Banded Controllers for Scalable POMDP Decision-Making

Kenneth Czuprynski and Kyle Hollins Wray

***Abstract*— This paper introduces a novel and computationally efficient policy representation, termed a banded controller, for Partially Observable Markov Decision Processes (POMDPs). The structure of a banded controller is obtained by restricting the number of successor nodes for each node in a finite state controller (FSC) policy representation; this is formally defined as the restriction of the controller's node transition matrices to the space of banded matrices. A gradient ascent based algorithm which leverages banded matrices is presented and we show that the policy structure results in a computational structure that can be exploited when performing policy evaluation. We then show that policy evaluation is asymptotically superior to a general FSC and that the degrees of freedom can be reduced while maintaining a large amount of expressivity in the policy. Specifically, we show that banded controller policy representations are equivalent to any FSC policy which is permutation similar to a banded controller. Meaning that banded controllers are computationally efficient policy representations for a class of FSC policies. Lastly, experiments are conducted which show that banded controllers outperform state-of-the-art FSC algorithms on many of the standard benchmark problems.**

I. Introduction

Partially Observable Markov Decision Processes (POMDPs) are models for single agent decision making problems [1]. They provide a general framework for planning under uncertainty and have been applied to numerous applications throughout robotics [2]; some examples include self-driving cars [3], target tracking [4], and aircraft collision avoidance systems [5]. Even outside of robotics, POMDPs have been used successfully in modeling cognitive radio [6] and other controlled sensing applications [7]. The ability to incorporate uncertainty make POMDPs a powerful modeling tool. The primary drawback of POMDP models is that they are computationally expensive to solve. Indeed they are known to be PSPACE complete [8] and the design of scalable algorithms is challenging. In this paper, we introduce a novel policy representation which has provably superior asymptotics when compared to general finite state controller (FSC) policy representations and present its use within a gradient ascent based algorithm.

Algorithms for POMDPs are often classified into offline vs. online algorithms with belief point based methods [9] and finite state controllers [10] being typical offline approaches. The belief based framework relies on the reformulation of the POMDP as a continuous state MDP which is parameterized by the space of probability distributions over the state space. Belief based methods work by constructing quantizations of belief space and generating approximations of the value function ([11], [12], [13]). The value functions themselves are parameterized by belief and the resulting polices are implicit in their representation. One drawback in the approach is that the number of parameters for the value function representation may grow indefinitely.

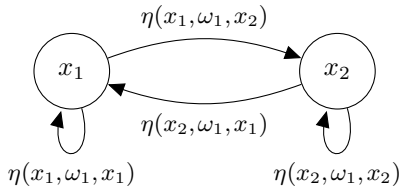

Fig. 1. Example of a FSC with two nodes $X=\{x_1,x_2\}$, one observation $\Omega=\{\omega_1\}$, and a finite set of actions A. The stochastic node transitions are given by $\eta: X\times\Omega\times X\to[0,1]$ and associated with each node $x_i\in X$, we have an action selection distribution $\psi: X\times A\to[0,1]$. We remark that the probability of observing ω_1 is dependent on the action selected by ψ.

FSCs provide memory efficient policy representations which can be represented as directed graphs [10]. Transitions between the nodes of the graph are observation dependent and stochastic where each node has an associated action selection distribution (cf. Figure 1). Controller size can be efficiently increased [14] and hierarchical approaches can be used to improve performance [15], [16]. A number of approaches have been pursued over the years, these include: nonlinear programming formulations (NLPs) [17], gradient based methods [18], and policy iteration methods [14], [19].

The structure of the controller can also be tailored to incorporate domain knowledge such as periodicity [20] which can also induce computational structure into the optimization framework [21]. This work expands on the idea of structured policy representations for fixed size controllers. It introduces a novel policy representation which aims to be general enough to perform well over a variety of domains while still containing exploitable computational structure, ultimately resulting in a more scalable controller.

When considering scalable methods, it is important to mention that online solvers employing search tree techniques have achieved impressive results [22], [23]. One drawback, however, is that execution is computationally intensive. This limits their use on energy constrained systems (e.g. smartphones) [19]. FSCs, on the other hand, are ideal for this application area, as policy execution reduces to traversal of the directed graph. This makes the development of scalable FSC methods an important area of research.

The contributions of this work are: (1) a formal definition of a banded controller (2) a projected gradient ascent based algorithm that exploits the policy structure (3) theoretical analysis of the equivalence of banded controllers to a general class of FSCs (4) analysis of the aymptotics of policy evaluation and (5) evaluation against state-of-the-art FSC algorithms over standard benchmarks, and demonstration on an real robot navigating a household environment.

II. Background

POMDPs provide a framework for determining optimal policies for single agent decision making problems in the presence of uncertainty. Formally, we express a POMDP as the tuple $\langle S, A, \Omega, T, O, R\rangle$, where S, A, and Ω are finite sets that denote the states, actions, and observations, respectively. The function $T: S \times A \times S \to [0,1]$ defines the state transition model for the POMDP where

$$T(s,a,s') = P(s'|s,a)$$

reflects the probability of transitioning to a new state s' after taking action a while in state s. The observation model O: $A \times S \times \Omega \to [0,1]$ is given by

$$O(a,s',\omega) = P(\omega|a,s')$$

and denotes the probability of observing ω after taking action a and transitioning to state s'. Lastly, the function $R: S \times A \to \mathbb{R}$ represents the reward model of the POMDP.

In the POMDP context, the states are not directly observable. Instead, one must rely on the action and observation histories to infer information about the state. Specifically, the action/observation histories are used to define a probability distribution over states of the POMDP $b \in \triangle^{n-1}$, where $\triangle^{n-1}$ denotes the $n-1$ simplex with $n = |S|$. This distribution is referred to as the belief and acts as a sufficient statistic for the history. At each time-step, the belief is updated to encode each new action/observation and the probability that the system is in state s is then given by $b(s)$.

A. Policy Representations

There are two primary approaches for policy representation: belief point based methods and finite state controllers. In the belief based approach, the policy representation is defined by directly mapping belief to action. These policies are defined in terms of the dynamic programming updates

$$V^*(b) = \max_{a \in A} \Big[\sum_{s \in S} b(s) R(s,a) + \gamma \sum_{\omega \in \Omega} P(\omega|b,a) V^*(b') \Big] \tag{1}$$

where b' denotes the successor belief obtained after taking action a and observing ω, and $\gamma \in [0,1)$ the discount factor. This denotes the expected discounted reward in terms of belief. The argument maximizing Equation 1 is then used to define the mapping between belief and action, $\pi(b) = a$.

The FSC representation of a policy π is defined using a controller with a finite set of nodes X. Each node in the controller has an associated action selection distribution $\psi: X \times A \to [0,1]$. The observation dependent transitions between the nodes are defined by $\eta: X \times \Omega \times X \to [0,1]$. The expected value of a policy $\pi = (\psi, \eta)$ is then

$$V^\pi(x,s) = \sum_{a \in A} \psi(x,a) \Big[R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \sum_{\omega \in \Omega} O(a,s',\omega) \sum_{x' \in X} \eta(x,\omega,x') V^\pi(x',s') \Big]$$

where the optimal policy maximizes over ψ and η for an initial controller node $x^0 \in X$ [10].

B. Gradient Ascent

For initial belief b^0 and controller node x^0 let $\beta^0 \in \mathbb{R}^{|X \times S|}$ where $\beta^0(\langle x,s\rangle) = b^0(s)[x = x^0]$ using Iverson bracket $[\cdot]$. Then, using matrix-vector notation, the optimal policy maximizing the expected discounted reward is given by

$$\pi^* = \arg\max_\pi \beta_0 \cdot \mathbf{v}^\pi \tag{2}$$

and $\mathbf{v}^\pi$ is the value function satisfying the Bellman equation

$$\mathbf{v}^\pi = \mathbf{r}^\pi + \gamma \mathbf{M}^\pi \mathbf{v}^\pi \tag{3}$$

and $\mathbf{r}^\pi$ is the reward function

$$\mathbf{r}^\pi(\langle x,s\rangle) = \sum_a \psi(x,a) R(s,a).$$

The matrix $\mathbf{M}$ is the cross-product MDP matrix defined as

$$\mathbf{M}^\pi(\langle x,s\rangle, \langle x',s'\rangle) = \sum_{a \in A} \psi(x,a) T(s,a,s') \sum_{\omega \in \Omega} O(a,s',\omega) \eta(x,\omega,x'). \tag{4}$$

When constructing the matrix $\mathbf{M}$, an enumeration of the node-state pairings is necessary and the choice of enumeration will impact the structure of the matrix. Throughout this paper, we assume a state-major ordering. This means that for a fixed controller node x we enumerate all s before moving to the next controller node in the vectorization.

To find the policy which maximizes Equation 2 we perform gradient ascent in policy space [18]. This requires differentiation of the function $\mathbf{v}^\pi$ with respect to π. We solve for $\mathbf{v}^\pi$ starting from Equation 3, this yields

$$\mathbf{v}^\pi = (\mathbf{I} - \gamma \mathbf{M}^\pi)^{-1} \mathbf{r}^\pi. \tag{5}$$

Differentiating Equation 5 then gives

$$\frac{\partial \mathbf{v}^\pi}{\partial \pi} = \mathbf{Z}^{-1} \Big(\frac{\partial \mathbf{r}^\pi}{\partial \pi} + \frac{\partial \mathbf{Z}}{\partial \pi} \mathbf{Z}^{-1} \mathbf{r}^\pi \Big) \tag{6}$$

where we have used $\mathbf{Z} = \mathbf{I} - \gamma \mathbf{M}^\pi$ for ease of presentation. We note that $\mathbf{Z}$ in Equation 6 is diagonally dominant (cf. Proposition 4) which implies $\mathbf{Z}^{-1}$ is well defined. The FSC representation of a policy π is defined in terms of the action selection distribution ψ and node transition matrix η. These representations of the policy in combination with Equation 6, result in the following iterate for gradient ascent with a FSC policy representation

$$\psi^{(k+1)}(x,a) = \psi^{(k)}(x,a) + \alpha \beta^0 \cdot \frac{\partial \mathbf{v}^{\pi^{(k)}}}{\partial \psi^{(k)}(x,a)} \tag{7}$$

$$\eta^{(k+1)}(x,\omega,x') = \eta^{(k)}(x,\omega,x') + \alpha \beta^0 \cdot \frac{\partial \mathbf{v}^{\pi^{(k)}}}{\partial \eta^{(k)}(x,\omega,x')} \tag{8}$$

where α denotes the step-size.

Lastly, we remark that the policies produced by gradient ascent must satisfy constraints. In particular, ψ and η must be valid probabilities and $\mathbf{v}^\pi$ must satisfy Equation 3. Satisfaction of the Bellman equation can be done implicitly by expressing Equation 2 as

$$\pi^* = \arg\max_\pi \beta_0 \cdot (\mathbf{I} - \gamma \mathbf{M}^\pi)^{-1} \mathbf{r}^\pi \tag{9}$$

whereas the simplex constraints are satisfied via projecting each new iterate in Equations 7 and 8 onto the simplex.

III. Banded Controllers

In this section, the notion of a banded controller is introduced. We begin by defining the relevant underlying linear algebra which is then used to formally define a banded controller. The utility of the representation is then discussed.

A. Banded Matrices

Informally, a banded matrix is defined as being zero after some limiting point along the upper and lower diagonals. Formally, we say that a matrix $\mathbf{B}$ has lower bandwidth p if $b_{i,j}=0$ for all $i>j+p$ and upper bandwidth q if $b_{i,j}=0$ for all $j>i+q$. As a result, the set

$$\mathcal{B}^n_{p,q}:=\left\{\mathbf{B}\in\mathbb{R}^{n\times n}\,|\,b_{i,j}=0 \text{ if } i>j+p \text{ or } j>i+q\right\}$$

represents all real valued $n\times n$ banded matrices with lower bandwidth p and upper bandwidth q. As an example, a matrix $\mathbf{B}\in\mathcal{B}^5_{1,2}$ has the form

$$\mathbf{B}=\begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} & 0 & 0\\ b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} & 0\\ 0 & b_{3,2} & b_{3,3} & b_{3,4} & b_{3,5}\\ 0 & 0 & b_{4,3} & b_{4,4} & b_{4,5}\\ 0 & 0 & 0 & b_{5,4} & b_{5,5}\end{pmatrix}.$$

Another common example is when $p=q=1$ which results in a tri-diagonal matrix. However, the above space $\mathcal{B}^n_{p,q}$ is very expressive and contains a large number of structured matrices. Selecting $p=q=0$ corresponds to diagonal matrices; $p=n-1, q=0$ are lower triangular matrices; and $p=q=n-1$ corresponds to full $n\times n$ matrices.

One of the useful properties of banded matrices is that the complexity of their direct solution is directly related to the bandwidth of the matrix.

Proposition 1: Let $\mathbf{B}\in\mathcal{B}^n_{p,q}$ then the direct solution of $\mathbf{Bx}=\mathbf{b}$ is $O(pqn)$ [24].

This allows for significant savings when solving linear systems. We will later show how this relates to efficient policy evaluation.

B. The Space of Banded Controllers

Recall that a finite state controller policy π is defined in terms of an action selection distribution ψ and node transition function η.

Definition 1: A finite state controller defined by the pair (ψ,η) is a **banded controller** if $\eta(\cdot,\omega,\cdot)\in\mathcal{B}^n_{p,q}$ for $\omega\in\Omega$.

This results in structurally constrained policy representations which: (1) will reduce the size of the search space and (2) will result in more efficient methods for policy evaluation. Further, the size of the search space is controlled by the bandwidth of the matrices in $\mathcal{B}^n_{p,q}$. As a result, the search space can be made more expressive by widening the allowable bandwidth or more computationally efficient by restricting the overall bandwidth.

C. Policy Representation

Imposing structure onto the policy representation reduces the search space in the underlying optimization problem. This is computationally advantageous but excludes policy representations which do not contain this structure. In this section we show that banded controllers are computationally efficient representations for a class of controllers with no *a priori* structure. We illustrate this with a simple example, then provide a more general statement on the class of controllers this applies to.

Let $\mathcal{A}=\{left, right, up, down\}$ denote our action space and consider a controller with four nodes. Assume that the optimal controller results in the following action selection distributions for each node

$$\psi(x_1,\cdot)=(1,0,0,0) \qquad \psi(x_2,\cdot)=(0,0,0,1)$$
$$\psi(x_3,\cdot)=(0,0,1,0) \qquad \psi(x_4,\cdot)=(0,1,0,0)$$

where the action distribution ordering follows the definition of $\mathcal{A}$. That is node x_1 selects action $left$, x_2 selects action $down$ and so on. Further, assume the node transition structure is given by

$$\eta(\cdot,\omega,\cdot)=\begin{pmatrix} 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ 0 & 0.5 & 0 & 0.5\\ 0.5 & 0 & 0.5 & 0\end{pmatrix}. \tag{10}$$

This matrix is sparse but not banded. The directed graph representation of the controller (ψ,η) is given in Figure 2.

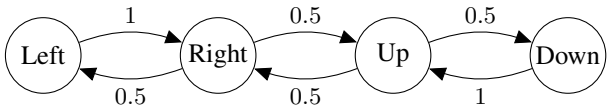

Fig. 2. Controller representation given by Equation 10 for fixed ω.

Next, we note that Equation 10 can be transformed into a banded matrix by interchanging columns two and four and rows two and four, resulting in

$$\eta_{\mathcal{B}}(\cdot,\omega,\cdot)=\begin{pmatrix} 0 & 1 & 0 & 0\\ 0.5 & 0 & 0.5 & 0\\ 0 & 0.5 & 0 & 0.5\\ 0 & 0 & 1 & 0\end{pmatrix}. \tag{11}$$

Formally, the relationship between η and $\eta_{\mathcal{B}}$ can defined in terms of a similarity transformation. That is, we have

$$\eta_{\mathcal{B}}=P^{-1}\eta P$$

for the permutation matrix

$$P=\begin{pmatrix} 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\end{pmatrix}.$$

Applying the same reordering to ψ by $\psi_{\mathcal{B}}(\cdot,\cdot)=P\psi(\cdot,\cdot)$

$$\psi_{\mathcal{B}}(x_1,\cdot)=(1,0,0,0) \qquad \psi_{\mathcal{B}}(x_2,\cdot)=(0,1,0,0)$$
$$\psi_{\mathcal{B}}(x_3,\cdot)=(0,0,1,0) \qquad \psi_{\mathcal{B}}(x_4,\cdot)=(0,0,0,1)$$

where the initial controller node x^0 is similarly permuted to obtain $x^0_{\mathcal{B}}$. One can confirm that $(x^0_{\mathcal{B}}, \psi_{\mathcal{B}}, \eta_{\mathcal{B}})$ is an equivalent controller policy to (x^0, ψ, η). This example illustrates that banded controllers are policy representations for a class of controllers which are not *a priori* banded. More generally, these representations provide equivalent controller representations for all permutation similar controllers.

Definition 2: A controller (ψ, η) is **permutation similar** to a banded controller, if there exists a permutation matrix P satisfying $P^2 = I$ such that $P^{-1}\eta P \in \mathcal{B}$.

That is, any controller η which can be transformed into a banded matrix via row and column swaps is in the search space of banded controllers. The underlying insight is that an action distribution can be associated with any controller node as long as the relationship between each action distribution is preserved.

D. Policy Evaluation

Policy evaluation is an essential part of POMDP solution frameworks. One of the primary motivations for introducing the notion of a banded controller is that policy evaluation can be done much more efficiently.

Let $C_{p,q}$ denote a banded controller with lower and upper bandwidth p and q, respectively. Further, let $n_x = |X|$ and $n_s = |S|$ denote the number of nodes of the FSC and states of of the POMDP, respectively. Starting from Equation 3, policy evaluation reduces to solving the linear system of equations

$$(\mathbf{I} - \gamma \mathbf{M}^{\pi})\mathbf{v}^{\pi} = \mathbf{r}^{\pi}.$$

In the canonical representation of a FSC, there is no underlying structure in the matrix $(\mathbf{I} - \gamma \mathbf{M}^{\pi})$ and the solution time for a direct solve is $O(n_x^3 n_s^3)$. When considering the banded controller $C_{p,q}$, it can be shown that

$$\mathbf{Z} = \mathbf{I} - \gamma \mathbf{M}^{\pi}$$

is a banded matrix with lower and upper bandwidth $(p+1)n_s - 1$ and $(q+1)n_s - 1$, respectively (cf. Proposition 3). It then follows from Proposition 1 that policy evaluation can be performed in $O(n_s^3 n_x pq)$. We remark that this is the complexity for the direct solution and, as a result, there are no potential issues with iteration convergence and tolerances.

Further, it can be shown that $\mathbf{Z}$ is diagonally dominant (cf. Proposition 4). This allows Gaussian Elimination without pivoting to be used which preserves the banded structure in the factored system (cf. [24]).

Proposition 2: Let $\mathbf{B} \in \mathcal{B}^n_{p,q}$ and let $\mathbf{B} = \mathbf{L}\mathbf{U}$ denote it LU factorization. If $\mathbf{B}$ is diagonally dominant, then $\mathbf{L}$ has lower bandwidth p and $\mathbf{U}$ has upper bandwidth q.

Importantly, this means that all subsequent uses of $\mathbf{Z}^{-1}$ in, for example, the gradient computation can be done efficiently.

It it worth remarking that the asymptotics of iterative solutions are $O(n_x^2 n_s^2)$. However, in practice for iterative methods to be effective, a preconditioner is typically necessary. In the context of policy evaluation, finding an effective preconditioner is difficult because the matrix is parameterized by π. That is, the linear system changes at each iteration of gradient ascent which would likely require a new preconditioner.

IV. A Banded Controller Gradient Ascent Algorithm

In this section we begin by reformulating the original optimization given by Equation 9 in terms of a banded controller. We begin by expressing the simplex constraints in vector notation and then present the formulation in terms of in-band node transitions only.

Letting $\boldsymbol{\psi}$ and $\boldsymbol{\eta}$ denote the vector form of ψ and η, the simplex constraints can be expressed as

$$\mathbf{J}_{\psi}\boldsymbol{\psi} = \mathbf{1}, \quad \boldsymbol{\psi} \geq \mathbf{0} \quad \text{and} \quad \mathbf{J}_{\eta}\boldsymbol{\eta} = \mathbf{1}, \quad \boldsymbol{\eta} \geq \mathbf{0}$$

where the matrices $\mathbf{J}_{\psi}$ and $\mathbf{J}_{\eta}$ are defined to enforce the appropriate summations to unity. By defining

$$\boldsymbol{\pi} = \begin{pmatrix} \boldsymbol{\psi} \\ \boldsymbol{\eta} \end{pmatrix} \quad \text{and} \quad \mathbf{J} = \begin{pmatrix} \mathbf{J}_{\psi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_{\eta} \end{pmatrix}$$

the original optimization problem defined by Equation 9 augmented with the banded controller constraint is given by

$$\begin{aligned} \underset{\pi}{\text{maximize}} \quad & \beta_0 \cdot (\mathbf{I} - \gamma \mathbf{M}^{\pi})^{-1} \mathbf{r}^{\pi} \\ \text{subject to} \quad & \mathbf{J}\boldsymbol{\pi} = \mathbf{1} \\ & \boldsymbol{\pi} \geq \mathbf{0} \\ & \eta(\cdot, \omega, \cdot) \in \mathcal{B}_{p,q}, \qquad \text{for } \omega \in \Omega. \end{aligned} \tag{12}$$

The minimizer of Equation 12 is the optimal policy in the constrained space of banded controllers.

A. Reduced System

Next, we express the above in terms of only the in-band node transitions, i.e. the non-zero parameters of the banded controller. Given $\eta(\cdot, \omega, \cdot) \in \mathcal{B}_{p,q}$, without loss of generality, define its vector form as

$$\boldsymbol{\eta} = (\boldsymbol{\eta}_{\mathcal{B}}^T, \boldsymbol{\eta}_0^T)^T$$

where $\boldsymbol{\eta}_{\mathcal{B}}$ corresponds to all of the in-band node transitions, and $\boldsymbol{\eta}_0$ corresponds to all of the off-band (i.e zero valued) node transitions.

The goal is to replace $\boldsymbol{\eta}$ with $\boldsymbol{\eta}_{\mathcal{B}}$ in Equation 12. This is done as follows. First, we note that the simplex constraint implies

$$\begin{pmatrix} \boldsymbol{J}_{\eta_{11}} & \boldsymbol{J}_{\eta_{12}} \\ \boldsymbol{J}_{\eta_{21}} & \boldsymbol{J}_{\eta_{22}} \end{pmatrix} \begin{pmatrix} \boldsymbol{\eta}_{\mathcal{B}} \\ \boldsymbol{\eta}_0 \end{pmatrix} = \begin{pmatrix} \mathbf{1} \\ \mathbf{1} \end{pmatrix}.$$

Because $\boldsymbol{\eta}_0$ is identically zero, this implies that $\boldsymbol{J}_{\eta_{11}}\boldsymbol{\eta}_{\mathcal{B}} = \mathbf{1}$. As a result, by defining

$$\boldsymbol{\pi}_{\mathcal{B}} = \begin{pmatrix} \boldsymbol{\psi} \\ \boldsymbol{\eta}_{\mathcal{B}} \end{pmatrix} \quad \text{and} \quad \mathbf{J}_{\mathcal{B}} = \begin{pmatrix} \mathbf{J}_{\psi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{J}_{\eta_{11}} \end{pmatrix}$$

Equation 12 can be written as

$$\begin{aligned} \underset{\pi_{\mathcal{B}}}{\text{maximize}} \quad & \beta_0 \cdot (\mathbf{I} - \gamma \mathbf{M}^{\pi_{\mathcal{B}}})^{-1} \mathbf{r}^{\pi_{\mathcal{B}}} \\ \text{subject to} \quad & \mathbf{J}_{\mathcal{B}}\boldsymbol{\pi}_{\mathcal{B}} = \mathbf{1} \\ & \boldsymbol{\pi}_{\mathcal{B}} \geq \mathbf{0}. \end{aligned} \tag{13}$$

This form is equivalent to Equation 12 but has removed the identically zero components $\boldsymbol{\eta}_0$. Further, the banded constraint is implicitly embedded in the optimization which means no explicit projections onto the space of banded controllers is necessary.

B. Projected Gradient Ascent

We solve Equation 13 using projected gradient ascent (PGA). For ease of presentation, let

$$f(\boldsymbol{\pi}_{\mathcal{B}}) = \beta_0 \cdot (\mathbf{I} - \gamma \mathbf{M}^{\pi_{\mathcal{B}}})^{-1} \mathbf{r}^{\pi_{\mathcal{B}}}$$

denote the objective function and

$$\mathcal{S} = \{\boldsymbol{\pi}_{\mathcal{B}} \,|\, \mathbf{J}_{\mathcal{B}} \boldsymbol{\pi}_{\mathcal{B}} = \mathbf{1} \text{ and } \boldsymbol{\pi}_{\mathcal{B}} \geq \mathbf{0}\}$$

denote the set of constraints. PGA produces an intermediate policy update at iterate k as

$$\boldsymbol{\pi}_B^{(k+1/2)} = \boldsymbol{\pi}_B^{(k)} + \alpha^{(k)} \nabla f(\boldsymbol{\pi}_B^{(k)})$$

where $\alpha^{(k)}$ denotes step size and $\nabla f(\boldsymbol{\pi}_{\mathcal{B}}^{(k)})$ is composed of all the partial derivatives

$$\frac{\partial f(\pi)}{\partial \pi_{\mathcal{B},i}} = \beta_0 \cdot \left(\mathbf{Z}^{-1} \left(\frac{\partial \mathbf{r}^{\pi}}{\partial \pi_{\mathcal{B},i}} + \frac{\partial \mathbf{Z}}{\partial \pi_{\mathcal{B},i}} \mathbf{Z}^{-1} \mathbf{r}^{\pi} \right) \right)$$

where $\mathbf{Z} = \mathbf{I} - \gamma \mathbf{M}^{\pi}$. The next policy iterate is then obtained by projecting onto the set of constraints $\mathcal{S}$

$$\boldsymbol{\pi}_B^{(k+1)} = \mathcal{P}_{\mathcal{S}} \left(\boldsymbol{\pi}_B^{(k+1/2)} \right)$$

where $\mathcal{P}_S(\cdot)$ denotes the projection operator.

Algorithm 1 PGA with line search for a banded controller.

Require: ℓ: The number of nodes.
Require: ϵ: The convergence criterion.
1: $\pi_{\mathcal{B}}^{(0)} \leftarrow \langle X = \{1, \ldots, \ell\}, \psi^{(0)} = \text{RAND}(\,), \eta_{\mathcal{B}}^{(0)} = \text{RAND}(\,) \rangle$
2: $\mathbf{v}^{(0)} \leftarrow \text{POLICYEVALUATION}(\pi_{\mathcal{B}}^{(0)})$
3: $k \leftarrow 0$
4: **do**
5: $\quad \frac{\partial \mathbf{v}^{\pi_{\mathcal{B}}^{(k)}}}{\partial \pi_{\mathcal{B}}^{(k)}} \leftarrow \text{COMPUTEGRADIENT}(\pi_{\mathcal{B}}^{(k)})$
6: $\quad \pi_{\mathcal{B}}^{(k+1)}, \mathbf{v}^{(k+1)} \leftarrow \text{LINESEARCH}(\pi_{\mathcal{B}}^{(k)}, \mathbf{v}^{(k)}, \frac{\partial \mathbf{v}^{\pi_{\mathcal{B}}^{(k)}}}{\partial \pi_{\mathcal{B}}^{(k)}})$
7: $\quad \pi_{\mathcal{B}}^{(k+1)} \leftarrow \text{PROJECTTOSIMPLEX}(\pi_{\mathcal{B}}^{(k+1)})$
8: $\quad k \leftarrow k+1$
9: **while** $\text{RELATIVEERROR}(\mathbf{v}^{\pi_{\mathcal{B}}^{(k)}}, \mathbf{v}^{\pi_{\mathcal{B}}^{(k-1)}}) > \epsilon$
10: **return** $\text{PROJECTTOSIMPLEX}(\pi_{\mathcal{B}}^{(k)})$

C. Line Search

In order to minimize the number of iterates when performing gradient ascent, we aim to increase the quality of each step by performing a line search algorithm along the gradient direction. Line search methods typically require evaluation of the objective function along the direction of the gradient. In this context, evaluating the objective function requires us to evaluate the policy at each sample point. As mentioned in the previous sections, policy evaluation is expensive; as a result, we employ Golden Section Search which has been used successfully the PGA context [21]. Golden section search coupled with the computational efficiency of policy evaluation for a banded controller allows us to feasibly generate higher quality gradient updates.

V. Theoretical Analysis

In this section, we prove algorithmically relevant theoretical results referenced throughout the paper. Specifically, we establish that the cross-product MDP matrix is banded and that the properties needed for efficient policy evaluation hold.

Proposition 3: Assume that the cross-product MDP matrix $\mathbf{M}^{\pi}$ in Equation 4 is constructed in state-major ordering. Then given a banded controller $C_{p,q}$, the matrix $\mathbf{M}^{\pi}$ is banded with lower bandwidth $(p+1)n_s - 1$ and upper bandwidth $(q+1)n_s - 1$.

Proof: We begin by writing $\mathbf{M}^{\pi}$ as a block matrix with the blocks indexed by the controller node indices. We have

$$\mathbf{M}^{\pi} = \begin{pmatrix} \mathbf{M}_{x_1,x_1} & \mathbf{M}_{x_1,x_2} & \cdots & \mathbf{M}_{x_1,x_{n_x}} \\ \mathbf{M}_{x_2,x_1} & \mathbf{M}_{x_2,x_2} & \cdots & \mathbf{M}_{x_2,x_{n_x}} \\ & \vdots & & \\ \mathbf{M}_{x_{n_x},x_1} & \mathbf{M}_{x_{n_x},x_2} & \cdots & \mathbf{M}_{x_{n_x},x_{n_x}} \end{pmatrix}$$

where each block is an $n_s \times n_s$ matrix

$$\mathbf{M}_{x_i,x_j}(s,s') := \mathbf{M}^{\pi}(\langle x_i, s \rangle, \langle x_j, s' \rangle)$$

with fixed $x_i, x_j \in X$. By definition we have

$$\mathbf{M}_{x_i,x_j}(s,s') = \sum_{a \in A} \psi(x,a) T(s,a,s') \sum_{\omega \in \Omega} O(a,s',\omega) \eta(x_i,\omega,x_j).$$

Since $\eta(x_i,\omega,x_j)$ is banded, we have that $\eta(x_i,\omega,x_j) = 0$ for $i > j+p$ or $j > i+q$. This implies that $\mathbf{M}_{x_i,x_j}$ is identically zero if $i > j+p$ or $j > i+q$, meaning that the banded structure of the controller is inherited by the block structure of $\mathbf{M}^{\pi}$. Because each block is of size n_s, it follows that $\mathbf{M}^{\pi}$ has lower bandwidth $(p+1)n_s - 1$ and upper bandwidth $(q+1)n_s - 1$. ■

The next proposition has important consequences for the factorization of banded systems.

Proposition 4: The matrix $\mathbf{Z} = \mathbf{I} - \gamma \mathbf{M}^{\pi}$ is banded and diagonally dominant.

Proof: The subtraction of $\gamma \mathbf{M}^{\pi}$ from $\mathbf{I}$ has no impact on the band structure and therefore $\mathbf{Z}$ has the same bandwidth as $\mathbf{M}^{\pi}$. Further, $\mathbf{M}^{\pi}$ is a stochastic matrix which implies all row summations are unity. The result follows by noting that $\gamma < 1$. ■

Corollary 1: For $\mathbf{Z} = \mathbf{I} - \gamma \mathbf{M}^{\pi}$ let $\mathbf{LU} = \mathbf{Z}$ denote its LU factorization. Then $\mathbf{L}$ has lower bandwidth $(p+1)n_s - 1$ and $\mathbf{U}$ has upper bandwidth $(q+1)n_s - 1$.

The corollary follows by combining Propositions 2 and 4. We note that the LU factorization of a banded matrix does not necessarily inherit the same bandwidth structure due to partial pivoting when performing Gaussian Elimination. However, because the FSC formulation is always diagonally dominant, the LU factorization can be obtained without partial pivoting; as a result, the LU factorization inherits the same bandwidth structure as the original matrix. This is important because it allows efficient memory use in computations of $\mathbf{Z}^{-1}$ which is needed at several points in the above algorithm.

Domain	$\|S\|$	$\|A\|$	$\|\Omega\|$	$\|X\|$	NLP Baseline [17]			Gradient Ascent Baseline [18]			Banded Gradient Ascent		
					ADR	T	$\|\pi\|$	ADR	T	$\|\pi\|$	ADR	T	$\|\pi\|$
toy	3	2	1	5	**0.08**	**0.02**	25	**0.08**	0.03	25	**0.08**	**0.02**	**20**
	3	2	1	10	**0.08**	**0.04**	100	**0.08**	0.16	100	**0.08**	**0.05**	**40**
qcd	3	2	3	5	**0.0**	0.04	65	−0.31	0.06	65	−0.22	**0.04**	**50**
	3	2	3	10	**0.0**	0.11	280	−0.31	0.65	280	−0.24	**0.08**	**100**
milos-aaai97	20	6	8	5	**22.1**	789.0	185	7.86	293.9	185	9.41	**167.4**	**145**
	20	6	8	10	**20.1**	253.1	770	13.4	947.4	770	16.1	**307.4**	**290**
query.s3	27	3	3	5	261.3	**0.98**	70	**318.2**	97.3	70	311.3	38.1	**55**
	27	3	3	10	261.7	288.3	290	**338.5**	167.6	290	311.5	**58.1**	**110**
tiger-grid	36	5	17	5	**0.0**	77.3	360	**0.0**	27.2	360	**−0.01**	**28.7**	**275**
	36	5	17	10	—	—	—	**0.0**	103.3	1570	−0.11	**61.8**	**550**
home-healthcare	64	5	2	5	**9.96**	2627.4	60	9.86	212.7	60	**9.95**	**110.8**	**50**
	64	5	2	10	—	—	—	**9.98**	408.9	220	**9.98**	**220.7**	**100**
query.s4	81	4	3	5	267.0	2111.8	75	**328.7**	268.8	75	283.1	**108.9**	**60**
	81	4	3	10	—	—	—	**342.0**	1079.9	300	281.6	**119.8**	**120**

TABLE I

RESULTS FROM SIMULATION. ALGORITHMS: THE PROPOSED BANDED GRADIENT ASCENT, TWO BASELINES: NONLINEAR PROGRAMMING (NLP) BASELINE [17] AND VANILLA GRADIENT ASCENT BASELINE [18]. DOMAINS: SEVEN BENCHMARKS, EACH VARYING NUMBER OF NODES ($|X|$). METRICS: AVERAGE DISCOUNTED REWARD (ADR), AND TIME IN SECONDS (T).

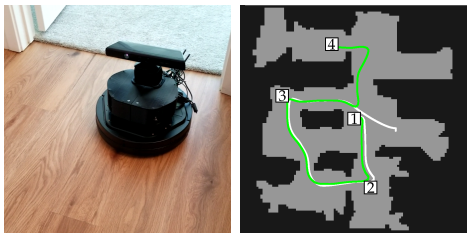

Fig. 3. Experimental results for banded gradient ascent's controller policy on a real robot acting in a real household environment. This fully implements the *home-healthcare* POMDP domain on an actual robot. The green path shows the controller's path computed by banded gradient ascent. The white path shows the controller's path from the gradient ascent baseline.

VI. EXPERIMENTS

A. Experimental Setting and Domains

We evaluate the novel banded controller gradient ascent with line search algorithm. It is compared against two baseline controller algorithms. The nonlinear programming (NLP) baseline [17] solves Equations 2 and 3. The vanilla gradient ascent with line search baseline [18] performs gradient ascent using Equation 6 with the same golden section line search. Each of the baseline algorithms work with the same optimization formulation in policy space and all three controller algorithms are directly compared with the same fixed number of nodes $|X|$ (5 and 10).

Standard metrics are also used [11], [17]: (1) average discounted reward (ADR), (2) time to solve in seconds, and (3) policy size in terms of the number of parameters. Seven standard POMDP benchmark domains are used, varying in size and complexity, ranging from the smaller *toy* and *qcd* to the larger *tiger-grid* and *query.s4*. Results were averaged over 10 trials for each combination of algorithm and domain. The algorithms were implemented and run in Julia 1.6. The experiments were done on an Intel Core i7-6700HQ CPU with 4 cores at 2.6GHz and 16GB of RAM.

B. Results and Discussion

Table I shows the results from our experiments. We first observe that no one FSC algorithm achieves a superior ADR over all benchmark problems. This can be attributed to the policy space formulation being non-convex. We observe that the banded gradient ascent ADR is comparable to the baseline algorithms and outperforms either NLP or vanilla gradient ascent for all but one benchmark problem. This is despite the baseline algorithms containing enough degrees of freedom to represent any banded controller. This suggests banded controllers are a useful way of constraining the search space in the underlying optimization problem.

We observe that banded gradient ascent is typically much faster than both NLP and gradient ascent baselines. For example, in larger domains such as *home-healthcare*, the banded approach is two times faster than vanilla gradient ascent, and nearly ten times faster than NLP. In some large domains, the NLP failed to converge within 2 hours, with its convergence behavior sporadic overall. To evaluate the effect that the banded controller's structure has on robot behavior, we analyze its use on real robot.

Figure 3 demonstrates a banded controller policy computed by the proposed approach that successfully searches the household environment. For comparison, a policy computed by gradient ascent is also shown. Both policies were computed using $|X|=10$ nodes. We observe that banded controllers are able to capture a very similar policy, but are computable up to an order of magnitude faster.

VII. CONCLUSION

This paper introduces a novel gradient based algorithm which leverages a new structured policy representation termed a banded controller. We show that the computational structure induced by the policy representation allows the algorithm to leverage banded matrices within the gradient updates for significant speedups. The algorithm is shown to outperform state-of-the-art finite state controller algorithms over standard benchmark problems as well as on a real robot. The theoretical and experimental results demonstrate that banded controllers are computationally efficient and expressive structured policy representations which increase the scalability of finite state controller algorithms for POMDPs.

REFERENCES

[1] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[2] M. Lauri, D. Hsu, and J. Pajarinen, "Partially observable markov decision processes in robotics: A survey," *IEEE Transactions on Robotics*, 2022.

[3] K. H. Wray, S. J. Witwicki, and S. Zilberstein, "Online decision-making for scalable autonomous systems," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 4768–4774.

[4] L. Dressel and M. Kochenderfer, "Hunting drones with other drones: Tracking a moving radio target," in *2019 International Conference on Robotics and Automation*, 2019, pp. 1905–1912.

[5] M. J. Kochenderfer, *Decision Making Under Uncertainty: Theory and Application*. MIT Press, 2015.

[6] S. K. Jayaweera, *Signal Processing for Cognitive Radios*. John Wiley & Sons, Hoboken, NJ, USA, 2014.

[7] V. Krishnamurthy, *Partially Observed Markov Decision Processes from Filtering to Controller Sensing*. Cambridge University Press, 2016.

[8] C. Papadimitriou and J. Tsitsiklis, "The complexity of markov decision processes," *Mathematics of Operations Research*, vol. 12, pp. 441–450, 1987.

[9] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based POMDP solvers," *Autonomous Agents and Multi-Agent Systems*, pp. 1–51, 2013.

[10] E. A. Hansen, "Solving POMDPs by searching in policy space," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 211–219.

[11] J. Pineau, G. Gordon, and S. Thrun, "Anytime point-based approximations for large POMDPs," *Journal of Artificial Intelligence Research*, vol. 27, pp. 335–380, 2006.

[12] M. Spaan and N. Vlassis, "Perseus: Randomized point-based value iteration for POMDPs," *Journal of Artificial Intelligence Research*, vol. 24, pp. 195–220, 2005.

[13] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces." in *Robotics: Science and systems*, 2008.

[14] P. Poupart and C. Boutilier, "Bounded finite state controllers," in *Proceedings of Advances in Neural Information Processing Systems 16*, 2004, pp. 823–830.

[15] E. A. Hansen and R. Zhou, "Synthesis of hierarchical finite-state controllers for pomdps." in *ICAPS*, 2003, pp. 113–122.

[16] M. Toussaint, L. Charlin, and P. Poupart, "Hierarchical pomdp controller optimization by likelihood maximization." in *UAI*, vol. 24, 2008, pp. 562–570.

[17] C. Amato, D. S. Bernstein, and S. Zilberstein, "Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs," *Autonomous Agents and Multi-Agent Systems*, vol. 21, no. 3, pp. 293–320, 2010.

[18] N. Meuleau, K.-E. Kim, L. P. Kaelbling, and A. R. Cassandra, "Solving POMDPs by searching the space of finite policies," in *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, 1999, pp. 417–426.

[19] M. Grzes and P. Poupart, "Incremental policy iteration with guaranteed escape from local optima in pomdp planning," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 1249–1257.

[20] J. K. Pajarinen and J. Peltonen, "Periodic finite state controllers for efficient POMDP and DEC-POMDP planning," in *Advances in Neural Information Processing Systems*, 2011, pp. 2636–2644.

[21] K. H. Wray and K. Czuprynski, "Scalable POMDP decision-making using circulant controllers," in *2021 International Conference on Robotics and Automation*, 2021.

[22] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," *Advances in neural information processing systems*, vol. 23, 2010.

[23] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "Despot: Online pomdp planning with regularization," *Advances in neural information processing systems*, vol. 26, 2013.

[24] J. W. Demmel, *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.