Developing Communication Strategy for Multi-Agent Systems with Incremental Fuzzy Model

Sam Hamzeloo, Mansoor Zolghadri Jahromi
Department of Computer Science and Engineering
Shiraz University
Shiraz, Iran

Abstract—**Communication can guarantee the coordinated behavior in the multi-agent systems. However, in many real-world problems, communication may not be available at every time because of limited bandwidth, noisy environment or communication cost. In this paper, we introduce an algorithm to develop a communication strategy for cooperative multi-agent systems in which the communication is limited. This method employs a fuzzy model to estimate the benefit of communication for each possible situation. This specifies minimal communication that is necessary for successful joint behavior. An incremental method is also presented to create and tune our fuzzy model that reduces the high computational complexity of the multi-agent systems. We use several standard benchmark problems to assess the performance of our proposed method. Experimental results show that the generated communication strategy can improve the performance as well as full-communication strategy, while the agents utilize little communication.**

Keywords—Multi-agent systems; decentralized partially observable Markov decision process; communication; planning under uncertainty; fuzzy inference systems

I. Introduction

One of the main goals of artificial intelligence is designing autonomous agents interacting in a domain. A Multi-Agent System (MAS) includes multiple autonomous agents operating in an uncertain environment in order to maximize their utility. In MAS, each agent independently perceives its local environment and influence the environment by executing its actions. Many artificial intelligence problems can take advantage of MAS design such as multiple mobile robots, sensor networks, disaster response teams, smart city and video games.

There are two types of problems in MASs, *self-interested* and *cooperative* settings [1]. In self-interested scenario the agents can have different and even conflicting goals. In cooperative setting, which we focus on it in this work, the agents cooperate to reach a shared target. In this case, each agent individually makes a decision based on its local observation, but the maximum reward will be achieved when the individual decisions are coordinated. Communication is an important factor to preserve coordinated behavior. However, communication is not always available, especially when the agents have limit on battery usage or the communication channel is noisy or limited. Therefore, one of the main challenges in MASs is to maintain coordination over a long period of time with minimal communication.

Various mathematical models have been used to characterize decision-making problems. In a stochastic fully observable environment, Markov Decision Process (MDP) provides a powerful modeling tool. Partially Observable Markov Decision Process (POMDP) is employed in problems with limited sensing capabilities. Decentralized Partially Observable Markov Decision Process (Dec-POMDP) is a powerful framework for collaborative multi-agent planning in an uncertain environment [2]. In this paper, we use Dec-POMDP to model cooperative MAS problems.

The strategies of multi-agent problems are categorized in two categories, finite-horizon and infinite-horizon Dec-POMDPs. Finite-horizon policies are usually represented by a decision tree and numerous techniques have been proposed to obtain or approximate the optimal policies [3]-[5]. Moreover, a number of methods have been developed to generate decentralized policies with minimal communication usage [1], [6]. On the other side, finite state controllers (FSCs) is a major model to represent infinite-horizon Dec-POMDP policy. Several optimization techniques have been used to approximate the parameters of FSCs, for example, Linear programming [7], nonlinear programming [8] and expectation-maximization [9], [10]. However, identifying best situations for communication has not been considered by existing methods. This paper focuses on solving this issue.

One of the powerful function approximators are fuzzy systems that can approximate any non-linear system to an arbitrary accuracy. A fuzzy system is capable of handling high level of uncertainty by a compact fuzzy rule-base. Therefore, presenting fuzzy model is desirable to solve a MAS in previous studies [11]. In [12] an incremental fuzzy controller has been introduced to find a solution of large MASs.

Our aim in this paper is to present an algorithm to identify best situations for making communication in MASs modelled by infinite-horizon Dec-POMDP. This method develops a strategy that helps the agents to maintain coordination with minimal communication. This communication policy is developed centralized in a training phase, where the communication is not restricted. The agents use this policy decentralized in a test environment that the communication channel is limited. This paper presents an incremental method to estimate the benefits of communication in every possible situation that the agents can have. Based on this estimation, the agents can decide when the communication has the most impact on the improvement of the final performance. The results show that the performance of the presented communication strategy is almost the same as the full communication.

The organization of the rest of paper is as follows. Section 2 formally defines the infinite-horizon Dec-POMDP and gives an overview of the Dec-POMDP solution methods. Section 3 presents the details of the proposed method. In Section 4 we evaluate the proposed communication strategy on several well-known Dec-POMDP problems. Finally, the conclusions are given in Section 5.

II. Background and Related Works

In this paper, we consider a group of agents cooperate with each other in an uncertain environment over infinite time steps. At each time step t, the agents take joint action $\vec{a}^t$ (action a_i^t for i-th agent) that causes the state of the environment to change from s^t to s^{t+1}. After that, each agent perceives its observation and receives a global reward from the environment. This cycle repeats over infinite steps. This type of MAS problems is properly modelled by infinite-horizon Dec-POMDP [12]. Fig. 1 displays the interaction of the agents and the environment.

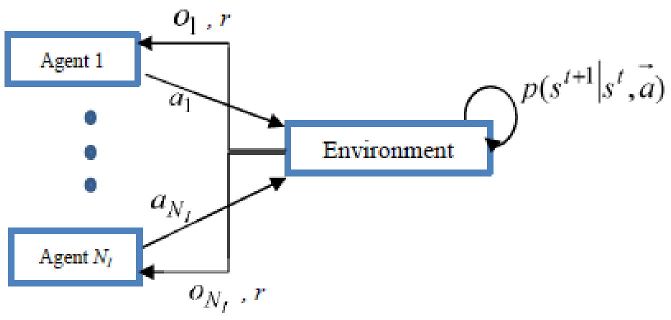


Fig. 1. Dec-POMDP Setup.

A. Infinite-Horizon Dec-POMDP

In infinite-horizon Dec-POMDP, a group of agents are considered that operate in an uncertain environment over infinite steps. Infinite-horizon Dec-POMDP is a tuple $\langle I,S,\{A_i\},\{\Omega_i\},P,O,R,b^0,\gamma\rangle$ where I is a finite set of agents and S is a finite set of states. Each state determines the specific situation of the environment. The number of agents is N_I and N_S is the number of states. A_i and Ω_i specify the finite set of actions and observations available for agent i. $\vec{a}=\langle a_1,\dots,a_{N_I}\rangle$ denotes a joint action ($\vec{A}=\times_{i\in I}A_i$) and $\vec{o}=\langle o_1,\dots,o_{N_I}\rangle$ denotes a joint observation ($\vec{\Omega}=\times_{i\in I}\Omega_i$). If the agents take joint action $\vec{a}^t$ in time step t, the state of the environment is transitioned from s^t to s^{t+1} with probability $P(s^{t+1}|s^t,\vec{a}^t)$. The probability of the joint observation $\vec{o}^{t+1}$ in state s^{t+1} after the agents perform joint action $\vec{a}^t$ is $O(\vec{o}^{t+1}|s^{t+1},\vec{a}^t)$. At the end of each time step, the environment gives the agents the global reward $R(s,\vec{a})$ for taking the joint action $\vec{a}$ in the state s. The initial state distribution is b^0 . The belief vector $b_i^t=\lfloor b_{i,1}^t\cdots b_{i,N_S}^t\rfloor$ determines the belief of i-th agent about the state of the environment in time step t. In fact, b_i^t is a probability distribution over S such that $b_{i,n}^t$ specifies the belief of i-th agent that the state of the environment is s_n. The belief space is an N_s-dimensional space defined by the belief vector.

For infinite-horizon Dec-POMDP problems with the initial state distribution b^0 , the solution is a joint policy δ that maximizes the expected infinite-horizon discounted reward $E\left[\sum_{t=0}^{\infty}\gamma^t R(s^t,\vec{a}^t)\middle|b^0\right]$, where a discount factor γ ($0\le\gamma < 1$) limits the summation of rewards in the infinite-horizon.

Finding the optimal solution for the infinite-horizon Dec-POMDP may not be practical, because of unbounded number of steps [13]. Previous researches have tried to find a sub-optimal solution by using a bounded policy representation. The most common policy representation is finite state controllers (FSCs). Several approaches have presented to estimate the parameters of FSCs such as linear programming [7], nonlinear programming [8] and expectation-maximization [9], [10]. Value function is another approach to represent the policy in infinite-horizon Dec-POMDP problems [14]. In our previous work [12] we have introduced an incremental method to learn a fuzzy model as a value function. It generates a compact fuzzy rule-base as a solution that offers scalability for large MAS problems.

As stated before, obtaining minimal communication to coordinate the behavior of the agents is one of the main challenges in cooperative MAS problems. Therefore, several methods have been introduced to determine the communication strategy. Most of these algorithms work for finite-horizon Dec-POMDP cases [15], [6]. F. Wu et al. [1] introduced an online planning approach to reduce the computational complexity. To cope with limited bandwidth, the agents communicate only when history inconsistency is detected. The presented method in [16] calculates divergence between the agents' belief to evaluate communication. Since this method has considered an imprecise assumption for calculating belief divergence, it cannot accurately estimate the value of communication.

B. Incremental Learning

An incremental learning is a method that creates a model by recursively extracting required information from sequence of incoming data. This learning method is able to start learning "from scratch". Its parameters and structure are tuned incrementally according to current information without memorizing previous observation. Thus, the model can be created using low computational complexity and limited memory size. Evolving fuzzy [17] and neuro-fuzzy [18] systems are the most popular approaches for incremental learning. Shahparast et al. in [19] proposed two fast methods for adapting certainty factors of fuzzy rules, based on the reinforcement learning and reward and punishment. In [20] a simple and fast method is proposed that uses gradient decent to tune the structure and parameters of a fuzzy classifier. D. Kangin et al. in [21] and [22] have introduced a group of incremental methods called TEDA that can be used for clustering, regression and classification. Incremental methods are also employed to find a policy for infinite-horizon Dec-POMDP. An incremental reinforcement learning algorithm is presented in [12] to create a compact fuzzy model as a solution of large MASs.

III. Our Proposed Method

In this paper, we introduce a method to find a communication strategy for cooperative MAS problems in which the communication is expensive or limited. This method estimates the benefit of communication by computing the effect of communication on increasing accumulated reward for each situation. This can be used to obtain minimal communication that is necessary for successful joint behavior.

In this paper, we extend our previous method presented in [12]. In that method, each agent makes use of an individual fuzzy rule-base to interact with the environment. These rule-bases that map the belief space to the value of the actions, are created and tuned by an incremental reinforcement learning algorithm regarding experiences of the agents.

In this paper, two phases are considered, learning and execution phase. In the learning phase, communication between the agents is not limited and the algorithm freely shares the information of the agents to tune the communication strategy. However, there is limited bandwidth in the execution phase and the agents use the learned strategy to identify the situations where the communication can be beneficial to improve the performance.

A. Learning Phase

In this phase, the agents interact with the environment and in addition to tuning their behavior according to the response of the environment, the communication strategy is adjusted. To do this, each agent has an individual decision making system to select the best action in every time step and there is a shared *communication rule-base*, that is used to learn the benefit of communication for each situation.

The benefit of communication, Q_c^t , is computed by comparing the outcomes of two different agent-environment interactions in the particular state of the environment. Since the state of the environment is not available in Dec-POMDPs, we approximate it with the belief vectors of the agents. In each time step, once, the agents select the action without using communication and once again, the actions are selected after sharing information. The difference between the value of these two selected actions (i.e. immediate reward plus the expected accumulation of future rewards) determines the benefit of communication. Therefore, there is a tuple for each time step that contains two parts: particular situation of the environment, which specified by belief vector and Q_c^t , the benefit of communication for this situation. We call this tuple an *experience*. Fig. 2 illustrates the process of producing an *experience* in time step *t*.

Since the agents interact with the environment many times and an *experience* is achieved for each time step, there is a sequence of *experiences* (one element for each time step). The *communication rule-base* is created and tuned using this sequence. The sequence of *experiences* theoretically is infinite in infinite-horizon Dec-POMDP problems. Therefore, we have introduced an incremental algorithm to develop communication strategy. We describe the process of producing an *experience* and updating mechanism according to an *experience* in following two sub-sections.

1) Producing an experience: At each time step t, first, the agents interact with the environment, using only local information. Each agent updates its previous belief vector b_i^{t-1} to local belief vector b_i^t that is computed based on its previous action a_i^{t-1} and local observation o_i^t .

$$\forall s' \in S \, , b_i^t(s') = \frac{\sum_{\vec{a}_{\neq i}^{t-1} \in A_{\neq i}} \sum_{\vec{o}_{\neq i}^{t} \in O_{\neq i}} O(\bar{o}^t \mid s', \bar{a}^{t-1}) \sum_{s \in S} P(s' \mid s, \bar{a}^{t-1}) b_i^{t-1}(s)}{\sum_{s'' \in S} \sum_{\vec{a}_{\neq i}^{t-1} \in A_{\neq i}} \sum_{\vec{o}_{\neq i}^{t} \in O_{\neq i}} O(\bar{o}^t \mid s'', \bar{a}^{t-1}) \sum_{s \in S} P(s'' \mid s, \bar{a}^{t-1}) b_i^{t-1}(s)} \quad (1)$$

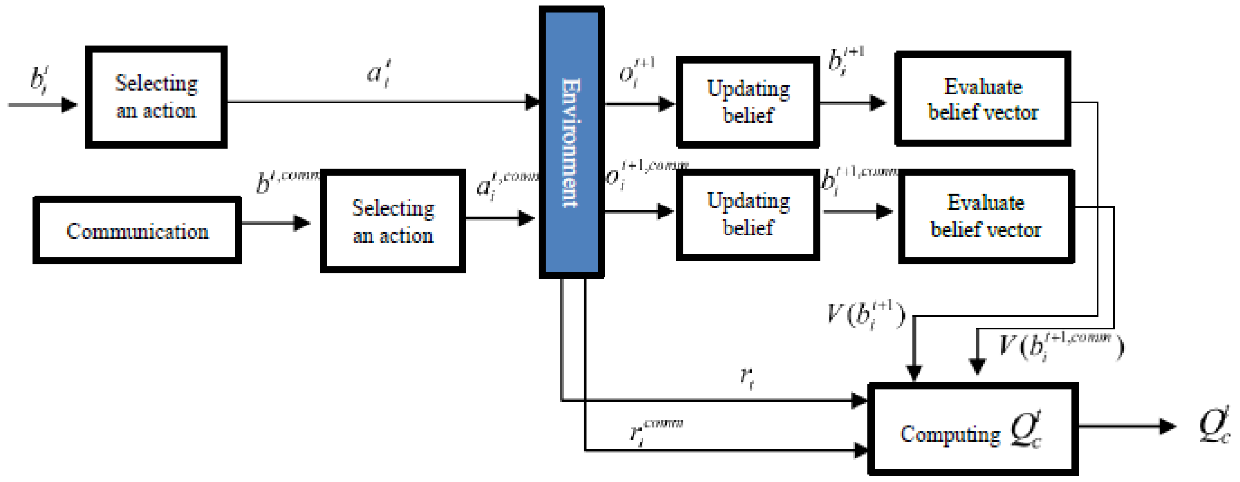


Fig. 2. The process of producing an *Experience* in time Step *t*.

Where, $\vec{o}_{\neq i}^{t}$ and $\vec{a}_{\neq i}^{t-1}$ are the joint observation and the joint action of all agents except agent *i* respectively. Also, $O_{\neq i}$ and $A_{\neq i}$ are all possible joint observations and all possible joint actions for the other agents, $\vec{o}^{t} = < o_i^t, \vec{o}_{\neq i}^t >$ and $\vec{a}^{t-1} = < a_i^{t-1}, \vec{a}_{\neq i}^{t-1} >$.

Then, according to b_i^t , the agent selects the best action. As stated before, we used our previuos work presented in [12] to determine the behavior of the agents. In this method, each agent has individual fuzzy rule-base to estimate the value of the actions according to its belief vector. In fact, fuzzy rule-base of *i-th* agent determines $Q_i(b_i^t, a_m)$, the expected value of action *m*. At each time step, the agents estimate the value of their actions and perform the action having maximum value.

$$a_i^t = \arg\max_{a_m} Q_i(b_i^t, a_m) \tag{2}$$

After obtaining a_i^t , the same process is done to determine the appropriate joint action if the agents share their local information. To do this, the algorithm considers $b^{t,comm}$ as global belief vector and update it by using joint action $\vec{a}^{t-1}$ and joint observation $\vec{o}^{t}$.

$$\forall s' \in S \text{ , } b^{t,comm}(s') = \frac{O(\vec{o}^t \mid s', \vec{a}^{t-1}) \sum_{s \in S} P(s' \mid s, \vec{a}^{t-1}) b^{t-1,comm}(s)}{\sum_{s'' \in S} O(\vec{o}^t \mid s'', \vec{a}^{t-1}) \sum_{s \in S} P(s'' \mid s, \vec{a}^{t-1}) b^{t-1,comm}(s)} \tag{3}$$

Using global belief vector $b^{t,comm}$, each agent selects the best action $a_i^{t,comm}$:

$$a_i^{t,comm} = \arg\max_{a_m} Q_i(b^{t,comm}, a_m) \tag{4}$$

Therefore, there are two joint actions in each time step for interacting with the environment; if the agents communicate to each other, $\vec{a}^{t,comm}$ is selected and if they make decision based on the local information, $\vec{a}^{t}$ is selected. The difference between the outputs of these two joint actions, determines the value of the communication in time step *t*.

Assume r_t^{comm} and $\vec{o}^{t+1,comm}$ are global reward and joint observation if the agents take joint action $\vec{a}^{t,comm}$; and if the agents perform joint action $\vec{a}^{t}$, they receive r_t and $\vec{o}^{t+1}$ from the environment. The difference between the outputs of these two joint actions is calculated as follow:

$$Q_c^t = \left\lfloor r_t^{comm} + \gamma V(b_i^{t+1,comm}) \right\rfloor - \left\lfloor r_t + \gamma V(b_i^{t+1}) \right\rfloor \tag{5}$$

Where b_i^{t+1} is updated for each agent using a_i^t , o_i^{t+1} and (1), and also $b_i^{t+1,comm}$ is updated using $a_i^{t,comm}$, $o_i^{t+1,comm}$ and same equation. $V(b_i^{t+1,-})$ (i.e. $V(b_i^{t+1}) \, or \, V(b_i^{t+1,comm})$) is the estimated value of the incoming situation. In fact, $V(b_i^{t+1,-})$ estimates accumulated reward that will be achieved in the future steps. $V(b_i^{t+1,-})$ is easily obtained by one-step look-ahead:

$$V(b_i^{t+1,-}) = \max_{a' \in A_i} Q_i(b_i^{t+1,-}, a') \tag{6}$$

In this manner, whenever each agent has the same belief vector as b_i^t , the benefit of the communication is Q_c^t (i.e. communication can increase the accumulated reward by Q_c^t). Our proposed algorithm uses this tuple $\left\lfloor b_i^t, Q_c^t \right\rfloor$ as an *experience* to tune the *communication rule-base*.

2) Updating mechanism: In the learning phase, the agents interact with the environment many times and an *experience* is achieved for each time step. Hence, there is a sequence of *experiences* that our algorithm uses to create and tune the *communication rule-base.* The proposed method combines the information of *experiences* by clustering the similar *experiences*, in which center of each cluster identifies a communication rule. Since the number of *experiences* in the learning phase is huge, we introduce an incremental approach to cluster the *experiences*. In the following, we present the incremental process of tuning the communication strategy according to an *experience*:

Each rule specifies the benefit of communication for a region of belief space. The *j-th* rule in *communication rule-base*, R_j^{comm} , have a following form:

$$R_j^{comm} \;:\; if \; b_i^t \; is \, like \; B_j^{comm} \; then \; Q = Q_j^{comm} \tag{7}$$

Where $B_j^{comm} = \left\lfloor B_{j,1}^{comm} \cdots B_{j,N_S}^{comm} \right\rfloor$ is a *reference belief vector* [12] of rule *j* that specifies the center of the region and Q represents the expected benefit of communication for this region.

Assume the *i-th* agent has an *experience* $\left\lfloor b_i^t, Q_c^t \right\rfloor$ in the time step *t*. The algorithm identifies the most similar reference belief vector to b_i^t . To do this, the similarity of b_i^t to the reference belief vector of all existing rules in the *communication rule-base* is computed as follows:

$$CosSim(b_i^t, B_j^{comm}) = \frac{\sum_{k=1}^{N_s} b_{i,k}^t B_{j,k}^{comm}}{\sqrt{\sum_{k=1}^{N_s}\left(b_{i,k}^t\right)^2}\sqrt{\sum_{k=1}^{N_s}\left(B_{j,k}^{comm}\right)^2}} \tag{8}$$

Where $CosSim(b_i^t, B_j^{comm})$ is the cosine similarity of these two vectors.

If maximum similarity of b_i^t to the existing rules is less than thershold $Sim_{\min}$, i.e. b_i^t considerably different with all reference belief vectors, so we consider $\lfloor b_i^t, Q_c^t \rfloor$ as a *new experience*. In this case, the proposed method adds a new rule to the *communication rule-base*, according to $\lfloor b_i^t, Q_c^t \rfloor$. The reference belief vector of the new rule is set to b_i^t ($B_{newRule}^{comm} \leftarrow b_i^t$) and the consequent part of the new rule is set to Q_c^t ($Q_{newRule} = Q_c^t$). It is noteworthy that if there is no rule in the *communication rule-base*, the same procedure is done to add the first rule.

Otherwise, if there is a similar reference belief vector to b_i^t, the nearest rule to b_i^t is determined:

$$w = \arg\max_j \left\{ CosSim(b_i^t, B_j^{comm}) \right\} \tag{9}$$

Where, w is the index of the most similar rule. Each rule is identified by averaging all similar *experiences* that agents have during the learning phase. Since the number of these *experiences* is huge, we use recursive formula to calculate the mean of group of similar *experiences*. For adjusting R_w^{comm} according to the *experience* $\lfloor b_i^t, Q_c^t \rfloor$, the antecedent of R_w^{comm} is updated regarding b_i^t by following recursive equation [21]:

$$B_{w(new)}^{comm} = \frac{(k-1)B_{w(old)}^{comm} + b_i^t}{k} \tag{10}$$

Where $B_{w(old)}^{comm}$ and $B_{w(new)}^{comm}$ are the reference belief vectors of R_w^{comm}, before and after updating, respectively. Similarly, the consequent of R_w^{comm} is updated as follow:

$$Q_{w(new)}^{comm} = \frac{(k-1)Q_{w(old)}^{comm} + Q_c^t}{k} \tag{11}$$

B. Execution Phase

The generated strategy is performed in the execution phase in which communication is limited. In this phase, the agents estimate the benefit of communication and if it is recognized beneficial, the agents share their local information.

In each time step, the agents compute the benefit of communication according to its belief vector as follow:

Assume the belief vector of *i-th* agent in time step *t* is b_i^t. Firing strength of all rules in *communication rule-base* are calculated using:

$$\omega_j^t = \prod_{n=1}^{N_S} \mu_{B_{j,n}^{comm}}(b_{i,n}^t) \tag{12}$$

Where ω_j^t is the firing strength of the rule *j*. These firing strengths are then used to calculate the benefit of communication:

$$Q_{comm}(b_i^t) = \frac{\sum_{j=1}^{N_r} \omega_j^t Q_j^{comm}}{\sum_{j=1}^{N_r} \omega_j^t} \tag{13}$$

Where N_r is the number of rules in the *communication rule-base* and $Q_{comm}(b_i^t)$ denotes the benefit of communication from the perspective of *i-th* agent. This agent propagates communication request if the estimated benefit is more than predefined threshold C_{comm} :

$$Q_{comm}(b_i^t) > C_{comm} \tag{14}$$

The values of C_{comm} depends on the characteristics of each problem. In the real-world problems, this parameter can be set according to the percentage of access to the communication. Also, in an application with the communication cost, this parameter can be used to balance the communication costs with the coordination benefits.

If communication is available, each agent propagates its sequence of action-observation from previous communication, up to the current time step. By sharing this information, the belief vectors of all agents are equivalent and thus the coordinated behaviours are guaranteed. In the absence of communication, the agent postpones its request until the communication is allowed. By using this strategy, the behaviours of the agents maintain coordinated with little communication.

IV. Experimental Results

We evaluated our proposed algorithm on several benchmark problems that have been widely used to rate multi-agent planning methods. These problems are Broadcast Channel [3], Meeting in a Grid 3×3 [4], Cooperative Box Pushing [5] and Stochastic Mars Rover [23]. We reported the accumulated discounted reward (Reward), percentage of communication (Comm. (%)) and the number of generated rules with different values of C_{comm} . In the real-world problems, C_{comm} can be set regarding the amount of access to the communication. Lower value of C_{comm} increases the communication usage. In an application with the communication cost, C_{comm} can be used to balance communication costs with coordination benefits. The discount factor is set to 0.9 and the results are averages over 50 runs.

To the best of our knowledge, this is the first attempt to find the communication behaviour in infinite-horizon Dec-POMDP problems. Therefore, we compare the performance of our communication strategy to the full-communication (Full-Comm.) strategy as an upper bound and the no-communication (No-Comm.) strategy as a lower bound. Since in real-world MAS problems the communication is limited, the main purpose of the experiments is to test whether our proposed communication behaviour can help the agents to approach the performance of full-communication, while using little communication.

A. Broadcast Channel Problem

In the Broadcast Channel problem two agents are connected in a network. In each time step, only one of them can use the connection and sends its message. To avoid collision, each agent has to decide whether send a message or not. This problem has 4 states, 2 actions and 5 observations. The results in Table I show that Broadcast Channel problem is very simple such that the agents can easily cooperate. Therefore, the performance of the various percentage of communication is almost the same and different values of C_{comm} have no effect on the performance.

TABLE I. BROADCAST CHANNEL RESULTS

Broadcast channel	C_{comm}	**Reward**	**Comm. (%)**	**No. of rules**
$\|S\|=4$	No-Comm.	9.1	0	-
$\|A_i\|=2$	0.5	9.11	0.0	2
$\|O_i\|=5$	0.1	9.18	83.16	1.16
	Full-Comm.	9.2	100	-

B. Meeting in a Grid Problem

In Meeting in a Grid problem, there are two agents on a 3×3 grid. They can move up, down, left or right, or stay on previous square. Each agent can sense whether there are walls around by noisy sensors with a 0.9 chance to perceiving the right observation. The goal of the agents is to spend as much time as possible on the same square. This problem has 81 states, 7 observations, 5 actions. The results in Table II show that low percentage of communication cannot significantly improve accumulated reward, however the performance of full-communication strategy can be achieved by making communication in almost half of time steps. Since the agents in Meeting in a Grid problem need the future planning information to cooperate, and in our method, the action-observation sequence is transferred, the proposed communication strategy cannot maintain the agents coordinated for a long time.

TABLE II. MEETING IN A 3×3 GRID RESULTS

Meeting in a 3×3 Grid	C_{comm}	**Reward**	**Comm. (%)**	**No. of rules**
$\|S\|=81$	No-Comm.	4.19	0	-
$\|A_i\|=5$	0.7	4.22	14.13	27.62
$\|O_i\|=7$	0.5	5.71	57.99	27.94
	Full-Comm.	5.82	100	-

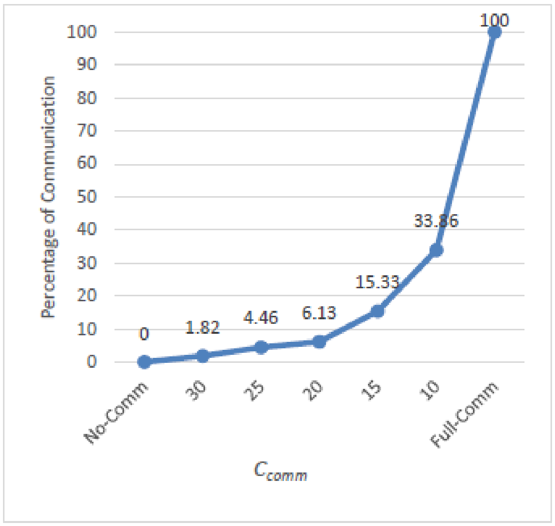
C. Cooperative Box Pushing Problem

In Cooperative Box Pushing problem, there are three boxes (two small and one large) on a 3×4 grid and two agents that can move the boxes. Each agent can push a small box alone. However, for moving the larger box, the agents need to cooperate. Whenever one of the boxes reaches into a goal area, a trial ends. If it is one of the small boxes, the agents gain a reward of +10, and if the large box move into the goal area, they get a reward of +100. However, if a box smashes into a wall or the large box is pushed by one agent, a penalty of -5 is received. The Box Pushing problem has 4 actions, 5 observations, 4 goal states and 96 non-goal states (100 states in total). According to the definition of this problem, communication has a significant impact on the performance. The reported results in Table III show the proposed communication strategy did significantly improve the performance with low percentage of communication. While the achieved accumulated reward with no communication is 177.11, this value can be increased to 218.97 by communicating in only 6.13% of time steps. Also, the accumulated reward has reached 225.19 by communicating in one third of time steps whereas it is 232.25 for the full-communication case.

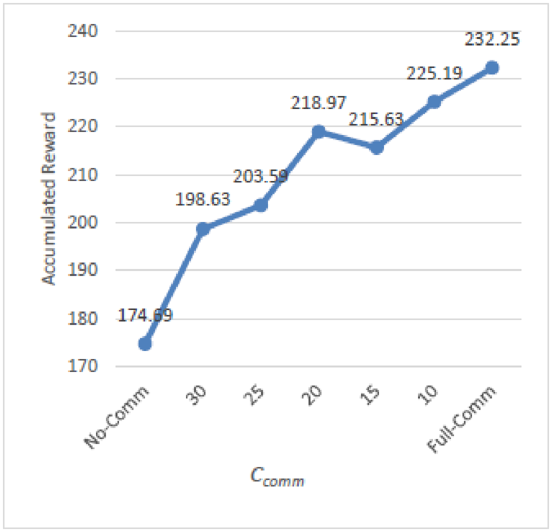
Fig. 3 demonstrates the effect of different values of parameter C_{comm} on the percentage of communication and the accumulated reward in solving Cooperative box pushing problem. In order to better illustration of the performance of our method, the values of the accumulated reward are shown between the achieved reward of the No-Comm. strategy as a lower bound and the Full-Comm. strategy as an upper bound. As stated before, the percentage of communication and accumulated reward are increased by decreasing C_{comm} . Moreover, regarding these figures it is obvious that the accumulated reward is significantly increased with a small increase in percentage of communication.

TABLE III. COOPERATIVE BOX PUSHING RESULTS

Cooperative box pushing	C_{comm}	**Reward**	**Comm. (%)**	**No. of rules**
$\|S\|=100$	No-Comm.	177.11	0	-
$\|A_i\|=4$	30	198.63	1.82	26.34
$\|O_i\|=5$	20	218.97	6.13	26.34
	10	225.19	33.86	26.56
	Full-Comm.	232.25	100	-



(a)



(b)

Fig. 3. The Effect of C_{comm} on (a) the Percentage of Communication and (b) the Accumulated Reward in Cooperative Box Pushing Problem.

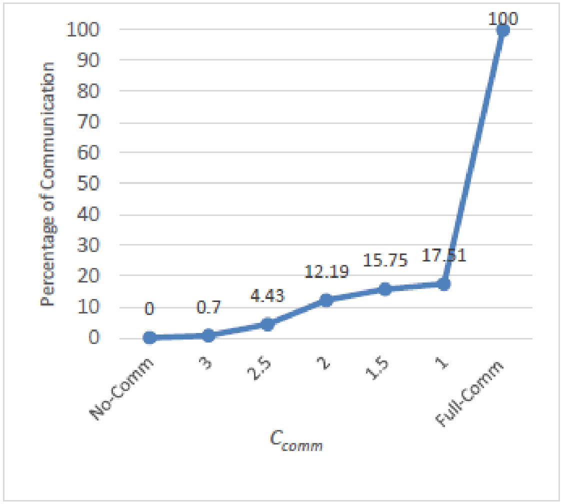
D. Mars Rover Problem

We evaluate the performance of our proposed method with a larger problem, Mars Rover problem. In This problem, there are two rovers experimenting at a 2×2 grid by independently drilling or sampling at each site or moving around. Two of the sites just need one agent to sample, while in the other sites, both agents must drill at the same time in order to get the maximum reward. The agents get a large penalty, if a site is drilled while it only needs to be sampled. When at least one experiment is performed at each site, the problem is reset. This problem has 256 states, 6 actions and 8 observations. As can be seen from Table IV, proposed communication strategy did very well for Mars Rover problem as a large MAS problem. The method achieves almost the same performance as the case of full-communication by making communication in less than one fifth of time steps (17.51%).

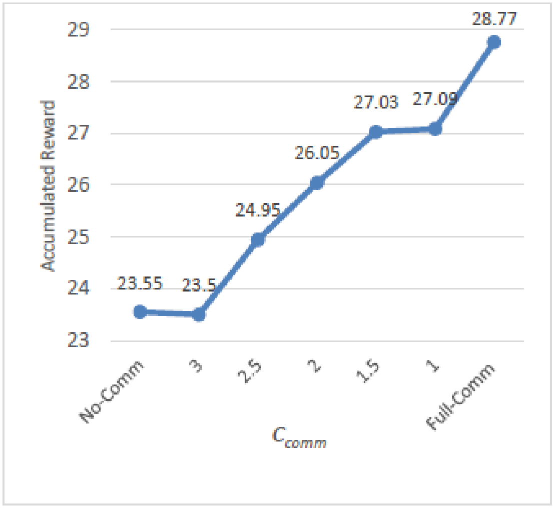
We have also demonstrated the results of accumulated rewards and the percentage of communication with different values of C_{comm} in solving Mars rover problem in Fig. 4. Fig. 4(a) illustrates the effect of C_{comm} on the percentage of communication and Fig. 4(b) shows the effect of this parameter on the accumulated reward. Again, in Fig. 4(b), the values of accumulated reward are shown between the reward of the No-Comm. strategy and the Full-Comm. strategy as the lower and upper bound, respectively. Fig. 4 clearly shows that with a small increase in percentage of communication, the accumulated reward is significantly increased.

TABLE IV. MARS ROVER RESULTS

Mars Rover	C_{comm}	Reward	Comm. (%)	No. of rules
$\lvert S\rvert=256$	No-Comm.	23.55	0	-
$\lvert A_i\rvert=6$	3	23.5	0.7	8.06
$\lvert O_i\rvert=8$	2	26.05	12.19	8.02
	1	27.09	17.51	8.24
	Full-Comm.	28.77	100	-



(a)



(b)

Fig. 4. The Effect of C_{comm} on (a) the Percentage of Communication and (b) the Accumulated Reward in Mars Rover Problem.

To summarize, our proposed algorithm to develop the communication strategy, performed very well in all the benchmark problems. Using this strategy can heavily reduce the amount of communication necessary for successful coordinated behaviour.

V. Conclusion

We introduced an algorithm to develop a communication strategy for cooperative multi-agent systems in which the communication is limited. This strategy identifies best situations for making communication in MASs modelled by infinite-horizon Dec-POMDP. This communication policy is developed centralized in a training phase, which the communication is not restricted. The agents use this policy decentralized in a test environment that the communication channel is limited. Our method generates a fuzzy model to approximate the benefit of communication for each situation. The agents can use this fuzzy model to obtain minimal communication that is necessary for coordinated behavior. We also introduced an incremental method to create and tune this fuzzy model. Our incremental method has reduced the high computational complexity of the multi-agent systems by constructing a compact fuzzy rule-base. We used several standard benchmark problems to evaluate the performance of our proposed method. Experimental results show that this communication strategy can help the agents to achieve almost the same performance as the full-communication strategy by using little communication. Therefore, in the real-world MAS problems that the communication is usually limited, our proposed algorithm can heavily reduce the amount of communication necessary for successful coordinated behaviour.

Many AI domains can take advantage of MAS design such as multiple mobile robots and disaster response teams. Developing a group of intelligent players or agents in video games is another interesting field in AI research. In our future work, we intend to customize our incremental model to create human-like players for real-time strategy games who can act and react intelligently against virtual environment and even real players.

References

[1] E Wu, S. Zilberstein and X. Chen, "Online Planning for Multi-Agent Systems with Bounded Communication," Artificial Intelligence, vol. 175, no. 2, p. 487–511, 2011.

[2] D. S. Bernstein, R. Givan, N. Immerman and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," in Mathematics of Operations Research 27, 2002.

[3] D. . S. Bernstein , . E. . A. Hansen and S. Zilberstein , "Bounded policy iteration for decentralized POMDPs," in Proceedings of the 19th international joint conference on Artificial intelligence , 2005 .

[4] C. Amato , J. S. Dibangoye and S. Zilberstein, "Incremental Policy Generation for Finite-Horizon DEC-POMDPs," in Proceedings of the 19th International Conference on Automated Planning and Scheduling, Thessaloniki, Greece, 2009.

[5] S. Seuken and S. Zilberstein, "Improved Memory-Bounded Dynamic Programming for Decentralized POMDPs," in Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI), Vancouver, British Columbia, 2007.

[6] M. Roth, R. Simmons and M. Veloso, "Reasoning about joint beliefs for execution-time communication decisions," in AAMAS '05 Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, 2005.

[7] D. S. Bernstein, C. Amato, E. A. Hansen and S. Zilberstein, "Policy Iteration for Decentralized Control of Markov Decision Processes," Journal of AI Research (JAIR), vol. 34, pp. 89-132, 2009.

[8] C. Amato , D. S. Bernstein, and S. Zilberstein, "Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs," Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS), vol. 21, no. 3, p. 293–320, 2010.

[9] J. K. Pajarinen and J. Peltonen, "Periodic Finite State Controllers for Efficient POMDP and DEC-POMDP Planning," in the 25th Annual Conference on Neural Information Processing Systems (NIPS 2011), 2011.

[10] A. Kumar and S. Zilberstein, "Anytime Planning for Decentralized POMDPs using Expectation Maximization," in Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI), Catalina Island, California, 2010.

[11] R. Sharma and M. T. J. Spaan , "Bayesian-Game-Based Fuzzy Reinforcement Learning Control for Decentralized POMDPs," IEEE Transactions on Computational Intelligence and AI in Games, vol. 4, no. 4, pp. 309 - 328 , 2012.

[12] S. Hamzeloo and M. Zolghadri Jahromi, "An incremental fuzzy controller for large dec-POMDPs," in Artificial Intelligence and Signal Processing Conference (AISP) , Shiraz, Iran, 2017.

[13] F. A. Oliehoek and C. Amato, A Concise Introduction to Decentralized POMDPs, Springer International Publishing, 2016.

[14] H. Kurniawati, D. Hsu and W. S. Lee, "Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in In Proc. Robotics: Science and Systems, 2008.

[15] R. Emery-Montemerlo, Game-theoretic control for robot teams, Doctoral Dissertation, Robotics Institute, Carnegie Mellon University, August 2005.

[16] S. A. Williamson, E. H. Gerding and N. R. Jennings, "Reward shaping for valuing communications during multi-agent coordination," in AAMAS '09 Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1 , Budapest, Hungary, May 10 - 15, 2009.

[17] P. P. Angelov and X. Zhou, "Evolving Fuzzy-Rule-Based Classifiers From Data Streams," IEEE Transactions on Fuzzy Systems, vol. 16, no. 6, 2008.

[18] S. Schliebs and N. Kasabov, "Evolving spiking neural network—a survey," Evolving Systems, vol. 4, no. 2, p. 7–98, 2013.

[19] H. Shahparast, S. Hamzeloo and M. Zolghadri Jahromi, "A Self-Tuning Fuzzy Rule-Based Classifier for Data Streams," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 22, no. 2, 2014.

[20] H. Shahparast and E. G. Mansoori , "An online fuzzy model for classification of data streams with drift," in Artificial Intelligence and Signal Processing Conference (AISP) , Shiraz, Iran, 25-27 Oct. 2017 .

[21] D. Kangin, P. Angelov and J. A. Iglesias, "Autonomously evolving classifier TEDAClass," Information Sciences, vol. 366, p. 1–11, 2016.

[22] D. Kangin and P. Angelov, "Evolving clustering, classification and regression with TEDA," in International Joint Conference on Neural Networks (IJCNN), 2015.

[23] C. Amato and S. Zilberstein , "Achieving goals in decentralized POMDPs," in Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems , 2009.