Learning to Communicate in Decentralized Systems

Martin Allen
Computer Science Department
University of Massachusetts
Amherst, MA 01003
mwallen@cs.umass.edu

Claudia V. Goldman
Caesarea Rothschild Institute
University of Haifa
Mount Carmel, Haifa, Israel 31905
clag@cri.haifa.ac.il

Shlomo Zilberstein
Computer Science Department
University of Massachusetts
Amherst, MA 01003
shlomo@cs.umass.edu

Abstract

Learning to communicate is an emerging challenge in AI research. It is known that agents interacting in decentralized, stochastic environments can benefit from exchanging information. Multiagent planning generally assumes that agents share a common means of communication; however, in building robust distributed systems it is important to address potential mis-coordination resulting from misinterpretation of messages exchanged. This paper lays foundations for studying this problem, examining its properties analytically and empirically in a decision-theoretic context. Solving the problem optimally is often intractable, but our approach enables agents using different languages to converge upon coordination over time.

Introduction

Learning to communicate in multi-agent systems is an emerging challenge for work in AI. Autonomous systems, developed separately, interact more and more often in contexts like distributed computing, information gathering over the internet, and wide-spread networks of machines using distinct protocols. In addition, systems may be called on to deal with new situations and information, as autonomy increases and environments grow more complex. As a result, we foresee the need for autonomous systems that can learn to communicate with one another in order to achieve cooperative goals. This raises a number of difficult questions, concerning such things as algorithms for learning to communicate effectively, and the properties of systems and environments that allow such learning to take place.

We make some first steps towards solving these problems. Coordination and communication while sharing resources has been extensively studied, in particular by the multi-agent systems community, as for example (Durfee 1988; Smith 1988; Pynadath & Tambe 2002; Peshkin *et al.* 2000; Mataric 1997). Such work on cooperative planning typically focuses upon maximizing global objectives, without deliberation about the *value* of communication. Often, systems resulting from such constrained attention feature no communication, where for instance agents follow predetermined social laws (Goldman & Rosenschein 1994). On the other hand are systems allowing free communication of well-understood messages, as in the SharedPlans (Grosz & Kraus 1996) and PGP (Lesser *et al.* 2004) frameworks, or the work of Balch (1994) and the KQML standard (Finin, Labrou, & Mayfield 1997). Evolution of common communication has also been studied from an ALife perspective, as for example MacLennan's work (1990) and adaptive language games (Steels & Vogt 1997). Recent work stays within the common-language paradigm, while analyzing tradeoffs between communication cost and the value of acquired information (Goldman & Zilberstein 2004; 2003; Nair *et al.* 2004). In contrast, the work here deals with cost-free communication in systems where communication is not completely shared.

Robust decentralized systems will often require agents to adapt their means of communicating in the face of new situations, or when mis-coordination arises. Such mis-coordination can be revealed in practice, or in simulation, and can serve as a signal to reinterpret messages received. In the context of this paper, agents attempt to learn correlations between languages with pre-existing simple semantics, distinguishing the approach from such as (Wang & Gasser 2002), in which agents collectively learn new shared concepts. The particular language learned will be directly related to system-utility, rather than to the cost of using that language (Gmytrasiewicz, Summers, & Gopal 2002). Further, agents learn to communicate while attempting to maximize some global objective, which need not be the specific one of learning to communicate itself, as opposed to work in which agents are specifically rewarded for pre-determined "correct" responses to messages (Yanco 1994).

We assume that agents already possess policies of action—mappings from states to possible actions—and leave the more general problem of learning how to act at the same time as learning to interpret messages for future work. Agents exchange messages, and interpret them by deciding upon a course of action after considering their contents. In our framework, agents presume that others involved in a shared cooperative task are communicating information relevant to that task. In this particular context, agents need to learn a mapping from the speaker's messages to their observations and actions, and then act based on an existing policy. The case of learning to communicate where agents begin with different possible contexts of communication, and need to determine when one particular context is inappropri-

ate, remains for future work.

The next section gives the formal framework of the problem, using decentralized Markov decision processes. Following that, we analyze some properties of the learning process, and make an attempt to characterize properties of stochastic environments that make learning to communicate possible in the first place. Lastly, we discuss empirical results based on an implementation of our ideas, and draw some conclusions.

The Decentralized Learning Framework

We study the learning problem in the context of *decentralized Markov Decision Processes* (Bernstein *et al.* 2002) with communication, involving a set of n agents, $\alpha_1, \ldots, \alpha_n$.

Definition 1 (Dec-MDP-Com). An n-agent *decentralized MDP with direct communication* (Goldman & Zilberstein 2004) is given by the tuple

$$M = \langle S,\, A,\, P,\, R,\, \Sigma,\, C_\Sigma, \Omega,\, O,\, T \rangle,$$

with elements as follows:

- S is a finite set of *states*, with initial state s^0.
- $A = \{A_i \mid A_i \text{ is a finite set of } \textit{actions } a_i \text{ for agent } \alpha_i\}$.
- P is a *transition function*, giving the probability $P(s' \mid s, a_1, \ldots, a_n)$ of moving from state s to state s', given actions $a_1, \ldots, a_n$.
- R is a *global reward function*, giving the system-wide reward $R(s, a_1, \ldots, a_n, s')$ when actions $a_1, \ldots, a_n$ cause the state-transition from s to s'.
- $\Sigma = \{\Sigma_i \mid \Sigma_i \text{ is a finite set of } \textit{messages } \sigma_i \text{ for } \alpha_i\}$.
- $C_\Sigma : \cup\Sigma_i \rightarrow \Re$ is a *cost function*, giving the cost of transmission for any message sent.
- $\Omega = \{\Omega_i \mid \Omega_i \text{ is a finite set of } \textit{observations } o_i \text{ for } \alpha_i\}$.
- O is an *observation function*, giving the probability $O(o_1, \ldots, o_n | s, a_1, \ldots, a_n, s')$ that each agent α_i observes o_i when actions $a_1, \ldots, a_n$ cause the state-transition from s to s'.
- T is the *time-horizon* (finite or infinite) of the problem.

We note an important constraint upon the observation function O, namely that a Dec-MDP-Com (unlike a Dec-*POMDP*-Com) is *jointly fully-observable*. That is, the combined observations of the agents fix the global state. In other words, there exists a mapping $J : (\Omega_1 \times \cdots \times \Omega_n) \rightarrow S$ such that if $O(o_1, \ldots, o_n \mid s, a_1, \ldots, a_n, s')$ is non-zero, then $J(o_1, \ldots, o_n) = s'$. This does not mean that each agent alone observes the entire state—the problem is genuinely decentralized—but that the observations of all agents taken together suffice to determine that state. Section of this paper contains an implemented example Dec-MDP-Com. For further analysis of the properties of Dec-MDPs, both with communication and without, see (Goldman & Zilberstein 2004).

Obviously, where agents use the same messages to mean the same thing, learning to communicate is not a problem. Rather, optimal linguistic action is a matter of deciding *what* and *when* to communicate, if at all, given its costs relative to the benefits information-sharing might bring. However, where agents utilize partially or completely different sets of messages, and do not fully understand one another, simple message-passing is not enough. Rather, agents need to learn how to respond to the messages that are passed between them—in a sense, learning what those messages mean.

While this sense of "meaning" is limited, it is not uninteresting. In natural-language contexts, of course, the meaning and structure of messages can be highly complex, and greatly varied—but an attempt to solve the problem of mechanical communication in natural language is well beyond the scope of our approach here. However, even where agents are limited to communicating about such things as their own basic actions and observations within the framework of a decentralized MDP, learning to correlate messages with appropriate responses can still be very challenging. Thus, when we discuss meanings here, we relate them strictly to the actions agents take in response to receiving messages, and otherwise leave them unanalyzed.

We represent the degree to which agent α_i understands agent α_j by a correspondence between messages sent by α_j, and those that α_i might itself send. As far as α_i is concerned, the meaning of some received message is a distribution over its own possible messages.

Definition 2 (Translation). Let Σ and Σ' be sets of messages. A *translation*, τ, between Σ and Σ' is a probability function over message-pairs: for any messages σ, σ', $\tau(\sigma, \sigma')$ is the probability that σ and σ' mean the same. $\tau^+_{\Sigma,\Sigma'}$ is the set of all translations between Σ and Σ'.

An agent translates between its own set of messages and another by establishing a probabilistic correlation between them. Each agent may in fact need to consider multiple possible translations between messages; that is, agents possess beliefs regarding which translation might in fact be the correct one to utilize in any given situation.

Definition 3 (Belief-state). Let agents α_1 and α_2 utilize message-sets Σ_1 and Σ_2, respectively. A *belief-state* for agent α_i is a a probability-function β_i over the set of translations $\tau^+_{\Sigma_i,\Sigma_j}$ ($i \neq j$). That is, for any translation τ between Σ_i and Σ_j, $\beta_i(\tau)$ is the probability that τ is correct.

That is, agents maintain beliefs in the form of probability distributions over translations, which are themselves probability distributions over message-pairs. Given any pair of messages, (σ_i, σ_j), an agent α_i assigns that pair a likelihood of having the same meaning, equal to the weighted sum of their assigned probabilities for each translation considered possible; we write $\beta_i^+(\sigma_i, \sigma_j)$ for that overall probability.

$$\beta_i^+(\sigma_i, \sigma_j) = \sum_{\tau \in \tau^+_{\Sigma_i,\Sigma_j}} \beta_i(\tau) \cdot \tau(\sigma_i, \sigma_j). \qquad (1)$$

In our model, learning to communicate is therefore the process of systematically updating belief-states with respect to translations. Agent α_i chooses an action, a_i, based upon its local observation, o_i, any messages received, and the current belief-state, β_i, about how to translate those messages. The choice of a_i, along with the actions chosen by other agents, leads to some state-transition, which in turn results

in some new observation, o_i'. This observation then leads to an update to a new belief-state, β_i', further affecting how later messages are translated, and thus influencing future actions. The procedure governing the update from belief-state β_i to β_i' comprises the agent's *language-model*: a function from actions, messages, and observations, to distributions over translations. Such models may be highly complex, and the prescribed updates can be difficult to compute correctly, especially where the languages are complicated, or the environment is only partially observable; our ongoing work considers a formal framework for learning in the latter case. Here we concentrate upon special—but interesting—cases for which it is much more straightforward to generate the probabilities in question.

Properties of the Problem

Optimal solution of Dec-MDPs without communication (or where communication is costly) is generally intractable (Bernstein *et al.* 2002; Goldman & Zilberstein 2004). Thus, we consider problems that are reducible to a more straightforward multiagent extension of the conventional MDP, first introduced by Boutilier (1999).

Definition 4 (MMDP). A *multiagent Markov decision process* is a 5-tuple:

$$M = \langle \overline{\alpha}, A^k, S, P, R \rangle,$$

with each element as follows:

1. $\overline{\alpha} = \{\alpha_1, \ldots, \alpha_k\}$ is a set of *agents*.
2. A^k is the combined set of *individual actions* for those agents; a *joint action* is a k-tuple $\langle a_1, \ldots, a_k \rangle \in A^k$ of actions, one for each agent.
3. $\mathcal{S}$ is a set of *states*.
4. P is a *state-action transition function*, returning the probability $P(s, \langle a_1, \ldots, a_k \rangle, s')$ that the system moves to state s', given joint action $\langle a_1, \ldots, a_k \rangle$ in state s.
5. $R : S \rightarrow \mathbb{R}$ is the *reward function*.

Simply put, an MMDP consists of a set of agents operating in a fully- and commonly-observed environment; transitions between states in that environment depend upon the *joint actions* of all of the agents, and a single, common reward is shared by the system as a whole. Given the common reward-function, and the fact that all agents can observe the full system state at any time, an MMDP with either a finite or infinite time-horizon can be solved as if it were a single-agent problem, since the value of a state at any point in time will obey the usual Bellman equations, solvable using standard iterative methods (Puterman 1994). The *optimal joint action*, $\langle a_1, \ldots, a_k \rangle^\star$, at any state s and time t is that which maximizes expected future value, and the *optimal joint policy*, $\pi^\star$, is a function taking each state-time pair to an optimal joint action for that point in the process.

However, while it is possible for individuals to calculate an optimal joint policy for such a process offline, *deriving* the optimal policy is not the same thing as *implementing* it. Unless agents can coordinate their individual actions in actual practice, there is no guarantee that they can always be sure of following a jointly optimal course of action (since in his model communication between agents is not allowed, or is unreliable). Boutilier makes this idea precise as follows:

Definition 5 (PIO actions). For agent α_i in state s of MMDP M at time t, action a_i is *potentially individually optimal* (PIO) iff there exists some joint action $a^\star = \langle a_1, \ldots, a_i, \ldots, a_k \rangle$ that is optimal for s at t.

If each agent in an MMDP has exactly one PIO action at each state-time pair, then implementing an optimal policy is straightforward, once calculated: each agent simply takes its sole optimal action, and the result is an optimal joint policy. However, where agents have multiple PIO actions at any point, coordination problems may arise, since not all combinations of PIO actions are themselves optimal.

Definition 6 (Coordination problem). MMDP M contains a *coordination problem* iff there exists state s, time t, and actions a_i, $1 \leq i \leq k$ such that each a_i is PIO, but joint action $\langle a_1, \ldots, a_k \rangle$ is not optimal.

An example arises, for instance, in a two-agent MMDP for which agent 1 has available actions a_1, b_1 and agent 2 has available actions a_2, b_2, and there exists some state-action pair for which the joint actions $\langle a_1, a_2 \rangle$ and $\langle b_1, b_2 \rangle$ are both optimal, but for which neither $\langle a_1, b_2 \rangle$ nor $\langle b_1, a_2 \rangle$ is optimal. In such a case, an individual agent may be able to calculate an optimal policy without necessarily being able to reliably implement it, unless there exists some means of ensuring that the requisite PIO actions "match up."

In his own work, Boutilier considers various mechanisms of coordination, and shows how optimal policies of action need to take those mechanisms into account. For instance, he considers free communication between agents over stochastically noisy channels, allowing them to share information regarding their intended courses of action, and points out how an effectively optimal policy will need to consider the probability of coordination following a noisy passage of messages. Other mechanisms are considered, and he shows how dynamic programming can be extended to calculate optimal policies that take into account the status of the coordination mechanism, at some cost in efficiency.

Unfortunately, the added burden of computing optimal policies for domains where coordination is difficult is only part of the story. In many interesting problems we must deal not only with agents who must coordinate their actions, but also with instances where each agent can only view its own local state-space, and the problem is genuinely decentralized in such a way as to make solving it generally intractable. As Goldman and Zilberstein (2004) point out, however, if agents can communicate among themselves *freely*, and hence at every time step, it is possible to reduce many such problems to the much easier one of an MMDP without coordination problems. If agents convey their own intended course of action (or can coordinate those actions in advance), and if their local views can be combined to form a global view of the system, communication of those local views and intended actions provides a means of ensuring that optimal multiagent policies can be enacted.

In some cases, however, there is an additional complication, namely that agents are not able to communicate in full, since they do not understand one another. We are interested

in understanding the general features of multiagent decision processes that allow agents to learn to communicate and so to optimize their joint actions. As a first step, we concentrate upon those decentralized problems that can be reduced to MMDPs. For such problems, it is possible to compute an optimal joint policy offline, under the assumption that agents *can in fact* communicate all necessary details. Agents can then, in certain circumstances, learn to communicate so that such an optimal policy can be implemented.

We presume that agents in a Dec-MDP-Com possess a noise-free channel of communication, and that there is an established protocol for sharing messages before actions are taken. While the issue of noisy communication is interesting, it is beyond the scope of what we can deal with here; instead, we concentrate upon the case where agents must learn to deal with messages that they clearly receive, but do not fully understand. Similarly, learning policies for when and what to communicate is also an interesting area, but not of concern here (for research on these latter two issues, see (Goldman & Zilberstein 2003; 2004)). Given these two presumptions then, we examine conditions allowing the reduction of a Dec-MDP-Com to a somewhat simpler problem.

Definition 7 (Fully-describable). A Dec-MDP-Com is *fully-describable* if and only if each agent α_i possesses a language Σ_i that is sufficient to communicate both: (*a*) any observation it makes, and (*b*) any action it takes.

Definition 8 (Freely-describable). A Dec-MDP-Com is *freely-describable* if and only if the cost of communicating any message σ is 0.

Claim 1. A Dec-MDP-Com is equivalent to an MMDP without coordination problems if (*a*) it is fully- and freely-describable; and (*b*) agents share a common language.

Proof: Straightforward. Since a Dec-MDP-Com is jointly fully-observable, the observations of each agent together determine the global state, and it is possible to calculate an optimal joint policy for each such state offline. Thus, agents that can freely and fully communicate their observations and intended actions in a shared language can also implement such a policy, without coordination problems. □

In what follows, we assume that each Dec-MDP-Com we deal with is freely- and fully-describable. In solving such problems, agents can *assume*, for the sake of calculating an optimal joint policy offline, that all agents do in fact share a common language, and that all information about observations and actions is shared. However, where agents must in fact learn to communicate, this assumption does not actually hold true, and so actual *implementation* of such policies requires more cooperation from the environment. Rather, the environment must provide enough in the way of observation and reward that agents can update their translations appropriately over time. In order to make this notion precise, we introduce some notation.

Notation 1. Let M be an n-agent Dec-MDP-Com. In some state s, at time t, suppose each agent α_j observes o_j and intends to take action a_j, communicating both facts to other agents by messages σ_j^o and σ_j^a. Then, for any agent α_i,

$$P_i^\sigma(o_j \mid \sigma_j^o,\, \beta_i^t) \tag{2}$$

is the probability, assigned by α_i, that α_j observes o_j, given message σ_j^o and α_i's current belief-state β_i^t. Similarly,

$$P_i^\sigma(a_j \mid \sigma_j^a,\, \beta_i^t) \tag{3}$$

is the probability that α_j will take action a_j, given message σ_j^a and α_i's current belief-state. Finally, we write $\max_i^\sigma(o_j)^t$ and $\max_i^\sigma(a_j)^t$ for the observation and action maximizing expressions (2) and (3), respectively (i.e., the observation and action that α_i considers *most likely* for α_j).

Notation 2. Let M be an n-agent Dec-MDP-Com. In some state s, at time t, suppose each agent α_j observes o_j and takes action a_j, causing a transition to state s', with observations $\langle o_1', \ldots, o_n' \rangle$, and reward $r' = R(s, a_1, \ldots, a_n, s')$ at time $t+1$. Then, for any agent α_i,

$$P_i^o(o_j \mid o_i',\, r')^{t+1} \tag{4}$$

is the probability, assigned by α_i, that agent α_j previously observed o_j, given that α_i now observes o_i' and the system receives reward r'. Similarly,

$$P_i^a(a_j \mid o_i',\, r')^{t+1} \tag{5}$$

is the probability that α_j took action a_j given α_i's current observation and the system-reward.

Given these notational conventions, we can now give sufficient conditions for a Dec-MDP-Com allowing each of its agents to learn the language of the others.

Definition 9 (Suitability). Let M be any fully- and freely-describable Dec-MDP-Com in which agents do not share a common language. In any state s at time t, let each agent α_i observe o_i and take action a_i, communicating both to other agents using messages σ_i^o and σ_i^a, and jointly causing a transition to state s'.

We say that M is *suitable* for learning to communicate iff, for any agents α_i and α_j, if $o_j \neq \max_i^\sigma(o_j)^t$, then for any time $t' \geq t$ at which α_j observes o_j (the same observation as at time t),

$$P_i^o(o_j \mid o_i'',\, r'')^{t'+1} > P_i^o(\max_i^\sigma(o_j)^t \mid o_i'',\, r'')^{t'+1}, \tag{6}$$

and similarly for $a_j \neq \max_i^\sigma(a_j)^t$,

$$P_i^a(a_j \mid o_i'',\, r'')^{t'+1} > P_i^a(\max_i^\sigma(a_j)^t \mid o_i'',\, r'')^{t'+1}. \tag{7}$$

That is, in a suitable Dec-MDP-Com, suppose agent α_j observes o_j and communicates that fact using message σ_j^o. However, agent α_i, based on its current belief-state β_i^t, incorrectly considers another observation $\max_i^\sigma(o_j)^t \neq o_j$ most likely for α_j. In such a case, at any later state (including the next one), α_i's resulting observation o_i'' and system-reward r'' "correct" the situation; that is, they are such that at any later stage of the process, whenever they are observed, α_i will consider the actual observation o_j more likely than the incorrect one thought most likely before. (And similarly for the action a_j taken by α_j.)

We stress that the given definition is but a first attempt to isolate conditions sufficient for agents to learn to communicate in a decentralized setting, and make no claim that such conditions are in fact necessary. However, while suitability as given is somewhat difficult to formalize precisely, we do not consider it to be an overly strong or artificial condition. For instance, domains in which agents have no idea what actions others are taking, but can positively eliminate candidates by observing their immediate effects, can be suitable with respect to those actions (given the proper conditions on communication): the evidence after any action is taken will eventually eliminate incorrect candidates, while increasing the probability of the correct action towards eventual certainty. Similarly, environments in which one agent observes some state variable a time step before another can be suitable with respect to observation, since the latter agent will eventually be given positive evidence allowing the determination of the correct observations. Section contains an example implementation of a relatively complicated, but still suitable, Dec-MDP-Com.

Our prior work (Goldman, Allen, & Zilberstein 2004) deals with some relatively simple examples of suitable problems, where agents do not need to communicate their actions, only their observations. In that domain, two agents work to meet at points in a grid-world environment, following a relatively simple procedure, with each acting in turn, according to the best estimate of the location of the other. Messages describing each agent's location are exchanged, and translations of those messages are updated after each step, depending upon whether or not the agents do in fact meet one another. Since agents are certain after checking some grid-location whether or not the other agent was in fact at there, the probability that the other observed that location is either 0 or 1, and the suitability of the Dec-MDP-Com follows immediately. We now give a more general version of this process, including actions.

Definition 10 (Elementary action protocol). Let s be a state of Dec-MDP-Com M, at time t, where agent α_i observes o_i. Each α_i follows the *elementary action protocol*:

(1) α_i communicates o_i to the others, using message σ_i^o.

(2) α_i calculates the *most likely observation sequence*,

$$o^\star = \langle \max_i^\sigma(o_1)^t, \ldots, o_i, \ldots, \max_i^\sigma(o_n)^t \rangle$$

and *most likely state*, $s^\star = J(o^\star)$. (Recall that J is the function from observations to global states, in accord with joint full observability; see p. 2.)

(3) Proceeding in turn, α_i chooses an action by:

(a) Calculating the *most likely action sub-sequence*,

$$a^\star = \langle \max_i^\sigma(a_1)^t, \ldots, \max_i^\sigma(a_{i-1})^t \rangle.$$

(b) Choosing action a_i such that some joint action,

$$a^+ = \langle a^\star, a_i, a_{i+1}, \ldots, a_n \rangle$$

maximizes value for likely state $s^\star$ at time t.

(c) Communicating a_i to the others by message σ_i^a.

(4) α_i takes action a_i after *all agents* complete step (3). (Agents choose actions based upon the *observations* of all others, but the *actions* of only those that precede them. The reader can confirm that this allows agents who already understand each other to coordinate optimally, avoiding the coordination problems Boutilier sketches. Agents who are still learning the language act in the way they believe *most likely* to be coordinated.)

(5) The state-transition from s to s' caused by joint action $\langle a_1, \ldots, a_n \rangle$ follows, generating new observation sequence $\langle o'_1, \ldots, o'_n \rangle$ and reward r' at time $t+1$. Agent α_i then updates its belief-state so that for any messages σ_j^o and σ_j^a received on the prior time step, and any possible observation o_j and action a_j, both:

$$P_i^\sigma(o_j \mid \sigma_j^o, \beta_i^{t+1}) = P_i^o(o_j \mid o'_i, r')^{t+1}. \quad (8)$$

$$P_i^\sigma(a_j \mid \sigma_j^a, \beta_i^{t+1}) = P_i^a(a_j \mid o'_i, r')^{t+1}. \quad (9)$$

That is, in the agent's new belief-state, the probability assigned an observation or action given the most recently received messages—i.e., the *meaning* of the messages—is identical to the probability that the other agent actually made that observation or took that action. It is assumed that the translation of all other messages from each other agent is adjusted only to account for normalization factors. Section briefly describes the use of a Bayesian Filtering algorithm to actually accomplish these sorts of updates in practice.

It is important to note that, for the general case of multi-agent coordination, such a straightforward procedure is not necessarily optimal, nor even necessarily close to optimal. As Boutilier points out, correct choice of action in the presence of unreliable or inaccurate communication must consider how each action may affect that communication, along with the other more immediate rewards to be had. Thus, in our case, it might sometimes be better for agents to choose their actions based not simply upon what they thought the most likely state might be, but also upon how certain expected outcomes would affect their translations for future instances of the problem, perhaps trading immediate reward for expected long-term information value.

As already discussed, however, computing optimal policies for legitimately decentralized problems is generally intractable, and so other methods and approximations for these cases are necessary. (Potential problems with attempts to solve such problems optimally, even where they can be treated as MMDPs that are not wholly decentralized, are discussed briefly at the end of Section .) Furthermore, where the problem is suitable, the elementary action protocol has the advantage that agents who follow it can eventually come to communicate clearly, and so act properly.

Claim 2. Given an infinite time-horizon, agents acting according to the elementary action protocol in a suitable Dec-MDP-Com will eventually converge upon a joint policy that is optimal for the states they encounter from then on.

Proof: The claim follows from suitability. As agents act, they choose actions based always on the observations and actions of others that they consider most likely. Since the problem is suitable, at any later time step the correct such

observations and actions will be more likely than any particular ones previously thought most likely. Furthermore, since updates of messages proceed directly in accord with these probability assignments, or as required for normalization, once a correct translation of any message is the most likely translation, it will remain so for all future time-steps. Thus, since the number of possible actions and observations for any agent is finite by definition, agents will, when given enough time, choose the correct entries, since these will be most probable. Agents will then implement a policy that is optimal from then on, since they are now acting based upon the actual states and next actions of the problem. □

Empirical Results

To explore the viability of our approach, we implemented our language-learning protocol for a reasonably complex Dec-MDP-Com. Each instance of the domain involves two (2) agents, each in control of a set of n pumps and m flow-valves in a factory setting, with parameters n and m varied to generate problem instances of different sizes. At each time step, each agent separately observes fluid entering the system from one of two different inflow ducts, along with the pumps and valves under its own control.

The task is then to maximize flow out of the system through one of several outflow ducts, subject to the constraint that the number of ducts be minimized. Accordingly, reward is directly proportional to outflow amount, minus the number of ducts used. Probabilistic effects arise because each of the pumps and valves is susceptible to variations in throughput, dependent upon whether the particular component was used to route flow in the prior time step. Any excess flow not routed through the system on a given time step is considered wasted, and is dropped from consideration.

Formally, we specify the problem as a Dec-MDP-Com:

$$M = \langle S,\, A,\, P,\, R,\, \Sigma,\, C_\Sigma, \Omega,\, O,\, T \rangle,$$

with elements as follows:

(*a*) S: the state-set is described by flow through the two inflow ducts, in_1 and in_2, two sets of pumps, $p_1^1, \ldots, p_n^1$ and $p_1^2, \ldots, p_n^2$, and two sets of valves, $v_1^1, \ldots, v_m^1$ and $v_1^2, \ldots, v_m^2$. Initially, all such flows are set to zero (0).

(*b*) A: at each time step each agent α_i chooses one action to control the pumps p_r^i (*on*, *off*, *forward*, *back*) or the valves v_s^i (*open*, *shut*).

(*c*) P: the transition function directs flow according to actions taken; however, pumps and valves fail to respond to commands probabilistically, based on whether or not they were used in the prior time step.

(*d*) R: the total reward is given by $(in/out) - d$, where in is the total units of inflow, out is the total units of outflow, and d is the number of outflow ducts used.

(*e*) Σ: each agent α_i possesses messages corresponding to each of its possible actions, identifying labels for every pump or valve in the system, as well as the observed units of inflow through duct in_i.

(*f*) C_Σ : the cost of all messages is zero (0).

(*g*) Ω: each agent α_i can observe its own inflow duct in_i, along with all pumps p_r^i and valves v_s^i that it controls.

(*h*) O: the observation-function takes any state of the system and returns the observable portions for each agent.

(*i*) T: the problem has an infinite time-horizon.

While the state-space of such a problem can be quite large, given the number of variables governing inflow and system settings, it is still efficiently solvable from a single-agent, centralized perspective. By taking the point of view of one agent observing all states globally, and acting in place of both agents simultaneously, the problem is solved offline, using typical dynamic-programming means.

Further, the environment is in fact an example of a suitable Dec-MDP-Com. The problem is both freely-describable, by the cost-function (*f*), and (for the purposes of solving the problem) fully-describable, as given by the set of messages (*e*). Furthermore, agents are aware of the overall structure of the pumping system, and can observe certain basic effects of each other's actions, by observing how many units of flow are routed through their own observable pumps and valves. These observations, combined with the total reward allow them to reason backwards to what those actions may have been, as well as to the total number of units of flow entering the system through the other agent's inflow duct. While certain actions may fail to have the desired effect, given pump or valve failure, actions never affect the wrong pump or valve; furthermore, no pump or valve fails permanently. Thus, the observed effect of any action taken by the other agent will either completely confirm which action was taken, or give the agent no evidence to update its translation of the last message. Taken together, these conditions ensure that incorrect interpretations are eventually eliminated in favor of correct translations. While this solution requires that agents know the overall structure of the domain, this is simply the same assumption required for usual optimal offline methods of solving such problems, and so we consider it no real defect in our method.

In line with the elementary action protocol, agents swap messages, choose actions based on their current beliefs, and act, repeating the process to converge towards mutual understanding and optimal action. Using their model of the environment, they update belief-states using a two-step Bayesian Filtering algorithm, first projecting possible belief-states before acting, then updating those belief-states given the results. This two-step simple update process is adapted from its applications in robotics (Thrun *et al.* 2001); our prior work (Goldman, Allen, & Zilberstein 2004) details the algorithm's use in communication in a simple grid-world setting; the interested reader is directed to that source for more on the specifics of the implementation. The experimental work in this paper expands upon that prior work, by including the language of actions, where before agents could only speak about state-observations, and by extending it to the more complicated domain presented here.

Agents interact until each learns the language of the other—achieved when each agent α_i achieves *certainty*, namely a belief-state β_i in which, for any message σ_j received from the other agent, there exists exactly one message

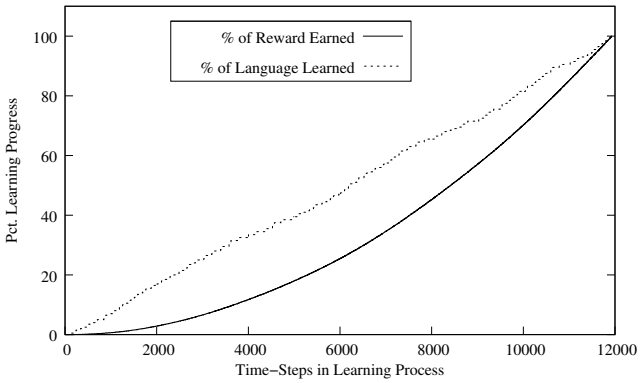

Figure 1: Reward accumulated as language is learned.

σ_i such that $\beta_i^+(\sigma_i, \sigma_j) = 1$. In this work, certainty provides a useful stopping condition, since the domain is one in which agents do in fact learn all of each other's messages in the course of optimizing action. We are now investigating cases in which complete certainty is not necessary, as when agents do not need to learn every part of another's language in order to achieve optimal performance, and convergence actually happens more quickly than where the entire set of messages is learned.

Our results show the elementary protocol to converge upon optimal policies in each of our problem-instances. Time of convergence depends upon the basic size of the problem, and thus the vocabulary of the agents necessary to describe all actions and observations, and also upon the frequency of certain rare states or actions. As conditions vary probabilistically, some states in the environment are encountered very infrequently, and agents do not have occasion to learn related terms in the other's language. By design, we insured that all states and actions are eventually encountered; current work also investigates cases where agents do not ever visit some parts of the state-space, and so whole parts of the language are unnecessary to optimal action.

The most interesting and suggestive results have to do with the rates at which agents accumulate reward, relative to how much of the language they have learned. Figure 1 gives one example, for a problem-instance featuring 100 vocabulary-items for each agent. The graph shows the percentage of total accumulated reward, and total shared vocabulary, at each time step in the process of learning and acting in the Dec-MDP-Com. In such a problem, agents converge upon a complete understanding of one another, and are able to act entirely optimally from then on, in approximately $12,000$ time steps, involving only a few minutes of computing time.

As can be seen, the language-learning process (top, dotted line) proceeds quite steadily. The rate of reward-accumulation, on the other hand, grows with time. Initially, language learning outpaces reward gain given that knowledge, as agents still find many actions and observations of others hard to determine. After about $2,900$ time steps, fully 25% of the language has been learned, but only just over 6% of the eventually accumulated reward. By the time 50% of the language has been learned, nearly $6,200$ steps in, things have improved somewhat, and some 27% of the reward has been earned. As time goes on, the rate of accumulation of reward actually increases to the point that it narrows the gap considerably, as agents now know much of what they need to communicate, and spend more time accumulating reward in already-familiar circumstances, without learning anything new about the language of the other agent. Essentially the same curves, although differing in their time of convergence, are exhibited by problem-instances of all sizes.

It is to be stressed that these results are first steps in the process of dealing with the problem of learning to communicate in decentralized settings. In particular, there are presently no ready candidates for comparison between different algorithms, since the communication problem is somewhat new. Our present work involves a detailed comparison between our method and the sort of offline optimal solution techniques proposed by (Boutilier 1999). It is our thought that a major obstacle to the application of these optimizing methods is the unavoidable blow-up in problem size. Essentially, such techniques would involve recreating the original problem in the form of an MDP with a state-space comprised of a cross-product of the original states with each possible belief-state. Since the number of latter such states will generally be exponential in the size of the descriptive language (and thus in the size of the original state-space), these methods will, it seems, often prove infeasible; thus, non-optimal methods may be necessary.

Conclusions and Extensions

We have presented learning to communicate in decentralized, multi-agent environments as a challenging problem for AI research. Our work makes some first steps toward a formal and systematic treatment of this problem. While solving decentralized MDPs optimally is generally intractable, we have shown that the presence of effective, free communication has the potential to make them much easier. Where communication is not initially possible, however, agents must learn to interpret one another before they can act effectively. Solving the problem of optimal action in the presence of language deficits is at least as hard as solving Dec-MDPs in general; in response, we show a relatively simple protocol that is non-optimal, but can allow agents to converge to optimal policies over time.

Analyzing the problem in the relatively familiar and rigorous context of MDPs, the definition of suitability presents a first attempt at identifying those characteristics of decentralized decision problems that allow some of them to be solved effectively. Our current work investigates such properties and when they arise in more detail. Further, our experimental results show the possibility of effective techniques allowing agents to learn to coordinate and communicate over time. We continue to investigate and compare other approaches, including analysis of the differences between possible optimal offline techniques and online learning methods. This opens the door for further study of the various parts of the problem. Of particular interest is the possibility for approximation, and we are currently interested in quantifying trade-

offs between the effort to learn another's language in full, and the marginal increases in utility such effort may bring.

Lastly, we note that the idea of "translations" has applications outside of the context of agents who begin from a blank slate with respect to one another's languages. The approach we use in our analysis and implementation does not require that agents begin with the presumption that they do not know what the other is saying. Rather, agents can begin from the position of partial, or even presumed total, understanding and proceed by checking and updating translations as they act, adjusting that understanding only as required by circumstance. Language *learning* can also therefore be a process of language *verification*. Artificial agents equipped with the ability to check whether their understanding of what is communicated to them matches up with the observed outcomes of their actions will be more resilient, able to catch errors in their specifications, and even adjust to them.

Acknowledgments

This work was supported in part by the National Science Foundation under grant number IIS-0219606 and by the Air Force Office of Scientific Research under grant number F49620-03-1-0090.

References

Balch, T., and Arkin, R. C. 1994. Communication in reactive multiagent robotic systems. *Autonomous Robots* 1:1—25.

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

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

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

Finin, T.; Labrou, Y.; and Mayfield, J. 1997. KQML as an agent communication language. In Bradshaw, J., ed., *Software Agents*. MIT Press.

Gmytrasiewicz, P. J.; Summers, M.; and Gopal, D. 2002. Toward automated evolution of agent communication languages. In *Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS-35)*.

Goldman, C. V., and Rosenschein, J. S. 1994. Emergent coordination through the use of cooperative state-changing rules. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 408–413.

Goldman, C. V., and Zilberstein, S. 2003. Optimizing information exchange in cooperative multi-agent systems. In *Proceedings of the Second International Conference on Autonomous Agents and Multiagent Systems*, 137–140.

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

Goldman, C. V.; Allen, M.; and Zilberstein, S. 2004. Decentralized language learning through acting. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 1006–1013.

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

Lesser, V.; Decker, K.; T.Wagner; Carver, N.; Garvey, A.; Horling, B.; Neiman, D.; Podorozhny, R.; Prasad, M. N.; Raja, A.; Vincent, R.; Xuan, P.; and Zhang, X. 2004. Evolution of the GPGP/TAEMS domain-independent coordination framework. *Autonomous Agents and Multi-Agent Systems* 9(1):87—143.

MacLennan, B. 1990. Evolution of communication in a population of simple machines. Technical Report CS90-99, University of Tennessee, Knoxville, Department of Computer Science.

Mataric, M. J. 1997. Learning social behaviors. *Robotics and Autonomous Systems* 20:191–204.

Nair, R.; Tambe, M.; Roth, M.; and Yokoo, M. 2004. Communication for improving policy computation in distributed POMDPs. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 1098–1105.

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

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: Wiley.

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

Smith, R. G. 1988. The contract net protocol: High level communication and control in a distributed problem solver. In Bond, A. H., and Gasser, L., eds., *Readings in Distributed Artificial Intelligence*. San Mateo, California: Morgan Kaufmann Publishers, Inc. 357–366.

Steels, L., and Vogt, P. 1997. Grounding adaptive language games in robotic agents. In *Proceedings of the Fourth European Conference on Artificial Life*.

Thrun, S.; Fox, D.; Burgard, W.; and Dellaert, F. 2001. Robust monte carlo localization for mobile robots. *Artificial Intelligence* 128:99–141.

Wang, J., and Gasser, L. 2002. Mutual online concept learning for multiple agents. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 362–369.

Yanco, H. A. 1994. Robot communication: Issues and implementation. Master's thesis, MIT.