Decentralized Control of Cooperative Agents

Claudia V. Goldman CLAG@CS.UMASS.EDU
Shlomo Zilberstein SHLOMO@CS.UMASS.EDU
Department of Computer Science,
University of Massachusetts Amherst, MA 01003 USA

UMass Computer Science Technical Report #03-36

Abstract

Decentralized control of a cooperative system is the problem faced by a group of decision makers who share a single global objective function. The difficulty in solving optimally such problems arises when the agents lack full observability of the global state of the system at execution time. The general problem has been shown to be NEXP-complete. In this paper, we identify classes of decentralized control problems whose complexity ranges between NEXP and P. In particular, we study problems characterized by independent transitions, independent observations, and goal-oriented objective functions. Two algorithms are shown to solve optimally useful classes of goal-oriented decentralized processes in polynomial time.

The second part of the paper focuses on information sharing among the decision-makers, which can improve their performance. We distinguish between three ways in which agents can exchange information: indirect communication, direct communication and sharing common features that are uncontrollable by the agents. Our analysis shows that for every class of problems we consider, introducing direct or indirect communication does not change the worst-case complexity. Finally, a general approximation scheme is introduced to solve decentralized control problems with direct communication. The approach is illustrated and evaluated using two polynomial algorithms (myopic-greedy and backward induction). These results offer one of the first practical approaches to address the complexity of decentralized control with communication.

1. Introduction

Markov decision processes have been widely studied as a mathematical framework for sequential decision making in stochastic domains. In particular, single-agent planning problems in stochastic domains were modeled as partially observable Markov decision processes (POMDPs) or fully-observable MDPs (Dean, Kaelbling, Kirman, & Nicholson, 1995; Kaelbling, Littman, & Cassandra, 1998; Boutilier, Dearden, & Goldszmidt, 1995). Borrowing from Operations Research techniques, optimal plans can be computed by solving the corresponding Markov decision problem. There has been a vast amount of progress in solving individual MDPs by exploiting domain structure (e.g., (Boutilier et al., 1995; Feng & Hansen., 2002)). Approximations of MDPs have also been studied, for example, Guestrin et al. (Guestrin, Koller, Parr, & Venkataraman, 2003) assuming that the reward of the system can be decomposed into local reward functions each depending on only a small set of variables.

We are interested in a single Markov decision process that is collaboratively controlled by multiple decision-makers. The group of agents cooperate in the sense that they all want to maximize a global objective (or minimize the cost of achieving it). Nevertheless, the decision-makers do not have full observability of the whole system at the time of execution. These processes can be found in many practical applications such as multi-robot problems, flexible manufacturing and information gathering. These robots or software agents need to compute the sequence of actions that when performed will optimize some global objective. For example, a set of robots like the rovers sent by NASA to Mars[1] may be assigned a list of experiments to carry out on the planet before meeting, as soon as possible. The robots, then, need to decide what experiments to perform and how much time they should invest on each one given the constraints of their battery's life and the time remaining to meet. Another example is found in information gathering systems. Assume a user submits a query, and several software agents have access to different servers that may provide answers. These agents' global objective is to give the user the answer with the highest quality as early as possible, given the load on their servers, and maybe preferences given by the user. These decisions are not trivial since these agents face uncertainty of the environment (e.g., the load on the communication connection between the servers in the information gathering example may behave stochastically), of their actions' outcomes (e.g., the rovers in Mars may encounter inaccuracies in their movements or hardware), and of the other agents' actions and observations. All these types of uncertainty are taken into account when solving such decentralized problem.

These processes are called decentralized partially-observable Markov decision processes (Dec-POMDPs) or decentralized Markov decision processes (Dec-MDPs)[2]. The complexity of solving these problems has been studied recently (Bernstein, Givan, Immerman, & Zilberstein, 2002; Pynadath & Tambe, 2002). Bernstein et al. showed that solving optimally a Dec-MDP is NEXP-complete by reducing the control problem to the tiling problem. Rabinovich et al. (Rabinovich, Goldman, & Rosenschein, 2003) have shown that even approximating the off-line optimal solution of a Dec-MDP remains NEXP. Researchers have attempted to *approximate* the coordination problem by proposing *on-line* learning procedures. Peshkin et al. (Peshkin, Kim, Meuleau, & Kaelbling, 2000) have studied how to approximate the decentralized solution based on a gradient descent approach for on-line learning (when the model is not known by the agents). Schneider et al. (Schneider, Wong, Moore, & Riedmiller, 1999) assume that each decision maker is assigned a local optimization problem. Their analysis is how to approximate the global optimal value function. Agents may exchange information about their local values at no cost. Neither convergence nor bounds are given. Wolpert et al. (Wolpert, Wheeler, & Tumer, 1999) assume that each agent runs a predetermined reinforcement learning algorithm, and transform the coordination problem into how to update the local reward functions to maximize the global reward function. Again, this algorithm is for on-line learning, it is an approximation, agents may communicate at no cost, and no convergence or bounds are given. Studies of off-line approximations were done by Guestrin et al. (a centralized approach (Guestrin, Koller, & Parr, 2001), and a distributed approach (Guestrin & Gordon, 2002)), where a known structure of the agents' action dependencies is assumed that induces a message passing structure (at

1. mars.jpl.nasa.gov/mer/
2. These problems will be accurately defined in Section 2 and Definition 4.

no cost). They assume that the value function of the system can be represented by a set of compact basis functions, which they approximate. The complexity of the algorithm is exponential in the width of the coordination graph. The order of elimination is needed beforehand because it has a great effect on the result. The agents choose their actions in turns.

These works have departed from the assumption that each agent has a known local reward function. The questions that they attempt to answer, hence, take the form of how to design or manipulate these local functions to approximate the actual system reward function.

We are interested in solving the decentralized control problem off-line without assuming any particular assumptions on the rewards of each agent, i.e., the problem is analyzed from a decentralized perspective. We have developed a theoretical formal model for decentralized control extending current models based on Markov Decision Processes. We refer to the most general problem where information sharing between the agents can result from indirect communication (i.e., via observations), by direct communication (i.e., via messages) or by sharing common uncontrollable features (that will be precisely defined in Section 2.2). When direct communication is possible, we assume that communication incurs a cost. On the one hand, communication can assist the agents to better control the process; on the other hand, communication may not be possible or desirable at every moment. Exchanging information may incur a cost associated with the required bandwidth, the risk of revealing it to competing agents or the complexity of solving an additional problem related to the communication (e.g., computing the messages). Assuming that communication may not be reliable adds another dimension of complexity to the problem.

Becker et al. (Becker, Zilberstein, Lesser, & Goldman, 2003) presented the first algorithm for optimal off-line decentralized control when a certain structure of the joint reward was assumed. There is no known efficient algorithm (short of evaluating all policies) to date that can solve the problem of control and communication optimally. Pynadath and Tambe (Pynadath & Tambe, 2002) studied a similar model to ours, although they did not propose an algorithm for solving the decentralized control problem. Claus and Boutilier (Claus & Boutilier, 1998) studied a simple case of decentralized control where agents share information about each other's actions during the off-line planning stage. The solution presented in their example includes a joint policy of a single action for each agent to be followed in a stateless environment. The agents learn which equilibrium to play. In our model, partial observability is assumed and the scenarios studied are more complex and include multiple states. Centralized multi-agent systems (MAS) were also studied in the framework of MDPs (e.g., (Boutilier, 1999)), where both the off-line planning stage and the on-line stage are controlled by a central entity, or by all the agents in the system, who have full observability.

Our work focuses on decentralized cooperative MAS. Agents in most cooperative MAS are limited by not being able to fully communicate during execution (due to the distributed nature of the system). Due to the cooperative nature of the MAS, in many situations these constraints do not apply to the pre-execution stage. Thus, cooperative agents are able to share information at the off-line planning stage as if they were centrally controlled. But unlike the centralized approach, these agents will be acting in real-time in a decentralized manner. The agents must take this into account while planning off-line. We do not as-

sume the existence of local processes or local rewards; we analyze the problems from the decentralized perspective.

Sub-classes of Dec-POMDPs can be characterized based on how the global states, transition function, observation function, and reward function relate to the partial view of each of the controlling agents. In the simplest case, the global states can be factored, the probability of transitions and observations are independent, the observations combined determine the global state, and the reward function can be easily defined as the sum of local reward functions. In this extreme case we can say that the Dec-POMDP is equivalent to the combination of n independent MDPs. This simple case is solvable by combining all the optimal solutions of the independent MDPs. We are interested in more complex Dec-POMDPs, in which some or all of these assumptions are violated. In particular, in the first part of this paper, we characterize Dec-POMDPs, which may be jointly fully-observable, may have independent transitions and observations and may result in goal-oriented behavior. We analyze the complexity of solving these classes off-line and optimally, and reveal interesting results on the complexities of the different classes which ranges from NEXP to P. We also identify different forms of information sharing. In particular, we prove, that when direct communication is possible, exchanging local observations is sufficient to attain optimal decentralized control.

In the second part of this paper, we study decentralized control with direct communication. We develop a practical approximation technique based on meta-level control of communication, motivated by a similar decision-theoretic approach to meta-level reasoning that was developed by Russell and Wefald (Russell & Wefald, 1991). We assume that the designer of the system also designs a mechanism for communication. This mechanism stipulates how to decompose the global problem into local (single-agent) and temporary problems (e.g., each agent is assigned an individual Markov decision problem). Each decision-maker computes its policy of communication that when executed, will lead to the synchronization of the agents' information, i.e., the global state of the system becomes fully observable as a result of communication. Each time the global information is known, the mechanism is applied to allow the agents to work independently, until the policy of communication instructs them to synchronize their information again.

The contributions of the paper are as follows: formalizing the decentralized control problem with information sharing (Section 2), identifying classes of decentralized control that are critical in decreasing the NEXP complexity (Section 3), designing algorithms for controlling optimally a decentralized process with goal-oriented behavior (Section 4), designing an algorithm for optimizing the control and the exchange of information in a decentralized problem (Section 5), developing a practical approximation scheme to solve this problem by decomposing it into temporary smaller problems, which each decision-maker can solve independently and optimally (Section 6). Overall, these results offer a comprehensive approach to cooperative systems that are composed of communicative agents.

2. The Dec-POMDP model

We are interested in a stochastic process that is cooperatively controlled by a group of decision-makers who lack a central view of the global state. Nevertheless, these agents share a set of objectives and all of them are interested in maximizing the utility of the

system. The process is decentralized because none of the agents can control the whole process, and neither of the agents has a full view of the global state. The formal framework in which we study such decentrally controlled processes, named Dec-POMDP is presented below (originally presented in (Bernstein et al., 2002)). For simplicity of exposition, the formal model is presented for two agents, although it can be extended to any number of agents.

$M =< S, A_1, A_2, P, R, \Omega_1, \Omega_2, O, T >$ where:

- S is a finite set of world states, i.e., the global states of the decentralized process. s^0 is the initial state of the system.
- A_1 and A_2 are finite sets of control actions. a_i denotes the action performed by agent i.
- P is the transition probability function. $P(s'|s, a_1, a_2)$ is the probability of moving from state $s \in S$ to state $s' \in S$ when agents 1 and 2 perform actions a_1 and a_2 respectively.
- R is the global reward function. $R(s, a_1, a_2, s')$ represents the reward obtained by the system as a whole, when agent 1 executes action a_1 and agent 2 executes action a_2 in state s resulting in a transition to state s'.
- Ω_1 and Ω_2 are finite sets of observations.
- O is the observation function. $O(o_1, o_2|s, a_1, a_2, s')$ is the probability of observing o_1 and o_2 (respectively by the two agents) when in state s agent 1 takes action a_1 and agent 2 takes action a_2, resulting is state s'.
- If the Dec-POMDP has a finite horizon, it is represented by a positive integer T.

We will illustrate our definitions and results through the Meeting under Uncertainty example. In this scenario, we assume for simplicity that there are two robots acting in a two-dimensional grid. The state of the system is given by the locations of each one of the robots, $s = [(x_1, y_1)(x_2, y_2)]$. The robots cannot recognize each other, and the movement actions they can perform have uncertain outcomes (e.g., with probability P the robot will successfully move to the next location, and with probability $1 - P$ the robot will remain at the same location where it took the action). The robots' objective is to minimize the time to meet. The observation of robot i corresponds to i's x and y coordinates. Solving optimally such a decentralized problem means to find the sequence of moves for each agent such that they meet as soon as possible.

Given the Dec-POMDP model, a *local policy* of action for a single agent is given by a mapping from sequences of observations to actions. In our example, a robot's local policy instructs it to take a certain movement action given the sequence of locations it has observed so far. A *joint policy* is a tuple composed of these local policies, one for each agent. To solve a decentralized POMDP problem is, then, to find the optimal joint policy, that is, the one with maximum value (for example given by the maximum expected accumulated global reward). Notice that the agents' observations can be dependent on each other allowing the agents to *know* what other agents are observing and in some sense enabling the agents

to obtain full observability of the system state. That is, even though the agents may not communicate directly, when the observations are dependent, agents may be able to obtain information about the others' without receiving direct messages. For example, assume that in our scenario there are certain locations which can host only one robot at a time. If one robot observes that it is located at anyone of these sites then it knows that the other robot cannot be located there even though this robot does not actually see the other nor receive any information from it.

In the next section, we characterize certain properties that a decentralized process may have. These properties will play an important role when analyzing the complexity of solving different classes of decentrally controlled cooperative problems.

2.1 Classes of Dec-POMDPs

It is known that solving optimally *general* decentralized problems is very hard (Bernstein et al. (Bernstein et al., 2002) showed that these problems are NEXP-complete). We are interested in classifying the general problem into classes of decentralized problems with certain characteristics. As we show in Section 3, this classification reveals interesting complexity results that correspond to easier problems. The first two categories that we define involve independency of the transitions or the observations of the agents. These are figuratively shown in Figure 1. Notice that there are no arrows connecting s'_1 to s'_2 nor o_1 to o_2.

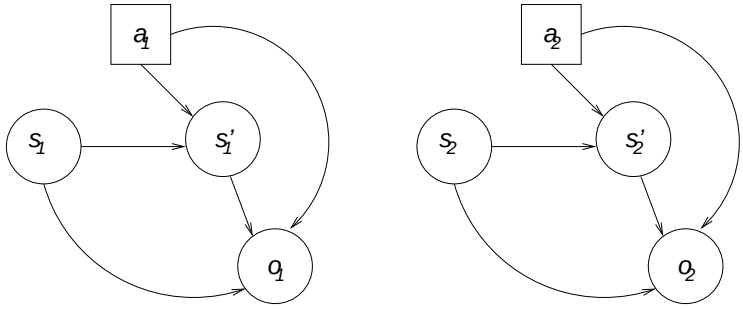

Figure 1: Independent Transitions and observations.

The formal definitions for decentralized processes with independent transitions and observations follow.

Definition 1 (A Dec-POMDP with Independent Transitions) *A Dec-POMDP has independent transitions if the set S of states can be factored into two components S_1 and S_2 such that:*

$$\forall s_1, s'_1 \in S_1, \forall s_2, s'_2 \in S_2, \forall a_1 \in A_1, \forall a_2 \in A_2,$$
$$Pr(s'_1|(s_1, s_2), a_1, a_2, s'_2) = Pr(s'_1|s_1, a_1) \wedge$$
$$\forall s_2, s'_2 \in S_2, \forall s_1, s'_1 \in S_1, \forall a_1 \in A_1, \forall a_2 \in A_2,$$
$$Pr(s'_2|(s_1, s_2), a_1, a_2, s'_1) = Pr(s'_2|s_2, a_2).$$

In other words, the transition probability P of the Dec-POMDP can be represented as $P = P_1 \times P_2$, where $P_1 = Pr(s'_1|s_1, a_1)$ and $P_2 = Pr(s'_2|s_2, a_2)$.

In the Meeting under Uncertainty example, if both robots are tied up to the one rope (e.g., connecting them to the spaceship) then each movement performed by a robot may *pull* the other robot, eventually affecting the next location of the other robot. In this case, the transitions are dependent. On the other hand, if each robot's location is affected only by its own movement action and previous location, then the transitions are independent.

Moreover, the observations of the agents can be independent, i.e., each agent's own observations are independent of the other agents' actions.

Definition 2 (A Dec-POMDP with Independent Observations) *A Dec-POMDP has independent observations if the set S of states can be factored into two components S_1 and S_2 such that:*

$$\forall o_1 \in \Omega_1, \forall s=(s_1,s_2), s'=(s'_1,s'_2) \in S, \forall a_1 \in A_1, \forall a_2 \in A_2, \forall o_2 \in \Omega_2,$$
$$Pr(o_1|(s_1,s_2),a_1,a_2,(s'_1,s'_2),o_2) = Pr(o_1|s_1,a_1,s'_1) \wedge$$
$$\forall o_2 \in \Omega_2, \forall s=(s_1,s_2), s'=(s'_1,s'_2) \in S, \forall a_1 \in A_1, \forall a_2 \in A_2, \forall o_1 \in \Omega_1,$$
$$Pr(o_2|(s_1,s_2),a_1,a_2,(s'_1,s'_2),o_1) = Pr(o_2|s_2,a_2,s'_2)$$
$$O(o_1,o_2|(s_1,s_2),a_1,a_2,(s'_1,s'_2)) = Pr(o_1|(s_1,s_2),a_1,a_2,(s'_1,s'_2),o_2) \times Pr(o_2|(s_1,s_2),a_1,a_2,(s'_1,s'_2),o_1).$$

In other words, the observation probability O of the Dec-POMDP can be decomposed into two observation probabilities O_1 and O_2, such that $O_1 = Pr(o_1|(s_1,s_2),a_1,a_2,(s'_1,s'_2),o_2)$ and $O_2 = Pr(o_2|(s_1,s_2),a_1,a_2,(s'_1,s'_2),o_1)$.

In the Meeting under Uncertainty example, if the robot's observation of its current location depends only on its transition from its previous location and on the action it performed then the observations are independent. But, more complex problems can arise if each agent's observation depends also on the other agent's location, or action. For example, assume a 3D version of the Meeting scenario where the robots can jump in addition to moving. Assume that performing a jumping action causes vibrations on the floor that may eventually change the location of the other agent. In such a case, the observation of one robot's location depends also on the effects of the jumping action performed by the other robot.

Throughout the paper, when we refer to a Dec-POMDP with independent transitions and observations we assume the same decomposition of the global states into S_1 and S_2. We refer to S_i as the *partial view* of agent i.

There are cases where agents may observe some common features of the global state, leading to dependent observations, although the information may be irrelevant to the agents, i.e., information that does not have any effect on the outcomes of the agents' actions and on the reward that the system obtains. Then, such problems can be reformulated to satisfy the property of independent observations. Assuming that only relevant information is handled by the agents, reduces the complexity of the problem as will be shown in Section 3.

One of the main difficulties in solving optimally Dec-POMDPs results from the fact that in such a model neither of the agents may have full observability of the complete global state. An agent has full-observability if it can determine with certainty the global state of the world from its local observation. For example, each time a robot observes where it is located, it also observes the other robot's location. Knowing both locations enables both agents to make the optimal decision as to where to move next to eventually meet earlier.

Definition 3 *A Dec-POMDP is* fully-observable *if there exists a mapping for each agent i, $F_i : \Omega_i \rightarrow S$ such that whenever $O(o_1, o_2|s, a_1, a_2, s')$ is non-zero then $F_i(o_i) = s'$.*

This paper analyzes decentralized problems where full-observability is not possible. Instead, we distinguish between two classes of problems in which we are able to restrict the total lack of observability of the system: 1) when *combining* both agents' partial views leads to the complete global state, and 2) when each agent's own partial view is fully observable. In the first case, even though neither of the agents has information about the other agent's partial view, each agent can assume that whatever it does not know about the global state is necessarily known by the other agent. We denote a Dec-POMDP with such property, a Dec-POMDP that is *jointly fully-observable*.

Definition 4 *A Dec-POMDP is* jointly fully-observable *(also referred to as Dec-MDP) if there exists a mapping $J : \Omega_1 \times \Omega_2 \rightarrow S$ such that whenever $O(o_1, o_2|s, a_1, a_2, s')$ is non-zero then $J(o_1, o_2) = s'$.*

Notice that both definitions 1 and 2 apply to Dec-MDPs as well as to Dec-POMDPs.

The Meeting under Uncertainty scenario as we presented in Section 2 is actually a Dec-MDP. The global state is given by the two pairs of coordinates. There is no other feature in the system state that is hidden from the agents. Notice, that even though the combination of the agents' observations result in the global state, each agent may still be uncertain about its own current partial view, it may have a belief where it is located. We define another class of problems where each agent is certain about its observations. These Dec-MDPs are referred as *locally fully-observable*. General Dec-MDPs consider only the *combination* of the agents' observations, but the definition does not say anything about *each* agent's observation.

Definition 5 *A Dec-POMDP with independent transitions is* locally fully-observable *if there exists a mapping for each agent i ($i = \{1, 2\}$), $L_i : \Omega_i \rightarrow S_i$ such that whenever $O(o_1, o_2|s, a_1, a_2, s')$ is non-zero then $L_1(o_1) = s'_1$ and $L_2(o_2) = s'_2$, where $s_i, s'_i \in S_i$ are the partial views of agent i.*

The Meeting under Uncertainty example is locally fully-observable, each robot knows with certainty where it is located. We may think of more realistic robots where due to hardware inaccuracies, there may be some error with respect to the robot's actual location.

Notice that a jointly fully-observable process which is also locally fully-observable is not necessarily fully-observable. In decentralized control problems, as studied in this paper, we do not have full observability of the system, at most the agents' observations combined determine with certainty the global state, and each such observation determines with certainty the partial view of each agent. The Meeting scenario is jointly fully-observable and locally fully-observable, but none of the agents know the complete state of the system.

Our next lemmas show interesting results concerning the relations between the classes identified so far. These lemmas will serve as the basis for reducing the complexity of solving certain Dec-POMDPs as shown in Section 3. The classes distinguished so far correspond to practical real-world scenarios, as multi-rover scenarios, multi-agent mapping,

and manufacturing where loosely-coupled robots can act in order to achieve a global objective. These problems may include dependencies in the reward structure, and they are not fully-observable.

Lemma 1 *If a Dec-POMDP is jointly fully-observable and it has independent observations and transitions then the Dec-POMDP is* locally fully-observable.

Proof. First, we show that in a Dec-MDP with independent transitions and observations, the following holds:

$$\forall s_1, s_1' \in S_1, s_2, s_2' \in S_2, a_1 \in A_1, a_2 \in A_2, o_1 \in \Omega_1, o_2 \in \Omega_2$$

$$Pr(s_1'|s_1, s_2, a_1, a_2, s_2', o_1, o_2) = Pr(s_1'|s_1, a_1, o_1)$$

Applying Bayes rule to $Pr(s_1'|s_1, a_1, o_1)$ (assuming all the parameters are quantified), we obtain:

$$Pr(s_1'|s_1, a_1, o_1) = Pr(o_1|s_1, a_1, s_1')Pr(s_1'|s_1, a_1)/Pr(o_1|s_1, a_1)$$

For any set of values for s_1, a_1 and o_1, the denominator is a constant, which we will refer as a normalizing factor α. Due to the independent transitions characteristic of the Dec-MDP:

$$Pr(s_1'|s_1, a_1, o_1) = \alpha Pr(o_1|s_1, a_1, s_1')Pr(s_1'|s_1, s_2, a_1, a_2, s_2')$$

Due to the independent observations characteristic of the Dec-MDP:

$$Pr(s_1'|s_1, a_1, o_1) = \alpha Pr(o_1|s_1, s_2, a_1, a_2, s_1', s_2', o_2)Pr(s_1'|s_1, s_2, a_1, a_2, s_2')$$

We also know that s_1' is independent of s_2, a_2 and s_2', given s_1, a_1 and s_1' because the Dec-MDP has independent transitions. The observation o_2 depends only on s_2, a_2 and s_2' because the Dec-MDP has independent observations. Therefore, s_1' cannot depend on o_2, given s_1, a_1 and s_1'. It follows that $Pr(s_1'|s_1, s_2, a_1, a_2, s_2') = Pr(s_1'|s_1, s_2, a_1, a_2, s_2', o_2)$.

So, after applying Bayes rule again, we obtain:

$$Pr(s_1'|s_1, a_1, o_1) = \alpha Pr(o_1|s_1, s_2, a_1, a_2, s_1', s_2', o_2)Pr(s_1'|s_1, s_2, a_1, a_2, s_2', o_2) =$$

$$= \alpha Pr(s_1'|s_1, s_2, a_1, a_2, s_2', o_1, o_2)$$

The lemma assumes a Dec-MDP, that is: $Pr(s'|o_1, o_2) = 1$. Since this probability is one, it is also true that $Pr(s'|o_1, o_2, s, a_1, a_2) = 1$. The lemma assumes independent transitions and observations, therefore the set of states is factored. Following conditional probabilities rules, we obtain:

$$1 = Pr(s_1', s_2'|o_1, o_2, s_1, s_2, a_1, a_2) = Pr(s_1'|o_1, o_2, s_1, s_2, a_1, a_2, s_2')Pr(s_2'|o_1, o_2, s_1, s_2, a_1, a_2)$$

Following our first result shown in this proof:

$$1 = Pr(s_1', s_2'|o_1, o_2, s_1, s_2, a_1, a_2) = Pr(s_1'|o_1, s_1, a_1)Pr(s_2'|o_2, s_2, a_2)$$

So, each agent's partial view is determined with certainty by its observation and own transition, i.e., the Dec-MDP is locally fully-observable.

□

From this lemma, we obtain that a local policy for agent i in a locally fully-observable Dec-MDP is a mapping from sequences of states in agent i's partial view to actions, as opposed to a mapping from sequences of *observations* to actions as in the general Dec-MDP case. Formally, $\delta_i : S_i^* \rightarrow A_i$ where S_i corresponds to the decomposition of global states assumed in definitions 1 and 2 for Dec-MDPs with independent transitions and observations.

Moreover, we can show that an agent does not need to map a *sequence* of partial views to actions, but it is sufficient to remember only the current partial view. This is shown in the next lemma.

Lemma 2 *The current partial view of a state s observed by agent i (s_i) is a sufficient statistic for the past history of observations ($\overline{o}_i$) of a locally fully-observable Dec-MDP with independent transitions and observations.*

Proof. Without loss of generality we do all the computations for agent 1. We define I_t^1 as all the information available to agent 1 about the Dec-MDP process at the end of the control interval t similarly to Smallwood and Sondik's original proof for classical POMDPs (Smallwood & Sondik, 1973). I_t^1 is given by the action a_{1_t} that agent 1 chose to perform at time t, the current resulting state s_{1_t}, which is fully-observable by agent 1 ($s_{1_t} = i_1$) and the previous information I_{t-1}^1. We assume a certain policy for agent 2, π_2 is known and fixed. $\pi_2(s_t)$ is the action taken by agent 2 at the end of control interval t.

We compute the belief-state of agent 1, that is the probability that the system is at global state j assuming only the information available to agent 1 (I_t^1).This computation tells us how to build a belief-state MDP for agent 1. Agent 1's optimal local policy is the solution that obtains the highest value over all the solutions resulting from solving all the belief-state MDPs built for each possible policy for agent 2.

We compute the probability that the system is in state $s_t = j = (j_1, j_2)$ at time t, given the information available to agent 1:$Pr(s_t = j|I_t^1) = Pr(s_t = (j_1, j_2)| < a_{1_t}, s_{1_t}, \pi_2(s_t), I_{t-1}^1 >)$. Applying Bayes rule leads to the following result:

$$Pr(s_t = (j_1, j_2)| < a_{1_t}, s_{1_t}, \pi_2(s_t), I_{t-1}^1 >) = \frac{Pr(s_t = (j_1, j_2), s_{1_t} = i_1|a_{1_t}, \pi_2(s_t), I_{t-1}^1)}{Pr(s_t = i_1|a_{1_t}, \pi_2(s_t), I_{t-1}^1)}$$

Since the Dec-MDP is locally fully-observable, the denominator equals one. We expand the numerator by summing all the possible states that could have lead to the current state j.

$$Pr(s_t = (j_1, j_2), s_{1_t} = i_1|a_{1_t}, \pi_2(s_t), I_{t-1}^1) =$$

$$\Sigma_k Pr(s_{t-1} = k|a_{1_t}, \pi_2(s_t), I_{t-1}^1) Pr(s_t = j|s_{t-1} = k, a_{1_t}, \pi_2(s_t), I_{t-1}^1) Pr(s_{1_t} = i_1|s_t = j, s_{t-1} = k, a_{1_t}, \pi_2(s_t), I_{t-1}^1)$$

The actions taken by the agents at time t do not affect the state of the system at time $t-1$, therefore the first probability term is not conditioned on the values of the actions. The second probability term is exactly the transition probability of the Dec-MDP. Since the Dec-MDP has independent transitions, we can decompose the system transition probability into two corresponding probabilities P_1 and P_2, following Definition 1. The last term is equal to one because the Dec-MDP is locally fully observable. Therefore, we obtain:

$$Pr(s_t = j|I_t^1) = \Sigma_k Pr(s_{t-1} = k|I_{t-1}^1) P(s_t = j|s_{t-1} = k, a_{1_t}, \pi_2(s_t) =$$

$$\Sigma_k Pr(s_{t-1} = k|I^1_{t-1})P_1(s_{1_t} = j_1|s_{1_{t-1}} = k_1, a_{1_t})P_2(s_{2_t} = j_2|s_{2_{t-1}} = k_2, \pi_2(s_t))$$

Since agent 1 fully observes $s_1 = i_1$ at time t, then the probability that the system is at state j and its first component j_1 is not i_1 is zero.

$$Pr(s_t = (j_1 \neq i_1, j_2)|I^1_t) = 0$$

$$Pr(s_t = (i_1, j_2)|I^1_t) = \Sigma_k Pr(s_{t-1} = k|I^1_{t-1})P_1(s_{1_t} = i_1|s_{1_{t-1}} = k_1, a_{1_t})P_2(s_{2_t} = j_2|s_{2_{t-1}} = k_2, \pi_2(s_t))$$

Agent 1 can compute the last term for the fixed policy for agent 2. We obtained that the probability of the system being at state j at time t depends on the belief-state at time $t-1$. □

Following this lemma, s_i, the current partial view of agent i is a statistic sufficient for the history of observations, so an agent does not need to remember sequences of observations in order to decide which actions to perform.

Corollary 1 *Agent i's local policy in a Dec-MDP with independent transitions and observations is a mapping from agent i's current partial view to actions:*

$$\delta_i : S_i \rightarrow A_i$$

The Meeting under Uncertainty scenario as described corresponds to a Dec-MDP with independent transitions and observations, therefore it is locally-fully observable. In such a case, for every possible location, a robot needs to find what is the optimal movement action it should take. This decision is not affected by the previous locations where the robot moved through.

We continue our classification of decentralized problems considering two additional dimensions: one is whether agents can share information and the other whether the agents' behavior is goal-oriented. These classes are further described in the next two sections.

2.2 Information Sharing

We distinguish among three possible ways in which agents can share information:

1. **Indirect Communication** — In the most general view, an action ($a_i \in A_i$) performed by an agent can result in three different consequences, and thus it serves any of these three purposes:

   (a) **Information Gathering** — Information about a state can be gathered by an agent that observes that state as a result of performing an action. For example, a robot in a three-dimensional scenario similar to our Meeting scenario may obtain information about the height of a certain location if it is capable of performing such a measure action.

   (b) **Change in Environment** — An agent can cause a direct change in the agent's environment by performing an action. For example, assume a modified version of our Meeting scenario where some of the locations are wells, and therefore can not be traversed by the robots. If the agent is able of covering a well, this action will change the topology of the environment.

(c) **Indirect Communication** — Agent i's actions can affect the observations that agent j observes, i.e., these observations can be captured as the messages that agent i wants to transmit to agent j. Assume, for example that a robot determines its location relatively to the other robot's location, which it observes. Then, the agents may have agreed on a meeting location based on their locations. If robot 1 sees robot 2 in location A, then they will meet at meeting place MA otherwise they will meet at meeting place MB. Even though the agents do not communicate directly, the dependencies between the observations can carry information that is shared by these agents.

Assuming no particular independency in the Dec-POMDP model, the general decentralized control problem includes also the problem of what to communicate and when, given that this communication is established as a consequence of an action performed by an agent, and the resulting observations in the other agents' partial views. Independent of the policy, this type of communication is limited to transferring only information about the features of the state. But in a more general context, the meaning of the communication can be embedded in the policy. That is, each time that an observation is made by agent i, this agent can infer what was meant by the communication in the domain and in the policy. This type of communication is assuming that the observations of the agents are indeed dependent, and this dependency is actually the means that enables each agent to transmit information.

2. **Direct Communication** — Information can be shared by the agents if they can send messages directly to each other. In this case, the observations can be either dependent or independent. We study decentralized processes with direct communication further in Section 5. For example, robot 1 sends a message to robot 2:"Bring tool T to location (x,y)".

3. **Common Uncontrollable Knowledge** — This is knowledge that can be acquired by both agents but does not result from any of these agents' actions. This common knowledge exists if there are features in the system state that are affected by the environment independently of the agents' actions. An example of such feature is the weather (assuming that neither of the agents can have any effect on whether it rains or it shines). Then, information about the weather can be made available to both agents if they share the same feature. Agents can then act upon the conditions of the weather, and thus coordinate their actions without direct exchanging messages. They may have already decided that when the sun shines they meet at location MA, and otherwise at location MB.

   Given that the global set of states S is factored, a *common feature* S^k is a feature of the global state that is included in the partial views of both agents.

   Definition 6 (Common Uncontrollable Knowledge) *A common feature is uncontrollable if:*

   $\forall a, b \in A_1, a \neq b, Pr(S^k|a, S) = P(S^k|b, S)$ *and* $\forall c, d \in A_2, c \neq d, Pr(S^k|c, S) = P(S^k|d, S)$.

In this paper, we focus on either indirect communication or direct communication when we allow information sharing. We exclude from the discussion uncontrollable state features because this knowledge could provide a form of dependency between the agents that we do not handle in this paper. A decentralized process that has independent transitions does not prevent agents from acquiring common uncontrollable knowledge and consequently becoming dependent.

Assumption 1 *We assume that all the changes in the system result necessarily from the agents' actions.*[3]

Finally, the next section presents our last classification of decentralized problems which have goal-oriented behavior. This classification is practical in many areas where the agents may incur some cost while trying to achieve a goal and may attain a global reward only when the goal is reached. This is different from most of the works done on single-agent MDPs where a reward is obtained for every action performed.

2.3 Goal-oriented Behavior

We characterize decentralized processes in which the agents' aim is to reach specific global goal-states. The Meeting under Uncertainty as we described has this feature, the agents' goal is to meet at some location. Other practical scenarios may include, assembling a machine, transferring containers from one location to a final destination, and providing a final answer to a query.

Definition 7 (Goal-Oriented Dec-POMDPs) *A Dec-POMDP is* goal-oriented *if the following two conditions hold:*

1. *There exists a special subset G of the global states that are global goal-states. The process ends when the agents reach any of these goal-states, i.e., no transitions are possible upon reaching a state in G ($\forall g \in G \subseteq S, \forall a_1 \in A_1, \forall a_2 \in A_2, P(g|g, a_1, a_2) = 1$).*

2. *The global reward is $R(s, a_1, a_2, s') = C(a_1) + C(a_2) + JR(s')$, where:*

   - *$C(a_i) < 0$ is the cost incurred by agent i when it performs action a_i. For simplicity, we assume in this paper that the cost of an action depends only on the action. In general, this cost may also depend on the state.*
   - *$JR(s') \in \Re$ is an arbitrary reward associated with each global goal-state and $JR(s') = 0$ for non-goal states ($s' \notin G$).*

The problem of solving a goal-oriented Dec-POMDP is the problem of maximizing the expected global reward. This definition is about global goal behavior, that is, it does not imply that each agent separately has certain goals to achieve.

In the next section, we analyze the complexity of interesting classes of Dec-POMDPs based on the characterization we have presented.

3. Deterministic features that never change their values, or change their values in a deterministic way (such as time that increases in each step) are allowed.

Process Class	Observations Needed by Agent i	Reference
Dec-POMDP	The local *sequence* of observations: $\overline{o}_i$	(Bernstein et al., 2002)
IT, IO Dec-POMDP	The *local sequence* of observations: $\overline{o}_i$	Claim 1
IT Dec-MDP (no IO)	The *local sequence* of observations: $\overline{o}_i$	Claim 1
IT, IO Dec-MDP	The *last local* observation: $o_i = s_i$	Lemma 2

Table 1: A summary of the information on which an optimal local policy is conditioned. IT stands for independent transitions and IO for independent observations.

3. A Taxonomy of Decentralized POMDPs: Complexity Results

We have distinguished between Dec-POMDPs and Dec-MDPs (where joint full-observability is assumed). In neither of these cases, the agents have full observability of the global state (at most they have *joint* full-observability and *local* full-observability, which is different from full observability). Therefore, each one of the agents has a belief about the global state of the system. This is the probability believed by each one of the agents that the system is at some global state s. Table 1 presents the information that each agent needs in order to update its belief about the global state of the system. Since each agent can solve its belief-state MDP assuming a fixed and known policy for the other agent, resulting in the optimal local policy that comprises the optimal joint policy, the information required by an agent to update each belief-state sheds light on the complexity of solving each class of corresponding decentralized control problems. All the complexity results presented in this section apply for decentralized processes controlled by n agents.

Claim 1 *The belief-state update that an agent will be required to perform in the first three cases that appear in Table 1 depends on (at most) the sequence of the agent's own observations.*

In the first two cases, this is true because there is no joint full-observability. In the third case, this is true because the agents' observations are dependent, so even though the decentralized process is jointly fully-observable, each agent does not have full observability of the global state. Notice that only for the last case in the table we have shown that s_i is a sufficient statistic (see Lemma 2). It is still an open question what is the least information that each agent needs to remember in each one of the first three cases, i.e., how the different independency assumptions affect the sufficient statistic for each case.

This section studies to what extent the classes characterized in the previous section represent different complexity classes. In all the results below we refer to the complexity of finding an optimal joint policy for the decentralized control problems handled in the lemmas. The lemmas are stated for the corresponding decision problems (i.e., given the decentralized process assumed in each one of the lemmas, the decision problem is to decide whether there is a joint policy whose value is equal or larger than a given constant K), but finding a solution cannot be easier than deciding the same problem.

All the results in this section correspond to problems given with a finite horizon T. We know from Bernstein et al.'s complexity result (Bernstein et al., 2002), that deciding a finite-horizon decentralized MDP is NEXP-Complete. In Section 2.2, we described indirect

communication, i.e., situations in which information can be shared when the observations are dependent. Therefore, the same complexity result applies for the decentralized control problem with indirect-communication as stated in the next Corollary.

Corollary 2 *Deciding a Dec-MDP or a Dec-POMDP with indirect communication (i.e., allowing the agents to communicate by acting and observing, when the observations are dependent) is NEXP-complete.*

We show in the next lemma, than adding only goal-oriented behavior to a general decentralized process does not change the complexity of the problem. In other words, a goal-oriented decentralized problem is not easier than a non goal-oriented decentralized problem.

Lemma 3 *Deciding a goal-oriented Dec-MDP is NEXP-complete.*

Proof. This case can be proved through the same reduction applied in Bernstein et al. (Bernstein et al., 2002). We can reduce the general goal-oriented Dec-MDP problem to the tiling problem by adding a goal-state to the last state of the Dec-MDP defined in the reduction. The agents reach this new goal-state and receive a reward of zero if the tiling was consistent. Otherwise the agents obtain a reward of -1 and do not reach the goal-state (but they do reach a terminal state and the process ends).

The main reason for this complexity result relies on the fact that each agent needs to remember a *sequence* of observations that it has observed (see Table 1). Adding only goal-states to the decentralized process (without assuming any further assumptions) does not make the control problem any easier. □

Since a Dec-POMDP is more general than a Dec-MDP, the same lower bound for the Dec-MDP is valid for an even harder problem.

Corollary 3 *Deciding a goal-oriented Dec-POMDP is NEXP-complete.*

The next lemma shows that by assuming that the transitions and observations are independent and that the agents have joint full-observability, the problem of solving a decentralized cooperative system, then becomes easier (the lemma does not assume goal-oriented behavior).

Lemma 4 *Deciding a Dec-MDP with independent transitions and observations is NP-complete.*

Proof. Since the Dec-MDP is jointly fully-observable and it has independent transitions and observations, it is also locally fully-observable (see Lemma 1). We have shown in Lemma 2 that for such Dec-MDPs, the current partial view of an agent is a sufficient statistic. Therefore, a local policy of an agent i is of size polynomial in $|S_i|$. There are $|A_i|^{|S_i|}$ policies (mappings from S_i to A_i). Each agent i needs to build a belief-state MDP with a number of states that is polynomial in $|S_i|$ (for a fixed and known policy for the other agent). Evaluating one such local policy can be done in polynomial time (by running dynamic programming on the belief-state MDP), but there are exponentially many such

policies for which this should be done. Therefore, the upper bound for the decision problem stated in this lemma is NP.

Figure 2 shows schematically the differences in the policies, which lead to the difference in the complexity classes of the control problems when independent transitions and observations are assumed. When no assumptions are made (as in the leftmost figure), a local policy is represented by a tree, where each node corresponds to a possible action to be taken and each edge corresponds to a possible observation (i.e., a possible transition). Each local policy needs to remember a sequence of observation as opposed to just the last observation as in the rightmost figure. In the belief-state MDP that each agent builds, there is an exponential number of states, that correspond to all the possible sequences of observations (this number is $|\Omega_i|^T$ when T is the finite horizon). Each such policy (of exponential size) can be evaluated with dynamic programming. There are a total of $|A_i|^{|\Omega_i|^T}$ local policies.[4]

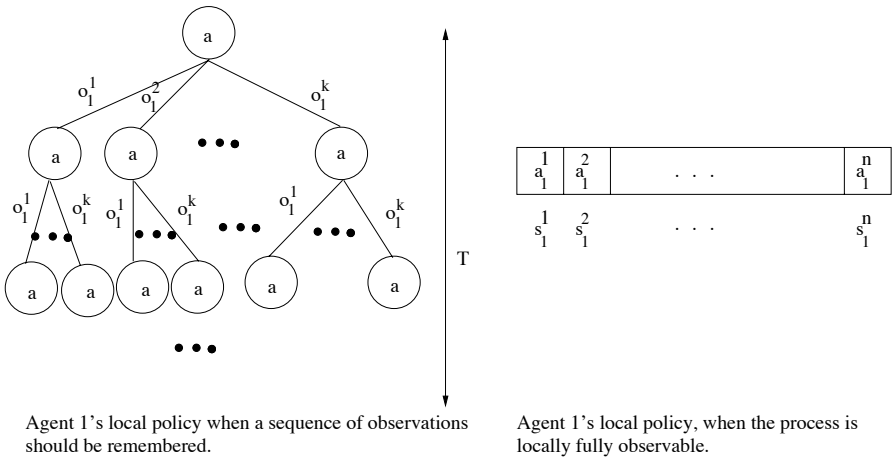

Figure 2: Exponential vs. polynomial sized policies.

We know already from Papadimitriou and Tsisiklis (Papadimitriou & Tsitsiklis, 1982, 1986) that a simple decentralized decision making problem for two agents is NP-hard (where $|A_i| \geq 2$ and $|A_j| \geq 3$). Therefore, the lower bound for the problem class stated in the lemma is also NP. □

It is an open question whether a Dec-POMDP with independent transitions and observations (without joint full-observability) results in a complexity class lower than NEXP.

The next lemmas show that by restricting the classes of goal-oriented decentralized problems, not to include any type of information sharing, while maintaining the independency assumptions, we may obtain easier problems that can be solved in polynomial time.

Lemma 5 *Deciding a goal-oriented Dec-MDP with independent transitions and observations, and a single global goal-state is P-complete.*

Proof. We denote the single global goal-state by $g = (g_1, g_2)$. The agents cannot obtain additional information about each other after they plan optimally towards g_1 and g_2

4. Assuming that T is similar in size to $|S|$, we obtain that the complexity of the brute-force search algorithm is double exponential in $|S|$. If $T << |S|$ the complexity can be NP.

respectively: Indirect communication is impossible since the observations are independent, direct communication is not allowed and no common uncontrollable knowledge is available. If the system has a single global goal-state, it is reasonable to assume that the model includes an action that allows the agents to make no transitions at no cost once they reached their corresponding component of the global goal-state. Due to the independent transitions, neither of the agents need to build a belief-state MDP, but a single-agent MDP is enough to solve optimally such a Dec-MDP. Each agent's MDP M_i is built as follows: $M_i =< S_i, A_i, P_i, R_i >$. S_i, A_i and P_i correspond to the partial view, actions and transition probability already defined by the Dec-MDP with independent transitions. R_i is defined as follows: $R_i = R_i(s_i, a_i, s'_i) = Cost(a_i) + GR(s'_i)$, where $GR(s'_i) \in \Re$ if $s'_i = g_i$ and zero otherwise. □

We show the algorithm that solves this problem in the next section. The same complexity result applies when there are many global goal-states, but no new information can be made available to the agents whose transitions and observations are independent.

Lemma 6 *Deciding a goal-oriented Dec-MDP with independent transitions and observations with at least one global goal-state is P-complete.*

Proof. Due to the independent transitions and observations, each agent can be assigned a single agent MDP for its corresponding component of each global goal-state. All the states of the system, including the goal-states are factored because of the independency assumptions. Each agent can solve these MDPs optimally. We also assume that there is at least one way to reach a global goal-state with probability one.

In the cases studied in this paper, there is a finite horizon T. The agents will be penalized if they did not reach a global goal-state by time T. This penalty can be determined so that agents will always prefer to reach the same global goal-state over not reaching it. Any policy that is not committed to the shortest path has a lower likelihood to reach the same global goal-state. In problems with finite-horizon as analyzed here, the penalty received by the system when the goal is not reached is a large negative number such that taking the risk of not reaching the same goal-state is not worthwhile. Agents will not prefer to switch goals, but will prefer to pursue the shortest paths to each one of the possible components of global goal-states.

Since we assume that no information sharing is possible, then solving each one of these possible MDPs (aimed at each one of the possible goal components) and choosing the global goal-state with the highest value solves optimally this problem. Any other solution that involves moving to any state that is not on the shortest path towards a single goal component will not be optimal because it has a positive probability of ending in a component of a global goal-state which was not reached by the other agents. The problem is an iterative version of the decentralized control problem with independent transitions and observations and a single global goal-state. Following Lemma 5, this problem can be solved in polynomial time. □

Assume that the Meeting scenario is given with a single meeting location (e.g., the spaceship in the rovers' case). Then, given the other independency assumptions, we can solve optimally this problem by building single-agent MDPs, each planning to achieve its component of the goal-state (in the example, each robot needs to reach the spaceship). If there is a larger set of global goal-states, for example, if there is a finite number of possible

meeting sites (e.g., the spaceship, the space station on the planet and some other particular site) then following the lemma, we can let each agent find its local optimal policy to each one of its corresponding component in these goal states. The optimal joint policy is the joint policy among those pairs of local policies with the highest value.

Notice that this result applies when the agents arrive at a component of a global goal and cannot move out of it, and also when moving between global goal-components is indeed possible. For example, in the basic Meeting under Uncertainty scenario, where two robots are trying to meet as soon as possible in a 2D grid, every grid location is a possible meeting place, but also these locations can be squares that the agents go through when moving to another target.

As we state in the next corollary, if any type of information sharing is indeed possible, then, a goal-oriented decentralized problem may not necessarily decompose into single agent processes with goal-oriented behavior. For example, agents may need to act towards a state which is not defined as a component of a goal state in order to achieve a global goal-state

Corollary 4 *If some type of information sharing is allowed while solving a goal-oriented Dec-MDP with many global goal-states then goal-oriented behavior does not necessarily induce individual goal-oriented behavior. The single-agent process resulting from the Dec-MDP has a clear set of states, given by the partial view of the agent (S_i), a transition probability P_i (due to the independent transitions), a set of actions A_i (as in the Dec-MDP) but the single-agent reward function R_i is not defined.*

The results presented so far for goal-oriented behavior assumed that the system state is jointly fully-observable. If this is not the case, then the resulting Dec-POMDP can be decomposed into single-agent POMDPs when the process is goal-oriented and information sharing is not possible. Similarly to the previous results, but assuming a Dec-POMDP process, we obtain that the complexity of deciding these corresponding Dec-POMDPs is equivalent to the complexity of deciding a single-agent POMDP, i.e., P-space (Papadimitriou & Tsitsiklis, 1987). This is stated in Corollaries 5 and 6.

Corollary 5 *Deciding a goal-oriented Dec-POMDP with independent transitions and observations with a single global goal-state is P-space.*

Corollary 6 *Deciding a goal-oriented Dec-POMDP with independent transitions and observations with at least one global goal-state is P-space.*

A summary of the complexity results presented in this Section appears in Table 2.

4. Algorithms for Decentralized Control

So far, the only known algorithm for controlling optimally Dec-MDPs with independent transitions and observations is the coverage-set algorithm described in (Becker et al., 2003). This algorithm assumes that the agents' actions could result in super-additive or sub-additive joint rewards. In the first case, the reward obtained by the system from agents doing certain actions is larger than the sum of each agent's local reward for those actions. In the second case, sub-additive joint rewards will be attained when the agents are penalized for doing redundant actions. As an example, we can look at a modified version of the

Process Class	Complexity Class	Reference
Dec-POMDP	NEXP-complete	(Bernstein et al., 2002)
Dec-MDP	NEXP-complete	(Bernstein et al., 2002)
Approximate Dec-POMDP	NEXP-complete	(Rabinovich et al., 2003)
Approximate Dec-MDP	NEXP-complete	(Rabinovich et al., 2003)
GO Dec-POMDP	NEXP-complete	Corollary 3
GO Dec-MDP	NEXP-complete	Lemma 3
IT,IO Dec-POMDP	NEXP	Section 3
GO,IT,IO, $\|G\| = 1$ Dec-POMDP	P-space	Corollary 5
GO,IT,IO, $\|G\| \geq 1$ Dec-POMDP,No info. Sharing	P-space	Corollary 6
IT,IO Dec-MDP	NP-complete	Lemma 4
GO,IT,IO,$\|G\| = 1$ Dec-MDP	P-complete	Lemma 5
GO,IT,IO,$\|G\| \geq 1$ Dec-MDP,No Info. Sharing	P-complete	Lemma 6
GO,IT,IO,$\|G\| \geq 1$ Dec-MDP, with Direct Comm.	NP	Lemma 10
GO,IT,IO,$\|G\| \geq 1$ Dec-MDP, with Direct Comm. Myopic-greedy Approximation	P	Lemma 11
GO,IT,IO,$\|G\| \geq 1$ Dec-MDP, with Direct Comm. Approximation to Monotonic Dec-MDPs	P	Lemma 13

Table 2: A summary of the complexity analysis for classes of decentralized control processes. We use the notation GO to denote a goal-oriented process, IT and IO for independent transitions and observations respectively.

Process Class	Optimal Algorithm	Reference
IT, IO Dec-MDP, No Information Sharing	Coverage-set	(Becker et al., 2003)
IT, IO Dec-MDP, with Direct Communication	Not Known Yet[5]	Section 5
IT, IO, GO Dec-MDP ($\|G\| = 1$)	$Opt1Goal$	Section 4.1
IT, IO GO Dec-MDP ($\|G\| \geq 1$), No Information Sharing	$OptNGoals$	Section 4.2
IT, IO GO Dec-MDP ($\|G\| \geq 1$), with Direct Comm.	Not Known Yet[5]	Section 5

Table 3: A summary of the algorithms known for controlling optimally decentralized MDPs.

Meeting scenario, where robots can move and can also run experiments at different sites. Then, a process may lead to sub-additive rewards if both agents run the same experiment, wasting their battery instead of doing non-overlapping tasks. Sometimes, the system is better off when both robots perform the same tasks, for example both agents run the same experiment at different times in the day, collecting eventually results with better quality. The class of problems handled by the coverage-set algorithm does not include necessarily goal-oriented decentralized processes. In this section, we present two tractable algorithms for controlling optimally Dec-MDPs with independent transitions and observations, which are also goal-oriented. A summary of the algorithms known to solve optimally decentralized control problems is presented in Table 3.

4.1 Single Goal, Goal-Oriented Dec-MDPs

The single global goal-state of the Dec-MDP is $g = (g_1, g_2)$. We have assumed that there exists at least one joint policy that reaches this goal, otherwise the agents are penalized with a negative large amount. $Opt1Goal$ (see Figure 3) is the algorithm that finds the optimal decentralized joint policy for such a Dec-MDP. The correctness of this algorithm is easily obtained from Lemma 5. Each agent i solves its corresponding single-agent MDP_i with a single goal-state given by g_i. MDP_i is given by the tuple $< S_i, A_i, P_i, R_i >$. For $i = \{1, 2\}$, the set of states S_i, the set of actions A_i and the transition probability P_i correspond to agent i's partial view, control actions and transitions resulting from the Dec-MDP as defined in Section 2.1. The local reward function R_i is defined as the reward that the agent will receive when taking an action a_i and moving from state s_i to s'_i. Similarly to the definition of the joint reward in a goal-oriented Dec-MDP (Definition 7), we define $R_i(s_i, a_i, s'_i) = C(a_i) + GR(s'_i)$, where:

- $C(a_i) < 0$ is the cost incurred by agent i when it performs action a_i.
- The local goal reward, $GR(s'_i) \in \Re$ is an arbitrary reward associated with each local goal-state and it is zero when the s'_i is a non-goal state.

4.2 Goal-Oriented Dec-MDPs with No Information Sharing

Due to the uncertainty of the outcomes of the agents' actions, an agent may decide to change its intention with respect to the global goal-state it is planning to. In this paper,

5. No algorithm was proposed short of full search with complexity NP as shown in Lemma 10.

function Opt1Goal(Dec-MDP)
returns the optimal joint policy δ^*,
inputs: Dec-MDP=$< S, A_1, A_2, P, R >$
G /* the set of global goal-states, $|G| = 1$, $g = (g_1, g_2) \in G \subseteq S$*/
/ * *Transition independence* $\Rightarrow S = S_1 \times S_2, P = P_1 \times P_2$ */
/ * $R(s, a_1, a_2, s') = Cost(a_1) + Cost(a_2) + JR(s')$ * /
/*ComputeLocalR computes the local rewards as follows:*/
/ * $R_1(s_1, a_1, s_1') = Cost(a_1) + GR(s_1')$ * /
/ * $R_2(s_2, a_2, s_2') = Cost(a_2) + GR(s_2')$ * /
/ * $GR(s_i') \in \Re$ *if* $s_i' = g_i$, *else* 0 * /

$R_1 \leftarrow ComputeLocalR(S_1, A_1, P_1, g_1)$
$MDP_1 =< S_1, A_1, P_1, R_1 >$
$R_2 \leftarrow ComputeLocalR(S_2, A_2, P_2, g_2)$
$MDP_2 =< S_2, A_2, P_2, R_2 >$
$\delta_1^* \leftarrow SOLVE(MDP_1)$
$\delta_2^* \leftarrow SOLVE(MDP_2)$
$\delta^* \leftarrow (\delta_1^*, \delta_2^*)$
return δ^*

Figure 3: The Opt1Goal algorithm.

we avoid this type of behavior because the agents prefer to reach the same global goal-state and otherwise will obtain a large negative penalty. There is no reason for an agent to take a longer path (through several visits to goal-states' components) instead of moving directly to each one of the goal-state's components.

The algorithm that optimally and decentrally solves a goal-oriented Dec-MDP problem with many global goal-states is *OptNGoals* which is presented in Figure 4. The correctness of this algorithm is obtained from Lemma 6, each agent, iteratively, solves its induced MDP towards each one of the possible components of each one of the global goal-states of the system. Finally, the optimal joint policy is the one with the highest value.

Lemma 7 *OptNGoals returns the optimal joint decentralized policy for a goal-oriented Dec-MDP with independent transitions and observations when no new information can be acquired by any agent.*

Proof. We assume that the process continues for T time steps.[6] The proof is by induction on the steps of the policy: At the basis of the induction, time is T; the agents cannot perform any more moves. Obviously, moving to an intermediate state is not beneficial.

6. If the process could stop before time T given that the agents reached an absorbing state, then there is a special kind of communication different from the ones we already mentioned. If an agent reaches a goal-state and the system did not stop, then this agent is actually getting new information, because it knows that the other agent did not reach the same goal-state.

function OptNGoals(Dec-MDP)
returns the optimal joint policy δ^*,
inputs: Dec-MDP=$< S, A_1, A_2, P, R >$
G /* the set of global goal-states, $|G| = N, g_i = (g_1^i, g_2^i) \in G, 1 \leq i \leq N$ */
/* $Transition\ independence \Rightarrow S = S_1 \times S_2, P = P_1 \times P_2$ */
/* $R(s, a_1, a_2, s') = Cost(a_1) + Cost(a_2) + JR(s')$ */
/* $R_1(s_1, a_1, s_1') = Cost(a_1) + GR(s_1')$ */
/* $R_2(s_2, a_2, s_2') = Cost(a_2) + GR(s_2')$ */
/* $GR(s_i') \in \Re\ if\ s_i' = g_i,\ else 0$ */

$Dec-MDP^1 = < S, A_1, A_2, P, R >$
$\delta^{*1} \leftarrow Opt1Goal(Dec-MDP^1, (g_1^1, g_2^1))$
$CurrOptJoint\delta \leftarrow \delta^{*1}$
$CurrMaxVal \leftarrow ComputeV(Dec-MDP^1, CurrOptJoint\delta, s^0)$
for $i \leftarrow 2$ **to** N
$Dec-MDP^i = < S, A_1, A_2, P, R >$
$\delta^{*i} \leftarrow Opt1Goal(Dec-MDP^i, (g_1^i, g_2^i))$
$CurrVal \leftarrow ComputeV(Dec-MDP^i, \delta^{*i}, s^0)$
if $(CurrVal > CurrMaxVal)$ **then**
$CurrOptJoint\delta \leftarrow \delta^{*i}$
$CurrMaxVal \leftarrow CurrVal$
return $CurrOptJoint\delta$

function ComputeV (Dec-MDP,δ, s^0)
returns the value of state s^0 following joint policy δ, $V_\delta(s^0)$
inputs: Dec-MDP, the current Dec-MDP being evaluated.
$\delta = (\delta_1, \delta_2)$, the joint policy found so far .
s^0, the initial state of the Dec-MDP.

if $s^0 \in G$ **then**
return $JR(s^0)$
else
$V \leftarrow \Sigma_{s'=(s_1', s_2')} P_1(s_1'|\delta_1(s^0), s^0) P_2(s_2'|\delta_2(s^0), s^0)(R(s^0, \delta_1(s^0), \delta_2(s^0), s') + ComputeV(Dec-MDP, \delta, s'))$

Figure 4: The OptNGoals algorithm.

We assume that if there are $k<T$ steps left, it is not beneficial for the agents to move towards an intermediate state instead of moving directly towards a goal-state. We show that if there are k+1 steps left, then the expected cost of a policy that instructs the agent to move to an intermediate state is larger than the cost of the policy that instructs the agent to move directly to the corresponding component of a goal-state if possible.

Denote by δ a local policy that instructs an agent to move directly towards a goal-state component g. δ finds the shortest path towards g. Now, assume another policy δ', which is different from δ at one state along the shortest path. $\delta(s_1) = a_1$ and $\delta'(s_1) = a'_1$. We assume that at state s_1 the agent cannot obtain any new information, therefore the agent should have preferred the shortest path and δ' cannot be more beneficial than δ. We know from the induction assumption that when the agent is at state s_1 it has less than k steps to the final T, and from s_1 it is not beneficial to move to intermediate states. □

5. Decentralized Control with Communication

Direct communication can be beneficial in decentralized control (i.e., the value of the optimal joint policy that allows communication may be larger than the value of the optimal joint policy without communication) because the agents lack full observability of the global state. We are interested in solving a decentralized control problem off-line taking into account that possible new information could be acquired on-line. Agents will consider this expected information while computing their optimal joint policy, thus deriving a policy for when and what to communicate.

If we assume that direct communication leads to full observability of the system state, that direct communication is free and that the observations are independent then obviously the agents will benefit most by constantly communicating, and thus having a fully observable decentralized process, which is equivalent to an MMDP (Boutilier, 1999). This problem is known to be P-complete (Papadimitriou & Tsitsiklis, 1987).

In real-world scenarios, it is reasonable to assume that direct communication has indeed an additional cost associated with it, given by the risk of revealing information to competitive agents, given by the bandwidth necessary for the transmission or even given by the complexity of computing the information to be transferred. Therefore, communication may not be possible or even desirable at all times.

We extend the model of decentralized partially-observable Markov decision process to include an explicit language of communication with an associated cost. We call this model Dec-POMDP-Com. It is given by the following tuple: $< S, A_1, A_2, \Sigma, C_\Sigma, P, R, \Omega_1, \Omega_2, O, T >$.

Σ is the alphabet of messages. $\sigma_i \in \Sigma$ denotes an atomic message sent by agent i (i.e., a letter in the language). $\overline{\sigma_i}$ denotes a sequence of atomic messages. A special message that belongs to Σ is the null message, which is denoted by ϵ_σ. This message is sent by an agent that does not want to transmit anything to the other agents. We omitted in this paper the details of the communication network that may be necessary to implement the transmission of the messages.

C_Σ is the cost of transmitting an atomic message: $C_\Sigma : \Sigma \rightarrow \Re$. The cost of transmitting a null message is zero. Communication cost models determine the flow of the information exchange and the cost of this communication. These models may include, for example, one-way communication models in which the cost C_Σ is incurred each time that one agent

sends information to another agent; two-way communication models where agents exchange messages each time at least one of them initiates communication, and the cost is incurred only once, each time. Other models may require additional messages like acknowledgments that may incur additional costs. We restrict ourselves in this paper to communication cost models based on joint exchange of messages and where communication leads to full observability of the global state. Note that when the observations are independent, and assuming that there is no common uncontrollable knowledge (Assumption 1), direct communication is the *only* means of achieving full-observability.

We define a Dec-MDP-Com as a Dec-POMDP-Com with *joint* full-observability, as we did with Dec-POMDPs and Dec-MDPs in Section 2.1. The Dec-POMDP-Com model can have independent transitions, independent observations, be locally fully-observable, and goal-oriented as the basic model presented in Section 2.

We describe the interaction among the agents as a process in which agents perform an action, then they observe their environment, and then send a message that is instantaneously received by the other agent.[7] Then, we can define the local policies of the controlling agents as well as the resulting joint policy whose value we are interested in optimizing. A *local policy* δ is composed of two policies, δ^A that determines the actions of the agents, and δ^Σ that states the communication policy. Notice that δ^A allows indirect communication if the observations of the agents are dependent, and that δ^Σ allows direct communication even if the observations are dependent. With direct communication, the agents' designer can enrich the agents' performance with additional messages.

Definition 8 *A local policy for action for agent i, δ_i^A, is a mapping from local histories of observations $\overline{o_i} = o_{i_1}, \ldots, o_{i_t}$ over Ω_i and histories of messages $\overline{\sigma_j} = \sigma_{j_1}, \ldots, \sigma_{j_t}$* received *$(j \neq i)$ since the last time the agents were synchronized to actions in A_i.*[8]

$$\delta_i^A : S \times \Omega^* \times \Sigma^* \rightarrow A_i$$

Definition 9 *A local policy for communication for agent i, δ_i^Σ, is a mapping from local histories of observations $\overline{o_i} = o_{i_1}, \ldots, o_{i_t}$ and o, the last observation perceived after performing the last local action, over Ω_i and histories of messages $\overline{\sigma_j} = \sigma_{j_1}, \ldots, \sigma_{j_t}$* received *$(j \neq i)$ since the last time the agents were synchronized to messages in Σ.*

$$\delta_i^\Sigma : S \times \Omega^* o \times \Sigma^* \rightarrow \Sigma$$

More complex cases result if the agents could communicate partial information about their partial views. This is left for future work.

Definition 10 *A joint policy $\delta = (\delta_1, \delta_2)$ is defined as a pair of local policies, one for each agent, where each δ_i is composed of the communication and the action policy for agent i.*

The complexity results we obtained in Section 3 apply also for the same classes of problems when direct communication is possible. Although agents achieve full observability

7. When agents exchange information there is a question whether information is obtained instantaneously or there are delays. For simplicity of exposition we assume no delays in the system.
8. In this paper, we study finite horizon processes, therefore time is included in the state representation.

each time they exchange information, the problem of finding the policy of communication off-line (when there is a cost associated with each communication act) remains as hard as the general problem with no communication. In the worst case, transmitting the messages can be prohibitively expensive. Therefore, adding direct communication does not simplify the problem. For all the cases shown to be in NEXP, adding direct communication cannot make them more difficult. The complexity of deciding a Dec-MDP when observations are independent and direct communication is allowed remains the same as when direct communication is not assumed, as shown in Lemma 10. The impact of direct communication on the classes of Dec-POMDPs with independent transitions and observations and with possible goal-oriented behavior remains an open question.

It is interesting to note that the decentralized control problem with direct communication can be reduced to the same problem with indirect communication when the observations are dependent. We assume that transmitting messages incur the same cost and that the language of messages is the language of observations. If the language of communication is different then the reduction does not apply.

Lemma 8 *A Dec-MDP with direct communication is polynomially-reducible to a Dec-MDP with indirect communication.*

Proof. We denote the Dec-MDP with direct communication Dec-D, and the Dec-MDP with indirect communication Dec-I. The reduction from Dec-D to Dec-I requires the addition of a single bit b to the global states of Dec-I. When b takes the value 1, the agents are in the communication mode. When b takes the value 0, the agents are performing control actions. A communication action a^c performed by agent i is agent i's local observation o_i. The transition probability of Dec-I, P_I is given as follows: $P_I([s,1],o_1,o_2,[s,0]) = 1$, no change is caused to the global state of the system besides flipping the value of b back to 0 each time the agents exchange information. The probability of observing o_1 and o_2 (respectively by the two agents) after performing communication acts when b equals 1 is one as long as o_1 is agent's 2 last observation, and o_2 is agent's 1 last observation. This probability is zero for any other action taken at $[s,1]$. $O(o_2,o_1|[s,1],o_1,o_2,[s,0]) = 1$. □

5.1 The Analytical Expression of the Optimal Solution to Dec-POMDP-Com

The agents send messages in a broadcast manner, and only one message is sent at each time. The agents in the system share the same language of communication. In a separate line of research, we are addressing the question of agents controlling a decentralized process where the agents develop a mutual understanding of the messages exchanged along the process.

Following the model presented in Section 5, we express the value of a state in the Dec-POMDP-Com model when no particular assumptions are made on the class of the problem. The optimal joint policy that stipulates for each decision-maker how it should behave and when it should communicate with other agents is the policy that maximizes the value of the initial state of the Dec-POMDP-Com. We will then study certain classes of this problem as we did with the case without communication.

In order to refer to a sequence of messages sent by an agent, two auxiliary functions are defined: f_1^l is the first l messages sent by agent 1. Similarly, f_2^l is defined for agent 2. f_1^l is a function of 1) the state in which the last message is sent, 2) the sequence of observations

seen by agent 1 (when $|\overline{o_1}| = l$, it is denoted by $\overline{o_1}^l$), and 3) the sequence of messages received from agent 2. These functions can be recursively defined: ($\cdot$ is the concatenation operator)

$$f_1^0 = \delta_1^\Sigma(s, \epsilon, \epsilon) \quad f_2^0 = \delta_2^\Sigma(s, \epsilon, \epsilon)$$
$$f_1^l = \delta_1^\Sigma(s, \overline{o_1}^{l-1}, f_2^{l-1}) \cdot f_1^{l-1}$$
$$f_2^l = \delta_2^\Sigma(s, \overline{o_2}^{l-1}, f_1^{l-1}) \cdot f_2^{l-1}$$

Definition 11 *The probability of transitioning from a state s to a state s' following the joint policy $\delta = (\delta_1, \delta_2)$ while agent 1 sees observation sequence $\overline{o_1}o_1$ and receives sequences of messages $\overline{\sigma_2}$, and agent 2 sees $\overline{o_2}o_2$ and receives $\overline{\sigma_1}$ of the same length, written $\overline{P_\delta}(s'|s, \overline{o_1}o_1, \overline{\sigma_2}, \overline{o_2}o_2, \overline{\sigma_1})$ can be defined recursively:*[9]

1. $\overline{P_\delta}(s|s, \epsilon, \epsilon, \epsilon, \epsilon) = 1$
2. $\overline{P_\delta}(s'|s, \overline{o_1}o_1, \overline{\sigma_2}\sigma_2, \overline{o_2}o_2, \overline{\sigma_1}\sigma_1) = \sum_{q \in S} \overline{P_\delta}(q|s, \overline{o_1}, \overline{\sigma_2}, \overline{o_2}, \overline{\sigma_1}) *$

$$P(s'|q, \delta_1^A(s, \overline{o_1}, \overline{\sigma_2}), \delta_2^A(s, \overline{o_2}, \overline{\sigma_1})) * O(o_1, o_2|q, \delta_1^A(s, \overline{o_1}, \overline{\sigma_2}), \delta_2^A(s, \overline{o_2}, \overline{\sigma_1}), s')$$

such that $\delta_1^\Sigma(s, \overline{o_1}o_1, \overline{\sigma_2}) = \sigma_1$ and $\delta_2^\Sigma(s, \overline{o_2}o_2, \overline{\sigma_1}) = \sigma_2$.

Then, the value of the initial state given by s^0 in the Dec-POMDP-Com after following a joint policy δ for T steps can be defined as follows:

Definition 12 *The value $V_\delta^T(s^0)$ after following policy $\delta = (\delta_1, \delta_2)$ from state s^0 for T steps is given by:*

$$V_\delta^T(s^0) = \sum_{(\overline{o_1}o_1, \overline{o_2}o_2)} \sum_{q \in S} \sum_{s' \in S} \overline{P_\delta}(q|s^0, \overline{o_1}, f_2^l, \overline{o_2}, f_1^l) * P(s'|q, \delta_1^A(s^0, \overline{o_1}, f_2^l), \delta_2^A(s^0, \overline{o_2}, f_1^l)) *$$
$$R(q, \delta_1^A(s^0, \overline{o_1}, f_2^l), \delta_1^\Sigma(s^0, \overline{o_1}o_1, f_2^l), \delta_2^A(s^0, \overline{o_2}, f_1^l), \delta_2^\Sigma(s^0, \overline{o_2}o_2, f_1^l), s')$$

where the observation and the message sequences are of length at most $T-1$, and both sequences of observations are of the same length l. The sequences of messages are of length $l+1$ because they considered the last observation resulting from the control action prior to communicating.

The problem of decentralized control with direct communication is to find an optimal joint policy δ^* for action and for communication such that $\delta^* = argmax_\delta V_\delta^T(s^0)$.

5.2 Languages of Communication

We start showing that under some circumstances the language of observations is as good as any other communication language. In the Meeting scenario, no matter what are the tasks assigned to the system, agents that exchange their current coordinates are guaranteed to find the optimal solution to the decentralized problem.

9. The notation $\overline{o} = o_1, \ldots, o_t$ and $\overline{o}o$ represents the sequence $o_1, \ldots, o_t o$. Similarly, the notation for sequences of messages: $\overline{\sigma_i}\sigma$ represents the sequence $\sigma_{i_1}, \ldots, \sigma_{i_t}\sigma$.

Theorem 1 *Given a Dec-MDP-Com with independent transitions and observations and constant message cost, the value of the optimal joint policy δ^* with respect to any Σ, $V^T_{\delta^*,\Sigma}(s^0)$ is not greater than the value of the optimal joint policy with respect to the language of observations ($\Sigma = \Omega$). That is:*

$$\forall \Sigma \; V^T_{\delta^*,\Sigma}(s^0) \leq V^T_{\delta'^*,\Sigma=\Omega}(s^0).$$

Proof. A Dec-MDP with independent transitions and observations is also locally fully-observable (see Lemma 1). Lemma 2 states that each agent's current partial view is a sufficient statistic for the history of observations needed to compute the optimal decentralized joint policy. Therefore, it is not beneficial for the agents to send any information in addition to their fully observable current partial view because the decentralized process is jointly fully-observable, so combining both agents' partial views (which each is fully-observable) results in the complete global state. Moreover, the theorem assumes a constant cost for every message, that is all non-null messages incur the same cost, there are not any messages that are either more expensive or cheaper to transmit than others. Therefore, the agents cannot benefit from exchanging information that is a strict subset of their partial views because we assume that the cost of sending any message is equal. □

Note that the theorem does not hold when different messages may incur different costs. In this case, sending less information might be cheaper, but equally valuable. For example, when agents observe their respective x and y coordinates, they may benefit from sending only one coordinate if it costs less than sending the complete location. Agents may also benefit from sending functions of their observations if this incurs a smaller cost. For example, agents may benefit from exchanging information about the Manhattan distance between their current location and some mutually-known location.

In general, it seems reasonable to introduce a language of communication to reduce complexity, but as the theorem shows, this cannot guarantee optimality when the language is comprised of messages different from the agents' observations. Messages different from observations may be exchanged to *approximate* the optimal decentralized solution at a lower complexity. Examples of such messages include: 1) commitments, which are constraints on the future behavior of the message sender, 2) instructions, which are constraints on the future behavior of the message hearer, 3) feedback which is an encouraging or punishing signal that is sent to another agent. The study of Dec-POMDP-Com problems with languages of communication different from observations is left for future work. Similarly, certain protocols of communication can restrict the optimal value of the policy of communication but may be easier to implement.

Moreover, the next lemma shows that an optimal policy of communication does not need to send sequences of observations, but it will instruct the agent to transmit its current observation or the null message.

Lemma 9 *Given a Dec-MDP-Com with independent transitions and observations, there is an optimal policy of communication such that whenever a non-null message is sent, it must be the agent's last observation.*

Proof. Since the Dec-MDP-Com is by definition jointly fully-observable and it has independent transitions and observations, then the Dec-MDP-Com is locally fully-observable

(see Lemma 1). Lemma 2 showed that the current locally fully-observable partial view is a sufficient statistic. Therefore, sending a non-null message that is an observation different from the last one cannot provide more information about the current state of the process than the last observation does. □

Since the current global state becomes fully observable each time that the agents communicate, all the necessary information is stored in the synchronized state s; the agents do not need to remember all the messages received so far when the Dec-MDP has independent observations and transitions. Thus, the local policies of action and communication for such a Dec-MDP-Com can be formalized as follows:

Corollary 7 *An optimal local policy of action for this problem, δ^A, can be represented as a mapping from synchronized states and current partial views to actions.*

$$\delta^A : S \times S_i \rightarrow A_i$$

Similarly, an optimal local policy of communication δ^Σ can be represented as a mapping from synchronized states and current partial views to two possible messages: either the current partial view or the null message.

$$\delta^\Sigma : S \times S_i \rightarrow S_i \cup \{\epsilon_\sigma\}$$

such that if $\delta^\Sigma(s, s_i) \neq \epsilon_\sigma$ then $\delta^\Sigma(s, s_i) = s_i$.

Agents need to remember only their current partial view and the last synchronized information to decide on their next action. This is a primary observation that affects the complexity of deciding Dec-MDP-Com with independent transitions and observations as shown in the next lemma.

Lemma 10 *Deciding a Dec-MDP-Com with independent transitions and observations is in NP.*

Proof. Following Corollary 7, each agent's policy is of size polynomial in $|S|$, and the number of possible policies is $2^{|S|^2} \times |A|^{|S|^2}$. In the worst case, a brute force algorithm can go through all the possible policies for agent 1 and for each one of them compute the optimal policy for agent 2. Agent 2 builds its belief-state MDP, where each node is the agent's belief that the global state is a state s. There is an edge for any possible action and message that the agents can choose. Agent 2 can choose any action $a_2 \in A_2$ and it can either send a null message, or a message with its last observation (s_2). For any possible policy of action and communication of agent 1, agent 2 can build such a belief-state MDP and solve it. This can also be done in time polynomial in the number of the belief-states. Whenever an agent sends a non-null message, then the belief-state MDP has a transition to a state that is fully observable with probability one. In any case, each agent needs only to remember its last current partial view, so the complexity of solving a Dec-MDP-Com with independent observations and transitions is in the NP class. □

We have seen so far that if the Dec-POMDP induces single-agent MDPs then the complexity drops to polynomial. In general, this is not the case. Assuming a Dec-MDP with independent transitions and observations results in a clear decomposition of the global states

set into two sets S_1 and S_2 and in a well-defined decomposition of the transition probability P into P_1 and P_2. If the observations are independent then O has a well-defined decomposition into O_1 and O_2. Each agent i has a set of actions A_i. Nevertheless, the problem of decomposing the global reward function is not trivial. For the set of problems already shown to belong to the P class, we can talk about the natural decomposition of the problem into single-agent local problems. In general, decomposing a decentralized process into single-agent processes is not a trivial task. The next section shows an approximation scheme to solve a Dec-MDP-Com by implementing a policy of communication and decomposing the decentralized problem into single-agent problems in-between communications.

6. Mechanism Design for Communication

We introduce the notion of mechanisms for communication as a practical approach for approximating the optimal joint policy for decentralized control with direct communication. We borrow from game theory and economics the notion of mechanism design (Moore, 1992). Mechanism design was originally studied in Game Theory to design games that yield outcomes with certain characteristics. Later, research in Computer Science has looked at adapting this approach to achieve social coordination and optimization of social welfare in distributed systems. We are interested in mechanisms that result in near-term behaviors that produce good approximations to the optimal control of a decentralized cooperative system.

Mechanism design or implementation theory is studied in Game Theory (Osborne & Rubinstein, 1994) in order to find rules for a game with certain characteristics. The players in this game, have each a preference function over the outcomes of the game. Given a choice rule from profiles of preferences to a subset of feasible outcomes, the question is whether a game can implement this choice rule in a such a way that a certain solution concept is attained. The players are self-interested and therefore information about their own preferences is kept private. A designer of the game looks for a mechanism that will produce the desired outcome (e.g., a Nash equilibrium) when the players reveal some part of their information as input to the designer. An algorithmic view to mechanism design is found in (Nisan & Ronen, 1999). The mechanism designer sets the algorithm for interaction among the agents and a payment structure that motivates the agents to participate in the interaction. This literature is concerned with agents that are self-interested and may hold privately known information about their preferences. We are interested in cooperative-systems.

In order to reduce the complexity of solving optimally the general decentralized control problem, we propose to design mechanisms for decentralizing the control, allowing the agents to synchronize their partial views from time to time through communication. That is, a mechanism reduces the optimization problem to two decision problems: 1) when to communicate and 2) what actions to choose between these communications. The local policies of action will be the solutions to each single-agent problem, induced by the mechanism. The local policy of communication will be obtained at the meta-level of control: i.e., the agents decide when to communicate and synchronize their partial views, once they know their policies of control. These mechanisms enable the agents to operate separately for certain periods of time. The question, then, is how to design mechanisms that will approx-

imate best the optimal joint policy of the decentralized problem. Notice that following our approach, each time that the agents apply the mechanism, they are faced with a problem to solve. In the economic approach the mechanism itself solves the problem.

Our Dec-POMDP-Com framework and mechanisms are general enough to capture models with discounted infinite horizon where the agents' objective is to maximize their joint reward. It can also capture scenarios with goal-oriented agents. When designing mechanisms for communication, we should also consider the relation between temporary goals adopted by each agent or the short-term accumulated rewards and the global goal or optimization function of the system. The goal-oriented Dec-POMDPs with possible information sharing are a special and difficult case. In these problems, a mechanism can serve as a method to impose local goal-states on the agents that are adopted from the global goal-states.

A decentralizing control mechanism (DCM) is a function from a decentralized process to two[10] single-agent problems. In the general case, the solution to each single-agent problem can be represented by a finite-state controller. The finite-state controller's transitions are over the agents' observations. In such case, the problem of finding a mechanism is not only how to decompose the joint reward functions into two local reward functions, but it also includes the problem of decomposing the set of global states into two sets of states, and the transition probability into two local transition probabilities on these local states. We represent the general mapping as follows:

$$DCM :< S, A_1, A_2, \Sigma, C_\Sigma, P, R, \Omega_1, \Omega_2, O > \rightarrow [(S_{A_1}, A_1, P_{A_1}, \Omega_{A_1}, R_1), (S_{A_2}, A_2, P_{A_2}, \Omega_{A_2}, R_2)]$$

In this paper, we restrict ourselves to single-agent problems that are represented by a Markov Decision Process. A Dec-POMDP-Com can be decomposed into two single-agent MDPs when it is jointly fully-observable. Assuming that the states S can be decomposed into two sets S_1 and S_2 when the transitions and observations are independent results in $S_{A_1} = S_1 = \Omega_{A_1}$, $S_{A_2} = S_2 = \Omega_{A_2}$, $P_{A_1} = P_1$ and $P_{A_2} = P_2$. In the simplest case, the Dec-MDP-Com is reward independent where R_1 and R_2 are clearly summed up to output R. In the general case, the Dec-MDP-Com is not reward independent, and R is given by a function of the local rewards that may add in a non-additive way (e.g., sub-additive or super-additive) that depends on both agents doing complementary or redundant actions.

The cost of communication C_Σ may include, in addition to the actual cost incurred by the communication, the cost resulting from the complexity of computing the decomposition (i.e., by applying the mechanism) as well as the cost resulting from the complexity of computing the agents' local policies.

The mechanism is applied each time the agents exchange information and thus, obtain full observability of the global state (we assume that the agents have full observability of the initial state s^0). Therefore, different approximations can be obtained for different policies of communication. Since we assume that the policy of communication of each agent is at the meta-level of control, any agent may initiate communication while solving its assigned local problem. These policies of communication trade-off the cost of communication with the value of the information obtained.

Figure 5 shows how both policies of action and communication are computed and executed given a mechanism for communication DCM. The optimal policy of action, δ_i^{A*},

10. In general, a mechanism can be applied to systems with n agents, in which case the decomposition of the Dec-POMDP-Com will be into n single-agent problems.

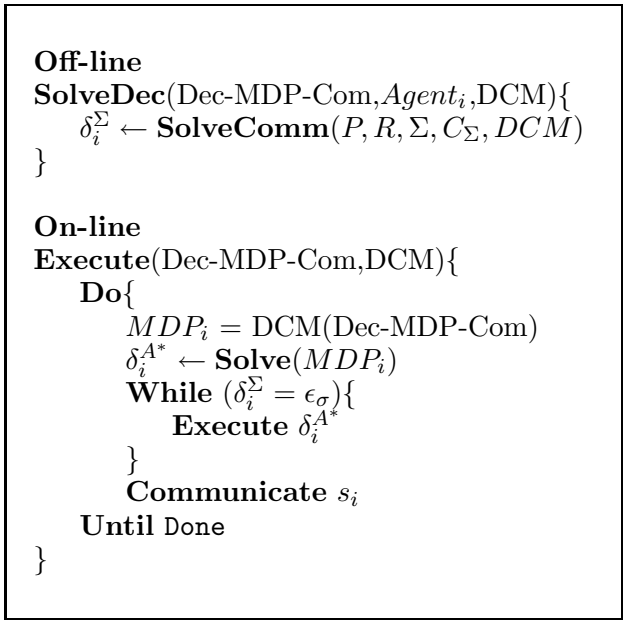

Figure 5: Mechanism design for communication-based control of decentralized cooperative processes

is found by solving the MDP induced by the mechanism. This computation considers the *local* reward function R_i (induced by the mechanism and the resulting single-agent problem). **SolveComm** is a function that computes the policy of communication δ_i^Σ (either an approximation or an optimal policy). The communication policy is computed based on the mechanism and the model of the decentralized problem, evaluated with the *joint* reward of the system. The mechanism is applied each time the agents communicate, allowing the agents to compute each its optimal policy of action that will be executed until the next time the agents' policies of communication instruct them to exchange information. In Sections 6.3 and 6.4, we study two algorithms for computing the policy of communication assuming a mechanism is given: one algorithm is based on a myopic-greedy approach and the other finds the optimal communication policy for a special class of monotonic Dec-MDPs.

6.1 Mechanism Characteristics

Following the economic approach, a good mechanism should have the following characteristics: *strategy-proof* , *efficient* and *budget-balanced* (Kfir-Dahav, Monderer, & Tenenholtz, 2000). A mechanism is strategy-proof if the agents are motivated to participate in it and will reveal their true preferences. A mechanism is efficient if its output state maximizes the utility of the system (i.e., the social-welfare is optimized taking into account the individual selfish utilities of the agents). A mechanism is budget-balanced if the total monetary transfer from the agents to the center (the system designer) is non-negative.

In our case, we are concerned with the design of mechanisms for communication in cooperative decentralized systems. Thus, the characterization of mechanisms is different.

Intuitively, agents exchange information to synchronize their knowledge and obtain full observability of the global state. Since communication has a cost associated with it, agents could only be synchronized from time to time. In between these periods agents work in a local manner on problems set by a mechanism such that eventually the agents approximate the actual global objective. Each local solution is computed optimally, and the policy of communication is an approximate or an optimal solution given a mechanism. In other words, the mechanism for communication is a means to interpret messages received and translate them into near-term problems that can be optimally solved locally.

A complete solution to a decentralized control problem comprises a mechanism (including the agents' optimal local policies of actions that solve the single-agent problems induced by the mechanism) and the policy of communication. In this section, we characterize the possible mechanisms for communication.

- **Stationary** — A mechanism is stationary if applying it on any two states that differ only in their time stamp, results always in the same decomposition of sub-problems.

- **Computational complexity** — The computation of the DCM mapping should be practical in the sense that the two single-agent problems will have complexity that is lower than the complexity of the decentralized problem with communication at free cost. There is a trade-off between the complexity of computing a mechanism and the global reward of the system. There may not be a simple way to split the Dec-POMDP into two separate processes. The whole motivation behind this approach is based on the idea that the mechanism itself has low computational complexity. In this paper, we assume that the mechanism can be implemented in constant time.

- **One-Pass** — A mechanism is, in general, a way to decompose one problem into two local problems. If this decomposition is ambiguous then the mechanism is not a one-pass mechanism. A one-pass mechanism does not require from the agents any additional information to fully determine their local MDPs, once information is exchanged and consequently the mechanism is applied. For example, if each resulting MDP has a single local goal, then the mechanism is indeed one-pass. Otherwise, agents may need to negotiate over which MDPs they should solve.

- **Complete** — If the Dec-POMDP-Com has a set of goal-states, then a mechanism is complete if there exists a communication policy such that it guarantees that the agents reach one of these goals whenever it is possible.

- **Efficient** — A mechanism DCM_1 is more efficient than another mechanism DCM_2 if the global reward attained by DCM_1 with some policy of communication is larger than the global reward attained by DCM_2 with any communication policy. A mechanism is *optimal* for a certain problem if it is at least as efficient as any other mechanism.

In the next section, we present more details about the meeting scenario that we studied to exemplify the practical approach to approximating the optimal solution of a decentralized cooperative process. In particular, we show how a mechanism is applied to that example. In this paper, we identify sufficient conditions under which an optimal mechanism exist in goal-oriented Dec-MDPs, when no information sharing is possible. In Sections 6.3 and

6.4, we present tractable (polynomial) algorithms that are aimed at approximating goal-oriented Dec-MDPs when direct communication is indeed feasible. Lemma 10 showed that the optimal solution for these problems is in NP.

6.2 Meeting under Uncertainty Example

We examine in more detail the Meeting under Uncertainty example used previously to illustrate our definitions and results so far. We consider the case that is modeled by a goal-oriented Dec-MDP-Com involving two agents that have to meet at some location as early as possible. The environment is represented by a 2D grid with discrete locations. The observations and the transitions are independent. The set of control actions includes moving North, South, East and West, and staying at the same location. The agents can initiate direct communication. Each agent's partial view (which is locally fully-observable) comprises the agent's location coordinates. There is uncertainty regarding the outcomes of the agents' actions. That is, with probability P_i, agent i arrives at the desired location after having taken a move action, but with probability $1 - P_i$ the agent remains at the same location. Due to this uncertainty in the effects of the agents' actions, it is not clear that setting a predetermined meeting point to which the agents will optimally move is the best strategy for designing these agents. Agents may be able to meet faster if they change their meeting place after realizing their actual locations. This can be achieved by exchanging information on the locations of the agents, that otherwise are not observable.

Adding direct communication to this setting allows the agents to attain full observability of the global state of the system. Each time the agents exchange information, a mechanism is applied to the decentralized process resulting in two single-agent goal-oriented MDPs that can be solved optimally. We have implemented the mechanism that leads the agents to adopt a single local goal: reach the location in the middle of the shortest Manhattan path between the agents' locations (this distance is revealed when information was exchanged). This mechanism is stationary, has low computational complexity (i.e., $O(1)$ since each agent computes the location in the middle of the Manhattan path with the information acquired by communication), and it is a one-pass mechanism. Once this goal location is determined, each agent can solve its own MDP and reach that location. Section 6.3 presents a myopic-greedy policy of communication for which this mechanism is complete, each agent indeed can reach its local goal.

Intuitively, it is desirable for a mechanism to set a meeting place in the middle of the shortest Manhattan path that connects the two agents because in the absence of communication, the cost to meet at that point is minimal. This can be shown by computing the joint expected time to meet, Θ_{nc}, for any pair of possible distances between the two agents and any location in the grid, when no communication is possible. The minimal value is attained when these distances are equal. To simplify the exposition we use a function that takes advantage of the specific characteristics of the example. In Section 6.3, we return to the notation of the general case. The notation is as follows: agent 1 is at distance d_1 from the meeting location, agent 2 is at distance d_2 from that location, the system incurs a cost of one at each time period if the agents have not met yet (i.e., this cost is the negative reward $R(s, a_1, a_2, s')$ attained from moving from state s to state s' where s' occurs at one time unit later than s) and P is the transition probability of the Dec-MDP-Com. If both

agents are at the meeting location, the joint expected time to meet is zero, $\Theta_{nc}(0,0) = 0$. If only agent 2 is at the meeting location, but agent 1 has not reached that location yet, then the joint expected time to meet is given by

$$\Theta_{nc}(d_1, 0) = P_1(-1 + \Theta_{nc}(d_1 - 1, 0)) + (1 - P_1)(-1 + \Theta_{nc}(d_1, 0))$$

i.e., with probability P_1 agent 1 succeeds in decreasing its distance to the meeting location by one, and with probability $1 - P_1$ it fails and remains at the same location. Recursively, we can compute the remaining joint expected time to meet with the updated parameters. Similarly for agent 2: $\Theta_{nc}(0, d_2) = P_2(-1 + \Theta_{nc}(0, d_2 - 1)) + (1 - P_2)(-1 + \Theta_{nc}(0, d_2))$. If none of the agents has reached the meeting place yet, then there are four different cases in which either both, only one, or none succeeded in moving in the right direction and either or not decreased their distances to the meeting location respectively:

$$\Theta_{nc}(d_1, d_2) = P_1 P_2(-1 + \Theta_{nc}(d_1 - 1, d_2 - 1)) + P_1(1 - P_2)(-1 + \Theta_{nc}(d_1 - 1, d_2)) +$$

$$+(1 - P_1)P_2(-1 + \Theta_{nc}(d_1, d_2 - 1)) + (1 - P_1)(1 - P_2)(-1 + \Theta_{nc}(d_1, d_2))$$

We computed $\Theta_{nc}(d_1, d_2)$ for all possible distances d_1 and d_2 in a 2D grid of size 10×10. the minimal expected time to meet was obtained when $d_1 = d_2 = 9$ and the expected cost was -12.16.

In summary, approximating the optimal solution to the Meeting under Uncertainty example when direct communication is possible and the mechanism applied is the one described above will unfold as follows: At time t_0, the initial state of the system s^0 is fully observable by both agents. The agents set a meeting point in the middle of a Manhattan path that connects them. Denote by d_0 the distance between the agents at t_0 and $g_{t_0} = (g^1_{t_0}, g^2_{t_0})$ the goal-state set at t_0. Each one of the agents can move optimally towards its corresponding component of g_{t_0} following the optimal policy of action each can compute. Each agent moves independently in the environment because the transitions and observations are independent. Each time t, when the policy of communication instructs an agent to initiate exchange of information, the current Manhattan distance between the agents d_t is revealed to both. Then, the mechanism is applied, setting a possibly new goal-state g_t, which decomposes into two components one for each agent. This goal-state g_t is in the middle of the Manhattan path that connects the agents with length d_t revealed through communication.

In the following two sections, we present two approaches to computing the policy of communication δ^Σ assuming a mechanism is given. One approach is the myopic-greedy approach and the other finds the optimal δ^Σ for monotonic Dec-MDPs that will be defined in the corresponding section. We first present these approaches in general (assuming some mechanism is given) and then present empirical results obtained for the Meeting under Uncertainty scenario and for the mechanism described in this section.

6.3 A Myopic-greedy Approach to Direct Communication

We consider a goal-oriented Dec-POMDP, which is jointly fully-observable and whose transitions and observations are independent. The global reward function is not constrained in any way. The first approximation that we present to the optimal decentralized control

problem with direct communication is myopic-greedy, i.e., each time an agent makes a decision, it chooses the action with maximum expected accumulated reward assuming that agents are only able to communicate once along the whole process. We denote the optimal policies induced by the mechanism applied δ_1^{A*} and δ_2^{A*} respectively. The complexity of computing these policies for action is in the P class (dynamic programming).

The expected global reward of the system, given that the agents do not communicate at all and each follows its corresponding optimal policy δ_i^{A*} is given by the value of the initial state s^0: $\Theta_{nc}^{\delta}(s^0, \delta_1^{A*}, \delta_2^{A*})$. This value can be computed by summing over all possible next states and computing the probability of each agent reaching it, the reward obtained then and the recursive value computed for the next states.

$$\Theta_{nc}^{\delta}(s^0, \delta_1^{A*}, \delta_2^{A*}) = \Sigma_{(s_1', s_2')} P_1(s_1'|s_1^0, \delta_1^{A*}(s_1^0)) P_2(s_2'|s_2^0, \delta_2^{A*}(s_2^0))$$

$$(R(s'|s^0, \delta_1^{A*}(s_1^0), \delta_2^{A*}(s_2^0)) + \Theta_{nc}^{\delta}(s', \delta_1^{A*}, \delta_2^{A*}))$$

At each state, each agent decides whether to communicate its partial view or not based on whether the expected cost from following the policies of action, and having communicated is larger or smaller than the expected cost from following these policies of action and not having communicated. We denote the expected cost of the system computed by agent i, when the the last synchronized state is s^0, and when the agents communicate once at state s and continue without any communication, $\Theta_c(s^0, s_i, \delta_1^{A*}, \delta_2^{A*})$:

$$\Theta_c(s^0, s_1, \delta_1^{A*}, \delta_2^{A*}) = \Sigma_{s_2} \overline{P}(s_2|s_2^0, \delta_2^A)$$

$$(\overline{R}((s_1, s_2)|s^0, \delta_1^{A*}(s_1^0), \delta_2^{A*}(s_2^0)) + \Theta_{nc}^{\delta}((s_1, s_2), \delta_1^{A*}, \delta_2^{A*}) + C_\Sigma * Flag)$$

Flag is zero if the agents reached the global goal-state before they reached state s. We denote by $t(s)$ the time stamp in state s. $\overline{P}(s|, s^0, \delta_1^A, \delta_2^A)$ is the probability of reaching state s from state s^0, following the given policies of action.

$$\overline{P}(s'|s, \delta_1^A, \delta_2^A) = \begin{cases} 1 & if\ s = s' \\ P(s'|s, \delta_1^A(s_1), \delta_2^A(s_2)) & if\ t(s') = t(s) + 1 \\ 0 & if\ t(s') < t(s) + 1 \\ \Sigma_{s''} \overline{P}(s'|s'', \delta_1^A, \delta_2^A) P(s''|s, \delta_1^A, \delta_2^A) & else \end{cases}$$

Similarly, $\overline{P_1}$ ($\overline{P_2}$) can be defined for the probability of reaching s_1' (s_2'), given agent 1 (2)'s current partial view s_1 (s_2) and its policy of action δ_1^A (δ_2^A).

The accumulated reward attained while the agents move from state s^0 to state s is given as follows:

$$\overline{R}(s^0, \delta_1^A, \delta_2^A, s) = \begin{cases} R(s^0, \delta_1^A(s_1), \delta_2^A(s_2), s) & if\ t(s) = t(s^0) + 1 \\ \Sigma_{s''} \overline{P}(s''|\delta_1^A, \delta_2^A, s^0) P(s|\delta_1^A, \delta_2^A, s'') & \\ (\overline{R}(s^0, \delta_1^A, \delta_2^A, s'') + R(s'', \delta_1^A(s_1''), \delta_2^A(s_2''), s)) & if\ t(s) > t(s^0) + 1 \end{cases}$$

Lemma 11 *Deciding a Dec-MDP-Com with the myopic-greedy approach to direct communication is in the P class.*

Proof. Each time the agreed-upon mechanism is applied each agent faces a single-agent MDP, which can be solved optimally in polynomial time. The complexity of finding the communication policy is the same as dynamic programming (based on the formulas above), therefore computing the policy of communication is also in P. There are $|S|$ states for which Θ^{δ}_{nc} and Θ_c need to be computed, and each one of these formulas can be solved in time polynomial in $|S|$. □

Lemma 12 $\Theta^{\delta}_{nc}(s^0, \delta_1^{A*}, \delta_2^{A*}) \leq \Theta_c(s^0, s_i, \delta_1^{A*}, \delta_2^{A*})$

Proof. Θ^{δ}_{nc} is the expected joint cost (negative) incurred by the joint policy assuming that the agents set a meeting location at time 0, and they do not communicate until they meet at such location. If the world were deterministic then the value of a joint policy computed by Θ^{δ}_{nc} will be equal to the value of a joint policy computed by Θ_c. In our case, there exists uncertainty in the outcome of the actions, i.e., the transition probability of the Dec-MDP can be larger than zero. Myopic-greedy agents may synchronize their information from time to time. Thus, they can correct their meeting location based on their actual locations revealed by the communication. When the agents do not communicate they do not have the chance to correct their policy with respect to another meeting location that may be closer to them, had they known their actual current locations. Therefore, the value of a joint policy computed with the myopic-greedy approach is at least as large as the value of the joint policy computed without any communication. □

6.3.1 Experiments - Myopic-greedy Approach

We present empirical results obtained when the myopic-greedy approach was implemented for the Meeting under Uncertainty example (explained in Section 6.2)[11] The messages in the language of communication Σ are the agents' own observations, i.e., their location coordinates. In all the experiments run, we assumed that $P_1 = P_2$ and we refer to these uncertainties as P_u. The mechanism that is applied whenever the agents communicate at time t results in each agent adopting a local goal-state, that is set at the location in the middle of the Manhattan path connecting the agents (the Manhattan distance between the agents is revealed at time t). We compare the joint utility attained by the system in the following four different scenarios:

1. No-Communication — The meeting point is fixed at time t_0 and remains fixed along the simulation. It is located in the middle of the Manhattan path that connects between the agents, known at time t_0. Each agent follows its optimal policy of action without communication to this location.

2. Ideal — This case assumes that C_Σ is zero, and that the agents communicate at every time step, this is the highest global utility that both agents can attain. Notice, though, that this is not the optimal solution we are looking for, because we do assume that communication is not free. Nevertheless, the difference in the utility obtained in these first two cases shed light on the trade-off that can be achieved by implementing non-free communication policies.

11. These results appeared also in (Goldman & Zilberstein, 2003).

3. Communicate SubGoals — A heuristic solution to the problem, which assumes that the agents have a notion of sub-goals. They notify each other when these sub-goals are achieved, eventually leading the agents to meet.

4. Myopic-greedy Approach — Agents act myopically optimizing the choice of when to send a message, assuming no additional communication is possible. For each possible distance between the agents, a policy of communication is computed such that it stipulates when it is the best time to send that message. By iterating on this policy agents are able to communicate more than once and thus approximate the optimal solution to the decentralized control with direct communication problem. The agents continue moving until they meet.

The solution to the No-Communication case is similar to the single global goal-oriented Dec-MDP case we analyzed in Lemma 5. This case can be solved analytically for the Meeting under Uncertainty example, by computing the expected cost[12] $\Theta_{nc}(d_1, d_2)$ incurred by two agents located at distances d_1 and d_2 respectively from the goal-state at time t_0 as computed in Section 6.2.

In the Ideal case, a set of 1000 experiments was run in which the cost of communication was assumed to be zero. Agents communicate their locations at every time instance, and update the location of the meeting place accordingly. Agents move optimally to the last synchronized meeting location.

For the third case tested (Communicate SubGoals) a sub-goal was defined by the cells of the grid with distance equal to $p * d/2$ from the fixed current meeting point. p is a parameter of the problem that determines the radius of the circle that will be considered a sub-goal. Each time an agent reaches a cell inside the area defined as a sub-goal, it initiates exchange of information (therefore, p induces the communication strategy). d expresses the Manhattan distance between the two agents, this value is accurate only when the agents synchronize their knowledge. That is at time t_0 the agents determine the first sub-goal as the area bounded by a radius of $p * d_0/2$ and, which center is located at $d_0/2$ from each one of the agents. Each time t that the agents synchronize their information through communication, a new sub-goal is determined at $p * d_t/2$. Figure 6 shows how new sub-goals are set when the agents transmit their actual location once they reached a sub-goal area. The meeting point is dynamically set at the center of the sub-goal area.

Experiments were run for the Communicate SubGoals case for different uncertainty values, values of the parameter p and costs of communication. These results show that agents can obtain higher utility by adjusting the meeting point dynamically rather than having set one fixed meeting point. Agents can synchronize their knowledge and thus they can set a new meeting location instead of acting as two independent MDPs that do not communicate and move towards a fixed meeting point (see Figure 7. Each data point represents the average over 1000 runs). Nevertheless, for certain values of p, the joint utility of the agents is actually smaller than the joint utility achieved in the No-Communication case (2 MDPs in the figure). This points out the need to empirically tune up the parameters needed in the implemented heuristic, as opposed to a formal approach to approximating the solution to the problem as is shown in the Myopic-greedy case.

12. Cost and utility are used interchangeably as appropriate meaning cost is minimized and utility is maximized.

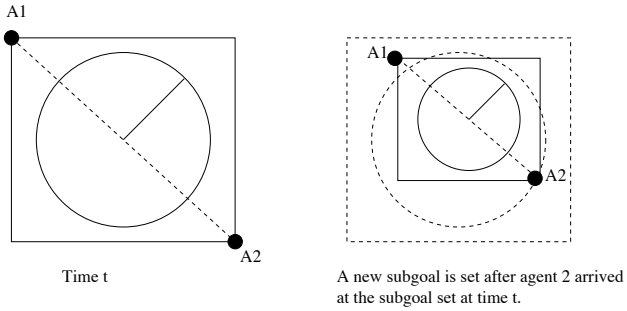

Figure 6: Goal decomposition into sub-goal areas.

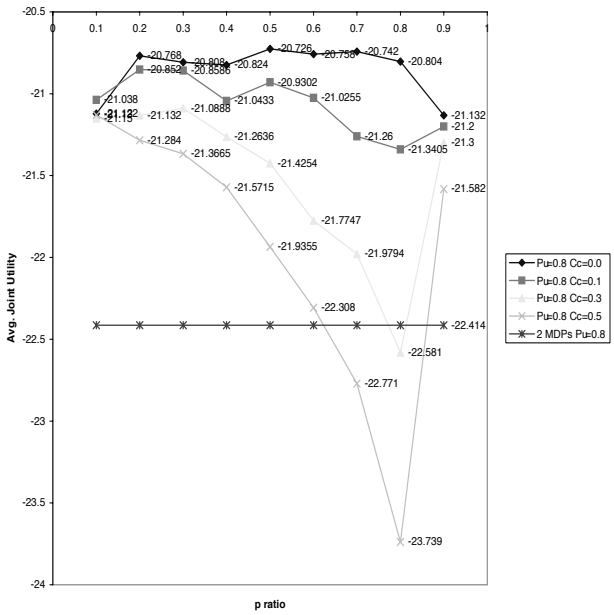

Figure 7: The average joint utility obtained when sub-goals are communicated.

In the Myopic-greedy case, we design the agents to optimize the time when a message will be sent assuming that they can communicate only once. At the off-line planning stage, the agents compute their expected joint cost, $\Theta_c(s^0, s_i, \delta_1^{A*}, \delta_2^{A*})$ for every possible synchronized distance that can occur between the two agents (s^0) and for every time t (in each agent's partial view s_i, up to some maximal constant). The local policies are the optimal local policies to the goals adopted after applying the mechanism each time the agents get synchronized. In the Meeting under Uncertainty scenario we study, Θ_c is the expected joint cost from taking control actions during t time steps, communicating at time $t+1$ if the agents have not met so far, and following the optimal policy of control actions towards the expected goal-state without communicating (at an expected cost of $\Theta_{nc}(d_1, d_2)$ as computed for the No-Communication case). In the case that the agents met before the t time steps, then the expected cost considers the relevant expected joint cost that the agents incur until they met, i.e., less than t.

At each time t, each one of the agents knows a meeting location, that is the goal location computed from the last exchange of information. Consequently, each agent moves optimally towards this goal-state. In addition, the myopic-greedy policy of communication is found by computing the earliest time t, for which $\Theta_c(d_1+d_2, s_1, \delta_1^{A*}, \delta_2^{A*}) < \Theta_{nc}(d_1, d_2)$, that is, what is the best time to communicate such that the expected cost is the least. The myopic-greedy policy of communication is a vector where each entry corresponding to a possible distance between the agents, specifies the time to communicate.

We found the myopic-greedy communication policies for the Meeting under Uncertainty problem where P_u takes any of the following values: $\{0.2, 0.4, 0.6, 0.8\}$, the cost of taking a control action is $R_a = -1.0$ and the costs of communicating C_Σ tested were $\{-0.1, -1.0, -10.0\}$. The resulting policies of communication are presented in Appendix A. For the smallest cost tested, it is always beneficial to communicate rather early, no matter the uncertainty in the environment, and almost no matter what d_0 is (the differences in time are between 2 and 4). For larger costs of communication and a given P_u, the larger the distance between the agents, the later they will communicate (e.g., when $P_u = 0.4$,$C_\Sigma = -1$ and $d = 5$, agents should communicate at time 4, but if $C_\Sigma = -10$, they should communicate at time 9). For a given C_Σ, the larger the distance between the agents is, the later the agents will communicate (e.g., when $P_u = 0.4$,$C_\Sigma = -10$ and $d = 5$, agents should communicate at time 9, but if $d = 12$, they should communicate at time 16). The results from averaging over 1000 runs show that for a given cost C_Σ as long as P_u decreases (the agent is more uncertain about its actions' outcomes), the agents communicate more times.

In the 1000 experiments run, the agents exchange information about their actual locations at the best time that was myopically found for d_0 (known to both at time t_0). After they communicate, they know the actual distance d_t, between them. The agents follow the same myopic-greedy communication policy to find the next time when they should communicate if they did not meet. This time is the best time found by the myopic-greedy algorithm given that the distance between the agents was d_t. Iteratively, the agents approximate the optimal solution to the decentralized control problem with direct communication by following their independent optimal policies of action, and the myopic-greedy policy for communication. Results obtained from averaging the global utility attained after 1000 experiments show that these myopic-greedy agents can perform better than agents who communicate sub-goals (that is a more efficient approach than no communicating at all).

The results for $C_\Sigma = 0.1$ are presented in Tables 4 and 5. Additional results obtained for other costs of communication appear in Appendix B.

	Average Joint Utility			
P_u	No-Comm.	Ideal	SubGoals[13]	Myopic-Greedy
0.2	-104.925	-62.872	-64.7399	-63.76
0.4	-51.4522	-37.33	-38.172	-37.338
0.6	-33.4955	-26.444	-27.232	-26.666
0.8	-24.3202	-20.584	-20.852	-20.704

Table 4: $C_\Sigma = -0.10$,$R_a = -1.0$.

The Myopic-greedy approach attained utilities statistically significantly greater than those obtained by the heuristic case when $C_\Sigma = -0.1$.[14] Ideal always attained higher utilities than Myopic-greedy, but when $C_\Sigma = -0.1$ and $P_u = 0.4$ both values were not significantly different with probability 98%. When $C_\Sigma = -1$ the utilities attained for the Myopic-greedy approach when $P_u < 0.8$ are significantly greater than the results obtained in the heuristic case and for $P_u = 0.8$, the heuristic case for the best p was found to be better than Myopic-greedy (Myopic-greedy obtained -21.3, and the SubGoals with $p = 0.1$ attained -21.05 (variance=2.18)). The utilities attained by the Myopic-greedy agents, when $C_\Sigma = -10$ and P_u in $\{0.2, 0.4\}$, were not significantly different from the SubGoals case for the best p with probabilities 61% and 82%, respectively. However, the heuristic case yielded smaller costs for the other values of $P_u = 0.6, 0.8$. One important point to notice is that these results consider the best p found for the heuristic, but in general a designer may not know this value. In all the settings tested, Myopic-greedy always attain utilities higher than the results attained in the SubGoals case with the worst p.

	Average Communication Acts Performed			
P_u	No-Comm.	Ideal $C_\Sigma = 0$	SubGoals	Myopic-greedy
0.2	0	31.436	5.4	21.096
0.4	0	18.665	1	11.962
0.6	0	13.426	1	8.323
0.8	0	10.292	1	4.579

Table 5: $C_\Sigma = -0.10$,$R_a = -1.0$.

For the same parameters tested so far, experiments were run with two deadlines, T in $\{8, 15\}$. Examples of the communication policies computed when the cost of communication was set to -10 are presented in Appendix C. In general, the myopic-greedy policy found may instruct the agent not to communicate if $\Theta_{nc} < \Theta_c$, i.e., had the agents communicated, unnecessary information had been exchanged. On the other hand, this policy may instruct an agent not to communicate, if given a deadline, the agent is not going to be able to reach the goal. In the first case, limiting the deadline to be earlier, results in policies of communication that stipulate that the agent should communicate earlier than in the case

13. The results are presented for the best p, found empirically.
14. Statistical significance has been established with t-test.

when no deadlines are added (for large values of d_0 with low uncertainties P_u). When no deadlines are assumed, the agents may benefit from exchanging information later. When a short deadline is assumed, if the agents have the chance to meet without communication given a later deadline, they will need to communicate earlier if the time stipulated in the policy with no deadlines is larger than the deadline. If the deadline is large enough for these agents to meet, they do not need to communicate at all. For shorter d values if the policy with no deadline allows the agent to communicate at a time smaller than the deadline the same policy holds.

In the second case, the agents may not communicate if they may not meet at all by the stipulated deadline. The empirical results show that by extending the deadline, agents benefit from communicating at a time that is later than the time found by the myopic-greedy policy when no deadlines were assumed. Since, in this case there is a chance of not meeting at all, agents need to wait more time until it becomes beneficial to communicate.

6.4 An Optimal Communication Policy

We characterize the set of *monotonic* goal-oriented Dec-MDPs for which we provide an algorithm that finds the optimal policy of communication given a mechanism. First, we define the *rank* of a global state to be a function $\rho : S \rightarrow \mathcal{N}$ such that when s is in G, $\rho(s) = 0$. For example, the rank can express the expected cost of the optimal policy to reach the global goal-state.

Definition 13 (Monotonic GO-Dec-MDPs) *A goal-oriented Dec-MDP is* monotonic *with respect to a given mechanism if there exists a ranking function ρ such that for all global states s and all global states s' reachable from s ($s' \neq s$), following the joint policy induced by the mechanism, $\rho(s') < \rho(s)$.*

Although the transitions are between states with non-increasing rank, the uncertainty about the outcomes of the agents' actions does exist, i.e., the agents' actions can fail.

Following our approach to decentralized control with mechanisms for communication, each agent can compute an optimal control policy, δ_i^{A*}, given a local goal-state, induced by the mechanism. The algorithm presented in this section finds the optimal policy of communication at the meta-level of control. This policy instructs the agents to synchronize the information in their partial views at the most beneficial time.

We assume a goal-oriented Dec-MDP with independent transitions and observations and a finite horizon T (time is discrete). Each agent i can choose an action a_i^j, $1 \leq j \leq m$, from its set of actions A_i. The notation we use in the algorithm is as follows: a_i^{j*} denotes the optimal action that agent i chooses given its partial view, its underlying MDP_i and δ_i^{A*}. After successfully performing the optimal control action a_i^{j*}, agent i moves to a state s_i' that is denoted by $a_i^{j*}(s_i, 1)$. $a_i^{j*}(s_i, 0)$ represents the resulting state when agent i fails to perform action a_i^{j*}. $EU^i(s, s_i, t)$ denotes the expected joint utility of the multi-agent system, computed by agent i at time t, when the synchronized global-state is s and agent i's partial view is s_i. The set of global-states are ordered by the rank assumed for monotonic Dec-MDPs. The states with rank k are represented by S^k (K is the largest rank that a state can have). For example, if the agents' goal is to meet, then the state of the Dec-MDP-Com may be given by the Manhattan distance between the agents, and the goal-state is reached

when this distance is zero. The rank is given, then, by the possible Manhattan distances between the agents given a 2D grid. The algorithm for computing the optimal policy of communication is based on backward induction. It is shown in Figure 8. EU_C and EU_{NC} are two temporary variables that denote the expected joint utility when agent i decides to communicate or when it does not. *Penalty* is the reward obtained when the agents do not achieve their goal by the time limit of the problem. $\Phi_C(P_1, P_2, R, C_\Sigma, s^0, s_i, t)$ computes the expected joint utility when agent i communicates its partial view s_i at time $t+1$, and the current synchronized global-state is s^0. This function computes the possible synchronized global-states in which the system could be in (given that agent i communicates), the expected costs incurred to arrive at these states, and the joint expected utility of these new states. Notice that since we deal with monotonic Dec-MDPs, the ranks of these new states are at most as high as the rank of the last synchronized state.

Theorem 2 *OptCom computes the optimal communication policy for a given monotonic goal-oriented Dec-MDP-Com with independent transitions and observations and a given mechanism.*

Proof. The correctness proof of the algorithm is given by induction. The induction is both on the time t that elapses and on the rank k of the global states.

Basis: If the synchronized state that is known by all the agents is a goal-state ($S^k = S^0$) then all agents are aware of having achieved this global goal-state. Therefore, it is optimal not to communicate then. It is also optimal not to communicate at time $t = T-1$. Based on the Dec-MDP-Com model, the agents *decide* to communicate at time t, but the actual communication act occurs at time $t+1$. If the time limit is T then it is not beneficial to decide to communicate at time $T-1$.

We assume that the algorithm OptCom computes the optimal time to communicate for any state $s \in S^k$ for any $0 \leq k \leq K'$ (for some $K' < K$, K is the largest rank of a global state), and for any time $0 \leq t < T$.

By induction on k and t, we prove that the *OptCom* algorithm presented in Figure 8 finds the optimal time to communicate for any state $s \in S^{K'+1}$ and time t. Following the algorithm, when the agent decides whether to communicate or not in state $s \in S^{K'+1}$, it compares its utility when it does not communicate (EU_{NC}) with its utility when it does communicate (EU_C). If the agent does not communicate, then it chooses the optimal control action a_i^{j*} based on its underlying Markov decision process induced by the given mechanism. The outcome of this action is given by the transition probability P_i, i.e., with probability P_i agent i moves to state $s'_i = a_i^{j*}(s_i, 1)$ and with probability $1 - P_i$ it moves to a state $s''_i = a_i^{j*}(s_i, 0)$. Therefore, $EU_{NC} = (1 - P_i)EU^i(s, s''_i, t + 1) + P_i EU^i(s, s'_i, t + 1)$. Since we assume that the Dec-MDP is monotonic, we know that $s''_i \preceq s_i$ and $s'_i \preceq s_i$, therefore $s''_i \wedge s'_i \in S^k$ for some $k < K' + 1$. Based on the assumption of the induction, these values are optimal and have taken into account the optimal decision when to communicate.

The expected utility if the agent decides to communicate is $EU_C = EU(s''', 0)$. A communication act always succeeds because we assume messages are reliable. Time becomes 0 because after communicating the agents become synchronized (thus they are reset). Since the Dec-MDP is monotonic, $s''' \preceq s$. Therefore, the expected utility of this state at time 0 is known and has been computed optimally. Therefore, the algorithm presented finds the

function OptCom(DCM, MDP_i, Dec-MDP-Com)
returns the optimal communication policy,
$Policy(s, s_i, t) \leftarrow 0$ if agent i should not
communicate at time $t+1$, when s is the last
synchronized state, and s_i is its current partial view,
otherwise i communicates at the time indicated by Policy().
inputs: DCM is the mechanism for communication.
MDP_i is the underlying MDP for agent i
resulting from applying DCM on Dec-MDP-Com.
Dec-MDP-Com=$< S, A_1, A_2, \Sigma, C_\Sigma, P, R, T >$

For each state $s \in S^k$ (for $k \leftarrow 0$ to K)
For time $t \leftarrow T-1$ **to** 0
For each $s_i \in S_i$
if $(k = 0)$ **then** /*agents reached the global goal-state*/
$Policy(s, s_i, t) \leftarrow 0$
$EU^i(s, s_i, t) \leftarrow 0$
else if $(t = T-1)$ **then** /*time is over*/
$Policy(s, s_i, t) \leftarrow 0$
$EU^i(s, s_i, t) \leftarrow Penalty$/*agents did not reach the global goal-state*/
else if $(t = 0)$ **then** /*agents are synchronized*/
$Policy(s, s_i, t) \leftarrow 0$
$EU^i(s, s_i, t) \leftarrow ComputeEU_{NC}(MDP_i, s, s_i, R, t)$
else
$EU_{NC} \leftarrow ComputeEU_{NC}(MDP_i, s, s_i, R, t)$
$EU_C \leftarrow \Phi_C(P, R, C_\Sigma, s^0, s_i, t)$
if $(EU_{NC} > EU_C)$ **then**
$Policy(s, s_i, t) \leftarrow 0$
$EU^i(s, s_i, t) \leftarrow EU_{NC}$
else /*communicate at t+1*/
$Policy(s, s_i, t) \leftarrow t + 1$
$EU^i(s, s_i, t) \leftarrow EU_C$
return Policy

function $ComputeEU_{NC}(MDP_i, s, s_i, R, t)$
returns the expected joint utility given that
the agent does not communicate.
inputs: MDP_i, underlying MDP for agent i.
s, the last Dec-MDP-Com synchronized state.
s_i, the current partial view of agent i
R, the Dec-MDP-Com reward function.
t, the current time.

$EU_{Succ} \leftarrow EU^i(s, a_i^{j*}(s_i, 1), t + 1)$
$EU_{Fail} \leftarrow EU^i(s, a_i^{j*}(s_i, 0), t + 1)$
return $(1-P_i)(R + EU_{Fail})+P_i(R + EU_{Succ})$

Figure 8: The OptCom algorithm for monotonic Dec-MDP-Com with independent transitions.

optimal policy of communication given a monotonic Dec-MDP with independent transitions. □

Lemma 13 *Deciding a monotonic Dec-MDP-Com with OptCom is in P.*

Proof. Each time the agreed-upon mechanism is applied each agent faces a single-agent MDP, which can be solved optimally in polynomial time. The complexity of finding the optimal communication policy by running the *OptCom* algorithm is the same as dynamic programming, therefore computing the resulting policy of communication is also in P. □

6.4.1 Experiments - Monotonic Goal-oriented Dec-MDPs

The performance of the *OptCom* algorithm is exemplified on the Meeting under Uncertainty example presented in Section 6.2. We compare here the *OptCom* results to the No-Communication, Ideal and Myopic-greedy cases.

The results from experimenting with different communication costs (and averaging over 1000 runs) appear in Table 6 and in Appendix D. The cost of taking a moving action was set to -1.0. In the Ideal case, C_Σ is zero. Note that the setup in this setting of experiments differs from the setting we studied in Section 6.3.1 because here the agents are penalized if they do not meet by the time limit.[15]

	Average Joint Utility			
P	No-Comm.	Ideal	Myopic-greedy	OptCom
0.2	-71.138	-62.968	-62.834	-63.226
0.4	-42.112	-37.372	-37.778	-37.734
0.6	-29.078	-26.518	-26.782	-26.642
0.8	-22.344	-20.52	-20.714	-20.574

Table 6: $C_\Sigma = -0.10$.

The results obtained by *OptCom* in Table 6 for $P = 0.2$ are not significantly different neither from Ideal (with probability 65%) nor from Myopic-greedy (with probability 48%). Table 7 shows the average number of messages exchanged in each one of the tested cases when the cost of communication was -0.1.

	Average Communication Acts Performed			
P	No-Comm.	Ideal	Myopic-greedy	OptCom
0.2	0	31.484	20.778	30.613
0.4	0	18.686	12.171	17.867
0.6	0	13.259	8.252	12.321
0.8	0	10.26	4.588	9.287

Table 7: $C_\Sigma = -0.10$.

15. Although we do have a program that can precisely compute the solution for the No-Communication case, the results presented were obtained from averaging over 1000 empirical tests, which result less time consuming than the analytical solution for the finite-horizon case.

7. Discussion

Decentralized control problems are very intriguing from a theoretical point of view as well as from a practical point of view. From a theoretical perspective, decentralized partially-observable Markov decision processes serve as a formal framework to study the foundations of multi-agent systems. A more solid formal footing is given to multi-agent systems' research (e.g., (Guestrin & Gordon, 2002), (Peshkin et al., 2000), (Pynadath & Tambe, 2002), (Claus & Boutilier, 1998)). Our study focuses on computing off-line decentralized policies of control for cooperative systems. The first part of this paper analyzes the complexity of solving these problems *optimally* for certain classes of decentralized control that are formally identified. We found critical transitions in complexity between classes of problems that range from NEXP to P. In the second part of the paper, we extend the decentralized process with the possibility of direct communication among the agents that incurs a certain cost. Communication allows the agents to synchronize their knowledge and thus eliminate the uncertainty about the global state of the world (at least at certain times).

From a practical perspective, decentralized control problems appear frequently in real-world applications where the decision-makers may be robots placed at separate geographical locations or computational processes distributed in information space. While the classes of Dec-POMDPs that we identify constrain the problems we can solve, the different categories seem to match many practical applications. Independent transitions and observations arise in examples such as multi-agent mapping, flexible manufacturing and multiple-rovers working on data-collection in uncertain terrains, when the agents' actions are not strongly coupled. Goal-oriented behavior is relevant in these examples when the agents' behavior is aimed at reaching specific states. Monotonic goal-oriented Dec-MDPs are also very interesting, and include many real-world applications such as: information-gathering, which are monotonic because information is always being added to what has previously been acquired. Another example involves a system that allocates tasks to agents, which is monotonic because previously completed tasks cannot be undone. Similarly, robots involved in a manufacturing process can be represented by a monotonic Dec-MDP as long as their actions cannot break any previously manufactured part in the production line. Actions may still fail, for example a robot may fail in assembling some hardware, but in such a case it remains in the same state that it was before it started to perform the action.

We analyzed the notion of information sharing in decentralized systems by distinguishing among three possible sources for information: indirect communication attained by agents observing *dependent* observations, direct communication achieved by adding an external language of communication, and common uncontrollable features, which are not affected by any of the agents' actions but can be observed by all the agents in the system. The typical distinction previously made in the literature is between systems with no communication and systems with a predefined language of communication, which typically does not incur any costs, overlooking the fact that dependent observations offer yet another form of communication (Pynadath & Tambe, 2002; Decker & Lesser, 1992; Grosz & Kraus, 1996; Durfee, 1988; Roth, Vail, & Veloso, 2003). Xuan et al.(Xuan, Lesser, & Zilberstein, 2001) address the problem of combining communication acts into the decision problem of a group of cooperative agents. Their framework is similar to ours but their approach is heuristic. We proved that the language of the observations is sufficient in order to reach an optimal

decentralized solution (assuming all the messages incur the same cost). This leads to the understanding that any other type of communication can serve as an approximation to the optimal solution, which may be easier to obtain. We study the trade-off between the cost of sharing information (in the agent's partial views) and the value of this information and its effect on the joint utility of the system.

In addition to presenting a formal framework of decentralized control, we introduced tractable algorithms for solving optimally certain classes of Dec-MDPs. The first algorithm to solve optimally decentralized MDPs with a certain reward structure appeared in (Becker et al., 2003). Here, we add two optimal algorithms aimed at goal-oriented decentralized control. We also suggested mechanism design as a tool to approximate optimal intractable decentralized solutions. Based on such mechanisms, we study two approximation algorithms to compute the policy of communication when direct communication is feasible: the myopic-greedy approach, and the optimal approach for monotonic Dec-MDPs. Both approximations have polynomial complexity.

The contribution of this paper is in framing and categorizing fundamental issues in decentralized control of cooperative systems. In particular, we characterize and study the complexity of goal-oriented behavior, jointly fully-observable processes and independent transitions and observations, which result in interesting and practical classes of control problems. We also study three sources for information sharing in such decentralized systems and provide algorithms that compute optimal solutions as well as tractable approximations for these problems. Future research will look at algorithms for decentralized control with direct communication achieved by implementing languages of communication different from the language of observations. We will study more general models of communication that allow exchanging partial information and handle unreliable communication.

Acknowledgments

The authors wish to thank Dan Bernstein for interesting discussions on the complexity of Dec-MDPs. This work was supported in part by the National Science Foundation under grants IIS-9907331, by the Air Force Office of Scientific Research under grant F49620-03-1-0090 and by NASA under grant NCC 2-1311. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of the NSF, AFOSR or NASA.

Appendix A. Policies of Communication - Myopic-greedy Approach

Tables 8, 9, and 10 present the complete policies of communication for agents acting in the Meeting under Uncertainty scenario (see Section 6.2). Each row corresponds to a different tested value for the transition probability of the process. Each column is a possible synchronized state given by the Manhattan distance between the agents moving in a 2D grid of size 10x10. Given a certain value for P_u and a certain distance, the entry in the table should be interpreted as the time when an agent should communicate its position. Each time that the agents reveal their actual distance, they can each check this table to figure out when is the next time to communicate. Time is reset to zero each time that the agents exchange information.

	d0=distance between agents when last synchronized, g located at $d0/2$																	
P_u	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0.2	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3
0.4	2	2	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3
0.6	2	2	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3
0.8	2	2	2	3	2	4	2	4	2	4	2	4	2	4	2	4	2	4

Table 8: Myopic-greedy policy of communication, where $C_\Sigma = -0.1$,$R_a = -1.0$.

	d0=distance between agents when last synchronized, g located at $d0/2$																	
P_u	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0.2	3	4	3	5	3	6	4	7	4	7	5	7	5	8	5	8	6	9
0.4	2	3	3	4	4	5	4	6	5	7	5	7	6	8	6	8	7	9
0.6	2	2	3	4	4	5	5	6	6	7	6	8	7	8	7	9	8	10
0.8	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10

Table 9: Myopic-greedy policy of communication, where $C_\Sigma = -1.0$,$R_a = -1.0$.

	d0=distance between agents when last synchronized, g located at $d0/2$																	
P_u	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0.2	9	9	11	13	14	17	18	20	21	23	25	27	28	30	32	34	35	37
0.4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0.6	4	4	5	6	6	7	8	9	9	10	11	12	12	13	14	15	15	16
0.8	3	3	4	4	5	5	6	7	7	8	8	9	10	10	11	11	12	12

Table 10: Myopic-greedy policy of communication, where $C_\Sigma = -10.0$,$R_a = -1.0$.

Appendix B. The Average Performance of the Myopic-greedy Approach

Tables 11 and 13 present the results obtained after running 1000 experiments when the cost of communication was zero, when sub-goals could be communicated and when the myopic-greedy approach was taken to compute the policy of communication. The analytical results computed following the formulas in Section 6.2 are presented when no communication was allowed (a meeting point was set in the middle of the grid at time 0). Tables 12 and 14 present the average number of communication acts performed in each one of these cases.

	Average Joint Utility				
P_u	No-Comm.	Ideal $C_\Sigma = 0$	Comm. SubGoals – Best p		Myopic-greedy
0.2	-104.925	**-62.872**	-65.906	0.3	-63.84
0.4	-51.4522	**-37.33**	-39.558	0.2	-37.774
0.6	-33.4955	**-26.444**	-27.996	0.2	-27.156
0.8	-24.3202	**-20.584**	-21.05	0.1	-21.3

Table 11: $C_\Sigma = -1.0$ in SubGoals and Myopic-greedy,$R_a = -1.0$.

	Average Communication Acts Performed			
P_u	No-Comm.	Ideal $C_\Sigma = 0$	Comm. SubGoals	Myopic-greedy
0.2	0	31.436	1.194	6.717
0.4	0	18.665	1	3.904
0.6	0	13.426	1	2.036
0.8	0	10.292	0	1.296

Table 12: $C_\Sigma = -1.0$ in Myopic-greedy and SubGoals,$R_a = -1.0$.

	Average Joint Utility				
P_u	No-Comm.	Ideal $C_\Sigma = 0$	Comm. SubGoals – Best p		Myopic-greedy
0.2	-104.925	**-62.872**	-69.286	0.1	-68.948
0.4	-51.4522	**-37.33**	-40.516	0.1	-40.594
0.6	-33.4955	**-26.444**	-28.192	0.1	-28.908
0.8	-24.3202	**-20.584**	-21.118	0.1	-22.166

Table 13: $C_\Sigma = -10.0$ in SubGoals and Myopic-greedy,$R_a = -1.0$.

P_u	Average Communication Acts Performed			
	No-Comm.	Ideal $C_\Sigma = 0$	Comm. SubGoals	Myopic-greedy
0.2	0	31.436	0	0.416
0.4	0	18.665	0	0.417
0.6	0	13.426	0	0.338
0.8	0	10.292	0	0.329

Table 14: $C_\Sigma = -10.0$ in Myopic-greedy and SubGoals,$R_a = -1.0$.

Appendix C. Myopic-greedy Policies of Communication with Deadlines

Table 15 presents the policy of communication computed following the myopic-greedy approach when the agents continue acting until they meet (no deadlines). Tables 16 and 17 show how this policy changes if different deadlines are added to the system with corresponding penalties for not having met by these deadlines.

P_u	d0=distance between agents when last synchronized, g located at $d0/2$																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0.2	9	9	11	13	14	17	18	20	21	23	25	27	28	30	32	34	35	37
0.4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0.6	4	4	5	6	6	7	8	9	9	10	11	12	12	13	14	15	15	16
0.8	3	3	4	4	5	5	6	7	7	8	8	9	10	10	11	11	12	12

Table 15: Myopic-greedy policy of communication, where $C_\Sigma = -10.0$,$R_a = -1.0$, No Deadline.

P_u	d0=distance between agents when last synchronized, g located at $d0/2$																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0.2	0	0	0	0	0	0	0	0	0	0	0	0	5	5	4	4	4	4
0.4	5	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.6	4	4	5	6	6	7	0	0	0	0	0	0	0	0	0	0	0	0
0.8	3	3	4	4	5	5	6	7	7	8	0	0	0	0	0	0	0	0

Table 16: Myopic-greedy policy of communication, where $C_\Sigma = -10.0$,$R_a = -1.0$, Deadline at T=8, Penalty=-100.0.

P_u	d0=distance between agents when last synchronized, g located at $d0/2$																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0.2	9	9	11	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.4	5	6	7	8	9	10	11	12	13	0	0	0	0	0	0	0	0	0
0.6	4	4	5	6	6	7	8	9	9	10	11	12	12	13	0	0	0	0
0.8	3	3	4	4	5	5	6	7	7	8	8	9	10	10	11	11	12	12

Table 17: Myopic-greedy policy of communication, where $C_\Sigma = -10.0$,$R_a = -1.0$, Deadline at T=15, Penalty=-100.0.

Appendix D. The Average Performance of the *OptCom* Algorithm

Tables 18 and 20 present the joint utilities attained by monotonic goal-oriented Dec-MDPs implemented in the Meeting under Uncertainty example when C_Σ took the values -1 and -10. We compare between the No-Communication case (where the meeting point is fixed at time 0 in the middle of the grid), the Ideal case with communication cost zero, the myopic-greedy case that punishes the agents if they did not meet by the finite-horizon, and the results obtained from running the *OptCom* algorithm (see Section 6.4). Tables 19 and 21 present the corresponding average number of communication acts in each case.

P	Average Joint Utility			
	No-Comm.	Ideal	Myopic-greedy	OptCom
0.2	-71.138	-62.55	-63.298	-62.776
0.4	-42.112	-37.292	-38.014	-37.622
0.6	-29.078	-26.716	-27.178	-26.692
0.8	-22.344	-20.57	-21.23	-20.622

Table 18: $C_\Sigma = -1.0$.

P	Average Communication Acts Performed			
	No-Comm.	Ideal	Myopic-greedy	OptCom
0.2	0	31.275	6.687	30.388
0.4	0	18.646	3.99	17.811
0.6	0	13.358	2.115	12.346
0.8	0	10.285	1.233	9.311

Table 19: $C_\Sigma = -1.0$.

P	Average Joint Utility			
	No-Comm.	Ideal	Myopic-greedy	OptCom
0.2	-71.138	-62.7	-69.516	-63.4
0.4	-42.112	-37.788	-40.994	-37.678
0.6	-29.078	-26.644	-28.974	-26.89
0.8	-22.344	-20.606	-22.09	-20.59

Table 20: $C_\Sigma = -10.0$.

P	Average Communication Acts Performed			
	No-Comm.	Ideal	Myopic-greedy	OptCom
0.2	0	31.35	0.444	28.957
0.4	0	18.894	0.428	16.032
0.6	0	13.322	0.33	11.474
0.8	0	10.303	0.301	9.2

Table 21: $C_\Sigma = -10.0$.

References

Becker, R., Zilberstein, S., Lesser, V., & Goldman, C. V. (2003). Transition-independent decentralized Markov decision processes. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pp. 41—48, Melbourne, Australia.

Bernstein, D., Givan, R., Immerman, N., & Zilberstein, S. (2002). The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, *27*(4), 819—840.

Boutilier, C. (1999). Sequential optimality and coordination in multiagent systems. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pp. 478–485, Stockholm, Sweden.

Boutilier, C., Dearden, R., & Goldszmidt, M. (1995). Exploiting structure in policy construction. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1104–1111, Montreal, Canada.

Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pp. 746–752, Madison, WI.

Dean, T., Kaelbling, L. P., Kirman, J., & Nicholson, A. (1995). Planning under time constraints in stochastic domains. *Artificial Intelligence*, *76*, 35–74.

Decker, K. S., & Lesser, V. R. (1992). Generalizing the partial global planning algorithm. *International Journal of Intelligent Cooperative Information Systems*, *1(2)*, 319–346.

Durfee, E. H. (1988). *Coordination of Distributed Problem Solvers*. Kluwer Academic Publishers, Boston.

Feng, Z., & Hansen., E. A. (2002). Symbolic heuristic search for factored Markov decision processes. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, Edmonton, Alberta, Canada.

Goldman, C. V., & Zilberstein, S. (2003). Optimizing information exchange in cooperative multi-agent systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pp. 137—144, Melbourne, Australia.

Grosz, B. J., & Kraus, S. (1996). Collaborative plans for complex group action. *Artificial Intelligence*, *86*(2), 269—357.

Guestrin, C., & Gordon, G. (2002). Distributed planning in hierarchical factored MDPs. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, Edmonton, Canada.

Guestrin, C., Koller, D., & Parr, R. (2001). Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems (NIPS-14)*, Vancouver, British Columbia.

Guestrin, C., Koller, D., Parr, R., & Venkataraman, S. (2003). Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, *19*, 399—468.

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*.

Kfir-Dahav, N., Monderer, D., & Tenenholtz, M. (2000). Mechanism design for resource-bounded agents. In *Proceedings of the Fourth International Conference on Multi-Agent Systems*.

Moore, J. (1992). Implementation, contracts, and renegotiation in environments with complete information. In Laffont, J.-J. (Ed.), *Advances in economic theory Sixth World Congress Volume 1*, pp. 182–282. Cambridge University Press.

Nisan, N., & Ronen, A. (1999). Algorithmic mechanism design. In *Proceedings of the Thirty First Annual ACM Symposium in Theory of Computing (STOC)*.

Osborne, M. J., & Rubinstein, A. (1994). *A Course in Game Theory*. MIT Press.

Papadimitriou, C. H., & Tsitsiklis, J. (1982). On the complexity of designing distributed protocols. *Information and Control*, *53*, 211–218.

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

Papadimitriou, C. H., & Tsitsiklis, J. (1987). The complexity of Markov decision processes. *Mathematics of Operations Research*, *12*(3), 441–450.

Peshkin, L., Kim, K.-E., Meuleau, N., & Kaelbling, L. P. (2000). Learning to cooperate via policy search. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI00)*, pp. 489–496, Stanford, CA.

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

Rabinovich, Z., Goldman, C. V., & Rosenschein, J. S. (2003). The complexity of multiagent systems: The price of silence. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pp. 1102—1103, Melbourne, Australia.

Roth, M., Vail, D., & Veloso, M. (2003). A world model for multi-robot teams with communication. In *Proceedings of IROS*.

Russell, S., & Wefald, E. (1991). Principles of metareasoning. *Artificial Intelligence*, *49*, 361–395.

Schneider, J., Wong, W.-K., Moore, A., & Riedmiller, M. (1999). Distributed value functions. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 371—378.

Smallwood, R. D., & Sondik, E. J. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, *21*(5), 1071—1088.

Wolpert, D. H., Wheeler, K. R., & Tumer, K. (1999). General principles of learning-based multi-agent systems. In *Proceedings of the Third International Conference on Autonomous Agents (Agents '99)*, pp. 77—83, Seattle, Washington.

Xuan, P., Lesser, V., & Zilberstein, S. (2001). Communication decisions in multi-agent cooperation: Model and experiments. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pp. 616–623, Montreal, Canada.