

Fuzzy Reinforcement Learning Control for Decentralized Partially Observable Markov Decision Processes

Rajneesh Sharma
Instrumentation and Control Division
Netaji Subhas Institute of Technology
New Delhi, India
rajneesh496@rediffmail.com

Matthijs T. J. Spaan
Institute for Systems and Robotics
Instituto Superior Tecnico
Lisbon, Portugal
mtjspaan@isr.ist.utl.pt

Abstract—Decentralized Partially Observable Markov Decision Processes (Dec-POMDPs) offer a powerful platform for optimizing sequential decision making in partially observable stochastic environments. However, finding optimal solutions for Dec-POMDPs is known to be intractable, necessitating approximate/suboptimal approaches. To address this problem, this work proposes a novel fuzzy reinforcement learning (RL) based game theoretic controller for Dec-POMDPs. The proposed controller implements fuzzy RL on Dec-POMDPs, which are modeled as a sequence of Bayesian games (BG). The main contributions of the work are the introduction of a game based RL paradigm in a Dec-POMDP settings, and the use of fuzzy inference systems to effectively generalize the underlying belief space. We apply the proposed technique on two benchmark problems and compare results against state-of-the-art Dec-POMDP control approach. The results validate the feasibility and effectiveness of using game theoretic RL based fuzzy control for addressing intractability of Dec-POMDPs, thus opening up a new research direction.

Keywords– Reinforcement learning, Fuzzy systems, Cooperative multiagent systems, Decentralized POMDPs

I. INTRODUCTION

Finding optimal strategies for multiple agents operating in an uncertain environment is an active field of research in Artificial Intelligence. Sequential decision making in fully observable stochastic environments has been tackled effectively via the well known Markov decision process (MDP) framework [1]. Sensor uncertainty can be addressed through partially observable Markov decision process (POMDP) framework [2] that extends the MDP model by incorporating observations and their probability of occurrence conditional on the state of the environment. Our focus herein is on learning planning and control for multiple agents in uncertain environments in a cooperative setting. This is formalized by the Decentralized POMDP (Dec-POMDP) model [3] which essentially generalizes a POMDP to multiple agents and can be used to model a team of cooperative agents. In a Dec-POMDP, agents have to cope with uncertainty regarding other agents in addition to imperfect information about the underlying state, making the task computationally intractable (NEXP-complete for the finite horizon case) [3, 4].

Reinforcement learning (RL) has now established itself as a major technique for direct adaptive optimal control of uncertain systems [5]. In the RL paradigm, agents learn to behave optimally by repeated trial and error interactions with the environment. A recent review of multiagent RL can be found in [6] with comprehensive details of such techniques for fully cooperative, fully competitive and mixed tasks. However, these methods typically assume full state observability and/or knowledge of the complete joint action at each time step [7]. In contrast, we propose and study one of the first RL approaches to Decentralized POMDPs.

Our motivation for the proposed approach is inspired by a recently proposed Q value function for Dec-POMDPs Q_{BG} , based on joint action-observation histories [8]. This Q value formulation allows us to extend RL based approximation to Dec-POMDPs akin to the MDPs for which Q learning [9] has been successfully applied for a variety of systems and processes [10]. Various approximate Q value formulations for Dec-POMDPs have been analyzed [8], which are significantly easier to compute than the optimal Q value function Q^* . These approximate Q value functions typically employ a joint belief over the state space of the underlying POMDP, which can be computed assuming agents share observations, but not in the decentralized setting.

In the proposed approach each agent g maintains tunable q^g values over a continuous global belief space of the underlying POMDP. We use fuzzy functions as a generic function approximator, which are capable of approximating any real function to an arbitrary accuracy [11]. Fuzzy systems offer an elegant way of representing Dec-POMDP policies in a compact form, i.e., with a finite amount of memory. While convergence guarantees exist for tabular Q-learning [9], the same guarantees do not hold when function approximation is introduced [10]. As has been the case with other approaches that have used function approximation for Q-learning [10, 12] or the Markov game based learning [13], theoretical convergence of the learning process with function approximation cannot be guaranteed.

To the best of our knowledge this work is the first study of applying fuzzy RL for generating a learning controller for

Dec-POMDPs in a game theoretic setting. Major contributions of our work being (i) Casting Dec-POMDPs as a game theoretic RL task (ii) Introducing Fuzzy Inference System (FIS) based generalization to Dec-POMDPs (pointed out by Seuken and Zilberstein [14] as a major need to improve scalability of Dec-POMDP solution techniques). We use RL to learn and progressively improve solution to the Dec-POMDP using FIS to effectively generalize over a continuous global belief space.

We empirically evaluate the proposed fuzzy Dec-POMDP control on the benchmark recycling robot [15] and the multi access broadcast channel [16] tasks and compare its performance against other state of the art infinite horizon Dec-POMDP control approach [15, 16, 17]. Results validate the feasibility of using a game theoretic RL controller for Dec-POMDPs, opening up a promising future research direction to improve scalability and tackle computational intractability being faced by majority of the current Dec-POMDP solution techniques.

The rest of the paper is organized as follows: Section II briefly sheds some light on background and related work, Section III describes the proposed fuzzy Dec-POMDP controller, Section IV presents empirical results on the recycling robot and multi access broadcast channel tasks, and Section V concludes the paper with a discussion on the future scope of the proposed approach.

II. BACKGROUND AND RELATED WORK

We assume prior reader familiarity with the basic concepts underlying Reinforcement Learning [5], POMDPs [2] and Fuzzy Systems [18] giving only a brief overview of the concepts that form the backbone of our work.

A. Dec-POMDPs

Multiagent sequential decision making problems under partial observability wherein agents cooperate to optimize joint performance are formalized in the Dec-POMDP model [3]. A Dec-POMDP models a number of agents that interact with their environment (system) at discrete time steps $t = 1, 2, \dots$. 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. The agents optimize a form of joint long term reward. The framework is decentralized as no agent can control the whole process and none of the agent has access to the global state. Further, agents know only their own actions and observations and there is no communication between the agents.

In this paper, we define a Dec-POMDP as the following tuple $\langle N_a, S, \{U_g\}, P, \{\Omega_g\}, O, R, T \rangle$ 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 $\bar{u} = \langle u_1, \dots, u_{N_a} \rangle$ 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 $\bar{o} = \langle o_1, \dots, o_{N_a} \rangle$ being 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 observing joint observation $\bar{o}$ given that joint action $\bar{u}$ was taken and led to state s' . $R : \bar{U} \times S \rightarrow \Re$ 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 $\sum_{t=0}^{\infty} \gamma^t r_t$ (with r_t being the reward received at time step t), which is what the agents optimize.

B. Dec-POMDPs as a sequence of Identical Payoff Bayesian games

A Bayesian game (BG) is an augmented normal form game in which players have some private information [19]. 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 $\langle N_a, \bar{U}, \Theta, P(\Theta), \{r^1, \dots, r^{N_a}\} \rangle$ 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 $P(\Theta)$ is defined, and $R : \Theta \times \bar{U} \rightarrow \Re$ is the payoff function.

In a BG, agents can condition their action on the private information they have. A joint policy in a BG is represented by $\beta = \langle \beta_1, \dots, \beta_{N_a} \rangle$ 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 [19]:

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

where $\beta(\theta) = \langle \beta_1(\theta), \dots, \beta_{N_a}(\theta) \rangle$ 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.

In modeling Dec-POMDPs as a sequence of BGs [20] the payoff function of BG (one stage) for an agent g is $Q_{BG}^g(\bar{\theta}', \bar{u})$ where $\bar{\theta}' = (\bar{o}^0, \bar{u}^0, \bar{o}^1, \dots, \bar{o}^{t-1}, \bar{u}^{t-1}, \bar{o}^t)$ is the joint action-observation history up to time t , where initial joint observation $\bar{o}^0$ is assumed to be an empty observation: $\bar{o}^0 = \bar{o}_{\emptyset} = \langle o_{1,\emptyset}, o_{2,\emptyset}, \dots, o_{N_a,\emptyset} \rangle$. Oliehoek et. al. [8], 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.

C. Overview of Dec-POMDP methods

As this work concerns designing controllers for infinite horizon Dec-POMDPs, we limit this brief overview to infinite horizon Dec-POMDPs. For finite horizon counterpart we refer the interested reader to [14]. For tackling infinite horizon Dec-POMDPs most prominent and by far most successful approach envisages Dec-POMDP policy representation as a finite state controller [15]. Finite state controllers offer an elegant way of representing a Dec-POMDP policy using finite amount of memory. Finite state controllers do not store whole observation histories, instead they encapsulate key pieces of information concerning observation history with a fixed number of nodes.

The first approach by Szer and Charpillet [16] requires high memory and running time and is limited as it searches for optimal deterministic controllers. The second one proposed by Bernstein et. al. called Bounded Policy Iteration for Dec-POMDPs (Dec-BPI) [17] is a multiagent extension of the BPI algorithm [21]. Dec-BPI uses linear programming for iterating through the nodes of each agents controller and attempts to find an improvement. The methodology is promising as it offers scalability. The last one, a recent approach by Amato et al. [15] formulates fixed size controller solution as a nonlinear program (NLP) and exploits nonlinear optimization techniques to solve the problem. They also use a certain ‘correlation device’ which is a shared source of randomness to improve the solution quality.

Yagan and Tham [22] propose an RL solution to approximate the joint optimal policies for Dec-POMDPs by exploiting locality of interaction among agents. Effectiveness of the proposed algorithm is shown on multiagent wireless multihop networks and for cooperative retransmission in wireless networks [23]. However, the algorithm is not applicable to general Dec-POMDPs as it assumes that neighboring agents can interact and exploits the broadcast nature of wireless medium for effective retransmission of a failed packet (a domain specific feature). Recently, Wu et al. [24] propose a novel decentralized rollout sampling policy iteration algorithm for Dec-POMDPs based on a RL technique called Monte Carlo methods. The algorithm samples a set of reachable belief states based on some heuristic policies and computes a joint policy for each belief state based on rollout estimations. Algorithm claims linear time complexity over no. of agents, bounded memory usage and good solution quality.

D. 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 [12], 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 Sharma and Gopal [13] 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., Duman et. al. [25] use fuzzy RL on a multiagent continuous pursuit domain. Tehrani et. al. [26] implement fuzzy RL on robotic soccer agents to enable them to coordinate their behavior locally and socially while learning from experience. To the best of our knowledge, until now there have been very few attempts at using fuzzy RL for effective learning in POMDPs and none for Dec-POMDPs.

III. FUZZY REINFORCEMENT LEARNING FOR DEC-POMDPs

In this paper, we propose a novel reinforcement learning approach for Dec-POMDPs. Given the overwhelming complexity of solving Dec-POMDPs, we propose to use a fuzzy RL based approach to learn controllers. Fuzzy membership functions are laid over the global belief space for each agent to generate what may be termed as a ‘Fuzzy Dec-POMDP’. Each agent maintains its tunable parameters or the fuzzy q^g values (agent g). The use of Fuzzy Inference System (FIS) for mapping the continuous belief space of the underlying Dec-POMDP allows us to define and tune the q^g values. The Dec-POMDP is modeled as a two stage Bayesian game [19], the matrix entries for each stage being the fuzzy Q_{BG}^g values [8] obtained from the fuzzy Dec-POMDP rule base.

A. Acting in two-stage BGs

As our proposed approach uses a fuzzy RL framework, we define Q_{BG}^g at stage t as in [8] for a two stage BG corresponding to agent g . Crucially, in such a BG, (joint) types correspond to one (joint) actions and following by a (joint) observation. We define the Q function at stage t to be constructed from fuzzy q^g parameters (for agent g) defined at stage $(t+1)$: $q^g(i, \bar{u}^t, \bar{o}^{t+1}, \beta(\bar{o}^{t+1}))$ where i is an index specifying the fuzzy

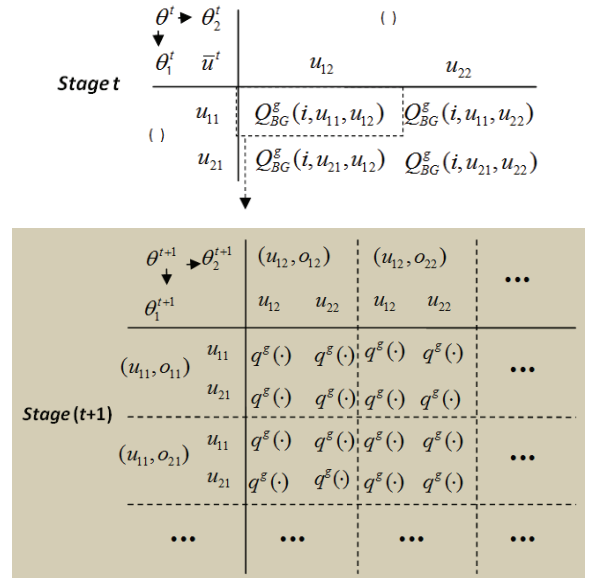


Figure 1 Fuzzy Dec-POMDP: Two stage Bayesian Game, where $q^g(\cdot) = q^g(i, \bar{u}^t, \bar{o}^{t+1}, \beta(\bar{o}^{t+1}))$

rule Y_i , $\bar{u}^t$ is the joint action at stage t , $\bar{o}^{t+1}$ is the joint observation at stage $(t+1)$, and $\beta(\bar{o}^{t+1})$ is the joint BG policy corresponding to $\bar{o}^{t+1}$ at stage $(t+1)$. This conforms to the modeling of Dec-POMDPs as a series of BGs as introduced in [8].

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

$$Q_{BG}^g(i, \bar{u}^t) = \max_{\beta} \sum_{\bar{o}^{t+1} \in O} p(\bar{o}^{t+1} | \theta^t, \bar{u}^t) q^g(i, \bar{u}^t, \bar{o}^{t+1}, \beta(\bar{o}^{t+1})) \quad (2)$$

The maximizing BG policy at next stage $(t+1)$ is given by

$$\beta_g^* = \arg \max_{\beta} \sum_{\bar{o}^{t+1} \in O} p(\bar{o}^{t+1} | \theta^t, \bar{u}^t) q^g(i, \bar{u}^t, \bar{o}^{t+1}, \beta(\bar{o}^{t+1})) \quad (3)$$

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

$$p(\bar{o}^{t+1} | \theta^t, \bar{u}^t) = \sum_{s^{t+1} \in S} \sum_{s^t \in S} O(\bar{o}^{t+1} | \bar{u}^t, s^{t+1}) p(s^{t+1} | \bar{u}^t, s^t) p(\theta^t) \quad (4)$$

It is to be noted that for updating type probability (4) the agent must have access to $\bar{o}^{t+1}$ and $\bar{u}^t$ (the joint observation and joint action, respectively) which are not available in standard Dec-POMDP setup. An agent g has access only to its own action (u_g^t) and observation (o_g^{t+1}) and the global reward signal (r_t) provided by the environment.

We propose to overcome this bottleneck by a judicious use of the information contained in the reward signal. Agent g samples action of other agents using the reward signal to get approximate $\bar{u}_{\neq g}^t \cong \hat{u}_{\neq g}^t$ ($\bar{u}_{\neq g}^t$ is the joint action of all agents except agent g). The sampled $\hat{u}_{\neq g}^t$ is combined with u_g^t to yield $\bar{u}^t = \langle \hat{u}_{\neq g}^t, u_g^t \rangle$. We then update type probabilities for all $\bar{o}^{t+1} \in O$ using (4). In contrast to [8], for instance, the proposed approach is an adaptive learning one based on the reinforcement learning technique. All RL techniques use the reinforcement signal (reward here) as the primary means to judge the goodness of taking a particular action. This enables the agent to discriminate amongst different actions and choose the one that produces a higher expected discounted reward [5].

Summarizing, we adapt a Dec-POMDP modeled as a sequence of BGs to fuzzy RL by using Q learning update for

updating the q^g values at $(t+1)$ stage BG matrix (Fig. 1). At each stage, we compute optimal $Q_{BG}^g(\cdot)$ value (2) and optimal joint policy β_g^* (3) corresponding to each fuzzy rule using a procedure referred to as the *forward sweep policy computation* [3].

B. Fuzzy Dec-POMDPs

In a POMDP an agent cannot observe state but it maintains a probability distribution over the state space called belief [2]. In a Dec-POMDP, a belief can typically not be computed as each agent only observes its own part of the joint observation. In the proposed fuzzy Dec-POMDP approach, we match an agent's perception of the global belief b_g^t (global belief state at time t) to fuzzy sets laid over the global belief space or in other words generate a fuzzy belief. We define fuzzy inference system for fuzzy Dec-POMDPs as consisting of a set of N rules (one set of rules for each agent) of the following form:

$$\begin{aligned} Y_i : & \text{ If } b_{g1}^t \text{ is } L_1^i \text{ and } \dots \text{ and } b_{gn}^t \text{ is } L_n^i \\ & \text{ then } u_1 = u_{11} \text{ and } \dots \text{ and } u_{N_a} = u_{1N_a} \text{ with } Q_{BG}^g(i, u_{11}, \dots, u_{1N_a}) \\ & \text{ or } u_1 = u_{21} \text{ and } \dots \text{ and } u_{N_a} = u_{2N_a} \text{ with } Q_{BG}^g(i, u_{21}, \dots, u_{2N_a}) \\ & \dots\dots\dots \\ & \text{ or } u_1 = u_{m1} \text{ and } \dots \text{ and } u_{N_a} = u_{mN_a} \text{ with } Q_{BG}^g(i, u_{m1}, \dots, u_{mN_a}) \end{aligned} \quad (5)$$

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 k th action in U_g or k th action of agent g , $b_g^t = [b_{g1}^t, \dots, b_{gs}^t, \dots, b_{gn}^t]$ is the global belief vector as perceived by agent g at time t , b_{gs}^t being the belief of agent g for state variable $s \in n$ (n being the number of state variables) at time t , L_s^i is the linguistic term (fuzzy label) of input variable b_{gs}^t in rule Y_i and its membership function denoted by $\mu_{L_s^i}$ corresponding to the state variable s . This matching leads to computation of rule firing strength ($\alpha^i(b_g^t) \forall i \in N$) as:

$$\alpha^i(b_g^t) = T(\mu_{L_1^i}(b_{g1}^t), \dots, \mu_{L_n^i}(b_{gn}^t)) \quad (6)$$

where the T-norm is implemented by the product

$$\alpha^i(b_g^t) = \prod_{j=1}^n \mu_{L_j^i}(b_{gj}^t) \quad (7)$$

For each rule Y_i , we solve the BG at the next time step $(t+1)$ as per (2) to get $Q_{BG}^g(i, \bar{u}^t)$ 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., $V_{BG}^i(b_g^t)$ is the maximal value of the BG matrix at

stage t corresponding to agent g and rule Y_i and is computed as:

$$V_{BG}^i(b_g^t) = \max_{\bar{u}^t \in \bar{U}} Q_{BG}^g(i, \bar{u}^t) \quad (8)$$

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

$$\bar{u}_g^*(i) = \arg \max_{\bar{u}^t \in \bar{U}} Q_{BG}^g(i, \bar{u}^t) \quad (9)$$

optimal one-step policy for agent g at stage t is thus

$$\pi_g^* = [\bar{u}_g^*(1), \dots, \bar{u}_g^*(i), \dots, \bar{u}_g^*(N)] \quad (10)$$

index i corresponds to rule Y_i .

C. Fuzzy RL for Dec-POMDPs

In order to explore set of possible actions and to acquire experience through RL, we use pseudo stochastic exploration/exploitation (EEP) policy [10]. In the EEP strategy, we gradually reduce the exploration parameter $\mathcal{E}$ according to some schedule, e.g., halve the $\mathcal{E}$ value every 50 iterations (in our case). We use an $\mathcal{E}$ -BG policy ($\mathcal{E}$ -greedy in Q learning) meaning that we choose a random action with probability $\mathcal{E}$:

$$\begin{aligned} \bar{u}_g^\dagger(i) &= \mathcal{E}\text{-BG } \bar{u}_g^*(i) \\ &= \begin{cases} \bar{u}^{rd}(i) & \text{with probability } \mathcal{E} \\ \bar{u}_g^*(i) & \text{with probability } (1-\mathcal{E}) \end{cases} \end{aligned} \quad (11)$$

$\bar{u}^{rd}$ is a random action, i.e., $\bar{u}^{rd} = [u_1^{rd}, \dots, u_{N_a}^{rd}]$ where each u_g^{rd} is an action chosen uniformly at random from the action set of the agent g : U_g , and the $\mathcal{E}$ -BG policy is

$$\pi_g^\dagger = [\bar{u}_g^\dagger(1), \dots, \bar{u}_g^\dagger(i), \dots, \bar{u}_g^\dagger(N)] \quad (12)$$

Next, we generate global $\mathcal{E}$ -BG action (agent g), $u_g^\dagger$ by summing the membership strength of each action in the action set of agent g under each rule and choosing the action that has the highest combined strength, i.e., one having maximum $\sum_{i \in N} \alpha^i(b_g^t)$ value. This procedure of computing $u_g^\dagger$ is repeated for each agent $g \in N_a$. The global $\mathcal{E}$ -BG joint action is then given by $\bar{u}^\dagger = [u_1^\dagger, u_2^\dagger, \dots, u_{N_a}^\dagger]$.

The computed joint action is then executed in a decentralized manner, i.e., each agent executes his component of $\bar{u}^\dagger$ or executes $u_g^\dagger$. Under this $\mathcal{E}$ -BG joint action $\bar{u}^\dagger$ 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 perception of global belief state b_g^{t+1} using the standard Bayesian update [2] which has been adapted herein to cope with nonavailability of joint observation ($\bar{o}^t$). In particular, we loop over all possible joint observation sets

given the local observation of agent g (o_g^t), by defining the joint observation as $\bar{o}^t = \langle o_g^t, \bar{o}_{\neq g}^t \rangle$:

$$b_g^{t+1}(s^{t+1}) = \frac{\sum_{\bar{o}_{\neq g} \in O_{\neq g}} O(\bar{o}^t | \bar{u}^t, s^{t+1}) \sum_{s' \in S} p(s^{t+1} | s', \bar{u}^t) b_g^t(s')}{\sum_{s^{t+1} \in S} \sum_{\bar{o}_{\neq g} \in O_{\neq g}} O(\bar{o}^t | \bar{u}^t, s^{t+1}) \sum_{s' \in S} p(s^{t+1} | s', \bar{u}^t) b_g^t(s')} \quad (13)$$

where $\bar{o}_{\neq g}^t$ is the joint observation of all agents except agent g and $O_{\neq g}$ being observation functions of respective agents barring agent g . Each agent uses this updated belief in conjunction with the fuzzy rule base for computing rule strength values $\alpha^i(b_g^{t+1})$. The current q^g parameter estimates are used to compute global fuzzy BG target value (as in FQL [12]) as per:

$$V_{BG}^*(b_g^{t+1}) = \frac{\sum_{i=1}^N V_{BG}^i(b_g^{t+1}) \alpha^i(b_g^{t+1})}{\sum_{i=1}^N \alpha^i(b_g^{t+1})} \quad (14)$$

Global Q_{BG}^{g*} value

For each rule Y_i , one of the $Q_{BG}^g(i, \bar{u}^t)$ values is selected to compute global Q_{BG}^{g*} value as per the $\mathcal{E}$ -BG policy (11) under each rule Y_i : $Q_{BG}^{g*}(i) = Q_{BG}^g(i, \bar{u}_g^\dagger(i))$. Global Q_{BG}^{g*} value is the weighted sum of these $\mathcal{E}$ -BG (rule) values weighted by the rule firing strengths, i.e.,

$$Q_{BG}^{g*}(b_g^t) = \frac{\sum_{i=1}^N Q_{BG}^{g*}(i) \alpha^i(b_g^t)}{\sum_{i=1}^N \alpha^i(b_g^t)} \quad (15)$$

q^g update

We calculate TD(0) error [10] for tuning q^g parameter values as:

$$\Delta Q^g = r_t + \gamma V_{BG}^*(b_g^{t+1}) - Q_{BG}^{g*}(b_g^t) \quad (16)$$

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

$$\begin{aligned} q^g(i, \bar{u}^t, \theta^{t+1}, \beta^*(\theta^{t+1})) &\leftarrow q^g(i, \bar{u}^t, \theta^{t+1}, \beta^*(\theta^{t+1})) \\ &+ \eta \Delta Q^g \frac{\alpha^i(b_g^t)}{\sum_{i=1}^N \alpha^i(b_g^t)}; \forall \theta^{t+1} \in \Theta \end{aligned} \quad (17)$$

where $0 < \eta \leq 1$ is the learning rate parameter. Every agent uses (17) to independently update its q^g parameter values at

the end of each iteration.

IV. EMPIRICAL RESULTS

This section describes the realization of the fuzzy Dec-POMDP controller and results of evaluating the approach on the recycling robot [15] and multiagent broadcast channel domains [16], which are two benchmark problems typically employed for comparing infinite-horizon Dec-POMDP methods. The first one involves robots that have the task of picking up cans in an office building, and has 4 states, 3 actions, and 2 observations. Each robot has three actions (i) search for a small can, (ii) search for a large can, or (iii) recharge the battery. The larger can be picked up only if both agents cooperate to pick it up at the same time. The robots have battery status of high and low, with an increased likelihood of transitioning to the low state or exhausting the battery after attempting to pick up the larger can.

The second problem considers two agents which need to share a communication channel, the Multiagent Broadcast Channel Access (MABC) problem. Each agent has a message buffer and it has to decide whether to send or not send the message through the channel. Only one message can go through the channel at one time, simultaneous sending results in a collision with the messages remaining in the buffers. We simulate the two agent case [16] with 4 states, 2 actions and 2 observations. New messages arrive with a rate of $p_1 = 0.9$ for agent 1 and $p_2 = 0.1$ for agent 2.

A. Recycling Robot

For the recycling robot problem, local state of either agent is battery low (BL) or battery high (BH). Global states being (BL,BL), (BL,BH),(BH,BL) and (BH,BH) or 4 states. We partition the belief corresponding to each state variable into three fuzzy subsets thereby generating $3^4 = 81$ fuzzy rules that map the global belief space to a fuzzy belief space. Linguistic terms for these fuzzy sets are Low(L), Medium(M), and High(H). The membership function for each fuzzy belief state variable (b_1, b_2, b_3, b_4) that correspond to belief over the state variables (s_1, s_2, s_3, s_4) are standard Gaussian membership functions defined by:

$$\mu_l(b_j) = e^{\frac{-(b_j - b_j^l)^2}{2(\sigma_j^l)^2}}; l = 1, 2, 3; j = 1, 2, 3, 4 \quad (18)$$

where l are fuzzy labels for each variable b_j ($b_j = p(s_j)$) being the belief variable for the state variable s_j , $p_j(s_i)$ is the probability distribution over s_j . Fuzzy label centers are defined as $b_j^l = c_j + d_j(l-1)$ with $c_{1,...,4} = 0, d_{1,...,4} = 0.5$ 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 $\eta = 0.9$. Exploration parameter ϵ is initialized from 0.9 and halved every 50 iterations. We refer to the time instant when one or both agents independently or jointly pick up a can generating a successful if the agents cooperate to lift the bigger can or pick

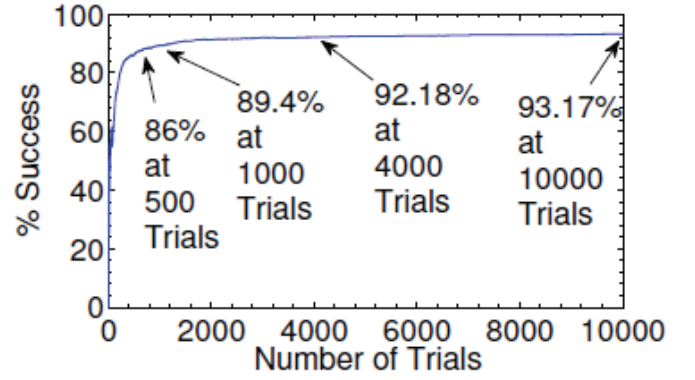


Figure 2 Fuzzy Dec-POMDP learning performance: Recycling Robot

global reward r_i as end of a trial. We consider the trial to be up smaller cans independently generating large reward. In all other cases the trial is considered a failure.

The figures 2 and 4 depict learning performance of the RL controller or how the performance improves as the controller attempts to discover the optimal policy by observing failures and rewards (while exploring). However, the figures 3 and 5 are about the performance of the learned policy, i.e., we use the final policy obtained at the end of 10000 trials and use it to find the cumulative reward (figure of merit) which is an indication of the quality of discovered policy. Episode means a set of trials; e.g., 100 trials (Fig. 2) and results have been averaged over 1000 such episodes. Figure 2 shows learning performance of the fuzzy Dec-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 (86% success) in just about 500 trials. The performance reached 92.18% at 4000 trials and peaked to 93.18% success rate at the end of 10, 000 trials.

We also plot infinite horizon discounted reward achieved by the fuzzy Dec-POMDP approach (Figure 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 22.80 in 20 trials and the final expected discounted infinite horizon value obtained is 26.07.

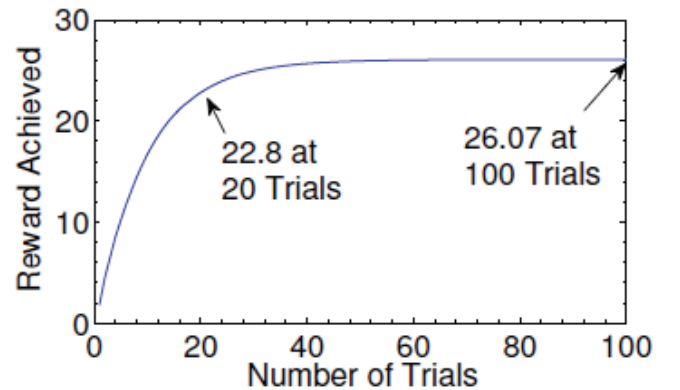


Figure 3 Quality of learned policy: Recycling robot

We compare performance of fuzzy Dec-POMDP against (i) Dec-BPI [17] which reaches a discounted reward of 26 to 27, and (ii) Finite state controller [15] which obtains 30 to 31 (read from figure). Our approach compares favorably against the Dec-BPI approach but is inferior in comparison to the fixed size controller. Fixed size controller's superior performance could be because the approach solves all the agent controllers centrally (in personal communication with the author of [15]) while in fuzzy Dec-POMDP control even the agent controllers are decentralized (solved independently). Furthermore, the performance of fixed size control approach is highly dependent on the controller size which is not the case with fuzzy Dec-POMDP control. Unlike Dec-BPI, performance of fuzzy Dec-POMDP approach does not depend on the finding appropriate fixed points on the belief space, instead it operates on the entire belief space. Even then our approach is able to find a comparable solution quality to the Dec-BPI.

B. Multiagent Broadcast Channel Access

In the MABC problem, each agent has two local buffer states (Empty, Full) resulting in 4 global states (cross product of the local buffer states). Here also we partition each belief variable into three fuzzy partitions leading to $3^4 = 81$ fuzzy rules that map the global belief space to a fuzzy belief space. Type of membership functions (18), their centers and widths are kept same as in the recycle robot task as also the RL parameters.

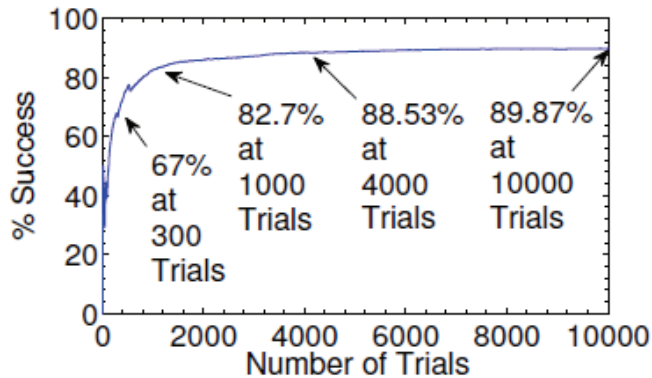


Figure 4 Fuzzy Dec-POMDP learning performance: Multiagent broadcast channel

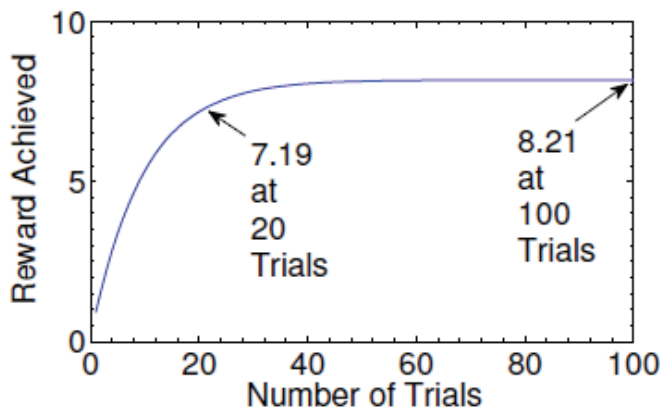


Figure 5 Quality of learned policy: Multiagent broadcast channel

Herein we consider a trial to be successful if the message (by either of the agents) gets transmitted through the channel successfully (without collision) generating a global reward of +1 or else the trial is a failure. Figure 4 shows learning performance of the fuzzy Dec-POMDP controller for the MABC task. We have averaged results over 100 episodes each of 10,000 trials. The controller achieved a performance level of 67% success in just about 300 trials. The performance rose to 88.53% at 4000 trials and became 89.87% at the end of 10,000 trials.

We also plot infinite horizon discounted reward achieved (Figure 5). Here also the results have been averaged over 1000 episodes of 100 trials each. As can be seen, the value achieved is 7.19 in 20 trials and the final expected discounted infinite horizon value obtained is 8.21. It is to be noted that the maximum value that can be achieved for this task with a discount factor $\gamma = 0.9$ is 9.0 [16]. The value of the proposed approach is superior to the one obtained by Szer and Charpillet [16], 8.0.

V. CONCLUSIONS

This paper presents a novel fuzzy reinforcement learning control scheme for Dec-POMDPs in a game-theoretic setting. We show how FIS based function approximation can be used as a principled means to effectively combat the time and space complexity of Dec-POMDPs. To the best of our knowledge, this work represents a first attempt in applying game theoretic learning to an FIS based Dec-POMDP setup. We elucidate feasibility of the proposed approach with empirical results on the benchmark recycling robot and multiagent broadcast channel problems. Proposed fuzzy Dec-POMDP controller successfully discovers a high quality optimal policy solution in a reasonable number of trials. Further, a comparison of the expected discounted reward value achieved against other recent state-of-the-art Dec-POMDP solution approaches [15, 16, 17] brings out the effectiveness of the proposed approach. In future work we plan to relax some of assumptions on knowledge of the system models, as well as identifying for which types of domains our approach is appropriate. Furthermore, the use of a fuzzy rule base opens the potential for generalization over continuous state spaces.

ACKNOWLEDGMENTS

This work was partially funded by Netaji Subhas Institute of Technology, New Delhi, India. This work was partially 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] M. L. Puterman. *Markov Decision Processes—Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, 1994.
- [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] 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, issue 4, pp. 819–840, 2002.
- [4] C. Boutilier, “Planning, learning and coordination in multiagent decision processes”, In *Theoretical Aspects of Rationality and Knowledge*, 1996.
 - [5] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
 - [6] L. Busoniu, R. Babuska, and B. De Schutter, “A comprehensive survey of multiagent reinforcement learning”, *IEEE Transactions on Systems, Man and Cybernetics — Part C: Applications and Reviews*, vol. 38, issue 2, pp. 156–172, 2008.
 - [7] J. R. Kok and N. Vlassis, “Collaborative multiagent reinforcement learning by payoff propagation”, *Journal of Machine Learning Research*, vol. 7, pp. 1789–1828, Sep. 2006.
 - [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] C. J. C. H. Watkins and P. Dayan, “Technical note: Q-learning”, *Machine Learning*, vol. 8, issue 3, pp. 279–292, 1992.
 - [10] J. Si, A. G. Barto, W. B. Powell and D. Wunsch. *Handbook of Learning and Approximate Dynamic Programming*. Wiley-IEEE Press, 2004.
 - [11] 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.
 - [12] L. Jouffe, “Fuzzy inference system learning by reinforcement methods”, *IEEE Transactions on Systems, Man, and Cybernetics — Part C: Applications and Reviews*, vol. 28, issue 3, pp. 338–355, 1998.
 - [13] 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, 2008.
 - [14] S. Seuken and S. Zilberstein, “Formal models and algorithms for decentralized decision making under uncertainty”, *Autonomous Agents and Multiagent Systems*, Feb. 2008.
 - [15] C. Amato, D. S. Bernstein, and S. Zilberstein, “Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs”, *Autonomous Agents and Multiagent Systems*, 2009.
 - [16] D. Szer and F. Charpillet, “An optimal best-first search algorithm for solving infinite horizon DEC-POMDPs”, In *European Conference on Machine Learning*, 2005.
 - [17] D. S. Bernstein, E. A. Hansen, and S. Zilberstein, “Bounded policy iteration for decentralized POMDPs”, In *Proc. Int. Joint Conf. on Artificial Intelligence*, 2005.
 - [18] T. J. Ross. *Fuzzy Logic with Engineering Applications*. Wiley & Sons Inc., 2nd edition, 2004.
 - [19] G. Owen. *Game Theory*. Academic Press, Orlando, FL, 2nd edition, 1982.
 - [20] R. Emery-Montemerlo, G. Gordon, J. Schneider and S. Thrun, “Approximate solutions for partially observable stochastic games with common payoffs”, In *Proc. of Int. Conference on Autonomous Agents and Multiagent Systems*, 2004.
 - [21] P. Poupart and C. Boutilier, “Bounded finite state controllers”, In *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.
 - [22] D. Yagan and C.-K. Tham, “Coordinated reinforcement learning for decentralized optimal control”, In *IEEE Int. Symp. on Approximate Dynamic Programming and Reinforcement Learning*, 2007.
 - [23] G. N. Shirazi, P.-Y. Kong, and C.-K. Tham, “A cooperative retransmission scheme in wireless networks with imperfect channel state information”, In *Proc. of the 2009 IEEE conf. on Wireless Communications & Networking Conference*, 2009.
 - [24] F. Wu, S. Zilberstein, and X. Chen, “Rollout sampling policy iteration for decentralized POMDPs”, In *Proc. of Uncertainty in Artificial Intelligence*, 2010.
 - [25] E. Duman, M. Kaya, and E. Akin, “A multiagent fuzzy reinforcement learning method for continuous domains”, In *Multi-Agent Systems and Applications IV*, LNCS, pp. 306–315. Springer, 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, issue 2, pp. 110–119, 2006.