prod_{t=0}^{T_k} \text{Dir}(\pi_{t} | \rho_{t}), \prod_{a=1}^{|A_n|} \prod_{o=1}^{|O_n|} \prod_{z_{\theta_n}} \prod_{|Z_{\theta_n}|} \prod_{\theta_{n,a,o}} \prod_{|\theta_{n,a,o}|} \text{Dir}(W_{n,a,o} | \omega_{n,a,o} | Z_{n,a,o}^{1:2})$$

\[4\]
with hyper-parameters $v_n = \nu_n^{1:|Z_n|}$, $\rho_n = \{\rho_n,z|z=1:|Z_n|\}$, and $\omega_n = \{\omega_n,o,a|o=1:|O_n|, a=1:|A_n|\}$.

The solution to (3) under the Dirichlet priors is given in Theorem 2.

**Theorem 2.** Let $p(\Theta)$ initially be the form of (4) with hyper-parameters ($\bar{\nu}, \bar{\rho}, \bar{\omega}$), then iterative application of the following updates leads to monotonic increase of (4), until convergence to a maxima. The updates of $\bar{\nu}$ and $\bar{\rho}_n,i,k(\cdot)$ are given by

$$\bar{\nu}_t^k = \frac{\gamma_t r_t^k \prod_{n=1}^{N} P(a_n^k, o_n^k|\Theta_n)}{\prod_{n=1}^{N} \prod_{t=0}^{\tau_t} \tilde{P}(\pi_{n,t}^{k}|h_{n,t}^k, \bar{\Theta}_n)} \forall t, k,$$

$$q_t^k(\cdot) = p(\cdot|\bar{\nu}_t^k, \bar{\rho}_n,i,k(\cdot), \tilde{P}(\pi_{n,t}^{k}|h_{n,t}^k, \bar{\Theta}_n)) \forall n, t, k,$$

where $\bar{\Theta} = \{\bar{\pi}, \bar{\rho}, \bar{\omega}, \bar{W}\}$ is a set of under-normalized probability (mass) functions with $\bar{\pi}_n^i = e^{v(\bar{\pi}_n^i)} - e^{v(\sum_{m=1}^{A_n^i} \bar{\rho}_n^m)}$, $\bar{\rho}_n,i,k(\cdot) = e^{\psi(\bar{\rho}_n,i,k(\cdot))} - e^{\psi(\sum_{z=1}^{Z_n} \bar{\omega}_n,z(\cdot))}$, and $\bar{W}(\pi_{n,t}^{k}|h_{n,t}^k, \bar{\Theta}_n)$ is the beta function. The hyper-parameters of the posterior distribution are updated as

$$\bar{\omega}_{n,i,a} = \omega_{n,i,a} + 1, \bar{\rho}_{n,i} = \rho_{n,i} + 1, \bar{\pi}_n^i = \frac{\sum_{z=1}^{Z_n} \bar{\omega}_n,z(\cdot)}{\sum_{z=1}^{Z_n} \bar{\omega}_n,z(\cdot)},$$

$$\alpha_{n,k}^{i,j}(i, j) = p(z_{n,\tau}^{k} = i | a_{n,\tau}^{k}, o_{n,\tau}^{k}, \Theta_n), \forall n, k, t, \tau.$$

The update equations in Theorem 2 constitutes the VB algorithm for learning a fixed-size joint FSCs under Dirichlet priors. In particular, (5) is a policy-evaluation step where the rewards are recomputed to reflect the improved marginal value of the new policy posterior updated in the previous iteration, and (7) is a policy-improvement step where the recomputed rewards are used to further improve the policy posterior. Both steps require (6), which are computed based on

$$\alpha_{n,k}^{i,j}(i, j) = p(z_{n,\tau}^{k} = i | a_{n,\tau}^{k}, o_{n,\tau}^{k}, \Theta_n), \forall n, k, t, \tau.$$
The stick-breaking process (SBP) is a general class of priors for discrete measures. It has been used as the prior for state transition probabilities in HMMs [17] and also for the graphical structures in topic modeling [7]. The basic properties of SBP are summarized in the supplements. It is noted that SBPs subsume Dirichlet Processes (DPs) [9] as a special case, when \( \sigma_{n,a,o}^{i,j} = 1, \lambda_n^i = 1, \forall i, j, n, a, o \) (in DEC-SBPR).

The purpose of using SBPs is to encourage a small number of FSC nodes. Compared to a DP, the SBP can represent richer patterns of sparse transition between the nodes of an FSC, because it allows arbitrary correlation between the stick-breaking weights (the weights are always negatively correlated in a DP). To see this mechanism, we calculate the expected length of a stick, \( E(\sum_{n,a,o} \lambda_n^i) = 1/(1 + \sum_{n,a,o} \sigma_{n,a,o}^{i,j}), \forall i, j \), which is seen to be inversely proportional to the random variable \( \Sigma_{n,a,o}^{i,j} \). For a fixed value of \( \sigma_{n,a,o}^{i,j} \), the node transition probability \( W_{n,a,o}^{i,j} \) is expected to decrease as \( \Sigma_{n,a,o}^{i,j} \) increases. Hence \( \Sigma_{n,a,o}^{i,j} \) reflects the prior about the frequency of transitions from node \( i \) to node \( j \). From the construction of SBP and the posterior updating equations [11], we can see \( \Sigma_{n,a,o}^{i,j} < \Sigma_{n,a,o}^{i,j+1} \), \( \forall m > 0 \), thus \( E(W_{n,a,o}^{i,j}) > E(W_{n,a,o}^{i,j+1}) \), i.e. the expected stick length of the state with a smaller index is greater than the stick length of the state with a larger index, therefore the state transition probabilities are biased towards the states with small indices. Similarly, \( \Lambda_n \) encodes the prior preference for initial node occupancy for agent \( n \). We impose gamma priors over \( \Sigma \) and \( \Lambda \) to allow Bayesian inference of hyper-parameters.

To accommodate an unbounded number of nodes, we apply the retrospective representation of SBPs [19] to DEC-SBPR. For agent \( n \), the stick breaking-prior is set with a truncation level \( |Z_n| \), taking into account the current occupancy as well as additional nodes reserved for future new occupancies.

To determine the occupancy of the nodes in a FSC, we compute \( |Z_n| \) directly by checking if there is a reward assigned to a node. For example, for action \( a \) and node \( i \), \( \hat{\rho}_{n,i}^a - \rho_{n,i}^a \) is the reward being assigned. If this quantity is greater then zero, then node \( i \) is visited. By summing over all actions, we can determine whether there is a value in visiting node \( i \). Hence we can directly compute \( |Z_n| \) based on the following formula

\[
|Z_n| = \sum_{t=1}^{\infty} 1(\sum_{a=1}^{|A|} (\hat{\rho}_{n,i}^a - \rho_{n,i}^a) > 0).
\]

It is shown in [10] that the random weights constructed by the stick-breaking prior are equivalently governed by a generalized Dirichlet distribution (GDD) and is therefore conjugate to the multinomial distribution; hence we can derive an efficient variational Bayesian algorithm for learning the decentralized policies.

The complete algorithm is described in Algorithm 1. With \( \xi_{t,r}^{n,k}(i,j) \) and \( \phi_{t,r}^{n,k}(i) \) defined the same as in [7], the update equations, derived in the Supplementary Material, are partly listed below:

\[
\begin{align*}
\hat{\rho}_{n,i}^a &= \hat{\rho}_{n,i}^a + \sum_{t=0}^{T_k} \sum_{r=1}^{T_k} \hat{\rho}_{n,i}^{\phi_{t,r}^{n,k}(i)}(a_{t-1}^r, a) \lambda_n^i + \eta_n^i, \quad \hat{\lambda}_n^i = \lambda_n^i + \sum_{t=1}^{|Z_n|} \eta_n^i \\
\hat{\sigma}_{n,a,o}^{i,j} &= \sigma_{n,a,o}^{i,j} + \sum_{t=1}^{|Z_n|} \xi_{t,r}^{n,k}(i,j) \sum_{l=1}^{|Z_n|} \xi_{t,r}^{n,k}(i,j) (a_{t-1}^r, a) \lambda_n^i + \eta_n^i \\
\hat{\xi}_{n,a,o}^{i,j} &= \sum_{t=1}^{|Z_n|} \xi_{t,r}^{n,k}(i,j) \sum_{l=1}^{|Z_n|} \xi_{t,r}^{n,k}(i,j) (a_{t-1}^r, a) \lambda_n^i + \eta_n^i
\end{align*}
\]

where \( \eta_n^i = \frac{1}{K} \sum_{k=1}^K \sum_{t=0}^{T_k} \hat{\rho}_{n,i}^{\phi_{t,r}^{n,k}(i)}(a_{t-1}^r, a) \), \( \xi_{n,a,o}^{i,j} = \frac{1}{K} \sum_{k=1}^K \sum_{t=0}^{T_k} \hat{\rho}_{n,i}^{\phi_{t,r}^{n,k}(i)}(a_{t-1}^r, a) \).

Upon the convergence of Algorithm 1 point estimates of the decentralized policies may be obtained by calculating the expectations: \( E(\hat{\rho}_{n,i}^a), E(\hat{\sigma}_{n,a,o}^{i,j}) \), and \( E(W_{n,a,o}^{i,j}) \) (see the supplements for details).

### 4.1 Computational complexity

The time complexity of Algorithm 1 in each iteration is summarized in Table 1 assuming the length of an episode is on the order of magnitude of \( T \), and the number of nodes per controller is on the order of magnitude of \( |Z| \). In Table 1 the worst case refers to the case where there is a nonzero
reward at every time step in an episode (dense rewards), while the best case is that nonzero reward is received only at the terminal step in each episode. Hence in general the algorithm scales linearly with the number of episodes and the number of agents. The time dependency on $T$ is between linear and quadratic.

4.2 Tradeoff between Exploration and Exploitation

Algorithm 1 assumes off-policy batch learning where trajectories are collected using a separate behavior policy. Off-policy learning is efficient if the behavior policy is close to optimal, as in the case when expert information is available to guide the agents. With a random behavior policy, it may take a long time for the policy to converge to optimality; in this case, the agents may exploit the policies learned so far to speed up the learning process.

An important issue concerns keeping a proper balance between exploration and exploitation, such that the action choices are sufficiently explored to prevent premature convergence to a suboptimal policy, and yet a proper level of exploitation is maintained to prevent repeated exploration. To address this issue, we define an auxiliary FSC, $\Psi_n = \{\mathcal{Y}, \mathcal{O}_n, \mathcal{Z}_n, W_n, h_n, \varphi_n\}$, to represent the policy of each agent in balancing exploration and exploitation. To avoid confusion, we refer to $\Theta_n$ as a primary FSC. The only two components distinguishing $\Psi_n$ from $\Theta_n$ are $\mathcal{Y}$ and $\varphi_n$, where $\mathcal{Y} = \{0, 1\}$ encodes exploration ($y = 1$) or exploitation ($y = 0$), and $\varphi_n = \{\varphi_y^{n,z}\}$ with $\varphi_y^{n,z}$ denoting the probability of agent $n$ choosing $y$ in $z$. One can express $p(y_n | h_n, \Theta_n)$ in the same way as one expresses $p(a_n | h_n, \varphi_n)$, the latter described in the paragraph above Section 2.2.

The behavior policy $\Pi_n$ of agent $n$ is constructed as

$$p^{\Pi_n}(a | h, \Theta_n, \Psi_n) = \sum_{y=0,1} p(a | y, h)p(y | h, \Psi_n),$$

where $p(a | y = 0, h) \equiv p(a | h, \Theta_n)$ is the primary FSC policy, and $p(a | y = 1, h)$ is a policy agent $n$ uses to explore action choices. For example, one may let $p(a | y = 1, h) \equiv 1/|\mathcal{A}_n|$, leading to uniformly random exploration. The $\epsilon$-greedy policy (12) is a special case of (14), corresponding to $p(a | y, h) \equiv \epsilon, p(a = 0 | y, h) \equiv 1 - \epsilon$.

The behavior policy in (12) has achieved significant success in the single-agent case $[6, 13]$. Here we extend it to the multi-agent case and show that, under a special construction of $\varphi_n$, the optimality of the primary joint FSC $\Theta_n$ is directly related to the exploration rate.

We define $\varphi_0^{n,z}$ as a random draw from a beta distribution, $\text{Beta}(u_0^{n,z}, u_1)$, and $\varphi_1^{n,z} = 1 - \varphi_0^{n,z}$, where $u_1$ is kept as a sufficiently large constant, and $u_0^{n,z}$ is updated as

$$u_0^{n,z} = \sum_{k=1}^{K} \sum_{t=0}^{T_k} \sum_{\tau=0}^{\tau_{k}} \varphi_0^{n,k}(z),$$

(13)

Recall that $\varphi_1^{n,z}$ is the probability of agent $n$ choosing exploration in $z$. Because $u_1$ is a large constant, the agent will always favor exploration in $z$ until $u_0^{n,z}$ has increased to a point where its value exceeds $u_1$. Rewriting (13) as

$$u_0^{n,z} = \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{k}} \left( \sum_{t=0}^{T_k} \varphi_0^{n,k}(z) \right),$$

(14)

it becomes clear that $u_0^{n,z}$ is the total amount of expected future reward received by agent $n$ in $z$ over all time steps in all episodes. Therefore, $u_1$ defines, up to a multiplicative constant, the total future reward required in $z$ for an agent to stop exploration in $z$. On the other hand, because $p(y | h, \Psi_n, \Theta_n) = \sum_z p(z | h, \Theta_n) p(y | z, \Psi_n, \Theta_n) = \sum_z \varphi_0^{n,z} p(z | h, \Theta_n)$, we deduce that $p(y = 0 | h, \Psi_n, \Theta_n) \gg p(y = 1 | h, \Psi_n, \Theta_n)$ implies $\varphi_0^{n,z} \gg \varphi_1^{n,z}$ and hence $u_0^{n,z} \gg u_1$, for any $z \in \{z : p(z | h, \Theta_n) \gg 0\}$. However, $u_0^{n,z}$ is the total amount of expected future reward received in $z$, as indicated by (14). As a result, when $u_1$ in (12) is sufficiently large, one can conclude

$$\mathcal{H}_0^{\text{known}} = \{ h : p(y = 0 | h, \Psi_n, \Theta_n) \gg p(y = 1 | h, \Psi_n, \Theta_n) \}$$

(15)

represents the set of local histories at which, agent $n$ has acquired large rewards or a large number of small rewards, and thus the primary FSC $\Theta_n$ has become optimal or close so. Let $u_1^{\text{min}}$ denote the minimum $u_1$ such that this is true.
4.3 Theoretical Analysis of Exploration and Optimality

Let $\mathcal{M}$ denote the true model of the DEC-POMDP. Then

$$V(\mathcal{M}; \Theta) = \sum_{t=0}^{\infty} \sum_{\bar{a}, \bar{o}, r} \gamma^t r_t p(\bar{a}_{0:t}, \bar{o}_{1:t}, r_t | \Theta, \mathcal{M}),$$

(16)

is the true value function of $\Theta$, where $r_t = 0, \forall t > T$, for an episode of length $T$. Denote by $R_{\max}$ the maximum immediate reward. The relation between exploration rate and policy optimality in Theorem 4 is proven in the Appendix.

**Theorem 4.** Let $\Theta^*$ be the optimal joint FSC for the underlying DEC-POMDP. Let $\Theta$ be the joint FSC learned from $D(K)$, and $\bar{q}$ be constructed as in Section 4.2, $u_1 \geq u_1^{\text{min}}$, and $\{u_0^{n, z}\}$ be updated as in (13). For any $\epsilon \geq 0$, if $V(\mathcal{M}; \Theta) < V(\mathcal{M}; \Theta^*) - \epsilon$, then

$$P_e = 1 - p(\bar{y}_{0:\infty} = 0 | \sigma, \Theta) > (1 - \gamma) \epsilon / R_{\max},$$

(17)

where $P_e$ denotes the probability of at least one agent choosing exploration, and $\bar{y}_{0:1} = 0$ is a shorthand for “$y_n, \tau = 0, \forall \tau \in [0, t], \forall n \in \mathbb{N}$”.

Theorem 4 shows that, when the value of the joint FSC is $\epsilon$ away from the optimal value, then, with probability of at least $(1 - \gamma) \epsilon / R_{\max}$, at least one agent will perform exploration. Conversely, when all agents perform exploitation with probability of at least $1 - (1 - \gamma) \epsilon / R_{\max} = (R_{\max} - \epsilon + \gamma \epsilon) / R_{\max}$, the value of the joint FSC is guaranteed to be $\epsilon$ close to the optimal value.

5 Experiments

We evaluate the performance of the proposed algorithms on five benchmark problems$^1$ and a large-scale problem (traffic control) $^2$. For all results reported here, we have followed the same experimental procedure as used in $^2$. For DEC-SBPR, the hyper-parameters in (9) are set to $c = e = 0.1$ and $d = f = 10^{-6}$ to promote sparse usage of FSC nodes.

![Figure 2: A demonstration of convergence and optimality of Algorithm 1 on the Mars Rover problem.](https://example.com/f2.png)

**Demonstration of convergence and optimality** The convergence of algorithm 1 is demonstrated on the Mars Rover problem, based on using $K = 500$ episodes to learn the FSCs and evaluating the policy by the discounted accumulative reward averaged over 100 test episodes of 1000 steps. The results are reported in Figure 2 which shows the exact value, its lower bound, and the numbers of FSC nodes learned for the two agents, as a function of iterations of the VB algorithm. It is seen that, as the iteration proceeds, the lower bound monotonically increases, while the value itself exhibits an overall improvement and converges to the optimal value. The FSCs initially use a large number of nodes and converges to small numbers around 10.

**Comparison with other methods** We compare the performance of the DEC-SBPR to those of several state-of-art methods, including: Monte Carlo EM (MCEM) $^2$, PBV1-BB-DecPOMDP $^4$ and Periodic EM (PeriEM) $^1$. Similar to DEC-SBPR, MCEM is also a model-free RL approach. We apply the exploration-exploitation strategy described in Section 4.2 and follow the same experimental procedure as in $^2$ to report the results. The testing rewards after the algorithm converges are summarized in Table 1. We can see DEC-SBPR achieves comparable or better policy values than MCEM. These results can be explained by the fact that EM algorithm is sensible to initialization and prone to local optima. Moreover, by fixing the size of controllers, the optimal policy from EM algorithms might be over/under represented. While by using Bayesian nonparametric priors, DEC-SBPR learns the policy with variable-size controllers, hence allows more flexibility for representing the optimal policy (though still prone to local optima).

$^1$After an episode terminates, the agent stays in an absorbing state with zero reward $^2$.

$^2$Downloaded from [http://rbr.cs.umass.edu/camato/decpomdp/download.html](http://rbr.cs.umass.edu/camato/decpomdp/download.html)
Table 1: Performances of DEC-SBPR on benchmark problems, in comparison to other state-of-art algorithms. Being reported here are the policy values (higher value indicates better performance) and CPU times of all competing algorithms, and the averaged controller size \(|Z|\) inferred by DEC-SBPR.

| Problems ((|S|, |A|, |O|)) | Model-Free | Model-Based |
|--------------------------|------------|-------------|
|                          | DEC-SBPR   | MCEM        | PBVI-BB-DECPOMDP | PEREM       |
|                          | VALUE, TIME, \(|Z|\) | VALUE, TIME | VALUE, TIME | VALUE, TIME |
| DEC-Tiger (2, 3, 3)      | −19.14, 96s, 6 | −32.31, 20s | 13.45, < 5h | 9.42, 6540s |
| Broadcast (4, 2, 5)      | 9.20, 7s, 2  | 9.15, 24s   | 9.2710, < 5h | ---          |
| Recycling Robots (3, 3, 2) | 31.26, 147s, 3 | 30.78, 19s | 31.9291, < 5h | 31.80, 272s |
| Box Pushing (100, 4, 5) | 77.63, 290s, 14 | 59.95, 32s | 224.1387, > 5h | 106.68, 7164s |
| Mars Rovers (256, 6, 8) | 20.62, 1286s, 10 | 8.16, 166s | ---          | 18.13, 6088s |

Finally, we compare to PBVI-BB-DECPOMDP [14], a state-of-art model-based method for generating controllers of the highest values. As such, the performance of PBVI-BB-DECPOMDP serves as the upper-bound for those of model-free methods. However PBVI-BB-DECPOMDP is not scalable to large problem sizes, and therefore it is unable to produce results for Mars Rovers [14]. By using a model-free reinforcement learning approach, DEC-SBPR solves this problem without being restricted by the large state space size.

**Scaling up to larger domains** To demonstrate the scalability to both large problem sizes and large numbers of agents, we test our algorithm on a traffic problem [23], which has $10^{20}$ states. In this domain, there are 100 agents controlling the traffic flow at $10 \times 10$ intersections with one agent located at each intersection. Each agent has 2 actions (aligning horizontally and aligning vertically), and 100 observations which indicate the number of traffic units waiting in the vertical and horizontal queues. The traffic in one direction can pass through if all the agents along that direction are aligned. The agents receive one unit of reward if one traffic unit passes through, and zero reward if the path is blocked. The goal for this problem is to coordinate the agents to maximize the total reward. Except for MCEM, no previous DEC-POMDPs algorithms are able to solve problems of such a large size. Since [23] uses, with a 10% chance, a hand-coded policy (comparing the traffic flow between two directions) as a heuristic for generating training trajectories, we also use such a heuristic for a fair comparison. Moreover, to examine the effectiveness of the exploration-exploitation strategy described in Section 4.2, we also consider the case without using the heuristic for generating training episodes; in this case, the initial behavior policy for each agent is to take random actions. From Figure 3, we can see that, with the help of the heuristic, DEC-SBPR can achieve the best performance. While without using the heuristic, by just using our exploration-exploitation strategy, in a few iterations, DEC-SBPR is able to produce a higher quality policy than MCEM. In addition, the inferred number of FSC nodes (averaged over all agents) is smaller than the number preselected by MCEM.

6 Conclusions

We have proposed a scalable nonparametric Bayesian policy representation and an associated learning framework (DEC-SBPR) for generating decentralized policies in DEC-POMDPs, and provided an exploration-exploitation method that extends the popular $\epsilon$-greedy method. Experimental results show DEC-SBPR achieves comparable and better results than the state-of-art model-free method, and has the additional benefit of inferring the number of nodes that is necessary for representing the optimal policy based on given experiences.
References


