Agent Interaction in Distributed POMDPs and its Implications on Complexity

Jiaying Shen Raphen Becker Victor Lesser
Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610
{jyshen, raphen, lesser}@cs.umass.edu

ABSTRACT

The ability to coordinate effectively is critical for agents to accomplish their goals in a multi-agent system. A number of researchers have modeled the coordination problem for multi-agent systems using decision theory. The most general models have proven to be extremely complex to solve optimally (NEXP-complete). Some of the more restricted models have been much more tractable, though still difficult (NP-complete). What is missing is an understanding about why some models are much easier than others. This work fills this gap by providing a condition that distinguishes between problems in NP and those strictly harder than NP. This condition relates to the quantity of information each agent has about the others, and whether this information can be represented in a succinct way. We show that the class of problems that satisfy this condition is NP-complete. We illustrate this idea with two interaction protocols that satisfy the condition. For those problems that do not satisfy this condition we demonstrate how our theoretical results can be used to generate an NP approximation of the original problem.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

DESIGN, THEORY

Keywords

Complexity, Interaction, Distributed POMDP

1. INTRODUCTION

In a multi-agent system, the agents do not act alone and must take into account the behaviors of the other agents in order to coordinate effectively and achieve their goals. Recently, multi-agent coordination problems have received attention from the decision theory community, and several distributed POMDP frameworks have been built to model these problems. These models assume that the agents have partial or noisy observations about the current world state, and that communication between the agents may have a cost. This partial observability of the world state leads to an increased complexity of NEXP-complete [4, 12]. However, two subclasses of distributed POMDPs have been identified whose complexity is much more tractable than the general case. One of them is the class of distributed POMDPs where the agents interact with each other only through a global reward function while their local transitions and observations remain independent [3]. In the other class of problems agents are allowed to communicate with each other, however, when they do they are required to transmit sufficient information such that all agents know the global state [6, 1]. Both subclasses have been proven to be NP-complete [7, 3].

The focus of this paper is to quantify the characteristics of a class of multi-agent coordination problems that determines its complexity. Our key result is that the complexity of the problem depends on the amount of important information each agent has about the other agents, and whether this information can be represented in a succinct way. Information is important if knowing it could allow the agents to achieve a higher expected reward, and by succinct we mean that the set of all possible important information the agents could acquire is polynomial in the number of states. We prove that this criteria is both sufficient and necessary for the class of problems to be NP-complete. We illustrate this idea with two examples from the literature and prove that both have this property.

Our goal in this paper is not to introduce new models or algorithms, but to change the way people view interactions between agents in the context of distributed POMDPs. Multi-agent researchers have long intuitively understood that the interaction between the agents is the cause of their high complexity. The theoretical results of this paper are significant in that they both formally justify this intuition as well as explain how the interaction affects the complexity. This new understanding of interaction and its relationship to complexity will help us to identify new classes of multi-agent systems with a lower complexity.

The significance of this theoretical result also has a more practical side. Most multi-agent systems are provably harder than NP and solving them optimally is very difficult. Much work has been put into developing good algorithms for approximating these problems. This work provides theoretical guidance in understanding how the approximations in a model limit the search space and reduce the complexity.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.
Copyright 2006 ACM 1-59593-303-4/06/0005 ...$5.00.

We demonstrate this on two problems that does not meet our condition by providing two approximations that impose additional restrictions on the interactions among the agents and reduce the complexity to no harder than NP.

There is existing work that looks at different types of interactions in distributed POMDPs. Synchronizing communication in an otherwise transition and observation independent DEC-MDP has been studied most extensively [6, 1]. Very little research has been done studying the transfer of only partial observation history between agents. Shen et al. [13] studies the problem of what to communicate in a Distributed Sensor Interpretation setting, where at every step only part of the local state is transferred between the agents. Some researchers [8, 5] have studied the case where the agents send their policies or actions to each other. In a sense, exchanging actions or policies between the agents accomplishes the same goal as transferring partial observation history. Both of them change each agent's belief of the global state and/or the other agent's local view of the global state without completely synchronizing the global views. The interaction history in these problems contains information that is not polynomial in the number of states and therefore they are harder than NP.

This work builds on previous work in the distributed POMDP field. Xuan and Lesser [14] proposed one of the first decentralized extensions to the single agent MDP, in which an agent has its local state in addition to the global state. Several more general distributed POMDP models have been independently developed and can be grouped into two categories. Partially Observable Identical Payoff Stochastic Games (POIPSG) [11], Decentralized Partially Observable Markov Decision Processes (DEC-POMDP) and Decentralized Markov Decision Processes (DEC-MDP) [4] do not explicitly model the communication actions but instead view them as a part of the normal action space. In contrast, Communicative Multi-agent Team Decision Problems (COM-MTDP) [12] and DEC-MDPs with Communication (DEC-MDP-Com) [6] make a clear distinction between a domain level action and an explicit communication action. In this paper, we prove that these two classes of models are equivalent. While making a clear distinction between explicit communication and domain level actions allows us to study the effects of communication more easily, this distinction does not increase the expressiveness of the model. Becker et al [2] made an early attempt to identify and exploit structure in the interactions between agents to reduce the complexity. This work explains the complexity results they found.

This paper is organized as follows. Section 2 presents two formal models that we use throughout the paper. Section 3 discusses different types of interactions between agents and presents the condition that determines the complexity of a distributed MDP. Two examples of NP-complete interaction protocols are found in Section 4, and Section 5 contains two examples of how harder problems can be approximated. Section 6 discusses future work and conclusions.

2. FORMAL MODELS

There have been a number of distributed decision-theoretic models for multi-agent systems published in the last few years. Many of these models describe the same class of problems. We will discuss two models, the DEC-MDP-Com and the DEC-MDP. While these two models are equivalent, they do represent interactions between the agents differently and it is therefore useful to examine both.

The models are based on the single agent Partially Observable Markov Decision Process (POMDP) extended to apply to systems with multiple agents. In this paper, we will define them for two agents i and j, but the definitions easily extend to any number of agents. The basic idea is that the state, now known as the global state, encompasses all of the necessary information about the state of each agent (local state) and the environment. Each agent has its own set of actions it takes and observations it makes. The transition function is from world states to world states given the actions taken by each agent, the joint action. The complexity of these models is NEXP-complete [4]. The reason that it is harder than the POMDP is that the agents are all receiving different observations, and only together could they identify the current global state. To predict what action agent i will take requires agent j to maintain not only a belief about the global state (similar to a POMDP) but also a belief about the belief agent i has. Interactions between the agents are built into the transition and observation functions. The likelihood that agent i will see a particular observation depends directly on the actions of both agents as well as the new global state, which in turn depends on the actions of both agents. The DEC-MDP stops there in its definition (see [4] for a formal definition), but the DEC-MDP-Com explicitly represents another form of interaction, communication.

Definition 1 (adapted from [7]). *A DEC-MDP-Com is a tuple: $\langle S, A, \Sigma, C_\Sigma, P, R, \Omega, O\rangle$, where:*

- *S is a finite set of global states, with a distinguished initial state s_0.*
- *$A = A_i \times A_j$ is a finite set of joint actions. A_i are the local actions for agent i.*
- *$\Sigma = \Sigma_i \times \Sigma_j$ is the alphabet of messages. $\sigma_i \in \Sigma_i$ denotes an atomic message sent by agent i. $\overline{\sigma}_i$ is a sequence of atomic messages sent by agent i. $\epsilon_\sigma \in \Sigma_i$ denotes not sending a message to the other agent.*
- *$C_\Sigma \rightarrow \Re$ is the cost of transmitting an atomic message.*
- *$P : S \times A \times S \rightarrow \Re$ is the transition function. $P(s'|s, a_i, a_j)$ is the probability of the outcome state s' when the joint action (a_i, a_j) is taken in state s.*
- *$R : S \times A \times S \rightarrow \Re$ is the reward function. $R(s, a_i, a_j, s')$ is the reward obtained from taking joint action (a_i, a_j) in state s and transitioning to state s'.*
- *$\Omega = \Omega_i \times \Omega_j$ is a finite set of joint observations. Ω_i is the set of observations for agent i.*
- *$O : S \times A \times S \times \Omega \rightarrow \Re$ is the observation function. $O(s, a_i, a_j, s', o_i, o_j)$ is the probability of agents i and j seeing observations o_i and o_j after the sequence s, (a_i, a_j), s' occurs.*
- *Joint full observability: the pair of observations made by the agents together fully determine the current state. If $O(s, a_i, a_j, s', o_i, o_j) > 0$ then $P(s'|o_i, o_j) = 1$.*

The policy for an agent is a mapping from all of the available information to a domain action and a communication action. Even though the global state is Markov, the agents base their decisions on the history of observations just like in a POMDP. However, if the agents ever communicate such that they both know the global state, i.e., they exchange their most recent observation, then they do not need to remember the prior history due to the Markov property. In effect, this synchronization resets the problem but with a new starting state.

Definition 2. *The local policy for agent i, π_i, is a mapping from the history of observations $\overline{\Omega}_i$, the history of messages sent $\overline{\Sigma}_i$, and the history of messages received $\overline{\Sigma}_j$ since the last synchronized world state S to a domain action and a communication action.*

$$\pi_i : S \times \overline{\Omega}_i \times \overline{\Sigma}_i \times \overline{\Sigma}_j \rightarrow A_i \times \Sigma_i.$$

The goal for a DEC-MDP-Com is to find a joint policy $\pi = \langle \pi_i, \pi_j \rangle$ that maximizes the expected value.

An important question is whether or not the communication explicitly represented in the DEC-MDP-Com increases the expressiveness of the model. It turns out that it does not – the communication actions are just special types of domain actions and the message received are just special types of observations.

Theorem 1. *DEC-MDP-Com is equivalent to DEC-MDP.*

Proof.

- **DEC-MDP $\leq_p$ DEC-MDP-Com.**

This reduction is trivial. To any DEC-MDP, we add $\Sigma = C_\Sigma = \emptyset$ and we get an equivalent DEC-MDP-Com.

- **DEC-MDP-Com $\leq_p$ DEC-MDP.**

We reduce a DEC-MDP-Com $\langle S, A, \Sigma, C_\Sigma, P, R, \Omega, O \rangle$ to an equivalent DEC-MDP $\langle \hat{S}, \hat{A}, \hat{P}, \hat{R}, \hat{\Omega}, \hat{O} \rangle$.

The basic idea is to introduce the two step process of the DEC-MDP-Com into the DEC-MDP by doubling the state space: $\hat{S} = S \times \{0, 1\}$. The states $\hat{s} = [s, 0]$ are for taking domain actions A_i and receiving observations Ω_i. The states $\hat{s} = [s, 1]$ are for taking communication actions Σ_i and receiving communications Σ_j. The total space of actions is therefore $\hat{A}_i = A_i \cup \Sigma_i$. The observations that agent i receives include both the messages sent by agent i and the messages received from agent j, i.e., $\hat{\Omega}_i = \Omega_i \times \Sigma_i \times \Sigma_j$. When taking domain actions nothing changes in the functions $\hat{P}$, $\hat{R}$ and $\hat{O}$:

$$\hat{P}([s, 0], a_1, a_2, [s', 1]) = P(s, a_1, a_2, s').$$
$$\hat{R}([s, 0], a_1, a_2, [s', 1]) = R(s, a_1, a_2, s').$$
$$\hat{O}([s, 0], a_1, a_2, [s', 1], o_1, o_2) = O(s, a_1, a_2, s', o_1, o_2).$$

When taking the communication actions, they do change:

$$\hat{P}([s, 1], \sigma_1, \sigma_2, [s, 0]) = 1.$$
$$\hat{R}([s, 1], \sigma_1, \sigma_2, [s, 0]) = C_\Sigma(\sigma_1) + C_\Sigma(\sigma_2).$$
$$\hat{O}([s, 1], \sigma_1, \sigma_2, [s, 0], \sigma_1\sigma_2, \sigma_2\sigma_1) = 1.$$

Therefore, the DEC-MDP-Com is equivalent to the DEC-MDP. □

Theorem 1 states that DEC-MDP-Com and DEC-MDP have the same expressiveness. However, the distinction between the communication actions and the domain actions can be very useful, as we will show in the next section.

3. POLYNOMIALLY ENCODABLE INTERACTIONS

The DEC-MDP-Com and related models allow for a very general form of interaction between the agents. The complexity for those problems has been proven to be NEXP-complete [4, 7]. At the other end of the spectrum would be to disallow all interactions between the agents. In effect, each agent would be independently solving a local MDP that represents its part of the system. MDPs are P-complete, so there is something about the interactions between the agents which is the cause of the complexity. As additional evidence, Becker et al [3] defined a class of multi-agent problems in which the agents were almost completely independent. Each agent had a local MDP that described its part of the system. The agents could not communicate in any way nor could they take an action that would influence another agent. However, the system received a reward that depended on the local states and actions of all of the agents, and the goal was to find a joint policy that maximized the sum of the expected local and global rewards. This class of problems in which the agents can only interact through the reward function proved to be NP-complete. Goldman and Zilberstein [7] also showed that by following certain communication protocols the agents from the previous example could communicate with each other and the problem remained at NP-complete.

This section will examine different types of interactions between agents and provide theoretical results to explain how and why the interactions affect the complexity of finding optimal solutions. The next section will elaborate on the two NP-complete examples introduced above and prove that they meet this condition.

We classify the actions agents can take into two groups: non-interacting (or independent) actions and interacting (or dependent) actions. Independent actions are those that do not affect the other agent and neither agent receives any information about the other. Dependent actions are those that affect the other agent in some way. For example, robot i could pick up robot j and move it, which would affect the local state of robot j. Communication is another example: agent i could send a message to agent j, which would change the knowledge agent j has. We can further subdivide dependent actions into explicit and implicit communication. Normally when one thinks about communication, i.e., sending a message, one is talking about explicit communication. This is the communication part of the DEC-MDP-Com. Implicit communication is the information an agent receives by a domain action, like the example of a robot picking up and moving another robot. The robot being picked up gains information about the local state and belief of the other robot, namely the location of the other robot and the fact that the other robot felt this was a useful action to take.

We will illustrate these interactions with a token collecting example. There is a $n \times n$ grid world, which is populated by two agents and a number of tokens. The agents can observe their own locations and the locations of the tokens. When an agent picks up a token, the system gets a positive reward. The goal of the system is to maximize the total reward within time T. This problem can be modeled in a DEC-MDP. The world state includes the locations of both agents, the locations of the tokens and the time left. The agents' observations at each time step include the agents' own location, the location of the tokens, and the time left. At every time step, each agent can either move to an adjacent square or pick up a token at its current location. If an agent moves, its action does not affect the other agent's observations, and therefore the movement actions are independent actions. However, if agent i picks up a token, agent j can observe the fact that one of the tokens just disappeared. By comparing its current observation and the observation at the last time step, agent j can infer the exact location of agent i and therefore has the complete knowledge of the current world state. As a result, the token collecting action is a dependent action even though there is no explicit communication in the system.

We defined a dependent action as an action that affects the observations of the other agent and therefore changes its belief about the global state. If a dependent action is explicitly modeled in a DEC-MDP-Com, its effect is recorded by the communication action σ_i itself. On the other hand, if it is not explicitly modeled in a DEC-MDP-Com, its effect is recorded by the observations of the agents. The observation history $\overline{\Omega}_i$ records the interaction history of agent i in the DEC-MDP. Consequently, in a DEC-MDP-Com where there are communication actions explicitly modeled, the interaction history of agent i is $\overline{\Omega}_i \times \overline{\Sigma}_i \times \overline{\Sigma}_j$.

Definition 3. *We call E_i an* **encoding** *of the interaction history of agent i, if a joint policy $\tilde{\pi} = \langle \tilde{\pi}_1, \tilde{\pi}_2 \rangle$ is sufficient to maximize the global value, where $\tilde{\pi}_i$ is of the form $S \times E_i \rightarrow A_i \times \Sigma_i$ for a DEC-MDP-Com, or of the form $E_i \rightarrow A_i$ for a DEC-MDP.*

The encoding represents removing all elements from the interaction history that are unnecessary to generate an optimal policy. The important characteristic is the size of the smallest encoding. The interaction history is normally considered to be exponential in $|S|$ because the length of the observation sequence is $O(|S|)$. In some problems, however, the smallest encoding is only polynomial in $|S|$.

Definition 4. *The interaction history of a DEC-MDP/DEC-MDP-Com is* **polynomially encodable** *if there exists an encoding E_i for each interaction history $\overline{\Omega}_i$ and a constant c_i, such that $|E_i| = O(|S|^{c_i})$.*

The criteria that determines the complexity of a multiagent system is whether the interaction history can be polynomially encoded. If it can, then the problem is in NP. If it cannot be polynomially encoded, then it is provably harder than NP. The following two theorems prove this relationship between the encoding and the complexity.

Theorem 2. *Deciding a polynomially encodable DEC-MDP/DEC-MDP-Com is NP-complete.*

Proof. Here we prove the DEC-MDP case, the DEC-MDP-Com is essentially the same.

To prove NP-completeness, we (1) provide a polynomial time verifier and (2) show a reduction from an NP-Complete problem to this one.

(1) A joint policy in a DEC-MDP can be evaluated by representing it as a belief state graph. Each node in the graph is composed of the state, the sequence of observations for agent i and for agent j. Each node has a single joint action, which is defined by the joint policy. The transitions between the nodes depends only on the transition and observation functions, and each transition has a reward defined by the reward function. The belief state graph can be evaluated using the standard MDP recursive value function and policy evaluation, which runs in time polynomial in the size of the graph. For a DEC-MDP, this size is $|S| \times |\Omega_i|^T \times |\Omega_j|^T$, where $T = O(|S|)$. However, since there exists a polynomial encoding E_i for each observation sequence $\overline{\Omega}_i$, the size of the graph is only $|S| \times |E_i| \times |E_j|$ and the policy evaluation takes $O((|S|^{c_i+c_j+1})^c)$, which is polynomial in the size of the state space for constants c_i, c_j and c.

(2) To prove the lower bound we will reduce the NP-complete problem DTEAM [9, 10] to this problem. DTEAM is a single-step discrete team decision problem with two agents. Agent i observes a random integer k_i, $1 \leq k_i \leq N$, and takes an action $\gamma_i(k_i) \in \{1, ..., M\}$. Their actions incur cost $c(k_i, k_j, \gamma_i(k_i), \gamma_j(k_j))$. The problem is to find policies γ_i and γ_j that minimize the expected cost:

$$\sum_{k_i=1}^{N} \sum_{k_j=1}^{N} c(k_i, k_j, \gamma_i(k_i), \gamma_j(k_j)).$$

The reduction is quite straightforward. In the initial state, the agents take a null action and transition to one of N^2 intermediate states that correspond to the random integers k_i and k_j. Agent i observes k_i and takes its action to reach the final state. The reward is negative the cost, $R(\cdot) = -c(\cdot)$. The size of the observation sequence $|\overline{\Omega}_i| = N = O(|S|)$ is polynomial in the size of the state space.

Given the polynomial time verifier and the reduction, a DEC-MDP with a polynomially encodable interaction history is NP-complete. □

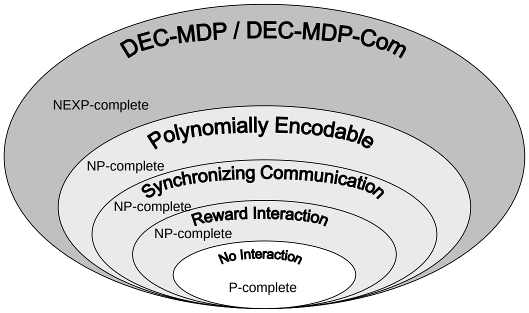

Figure 1: The relationships and complexity between various distributed MDP models.

Theorem 3. *Deciding a non-polynomially encodable DEC-MDP/DEC-MDP-Com is harder than NP.*

Proof. We prove this by contradiction. Assume that an arbitrary DEC-MDP without a polynomial encoding is in NP. This means that there exists a polynomial time verifier for any policy in the DEC-MDP. If a policy can be verified in polynomial time then it must have a polynomial sized representation. Since a policy is a mapping from $E_i \rightarrow A_i$, this polynomial sized representation is the encoding of the interaction history. Contradiction. Therefore, the DEC-MDP without a polynomial encoding is not in NP.

The proof for DEC-MDP-Com is similar. □

Figure 1 illustrates the relationship and complexity between the models discussed in this paper. Polynomially Encodable interaction histories are an NP-complete subset of DEC-MDPs. Other models, like the synchronizing communication are NP-complete subsets of polynomially encodable problems. No interaction between the agents is a P-complete class of problems.

4. EXAMPLES OF PROTOCOLS

In this section, we present two interaction protocols known to be NP-complete, and demonstrate how to prove the existence of a polynomial encodings.

4.1 Reward Dependence

The first example is a DEC-MDP in which the agents are mostly independent of each other. All of their actions are independent actions. The dependence between the agents comes from the reward function which depends on the world state and joint action. Becker et al. [3] formally defined this class of problems as a Transition Independent DEC-MDP (TI-DEC-MDP). It is a DEC-MDP with a factored state space, which means that there is a local state for each agent and the global state is the product of all of the local states, $S = S_i \times S_j$. The transition from one local state to the next depends only on the actions of that agent. Similarly, the observations of agent i depends only on i's local states and actions. We call these properties transition and observation independent.

Definition 5 ([3]). *A factored DEC-MDP is said to be* **transition independent** *if the new local state of each agent depends only on its previous local state and the action taken by that agent:*

$$P(s_i'|(s_i, s_j), (a_i, a_j), s_j') = P_i(s_i'|s_i, a_i).$$

Definition 6 ([3]). *A factored DEC-MDP is said to be* **observation independent** *if the observation an agent sees depends only on that agent's current and next local state and current action:* $\forall o_i \in \Omega_i$

$$P(o_i|(s_i, s_j), (a_i, a_j), (s_i', s_j'), o_j) = P(o_i|s_i, a_i, s_i').$$

At each step each agent fully observes its local state, and observes no information about the local state of the other agent.

Definition 7 ([3]). *A factored DEC-MDP is said to be* **locally fully observable** *if each agent fully observes its own local state at each step., i.e.,* $\forall o_i \ \exists s_i \ : P(s_i|o_i) = 1$.

The interaction history of the DEC-MDP is the sequence of local observations, which in this problem translates into the sequence of local states. However, due to the limited interaction, the most recent local state and the time step is sufficient to maximize the global value. Proving that this encoding is sufficient is the primary component in proving that this class of problems is polynomially encodable.

Theorem 4. *The interaction history of a DEC-MDP with independent transitions and independent observations is polynomially encodable.*

Proof. The interaction history of a transition independent and observation independent DEC-MDP is a sequence of local states $\overline{s}_i$ with an upper bound on the length being $T = O(|S|)$. We will prove that there exists an encoding of the interaction history composed of the last state in the sequence and the length of the sequence. This encoding $E_i = S_i \times T$ is polynomial: $|S_i \times T| = O(|S|^2)$.

The Q value of the belief state graph (see Theorem 2) is the Q value of taking the given action in the current state sequence and then following the given optimal policy $\overline{\pi}^* = \langle \overline{\pi}_1^*, \overline{\pi}_2^* \rangle$, where $\overline{\pi}_i^* : \overline{S}_i \rightarrow A_i$.

$$Q_{\overline{\pi}^*}(\overline{s}_1 s_1, \overline{s}_2 s_2, a_1 a_2) = \sum_{s_1' s_2'} P(s_1' s_2'|\overline{s}_1 s_1, \overline{s}_2 s_2, a_1 a_2) \times$$
$$[R(\overline{s}_1 s_1, \overline{s}_2 s_2, a_1 a_2, s_1' s_2') + \gamma V_{\overline{\pi}^*}(\overline{s}_1 s_1 s_1', \overline{s}_2 s_2 s_2')].$$

If for all possible histories $\overline{s}_1', \overline{s}_2'$ of length T that led to the current state, the Q value is the same, then the history of states is irrelevant and can be replaced in the policy by just the length of the sequence T.

To prove $Q_{\overline{\pi}^*}(\overline{s}_1 s_1, \overline{s}_2 s_2, a_1 a_2) = Q_{\overline{\pi}^*}(\overline{s}_1' s_1, \overline{s}_2' s_2, a_1 a_2)$, we have to show three things:

1. $P(s_1' s_2'|\overline{s}_1 s_1, \overline{s}_2 s_2, a_1 a_2) = P(s_1' s_2'|\overline{s}_1' s_1, \overline{s}_2' s_2, a_1 a_2)$.
From the definition of Markov,

$$P(s_1' s_2'|\overline{s}_1 s_1, \overline{s}_2 s_2, a_1 a_2) = P(s_1' s_2'|s_1 s_2, a_1 a_2). \qquad (1)$$

2. $R(\overline{s}_1 s_1, \overline{s}_2 s_2, a_1 a_2, s_1' s_2') = R(\overline{s}_1' s_1, \overline{s}_2' s_2, a_1 a_2, s_1' s_2')$.
From the definition of reward,

$$R(\overline{s}_1 s_1, \overline{s}_2 s_2, a_1 a_2, s_1' s_2') = R(s_1 s_2, a_1 a_2, s_1' s_2'). \qquad (2)$$

3. $V_{\overline{\pi}^*}(\overline{s}_1 s_1 s_1', \overline{s}_2 s_2 s_2') = V_{\overline{\pi}^*}(\overline{s}_1' s_1 s_1', \overline{s}_2' s_2 s_2')$
We show $V_{\overline{\pi}^*}(\overline{s}_1 s_1, \overline{s}_2 s_2) = V_{\overline{\pi}^*}(\overline{s}_1' s_1, \overline{s}_2' s_2)$ by induction.

Base case: s_1 and s_2 are final states and their values are always zero. $V_{\overline{\pi}^*}(\overline{s}_1 s_1, \overline{s}_2 s_2) = 0$, $V_{\overline{\pi}^*}(\overline{s}_1' s_1, \overline{s}_2' s_2) = 0$.

Inductive case: We assume it is true for

$$V_{\overline{\pi}^*}(\overline{s}_1 s_1 s_1', \overline{s}_2 s_2 s_2') = V_{\overline{\pi}^*}(\overline{s}_1' s_1 s_1', \overline{s}_2' s_2 s_2'), \qquad (3)$$

we need to show that it is true for

$$V_{\overline{\pi}^*}(\overline{s}_1 s_1, \overline{s}_2 s_2) = V_{\overline{\pi}^*}(\overline{s}_1' s_1, \overline{s}_2' s_2).$$

The value function is very similar to the Q function, except the current action is chosen from the policy.

$$\begin{array}{ll} & V_{\overline{\pi}^*}(\overline{s}_1 s_1, \overline{s}_2 s_2) \\ = & \sum_{s_1' s_2'} P(s_1' s_2'|\overline{s}_1 s_1, \overline{s}_2 s_2, a_1 a_2) \times \\ & [R(\overline{s}_1 s_1, \overline{s}_2 s_2, a_1 a_2, s_1' s_2') + \gamma V_{\overline{\pi}^*}(\overline{s}_1 s_1 s_1', \overline{s}_2 s_2 s_2')] \\ \overset{(1),(2)}{=} & \sum_{s_1' s_2'} P(s_1' s_2'|s_1 s_2, a_1 a_2) \times \\ & [R(s_1 s_2, a_1 a_2, s_1' s_2') + \gamma V_{\overline{\pi}^*}(\overline{s}_1 s_1 s_1', \overline{s}_2 s_2 s_2')] \\ \overset{(1),(2),(3)}{=} & \sum_{s_1' s_2'} P(s_1' s_2'|\overline{s}_1' s_1, \overline{s}_2' s_2, a_1 a_2) \times \\ & [R(\overline{s}_1' s_1, \overline{s}_2' s_2, a_1 a_2, s_1' s_2') + \gamma V_{\overline{\pi}^*}(\overline{s}_1' s_1 s_1', \overline{s}_2' s_2 s_2')] \end{array}$$

Since a_1' and a_2' are optimal actions, we have:

$$\begin{array}{ll} & \sum_{s_1' s_2'} P(s_1' s_2'|\overline{s}_1' s_1, \overline{s}_2' s_2, a_1 a_2)[R(\overline{s}_1' s_1, \overline{s}_2' s_2, a_1 a_2, s_1' s_2') \\ & +\gamma V_{\overline{\pi}^*}(\overline{s}_1' s_1 s_1', \overline{s}_2' s_2 s_2')] \\ \leq & \sum_{s_1' s_2'} P(s_1' s_2'|\overline{s}_1' s_1, \overline{s}_2' s_2, a_1' a_2')[R(\overline{s}_1' s_1, \overline{s}_2' s_2, a_1' a_2', s_1' s_2') \\ & +\gamma V_{\overline{\pi}^*}(\overline{s}_1' s_1 s_1', \overline{s}_2' s_2 s_2')] \\ = & V_{\overline{\pi}^*}(\overline{s}_1' s_1, \overline{s}_2' s_2) \end{array}$$

As a result, we have $V_{\overline{\pi}^*}(\overline{s}_1 s_1, \overline{s}_2 s_2) \leq V_{\overline{\pi}^*}(\overline{s}_1' s_1, \overline{s}_2' s_2)$. Due to symmetry, we can also show that $V_{\overline{\pi}^*}(\overline{s}_1' s_1, \overline{s}_2' s_2) \leq V_{\overline{\pi}^*}(\overline{s}_1 s_1, \overline{s}_2 s_2)$. Therefore $V_{\overline{\pi}^*}(\overline{s}_1 s_1, \overline{s}_2 s_2) = V_{\overline{\pi}^*}(\overline{s}_1' s_1, \overline{s}_2' s_2)$.

Since the value of taking a joint action while following the optimal policy does not depend on the history, the same joint action is optimal for all histories and the policy need not include it. The interaction history can be summarized by the current state and time. □

Theorem 4 implies that a DEC-MDP with independent transitions and observations can be decomposed into two independent MDPs, with local states only affected by the local actions. The policies are standard policies for MDPs with the addition of time, and the goal is to maximize the expected reward received from a global value function.

4.2 Synchronizing Communication

In the DEC-MDP with independent transitions and observations presented above, the agents interact only through reward function. This highly restricted form of interaction does not reveal any information about other agent's local state or observations. In this section, we look at a less restricted form of interaction, which we call **synchronizing communication**. We define it in the context of DEC-MDP-Com, since the explicit modeling of communication allows it to remain distinct from the domain actions.

Definition 8. *A communication protocol is said to be a* **synchronizing communication protocol** *if whenever any agent communicates, all agents send sufficient information to all other agents to unify their world views. Such an exchange is viewed as a single communication action with a single cost, even though there are potentially many messages sent.*

In a DEC-MDP-Com with a synchronizing communication protocol, whenever there is communication, each agent has the same view of the world. Since the world state is jointly fully observable, each agent has a complete view of the world state, and knows that the other agent has a complete view as well. The DEC-MDP-Com is essentially reset to an identical problem with a different start state, and the agents are safe to forget their past observation histories and communication histories. The communication actions essentially divide the DEC-MDP-Com into individual episodes, each of which is a DEC-MDP with no communication actions. The length of each episode varies depending on when it is optimal to communicate.

There are many applications in which synchronizing communication is an appropriate protocol. In certain problems, the communication setup cost is so high that it does not matter how much actual information is transferred. In other systems, the minimum packet size sent over the network may be larger than the messages the agents send, giving them a constant cost per message. For applications such as these, the amount of information contained in each message does not change its cost. A communication protocol is said to have **constant cost** if all the communication actions have the same cost. Specifically, a synchronizing communication action has the same cost as any other communication actions, no matter how many messages are actually exchanged to synchronize their partial views of the world state. Goldman and Zilberstein [7] proved that given a DEC-MDP-Com with constant communication cost, there is an optimal communication policy such that whenever there is communication, the agents exchange their last observations. Since a DEC-MDP-Com is jointly fully observable, when the agents exchange their last observations, they synchronize their views of the global state. As a result, if a DEC-MDP-Com has constant communication cost, there is an optimal communication policy such that whenever there is communication between the agents, it is synchronizing communication.

The second example of a polynomially encodable interaction protocol is the synchronizing communication in a DEC-MDP-Com with independent transitions and observations and a constant communication cost. This protocol has been studied in other work [7, 1].

Theorem 5. *The interaction history of a DEC-MDP-Com with independent transitions and observations and constant communication cost is polynomially encodable.*

Proof. From Definition 2, a local policy for a DEC-MDP-Com is of the form $\pi_i : S \times \overline{\Omega}_i \times \overline{\Sigma}_i \times \overline{\Sigma}_j \rightarrow A_i \times \Sigma_i$, where S is the last synchronized state, and $\overline{\Omega}_i$, $\overline{\Sigma}_i$ and $\overline{\Sigma}_j$ are the observation history and communication history since S. Since the DEC-MDP-Com has a constant communication cost, a synchronizing communication protocol is optimal. As a result, whenever there is communication the last synchronized global state S is updated to the newly synchronized state, and $\overline{\Omega}_i$ is set to $\emptyset$. When there is no communication, the observation is appended to $\overline{\Omega}_i$. Neither $\overline{\Sigma}_i$ nor $\overline{\Sigma}_j$ are used. Therefore, between communications the DEC-MDP-Com is equivalent to a DEC-MDP with independent transitions and observations, which is polynomially encodable (from Theorem 4). Since the interaction history between communications is polynomially encodable and communication resets the interaction history to $\emptyset$, the problem is polynomially encodable. □

Even though a DEC-MDP-Com with independent transitions and observations and constant communication cost has considerably more communication than a DEC-MDP with only reward dependence, Theorem 5 shows that its interaction protocol is still polynomially encodable, and therefore it remains in NP.

5. EXAMPLES OF APPROXIMATIONS

For applications where communication cost is constant, one can find an optimal policy that periodically synchronizes the world views of the agents. However, there are many other applications where the cost of message depends on its contents, and it may be beneficial to send less information than would synchronize the agents' world views. Unfortunately, most such interactions do not seem to be polynomially encodable because each piece of information changes an agent's belief about the local state of the other agent. A good guideline is that an agent needs to keep track of both the other agent's local state as well as the other agent's knowledge. However, we may still be able to design approximate encodings that are polynomial for such problems. The purpose of these encodings is to put extra restrictions on the interactions so that the complexity of approximating the optimal policy is reduced to at most NP. In this section, we show two examples of such approximations.

5.1 Constant Horizon

Consider the token collecting example. While there is no explicit communication between the agents, both of the agents observe the location of all available tokens. When agent i picks up a token at time t, agent j observes the fact that a token disappeared and can infer the location of agent

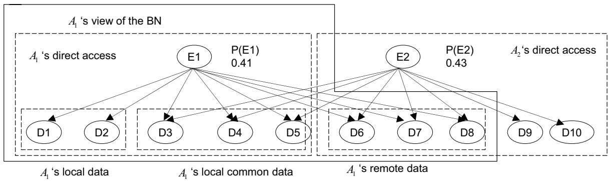

Figure 2: There are two events E_1 and E_2. Data $D_1, D_2, ...D_{10}$ are distributed between two agents. A_1 has access to $D_1, ...D_5$ and is responsible for solving E_1, while A_2 can see only $D_6, ...D_{10}$ and is responsible for solving E_2. The objective is for A_1 and A_2 to figure out what E_1 and E_2 are with required confidence and with minimum expected communication cost

i. With this observation, agent j knows the global state but agent i does not, so the interaction is not synchronizing but what we call asymmetric synchronizing. Asymmetric synchronization is where an agent either gains no information about the other agent's local state or complete information, giving it a belief of 1.0 about the current global state. The difference between synchronization and asymmetric synchronization is in the asymmetric case the agents do not necessarily both synchronize at the same time.

This difference precludes a polynomial encoding. In the token collecting example when agent i picks up a token both agents i and j know that i picked it up as well as both know the other knows, and so on. Agent i, then, must keep track of the information j has collected about i to best predict what j will do at least until j picks up a token itself and i learns where j is located. Remembering this sequence of information is exponential in the size of the state space in the worst case.

While the problem itself is harder than NP, it can be approximated by making assumptions that allow the interaction history to be polynomially encoded. For example, instead of keeping track of the entire history of interaction, one could assume that the last c interactions, for some constant c, was sufficient. The interaction history is now of size $|S|^c$, which is polynomial in the size of the state space. In the token collecting example this could correspond to agent i remembering the last 5 tokens it collected.

5.2 Flat Representation

Now let us look at a communication optimization problem in a Distributed Sensor Interpretation (DSI) system [13]. There are two agents in the system. Each agent directly observes data from a group of sensors and is responsible for solving a subset of the interpretation problem to best explain the sensor data collectively observed by the agents. Since each subproblem may relate to not only the sensor data collected by the agent itself, but also those collected by the other agent, the agents may need to communicate with each other in order to solve its local problem with required confidence. Figure 2 illustrates such a DSI system that is represented in a Bayesian Network. The top level nodes are the events that are the possible causes of the observed data, while the leaves are the raw data gathered by various agents. There are two classes of actions for each agent: SEND and REQUEST. An agent can only send one piece of data that the other agent has no knowledge of, and can only request one piece of the remote data that it has not yet acquired. The two agents alternate actions until the confidence threshold is reached by both. The communication optimization problem is, for the specified BN structure, to find the communication strategy whose expected cost is minimized in order to reach the required confidence for both agents.

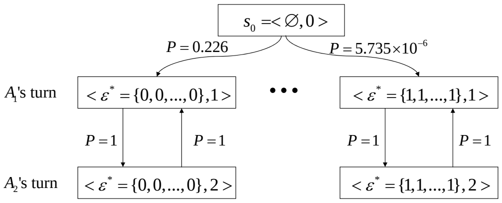
$s_0 =< \varnothing, 0 >$
$P = 0.226$
$P = 5.735\times10^{-6}$
A_1's turn
$< \varepsilon^* = \{0, 0, ..., 0\}, 1 >$
• • •
$< \varepsilon^* = \{1,1, ...,1\}, 1 >$
$P = 1$
$P = 1$
$P = 1$
$P = 1$
A_2's turn
$< \varepsilon^* = \{0, 0, ..., 0\}, 2 >$
$< \varepsilon^* = \{1,1, ...,1\}, 2 >$

Figure 3: The decentralized MDP generated for the problem in Figure 2.

Shen et al. [13] models this problem with a DEC-MDP as shown in Figure 3. Every global state is in the form of $s = \langle e^*, i\rangle$, where e^* is all of the sensor data values observed by the system, and i is an external feature that indicates it is agent i's turn to communicate. s_0 is a dummy start state $\langle \emptyset, 0\rangle$. When all the sensors collect their data, the start state transitions to one of the possible real global states $\langle e^*, 1\rangle$ before any communication takes place. A_1 and A_2 are action sets of the two agents, $A_i = \{$SEND x, REQUEST $y\}$ for local data x and remote data y. An observation o_i is the data value just sent or received by agent i. As a special case, after s_0 each agent observes the values of its local data.

In this DEC-MDP every action is explicit communication. Instead of explicitly modeling it as Σ in a DEC-MDP-Com, these actions are implicitly modeled in the DEC-MDP framework itself. The observation history $\overline{\Omega}$ records the interaction history of the system. Since every observation at every time step is needed to calculate the confidence level achieved, it is necessary for the agents to remember all the data values transferred between the agents in the past. Furthermore, remembering the order in which the data are exchanged is valuable because it carries useful information. An agent can infer why the other agent chose to transfer this piece of data before the other piece. Therefore, remembering the entire $\overline{\Omega}$ is essential to generating the optimal policy. We write the number of local sensor data as n, and assume that each sensor data has at most m possible values. As a result, $|S| = m^n$, and $|\overline{\Omega}| = O(n! \cdot m^n) = O(n^n) = O(|S|^{\log_2 n})$. Since n is not independent of $|S|$, $\overline{\Omega}$ is not polynomially encodable and therefore the DEC-MDP is harder than NP.

One way to approximate the optimal solution to this problem is that, instead of remembering the entire $\overline{\Omega}$, each agent only remembers the values of the data exchanged so far without remembering the order in which they were transferred. This is a reasonable approximation since to calculate the confidence level of the local interpretations, only the sensor data values are needed. In this approximation, the approximate encoding E_i of $\overline{\Omega}$ is of the size $O((m+1)^n) = O(|S|^{\log_m(m+1)})$. Since m is independent of $|S|$, E_i is a polynomial encoding, and therefore this approximation is no harder than NP.

6. CONCLUSIONS

Distributed POMDPs have been developed and employed to model various multi-agent coordination problems. Understanding the source of their high complexity is crucial to identifying new and more tractable models as well as developing appropriate approximations to otherwise intractable problems. This paper establishes that the interactions present among the agents is the cause of the high complexity of distributed POMDPs. We proved that deciding a distributed POMDP whose interaction history contains information of a size polynomial in the number of states is NP-complete, and that deciding a non-polynomially encodable distributed POMDP is harder than NP. We demonstrated how two subclasses of distributed POMDPs known to be NP-complete can be polynomially encoded. This is the first time that a well-defined condition has been identified that can distinguish between multi-agent problems in NP and those that are strictly harder than NP. It is an important step in mapping out the complexity hierarchy of multi-agent systems.

Our goal in this paper was not to introduce new models or algorithms, but to change the way people view interactions between agents in the context of distributed POMDPs. Multi-agent researchers have long intuitively understood that the interaction between the agents is the cause of their high complexity. The theoretical results of this paper are significant in that they both formally justify this intuition as well as explain how the interaction affects the complexity. This new understanding of interaction and its relationship to complexity will help us to identify new classes of multi-agent systems with a lower complexity.

The significance of this theoretical result also has a more practical side. Most multi-agent systems are provably harder than NP and solving them optimally is not possible. Much work has been put into developing good algorithms for approximating these problems. This work provides theoretical guidance in understanding how the approximations in a model limit the search space and reduce the complexity. We demonstrate this on two non-polynomially encodable problems by providing two assumptions that reduce the complexity to no harder than NP.

We are currently pushing this work in two directions. First, we are searching for new examples of polynomially encodable interaction protocols, as well as common protocols that we can prove are not polynomially encodable. Second, we are evaluating the performance of the approximations we presented here in addition to finding new approximations appropriate to the protocols that we can prove to be harder than NP.

7. ACKNOWLEDGMENTS

This work was supported in part by NASA under grant NCC 2-1311, by NSF under grants IIS-0219606 and IIS-0414711, and by AFOSR under grant F49620-03-1-0090. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of NASA, NSF or AFOSR.

8. REFERENCES

[1] R. Becker, V. Lesser, and S. Zilberstein. Analyzing myopic approaches for multi-agent communication. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 550–557, Compiegne, France, September 2005. IEEE Computer Society.

[2] R. Becker, S. Zilberstein, and V. Lesser. Decentralized Markov decision processes with structured transitions. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, volume 1, pages 302–309, New York City, New York, July 2004. ACM Press.

[3] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Solving transition independent decentralized MDPs. *Journal of Artificial Intelligence Research*, 22:423–455, 2004.

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

[5] M. Ghavamzadeh and S. Mahadevan. Learning to communicate and act in cooperative multiagent systems using hierarchical reinforcement learning. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 1114–1121, New York, July 2004. ACM Press.

[6] C. V. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 137–144, Melbourne, Australia, July 2003. ACM Press.

[7] C. V. Goldman and S. Zilberstein. Decentralized control of cooperative systems: Categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 22:143–174, 2004.

[8] H. Li, E. H. Durfee, and K. G. Shin. Multiagent planning for agents with internal execution resource constraints. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 560–567, New York, NY, USA, 2003. ACM Press.

[9] C. H. Papadimitriou and J. Tsitsiklis. On the complexity of designing distributed protocols. *Information and Control*, 53:211–218, 1982.

[10] C. H. Papadimitriou and J. Tsitsiklis. Intractable problems in control theory. *SIAM Journal on Control and Optimization*, 24(4):639–654, 1986.

[11] L. Peshkin, K.-E. Kim, N. Meuleau, and L. P. Kaelbling. Learning to cooperate via policy search. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 489–496, San Francisco, CA, 2000. Morgan Kaufmann.

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

[13] J. Shen, V. Lesser, and N. Carver. Minimizing communication cost in a distributed Bayesian network using a decentralized MDP. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 678–685, Melbourne, Australia, July 2003. ACM Press.

[14] P. Xuan and V. Lesser. Multi-agent policies: From centralized ones to decentralized ones. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 1098–1105. ACM Press, 2002.