A Bayesian Game based Adaptive Fuzzy Controller for Multiagent POMDPs

Rajneesh Sharma and Matthijs T. J. Spaan

Abstract— **This paper develops a novel fuzzy reinforcement learning (RL) based controller for multiagent partially observable Markov decision processes (POMDPs) modeled as a sequence of Bayesian games. Multiagent POMDPs have emerged as a powerful framework for modeling and optimizing multiagent sequential decision making problems under uncertainty, but finding optimal policies is computationally very challenging. Our aim here is twin fold, (i) introduction of a learning paradigm in infinite horizon multiagent POMDPs and (ii) scaling up multiagent POMDP solution approaches by introduction of fuzzy inference systems (FIS) based generalization. We introduce what may be called fuzzy multiagent POMDPs to overcome space and time complexity issues involved in finding optimal policies for multiagent POMDPs. The proposed FIS based RL controller approximates optimal policies for multiagent POMDPs modeled as a sequence of Bayesian games. We empirically evaluate the proposed fuzzy multiagent POMDP controller on the standard benchmark multiagent tiger problem and compare its performance against other state-of-the-art multiagent POMDP solution approaches. Results showcase the effectiveness of the proposed approach and validate the feasibility of employing Bayesian game based RL (in conjunction with FIS approximation) for addressing the intractability of multiagent POMDPs.**

I. INTRODUCTION

Optimization of sequential decision making problems under uncertainty has been an active area of research for over three decades now spanning diverse fields such as Artificial Intelligence, Operations Research and Control Theory. Considerable success has been achieved for fully observable environments via the well known Markov decision process (MDP) framework [1], and also optimizing partially observable domains formalized as partially observable Markov decision processes (POMDPs) [2] has seen many advances. The POMDP framework extends MDP model by incorporating observations and their probability of occurrence conditional on the state of the environment to deal with *perceptual aliasing* and limited sensing capabilities [8], [30]. However, POMDPs become intractable for situations where two or more agents (multiagent POMDPs) have to cooperate to optimize a joint reward, nicely formalized by various multiagent POMDP frameworks [3].

Reinforcement learning (RL) has now established itself as a major technique for direct adaptive optimal control of uncertain systems [4]. RL methods have their roots in the studies of animal learning and learning control work. In the RL paradigm, agents learn to behave optimally by repeated trial and error interactions with the environment. RL has been successfully applied to a broad range of systems and processes [5], [6] which clearly bring out its viability as an optimization technique. However, majority of the RL research has focused on the single agent fully observable scenario, i.e., the MDP framework.

This work embodies a specific RL technique called Q learning [5], [7] at its heart that has proven convergence. Feasibility of defining an optimal Q value function Q^* for decentralized POMDPs has been established in [8], which is defined over the space of histories (which grows exponentially with the time horizon). In general, for moderately large problem domains Q learning can be implemented using a lookup table. However, for larger domains it is either impractical (large state spaces) or infeasible (continuous state spaces). Fortunately this 'curse of dimensionality' can be tackled by several standard function approximation techniques such as neural networks, fuzzy inference systems (FIS) etc. Fuzzy systems, in particular, offer an effective generalization scheme as they are capable of approximating any real function to an arbitrary accuracy [9]. Furthermore, empirical results with FIS have established that they are capable of learning [10], [11].

Herein, we focus on a particular multiagent POMDP formalism referred to as Dec-POMDPs, first proposed by Bernstein et. al. [12]. In a Dec-POMDP, aim of each agent is to maximize a joint global reward function. It has been shown that even finite horizon Dec-POMDPs are provably intractable (NEXP-complete) [13]. As this work concerns infinite horizon multiagent POMDPs, we refrain from discussing their finite horizon counterparts and refer the interested reader to [3], [8] for a detailed discussion and description of finite horizon multiagent POMDPs and their solution approaches.

Given the intractability of solving general multiagent POMDPs, several approximate approaches that make assumptions about domain conditions have emerged as a viable solution [14]. One such approach uses the assumption of free communication to address intractability [15]. Free communication at every time step transforms a multiagent POMDP into a more tractable single agent POMDP, albeit exponentially-sized in the number of agents. Authors in [16] use this transformation to generate "centralized policies" for multiagent POMDPs.

In the proposed approach we assume free communication at policy generation time as in [15] to generate "centralized

Rajneesh Sharma is with the Intelligent Systems Lab, Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal (phone: +351-21-8418270; e-mail: rajneesh496@isr.ist.utl.pt).

Matthijs T. J. Spaan is with the Intelligent Systems Lab, Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal (e-mail: mtjspaan@isr.ist.utl.pt).

policies" for multiagent teams modeled as what may be referred to as fuzzy multiagent POMDPs. We employ a fuzzy inference system for generalizing a continuous belief space multiagent POMDP. We use an FIS based RL controller instead of a neural controller as our experience has shown [5], [11] that an FIS based approximation is both efficient and quick in comparison to a neural function approximator. At the policy generation stage a centralized joint policy is obtained by solving the fuzzy multiagent POMDP modeled as a series of Bayesian games [8]. The computed approximate optimal joint policy is then executed in a decentralized manner. For an in-depth discussion on the feasibility and effectiveness of using communication in multiagent POMDPs, we refer the reader to [3], [15], [16].

To the best of our knowledge, this work represents a first effort that seeks to hybridize broad areas of fuzzy systems, game theory, reinforcement learning, and multiagent POMDPs for devising a robust and effective solution approach to address space and time intractability of multiagent POMDPs in general, and infinite horizon multiagent POMDPs in particular. Besides our fuzzy RL method, another significant contribution of this work is the introduction of a compact representation of the belief space using FIS for multiagent partially observable systems (pointed out in [3] as a major need for improving scalability). We also show how using the fuzzy q values that are learned online, we can compactly represent a multiagent POMDP policy. Our proposed approach scores over other state-of-the-art multiagent POMDP solution approaches, in terms of scalability, compact controller realization (low memory requirement), adaptability (use of learning framework), and quality of policy obtained (in terms of total discounted finite/infinite horizon reward obtained).

The proposed approach is applicable to multiagent POMDP domains wherein each agent maintains a belief and communicates it to a central FIS at each stage, which forms a fuzzy mapping of the belief space of the underlying Multiagent POMDP. This fuzzy belief mapping is then used to solve a sequence of Bayesian games to generate an approximate optimal joint policy which is executed by each agent. Under this joint action the system transitions to the next state and each agent receives his own observation and a signal that indicates the goodness of executing the joint action (joint reward). This signal is then used to tune q values to reflect the consequence of taking that joint action as per standard Q learning.

We empirically test the proposed approach on standard benchmark multiagent tiger problem [17]. Simulation results and comparison of expected discounted reward values realized, against other recent multiagent POMDP solution approaches validate the effectiveness and feasibility of FIS based solution to multiagent POMDPs. Section II briefly describes some background and related work. Section III describes the proposed fuzzy multiagent POMDP solution approach. Section IV gives empirical results of applying the scheme on the multiagent tiger problem and section V concludes the paper with a discussion on future scope of the proposed methodology.

II. Background and Related Work

We assume prior reader familiarity with the basic concepts underlying Reinforcement Learning [4], POMDPs [2] and Fuzzy Systems and give only a brief overview of the concepts that form the backbone of our work.

A. Multiagent POMDPs

Multiagent sequential decision making problems under partial observability where agents cooperate to optimize performance can be modeled by several representations such as MTDP [19], POIPSG [20], I-POMDP [21], and Dec-POMDP [12]. A multiagent POMDP models a number of agents that interact with their environment (system) at discrete time steps $t = 1, 2, \ldots$. At each time step t every agent takes an action and the combination of these actions (joint action) makes the system transit to the next state. At the next time step, each agent receives a local observation of the environment. State transition and observation probabilities model the dynamics of the environment while global reward specifies desired behavior or the goodness of taking joint action in a particular state. For infinite horizon problems the most typical goal of agents is to maximize expected infinite horizon discounted global reward defined as $\sum_{t=0}^{\infty} \gamma^t r_t$ where r_t is the reward received at time step t and γ is a discount factor, $0 \leq \gamma < 1$. Herein, we assume that joint policy is computed in a centralized manner and then the computed policy is distributed to each agent who merely executes it.

In this paper, we use the notation for multiagent POMDPs as introduced in [12] which defines a Dec-POMDP as a tuple $< N_a, S, \{U_g\}, P, \{\Omega_g\}, O, R, T >$ where

- S is the finite set of states;
- N_a is the number of agents;
- U_g is the finite set of actions available to agent g and $\bar{U} = \times_{g \in N_a} U_g$ is the set of joint actions with $\bar{u} =< u_1, \ldots, u_{N_a} >$ being joint action;
- P is the state transition function with $p(s'|s, \bar{u})$ being the probability of landing in state s' when joint action $\bar{u}$ is taken at state s;
- Ω_g is the finite set of observations available to the agent g and $\bar{\Omega} = \times_{g \in N_a} \Omega_g$ is the set of joint observations with $\bar{o} = < o_1, \ldots, o_{N_a} >$ being a joint observation;
- $O : \bar{U} \times X \rightarrow P(\bar{\Omega})$ is an observation function with $O(\bar{o}|\bar{u}, s')$ being the probability of receiving joint observation $\bar{o}$ given that joint action $\bar{u}$ was taken and led to state s' (but each agent only observes its own o_g);
- $R : \bar{U} \times S \rightarrow \Re$ is the reward function;
- T is the horizon.

In the infinite horizon case, we do not have the parameter T but instead a discount factor $0 \leq \gamma < 1$ which limits the infinite horizon sum $\sum_{t=0}^{\infty} \gamma^t r_t$.

B. Solution Approaches for Multiagent POMDPs

In [22], Bernstein et. al. proposed Bounded Policy Iteration (BPI) for Multiagent POMDPs. The approach optimizes the value of fixed number of nodes for each controller including local controller for each agent and a correlation device. However BPI gets stuck in local optima as only one controller node is improved at a time while all others are held fixed [3]. A recent approach by Amato et. al. [23] optimizes fixed size controllers allowing for a specific start distribution over states. The resulting non linear program (NLP) can be solved efficiently using existing solvers. Generating globally optimal solutions is hard and sometimes infeasible. Nevertheless, they empirically show that use of non linear optimization techniques leads to a better controller than produced by BPI.

Cogill et. al. [24] proposed an approximate dynamic programming approach that uses Q functions. The approach uses a centralized solution to tackle decentralized problems. By incorporating problem specific human knowledge they transform it into a set of easier sub problems and approximate the optimal decentralized solution. The algorithm uses the notion of Q functions to generate a linear programming solution to the decentralized problem. However, the weights used in the approximate linear programming have a huge impact on the quality of policy obtained and finding good weights remains an open problem.

In [15] authors propose to tackle intractability of multiagent POMDPs by using communication at every time step to transform a multiagent POMDP into a more tractable single agent POMDP. They propose "centralized" policies for multiagent POMDPs at plan-time by assuming presence of free communication. The approach basically trades off the need to do some computation at execution time for ability to generate policies more tractably at plan time. For more information regarding this approach, we refer the reader to [15].

C. FIS based Reinforcement Learning

Fuzzy logic is a mathematical approach to emulate human way of thinking and learning. In MDPs fuzzy systems have been used as function approximators to facilitate generalization in state space and for generating continuous actions. In [10], Jouffe introduced fuzzy Q learning wherein a collection of fuzzy rules is considered as an agent. The approach produces an action by triggering some rules and cooperating. Following on this idea authors in [11] introduced fuzzy Markov games where FIS is used to introduce generalization in Markov games.

Fuzzy RL has also been used for multiagent systems, e.g., in [25] authors use fuzzy RL on a multiagent continuous pursuit domain. In [26], authors implement fuzzy RL on robotic soccer agents to enable them to coordinate their behavior locally and socially while learning from experience. For POMDPs, a neuro fuzzy approach is proposed in [27] to generate fast, robust and easily interpretable solutions. To the best of our knowledge, until now there have been very few attempts at using fuzzy RL for effective learning in POMDPs.

III. Fuzzy Multiagent POMDP Approach

As our approach uses the formulation where multiagent POMDPs are modeled as a sequence of Bayesian games [8], we give a brief overview of Bayesian games (BG) to help reader understanding of the concepts introduced later.

A. Bayesian games

A Bayesian game [28] is an augmented normal form game in which players have some private information. This private information defines the type of agent, i.e., a particular type $\theta_g \in \Theta$ of an agent g corresponds to that agent knowing some particular information. A BG is defined by the tuple $< N_a, \bar{U}, \Theta, P(\Theta), \{r^1, \dots, r^{N_a}\} >$ where

- N_a is the number of agents;
- $\bar{U}$ is the set of joint actions;
- $\Theta = \times_{g \in N_a} \theta_g$ is the set of joint types over which a probability distribution $P(\Theta)$ is defined;
- $R : \Theta \times \bar{U} \rightarrow \Re$ is the payoff function.

In a Bayesian game, agents can condition their action on the private information they have. A joint policy in a BG is represented by $\beta =< \beta_1, \dots, \beta_{N_a} >$ where individual policies are mappings from types to actions, i.e., $\beta_g : \theta_g \rightarrow U_g$ (U_g being the action set of agent g). The solution of a BG for identical payoffs is given by [8]:

$$\beta^* = \arg\max_{\beta} \sum_{\theta \in \Theta} p(\theta) r(\theta, \beta(\theta)) \qquad (1)$$

where $\beta(\theta) =< \beta_1(\theta), \dots, \beta_{N_a}(\theta) >$ is the joint action specified by β, $p(\theta)$ is the probability associated with joint type θ and $r(\theta, \beta(\theta))$ is the payoff received. β^* is the Pareto optimal Nash equilibrium joint policy [28].

B. Multiagent POMDP: Sequence of BGs

In modeling multiagent POMDPs as a sequence of BGs [29] the payoff function of BG (one stage) is represented by $Q(\bar{\theta}^t, \bar{u})$ where $\bar{\theta}^t = (\bar{o}^0, \bar{u}^0, \bar{o}^1, \dots, \bar{o}^{t-1}, \bar{u}^{t-1}, \bar{o}^t)$ is the joint action-observation history up to time t. The initial joint observation $\bar{o}^0$ is assumed to be an empty observation: $\bar{o}^0 = \bar{o}_\emptyset = < o_{1,\emptyset}, o_{2,\emptyset}, \dots, o_{N_a,\emptyset} >$. In [8], authors define a new approximate Q-value function called Q_{BG} wherein BG have types that correspond to single observations instead of complete action-observation histories leading to smaller and more tractable BGs.

As our proposed approach uses a fuzzy RL framework, we use Q_{BG} at stage t as defined in [8] for a two stage BG. The Q function at stage t can be constructed from fuzzy q parameters defined at stage $(t+1)$: $q(i, \bar{u}^t, \theta^{t+1}, \beta(\theta^{t+1}))$ where i is an index specifying the fuzzy rule Y_i, $\bar{u}^t$ is the joint action at stage t, θ^{t+1} is the joint type at stage $(t+1)$, and $\beta(\theta^{t+1})$ is the joint BG policy corresponding to type θ^{t+1} at stage $(t+1)$. This conforms with the modeling of Multiagent POMDPs as a series of BGs as introduced in [8].

C. Fuzzy Multiagent POMDPs

We propose fuzzy multiagent POMDPs as a generalization of fuzzy Q learning (FQL) [10] / fuzzy Markov games (FMG) [11] to a Multiagent POMDP setting. Following FQL / FMG, we define fuzzy inference system for fuzzy multiagent POMDPs as consisting of N rules of the following form:

$$\begin{aligned}
&Y_i: \quad \text{If } b_1^t \text{ is } L_1^i \text{ and } \dots \text{ and } b_{N_a}^t \text{ is } L_{N_a}^i \\
&\text{then } u_1 = u_{11} \text{ and } \dots \text{ and } u_{N_a} = u_{1N_a} \\
&\qquad\qquad\qquad \text{with } Q_{BG}(i, u_{11}, \dots, u_{1N_a}) \\
&\text{or} \quad u_1 = u_{21} \text{ and } \dots \text{ and } u_{N_a} = u_{1N_a} \\
&\qquad\qquad\qquad \text{with } Q_{BG}(i, u_{21}, \dots, u_{1N_a}) \\
&\qquad\qquad\qquad \vdots \\
&\text{or} \quad u_1 = u_{m1} \text{ and } \dots \text{ and } u_{N_a} = u_{mN_a} \\
&\qquad\qquad\qquad \text{with } Q_{BG}(i, u_{m1}, \dots, u_{mN_a})
\end{aligned} \tag{2}$$

where i is the index specifying rule Y_i, $m = |U_g| \forall g \in N_a$, i.e., action set of all agents is assumed to have same cardinality m. u_{kg} is the kth action in U_g or kth action of agent g, $[b_1^t, \dots, b_{N_a}^t]$ is the belief vector for agents $1, \dots, N_a$ at time t, L_g^i is the linguistic term (fuzzy label) of input variable b_g^t in rule Y_i and its membership function denoted by $\mu_{L_g^i}$.

Figure 1 shows the BG for time steps t and $(t+1)$ for a fictitious multiagent POMDP with two agents each having two actions and two observations. $Q_{BG}(i, u_{m1}, \dots, u_{mN_a}) := Q_{BG}(i, \bar{u}^t)$ is the solution of BG defined at the next stage $(t+1)$ corresponding to the tuple $< i, u_{m1}, \dots, u_{mN_a} >$. It is calculated as the maximizing sum of the entries of next time step BG weighted by their respective type probabilities, i.e.,

$$Q_{BG}(i, \bar{u}^t) = \max_{\beta} \sum_{\theta^{t+1} \in \Theta} p(\theta^{t+1} | \theta^t, \bar{u}^t) q(i,\ \bar{u}^t, \theta^{t+1}, \beta(\theta^{t+1})) \tag{3}$$

where θ^{t+1} is the joint type at stage $(t+1)$ and $\bar{u}^t$ is the joint action at stage t. The maximizing BG policy β^* at next stage $(t+1)$ is given by

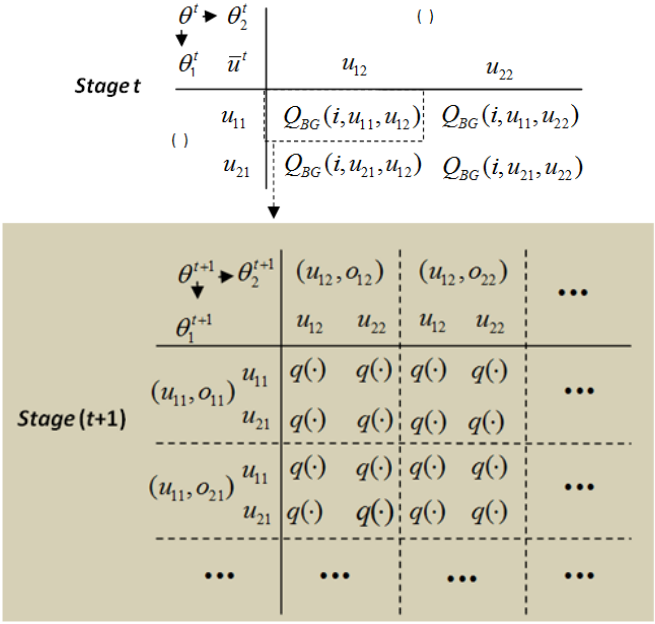
$$\beta^* = \arg\max_{\beta} \sum_{\theta^{t+1} \in \Theta} p(\theta^{t+1} | \theta^t, \bar{u}^t) q(i,\ \bar{u}^t, \theta^{t+1}, \beta(\theta^{t+1})) \tag{4}$$

Thus, in order to code control policies (as in FQL / FMG), we maintain a parameter vector q defined as $q(i, \bar{u}^t, \theta^{t+1}, \beta(\theta^{t+1}))$. These q values form the entries of the next step BG matrix, i.e., at time step $(t+1)$. Type probability $p(\theta^{t+1} | \theta^t, \bar{u}^t)\ \forall \theta^{t+1} \in \Theta$ is calculated as:

$$p(\theta^{t+1} | \theta^t, \bar{u}^t) = \sum_{s^{t+1} \in S} \sum_{s^t \in S} O(\bar{o}^{t+1} | \bar{u}^t, s^{t+1}) p(s^{t+1} | \bar{u}^t, s^t) p(\theta^t) \tag{5}$$

where $\theta^{t+1} = (\theta^t, \bar{u}^t, \bar{o}^{t+1})$.

We adapt a multiagent POMDP modeled as a sequence of BGs to fuzzy RL by using Q learning update for updating the q values at $(t+1)$ stage BG matrix (Fig. 1). At each stage, we compute optimal $Q_{BG}(\cdot)$ value (3) and optimal joint policy β^* (4) corresponding to each fuzzy rule using a procedure referred to as the *forward sweep policy computation* [29], [8]. In a multiagent POMDP the agents do not have access to the underlying state s^t but may have a belief over their local state referred to as the belief state [30].



where $q(\cdot) = q(i, \bar{u}^t, \theta^{t+1}, \beta(\theta^{t+1}))$

Fig. 1. Multiagent POMDP: Two stage Bayesian Game.

In the proposed fuzzy multiagent POMDP approach, we match each agent's belief b_g^t (belief state of agent g at time t) to fuzzy sets laid over its belief space or in other words generate a fuzzy belief. This matching leads to computation of rule firing strength ($\alpha^i(\bar{b}^t)\ \forall i \in N$) where $\bar{b}^t = [b_1^t, \dots, b_{N_a}^t]$ as: $\alpha^i(\bar{b}^t) = T(\mu_{L_1^i}(b_1^t), \dots, \mu_{L_{N_a}^i}(b_{N_a}^t))$ where the T-norm is implemented by the product $\alpha^i(\bar{b}^t) = \prod_{j=1}^{N_a} \mu_{L_j^i}(b_j^t)$. For each rule Y_i, we solve the BG at the next time step $(t+1)$ as per (3) to get $Q_{BG}(i, \bar{u}^t)$ values that form the entries of the BG matrix at stage t (Fig. 1). Optimal one-step target value for rule Y_i, i.e., $V_{BG}^i(\bar{b}^t)$ is the maximal value of the BG matrix at stage t and is computed as:

$$V_{BG}^i(\bar{b}^t) = \max_{\bar{u}^t \in \bar{U}} Q_{BG}(i, \bar{u}^t) \tag{6}$$

and optimal one-step joint action $\bar{u}_{BG}^*(i);\ i \in N$ is given by

$$\bar{u}_{BG}^*(i) = \arg\max_{\bar{u}^t \in \bar{U}} Q_{BG}(i, \bar{u}^t) \tag{7}$$

optimal one-step policy at stage t is thus

$$\pi_{BG}^* = [\bar{u}_{BG}^*(1), \dots, \bar{u}_{BG}^*(i), \dots, \bar{u}_{BG}^*(N)], \tag{8}$$

where index i corresponds to rule Y_i. In order to explore set of possible actions and to acquire experience through RL, we use a pseudo stochastic exploration/exploitation (EEP) policy [4]. In the EEP strategy, we gradually reduce the exploration parameter ε according to some schedule, e.g., halve the ε value every 50 iterations (in our case). We use an ε-BG policy (ε-greedy in Q learning) meaning that we choose a random action with probability ε:

$$\begin{aligned}\bar{u}^{\dagger}_{BG}(i) &= \varepsilon\text{−BG}\bar{u}^{*}_{BG}(i)\\ &= \begin{cases}\bar{u}^{rd}(i) \text{ with probability } \varepsilon\\ \bar{u}^{*}_{BG}(i) \text{ with probability } (1-\varepsilon)\end{cases}\end{aligned} \tag{9}$$

where $\bar{u}^{rd}$ is a random joint action, i.e., $\bar{u}^{rd} = [u^{rd}_1, \ldots, u^{rd}_{N_a}]$ and each u^{rd}_g is an action chosen uniformly at random from the action set of the agent g: U_g, and the ε-BG policy is

$$\pi^{\dagger}_{BG} = [\bar{u}^{\dagger}_{BG}(1), \ldots, \bar{u}^{\dagger}_{BG}(i), \ldots, \bar{u}^{\dagger}_{BG}(N)]. \tag{10}$$

Next, we generate global ε-BG action for each agent $u^{\dagger}_{BG}(g)$ $\forall g \in N_a$by summing the membership strength of each action in the action set of agent g and choosing the action that has the highest combined strength, i.e., one having maximum $\sum_{i\in N} \alpha^i(\bar{b}^t)$ value. Then global ε-BG joint action $\bar{u}^{\dagger}_{BG}$ is given by $\bar{u}^{\dagger}_{BG} = [u^{\dagger}_{BG}(1), \ldots, u^{\dagger}_{BG}(N_a)]$.

This joint action is then executed in a decentralized manner, i.e., each agent executes his component of $\bar{u}^{\dagger}_{BG}$ or executes $u^{\dagger}_{BG}(g)$. Under this ε-BG joint action $\bar{u}^{\dagger}_{BG}$ the environment transitions to the next state s^{t+1}. Each agent receives a local observation from the system, a global reward r_t and updates its local belief state b^{t+1}_g using the standard Bayesian update [30] as:

$$b^{t+1}_g(s^{t+1}) = \frac{O_g(o^t_g|u^t_g, s^{t+1}) \sum_{s^t\in S} p(s^{t+1}|s^t, \bar{u}^t) b^t_g(s^t)}{\sum_{s^{t+1}\in S} O_g(o^t_g|u^t_g, s^{t+1}) \sum_{s^t\in S} p(s^{t+1}|s^t, \bar{u}^t) b^t_g(s^t)} \tag{11}$$

where O_g is the observation function of agent g. Currently, we require every agent to have an independent observation model. Each agent communicates this updated belief to the fuzzy rule base for computing rule strength values $\alpha^i(\bar{b}^{t+1})$. These current q parameter estimates are used to compute global fuzzy BG target value (as in FQL [10]) as per:

$$V^{*}_{BG}(\bar{b}^{t+1}) = \frac{\sum_{i=1}^{N} V^{i}_{BG}(\bar{b}^{t+1})\alpha^i(\bar{b}^{t+1})}{\sum_{i=1}^{N} \alpha^i(\bar{b}^{t+1})}. \tag{12}$$

For each rule Y_i, one of the $Q_{BG}(i, \bar{u}^t)$ values is selected to compute global Q^{*}_{BG} value as per the ε-BG policy (9) under each rule Y_i: $Q^{*}_{BG}(i) = Q_{BG}(i, \bar{u}^{\dagger}_{BG}(i))$. Global Q^{*}_{BG} value is the weighted sum of these ε-BG (rule) values weighted by the rule firing strengths, i.e.,

$$Q^{*}_{BG}(\bar{b}^t) = \frac{\sum_{i=1}^{N} Q^{*}_{BG}(i)\alpha^i(\bar{b}^t)}{\sum_{i=1}^{N} \alpha^i(\bar{b}^t)} \tag{13}$$

We calculate TD(0) error [4] for tuning q parameter values as:

$$\Delta Q = r_t + \gamma V^{*}_{BG}(\bar{b}^{t+1}) - Q^{*}_{BG}(\bar{b}^t) \tag{14}$$

where r_t is the global reward at stage t. Finally, q parameter values are updated for all rules $i \in N$ as per:

$$\begin{aligned}q(i, \bar{u}^t, \theta^{t+1}, \beta^*(\theta^{t+1})) &\leftarrow q(i, \bar{u}^t, \theta^{t+1}, \beta^*(\theta^{t+1}))\\ &+ \eta\Delta Q \frac{\alpha^i(\bar{b}^t)}{\sum_{i=1}^{N} \alpha^i(\bar{b}^t)}; \forall\theta^{t+1} \in \Theta\end{aligned} \tag{15}$$

where $0 < \eta \leq 1$ is the learning rate parameter.

IV. Empirical Performance: Fuzzy Multiagent POMDP Control

This section describes realization of the fuzzy multiagent POMDP controller and results of evaluating the approach on a multiagent tiger problem [17]. We briefly outline the multiagent tiger problem domain.

A. Multiagent tiger

This problem has been a standard test bed for evaluating the performance of multiagent POMDP solution approaches as it is conceptually simple and yet includes all the finer intricacies of the multiagent POMDP setup. This problem was first introduced by Nair et. al. [17] and has been extensively used by several researchers for validating the performance of proposed multiagent POMDP solution approaches [3], [16], [23]. It is a modification of the single agent tiger problem introduced by Kaelbling et. al. [30].

The problem concerns two agents that are standing in a hallway with two doors. Behind one of the doors is a tiger while the other door has a treasure behind it. The task is to open the correct door to receive the treasure. The states are tiger behind left door (s_l) or tiger behind the right door (s_r). Each agent has three actions, open the left door (u_{OL}), open the right door (u_{OR}) and listen (u_{LI}). The agents can't observe each other's actions. Each agent can receive two observations: hear sound left (o_{HL}) or hear sound right (o_{HR}).

The problem starts with tiger uniformly located behind either door, i.e., state is s_l or s_r with probability 0.5. The state remains unchanged as long as no agent opens the door and resets the moment any door is opened. For detailed transition, observation and reward model, we refer the reader to [17]. When either agent opens the door that has treasure behind it gets reward however opening the door with tiger results in a penalty. Opening the wrong door simultaneously leads to lower penalty while opening correct door together gives higher reward.

B. Fuzzy Multiagent POMDP controller realization

Since the problem has only two states s_l and s_r, we can use only one probability $p(s_l)$ to specify belief as $p(s_r) = 1 - p(s_l)$. The belief space of either agent can be fully specified by a probability distribution over only s_l or belief $b = p(s_l)$. The belief space for each agent (b_1, b_2) is thus specified by $(0, 1]$. We partition the belief space of each agent into three fuzzy subsets thereby generating nine rules. Linguistic terms for these fuzzy sets are (TL, TNS, TR) where TL stands for "tiger left", TNS is "tiger not sure" and

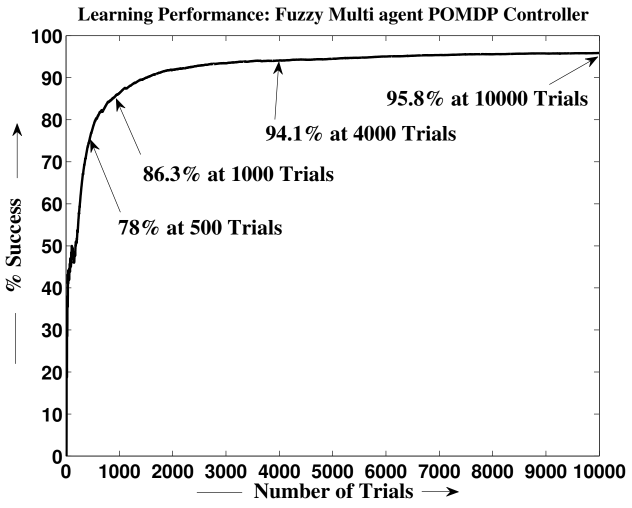


Fig. 2. Performance of fuzzy Multiagent POMDP controller.

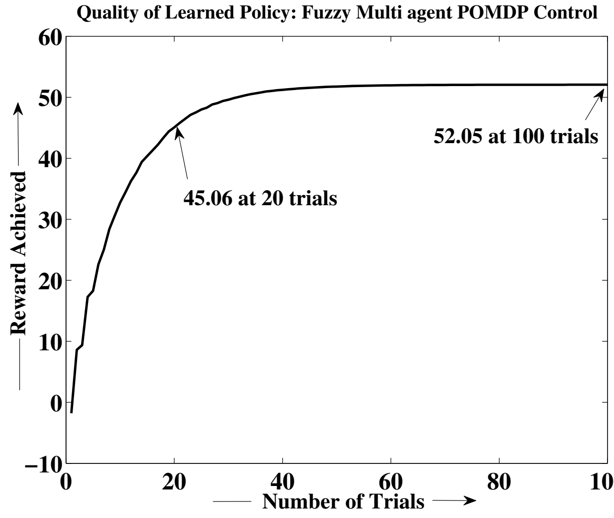


Fig. 3. Control quality of proposed fuzzy multiagent POMDP controller, showing the expected infinite horizon discounted reward.

TR stands for "tiger right". The membership function for each belief state variable (b_1, b_2) is the standard Gaussian membership function defined by:

$$\mu_l(b_j) = e^{\frac{-(b_j - b_j^l)^2}{2(\sigma_j^l)^2}}; \; l = 1, 2, 3; \quad j = 1, 2 \qquad (16)$$

where l are fuzzy labels for each variable $b_j (b_j = p_j(s_l))$ being the belief variable for agent j and $p_j(s_l)$ is the probability distribution over s_l corresponding to agent j. Fuzzy label centers are defined as $b_j^l = c_j + d_j(l-1)$ with $c_1 = c_2 = 0$, $d_1 = d_2 = 0.5$ and widths defined by $\sigma_j^1 = \sigma_j^3 = 0.15$, $\sigma_j^2 = 0.2$. The RL parameters being, discount factor $\gamma = 0.9$, learning rate $\eta = 0.8$. Exploration parameter ε is initialized from 0.8 and halved every 50 iterations.

We refer to the time instant when either agent opens a door (generating a global reward, r_t) as the end of a trial. When either agent opens the correct door a positive reward is received and the trial is termed successful. Opening wrong door results in penalty and the trial is referred to as a failure. If both agents listen then trial continues. We initialize q values as small random numbers between 0 and 1. Figure 2 shows learning performance of the fuzzy multiagent POMDP controller, i.e., shows % success as a function of number of trials.

We have averaged results over 100 episodes each of $10,000$ trials. The controller achieved a significant level of performance (78% success) in just about 500 trials. The performance reached 94.1% at 4000 trials and peaked to 95.8% success rate at the end of $10,000$ trials. It is to be noted that initially agents fail to open the correct door as (i) they are learning and q values are getting updated (ii) use of the EEP policy leads to random agent actions during the initial phase of the learning process. A 95.8% success, therefore, represents significant performance level as this includes initial failures as well. In fact, agents learn to open the correct door with 95% success in about 1000 trials.

TABLE I
COMPARISON OF MULTIAGENT POMDP CONTROLLERS

Algorithm/Approach	Expected discounted reward		
	$T=6$	$T=8$	$T=\infty$
Fuzzy multiagent POMDP	23.92	31.97	52.05
Free communication [15][16]	11.95	17	
DEC-COMM [16]	9.35		
Finite state controller [18]			5.2

We also plot infinite horizon discounted reward achieved by the fuzzy multiagent POMDP approach (Fig. 3). This value is a figure of merit for the quality of the learned policy, i.e., how much reward agents accumulate while following the discovered policy. We have averaged the results over 1000 episodes of 100 trials each. As can be seen, the value achieved is 45.06 in 20 trials and the final expected discounted infinite horizon value obtained is 52.05.

For effectively comparing performance of the proposed approach against other recent multiagent solution approaches, we applied multiagent POMDP control for different horizons, i.e., $T = 6$, $T = 8$, and infinite horizon ($T = \infty$). Table 1 gives a comparison of reward achieved by fuzzy multiagent POMDP against (i) centralized POMDP control with free communication, (ii) Dec-COMM (M. Roth et. al. [15], [16]) and (iii) Finite state controller (Amato et. al. [18]).

From Table 1 it can be observed that fuzzy multiagent framework leads to significantly higher performance levels. This may be attributable to a centralized rule generation (centralized joint policy) due to instantaneous communication leading to an almost perfect inter-agent cooperation [15]. Use of FIS allows agents to quickly and stably zero-in on the optimal joint policy. Realization of such high performance is an indicator of the quality of learned policy.

V. Conclusions and Scope for future work

This paper presents a novel fuzzy reinforcement learning control scheme for multiagent POMDPs in a game-theoretic setting. We introduce fuzzy RL in the multiagent POMDP framework to successfully address time and space complexity issues. It shows how FIS based function approximation can be used as a principled means to represent continuous multiagent belief space by what may be termed as fuzzy multiagent POMDPs. To the best of our knowledge, this work represents a first attempt in applying game theoretic learning to an FIS based multiagent POMDP setup. We elucidate feasibility of the proposed approach with empirical results on the benchmark multiagent tiger problem. The proposed fuzzy multiagent POMDP controller successfully discovers a high quality optimal policy solution in reasonable number of trials. Further, a comparison of the expected discounted reward value (figure of merit for controller evaluation) achieved against other very recent state-of-the-art multiagent POMDP solution approaches ([15], [16], [18]) brings out the effectiveness of the proposed approach.

Future work would involve testing the proposed approach on other benchmark problem domains such as meeting in a grid, box pushing [3], and real dynamical systems involving multiple robots, e.g., multi robot urban search and rescue operation. Another interesting future research direction could be adapting the approach to the partially observable stochastic game setup wherein each agent has an individual reward function. In our view, proposed fuzzy multiagent POMDP control is a promising new research direction that could hold the key for addressing the scalability and tractability of infinite horizon multiagent POMDPs.

Acknowledgments

This work was funded by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the PIDDAC Program funds and was supported by project PTDC/EEA-ACR/73266/2006.

References

[1] G. Strang, *Linear Algebra and its applications*. Academic Press, Orlando, FL, 1980.

[2] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

[3] S. Seuken and S. Zilberstein, "Formal Models and Algorithms for Decentralized Decision Making under Uncertainty," *Autonomous Agents and Multi-Agent Systems*, vol. 17(2), pp. 190-250, October, 2008.

[4] R. S. Sutton, and A.G. Barto, *Reinforcement learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[5] R. Sharma, and M. Gopal, "Reinforcement learning based game-theoretic neural controller," in *Proc. 2006 American Control Conference*, Minneapolis (USA), pp. 2975-2980, June 14-16, 2006.

[6] Jennie Si, A. G. Barto, W. B. Powell, and D. Wunsch, *Handbook of Learning and Approximate Dynamic Programming*. Willey-IEEE Press, August 2004.

[7] C. J. C. H. Watkins, and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279-292, 1992.

[8] F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis, "Optimal and Approximate Q-value functions for Decentralized POMDPs," *Journal of Artificial Intelligence Research*, vol. 32, pp. 289–353, 2008.

[9] L. X. Wang, and J.M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Transactions on Neural Networks*, vol. 3 , pp. 807–814, 1992.

[10] L. Jouffe, "Fuzzy inference system learning by reinforcement methods," *IEEE Transactions on Systems, Man, and Cybernetics —Part C: Applications and Reviews*, vol. 28(3), pp. 338-355, August 1998.

[11] R. Sharma, and M. Gopal, "A Markov Game Adaptive Fuzzy Controller for Robot Manipulators," *IEEE Transactions on Fuzzy Systems*, vol. 16, issue 1, pp. 171 – 186, Feb. 2008.

[12] D. S. Bernstein, S. Zilberstein, and N. Immerman, "The complexity of decentralized control of Markov decision processes," In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 32–37, Stanford, California, June 2000.

[13] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Mathematics of Operations Research*, vol. 27(4), pp. 819–840, 2002.

[14] F. A. Oliehoek, J. F. P. Kooij, and N. Vlassis, "The Cross-Entropy Method for Policy Search in Decentralized POMDPs," *Informatica* vol. 32, pp. 341–357, 2008.

[15] M. Roth, R. Simmons, and M. Veloso, "Decentralized Communication Strategies for Coordinated Multi-Agent Policies," In *Multi- Robot Systems: From Swarms to Intelligent Automata,* vol. *3* , 2005.

[16] M. Roth, R. Simmons, and M. Veloso, "What to Communicate? Execution-Time Decision in Multi-agent POMDPs," In *Proceedings of DARS-2006*, Minneapolis, MN, July 2006.

[17] R. Nair, M. Tambe, M. Yokoo, D. V. Pynadath, and S. Marsella , "Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings," In *Proc. of the International Joint Conference on Artificial Intelligence*, pp. 705–711, Acapulco, Mexico, August 2003.

[18] C. Amato and S. Zilberstein, "Achieving Goals in Decentralized POMDPs," In *Proceedings of the Eighth International Conference on Autonomous Systems and Multiagent Systems (AAMAS)*, pp. 593-600, Budapest, Hungary, 2009.

[19] D. V. Pynadath, and M. Tambe, "The communicative multiagent team decision problem: Analyzing teamwork theories and models," *Journal of Artificial Intelligence Research (JAIR),* vol. 16, pp. 389–423, 2002.

[20] L. Peshkin, K. E. Kim, N. Meuleau, and L. P. Kaelbling, "Learning to Cooperate via Policy Search," *Proceedings of the Sixteenth International Conference on Uncertainty in Artificial Intelligence,* 2000.

[21] P. J. Gmytrasiewicz, and P. Doshi, "A framework for sequential planning in multiagent settings," *Journal of Artificial Intelligence Research*, vol. 24, pp. 49–79, 2005.

[22] D. S. Bernstein, E. A. Hansen, and S. Zilberstein, "Bounded policy iteration for decentralized POMDPs," In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI),* pp. 1287–1292, Edinburgh, Scotland, July 2005.

[23] C. Amato, D. Bernstein, and S. Zilberstein, "Optimizing memory-bounded controllers or decentralized POMDP," In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*,Vancouver, Canada, July 2007.

[24] R. Cogill, M. Rotkowitz, B. van Roy, and S. Lall, "An approximate dynamic programming approach to decentralized control of stochastic systems," In *Proceedings of the Allerton Conference on Communication, Control, and Computing* , pp. 1040–1049, Urbana- Champaign, IL, 2004.

[25] E. Duman, M. Kaya, and E. Akin, "A multiagent fuzzy reinforcement learning method for continuous domains," *LNCS*, Springer, Berlin, pp. 306-315, 2005.

[26] A. M. Tehrani, M. S. Kamel, and A. M. Khamis, "Fuzzy reinforcement learning for embedded soccer agents in a multiagent context," *International Journal of Robotics and Automation*, vol. 21(2), pp. 110-119, April 2006.

[27] T. Karadoniz, and L. Akin "FDMS with Q learning: A neuro fuzzy approach to POMDPs," *International Journal of Advanced Robotic systems*, vol. 1, no. 3, pp. 251-262, 2004.

[28] G. Owen, in: 2nd Edition, *Game Theory*. Academic Press, Orlando, FL, 1982.

[29] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun, "Approximate solutions for partially observable stochastic games with common payoffs," In *Proc. of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 136–143, 2004.

[30] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101(1-2), pp. 99–134, 1998.