Learning to Communicate in a Decentralized Environment*

U. of Massachusetts, Amherst, Computer Science Dept., Tech. Report UM-CS-2006-16

Claudia V. Goldman[1] **Martin Allen**[2] **Shlomo Zilberstein**[2]
[1]Caesarea Rothschild Institute
University of Haifa, Mount Carmel, Haifa, 31905 Israel
ph.: 972-4-8288355
[2]Department of Computer Science
University of Massachusetts Amherst, Amherst, MA 01003
ph. : 1-413-5451985
clag@cri.haifa.ac.il
{mwallen,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 miscoordination 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. We establish a formal framework for the problem, and identify a collection of necessary and sufficient properties for decision problems that allow agents to employ probabilistic updating schemes in order to learn how to interpret what others are communicating. Solving the problem optimally is often intractable, but our approach enables agents using different languages to converge upon coordination over time. Our experimental work establishes how these methods perform when applied to problems of varying complexity.

1 Introduction

Cooperative decentralized planning is the problem of computing a set of local behaviors for a group of agents that act in the same environment while maximizing a global objective. Each local policy maps the information known or believed by the agent to actions (i.e., domain actions and possibly communication acts as well). It has been shown [4] that in the worst case, solving such multiagent problems optimally is significantly more complex than is the case for single-agent sequential decision problems. One of the main sources of this difficulty is the fact that each individual decision-maker lacks global information when they compute their local behaviors. Allowing agents to share information may reduce uncertainty about this global information, for instance by reducing the number of possible belief-states that each agent needs to consider. In extreme cases, when communication is free and mutually understood, decentralized planning becomes equivalent to single-agent planning (e.g., see the MMDP model [6]). However, in

*This work was supported in part by the National Science Foundation under grants number IIS-0219606 and IIS-0535061 and by the Air Force Office of Scientific Research under grants number F49620-03-1-0090 and FA9550-05-1-0254.

practice, communication has some cost, be it the actual bandwidth used by a transmission, or some other function that quantifies the resources required for the information exchange. We have previously shown that computing policies involving costly communication can be as hard as computing optimal solutions without communication [25]. Consequently, although communication can indeed be helpful in simplifying coordination at execution time, computing when to communicate and what to communicate may still be a complex process.

Beyond overcoming the computational problem of decentralized planning under uncertainty, we are interested in *robust* decentralized systems. Such robustness will often require agents to adapt their means of communicating in the face of new situations, or when miscoordination arises. 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. As a result, systems may be called on to deal with new situations and information, as autonomy increases and environments grow more complex. Agents that act cooperatively may not necessarily share the same language of communication, and may not simply be able to exchange a translation, mapping discrepancies in the languages, in an off-line manner. Such problems may occur, for instance, when the content of an agent's communication arises from observations available solely to that agent, in the midst of some shared task. Even agents with the same sensing apparatus may still lack the contextual information necessary to correctly interpret each other's messages. Alternatively, the ability to learn new meanings can guard against unintentional design-time errors. In many applications, the misinterpretation of messages can lead to miscoordination and an eventual decrease in performance. NASA's Climate Orbiter probe, for instance, crashed as a result of an unwitting use of different (metric and imperial) conventions of measure in calculations communicated between different design teams, causing the spaceship to follow an incorrect flight plan [31]. Such considerations also arise where users of a system may have different levels of competence—as in automated and interactive tutoring—or where it is practically infeasible to specify all necessary communication protocols at design time. The latter problem arises, for instance, in automated control and diagnosis. In such contexts, the range of ways a particular mechanism may go wrong cannot generally be known in advance, and encountered problems often require novel diagnoses and solutions [8]. Such problems are compounded when various mechanisms are combined as parts of a larger overall process, as is common in manufacturing plants.

The ability to communicate is thus a double-edged sword. While it has the potential to make multiagent coordination problems much easier to solve, the possibility of miscommunication opens up difficulties of its own. In this paper, then, we focus on the problem of how agents may learn how to interpret the messages they exchange, while acting together in an uncertain and decentralized environment. We isolate this problem from the problem of computing optimal policies with costly communication and concentrate on: (1) algorithms that update the understanding of messages exchanged while improving the value of the joint policies of behavior; and (2) the properties of systems and environments that allow such learning to take place.

The standard of success by which we will judge the particular process of learning to communicate will be directly related to the system-wide measure of utility, rather than to the individual cost of using that language [23]. In this context, agents attempt to learn correlations between languages with pre-existing simple semantics, distinguishing the approach from such as [43], in which agents collectively learn new shared concepts for the purpose of learning per se and not in the framework of a planning problem. Furthermore, 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 [45]. Finally, our work on cases where miscoordination arises is to be distinguished from such as that in verification systems [39], where the

aim is to identify inconsistencies between a software specification and its execution code, which can then be eventually fixed manually. Our purpose, on the other hand, is to explore methods by which agents automatically learn to correct a misinterpretation in addition to identifying it (on-line or by simulation), in the framework of decentralized control.

We provide a general practical and formal framework for studying the problem of learning to communicate, and isolate sufficient and necessary conditions under which agents can maximize their joint expected utility by successfully solving that problem. We show this in particular for agents that communicate their observations and actions; in the process, we shed light on the difficulties involved with learning languages in general. One of the contributions of this work is in understanding how hard this problem can be. The basic framework is decision-theoretic; agents operate with probabilistic models of the meaning of languages of communication, and base decisions on the given probabilities. The problem of determining message meanings may then become unsolvable for languages in general—since it may be impossible to generate a meaningful probabilistic model of a sufficiently rich language—and can be quite complex even for relatively simple languages. Another contribution is to provide an algorithm that, under the identified conditions, converges to some mutual understanding of a language which is not initially shared by the agents. We show how such convergence eventually leads to a situation in which agents can then maximize the value of their joint policy.

Sections 2 and 3 present the general language-learning process and show how the problem can be framed as a process of updating belief-states in the context of a decentralized Markov decision process. We explain this process further in Section 4. Important properties of decision problems and necessary and sufficient conditions for learning to communicate while acting are explained in Sections 5 and 6. These are followed by empirical evidence about the possibilities of the approach, presented for scenarios with increasing complexity, in Section 7. Finally, Sections 8 and 9 provide an overview of other work done on communication in multiagent systems and give our conclusions.

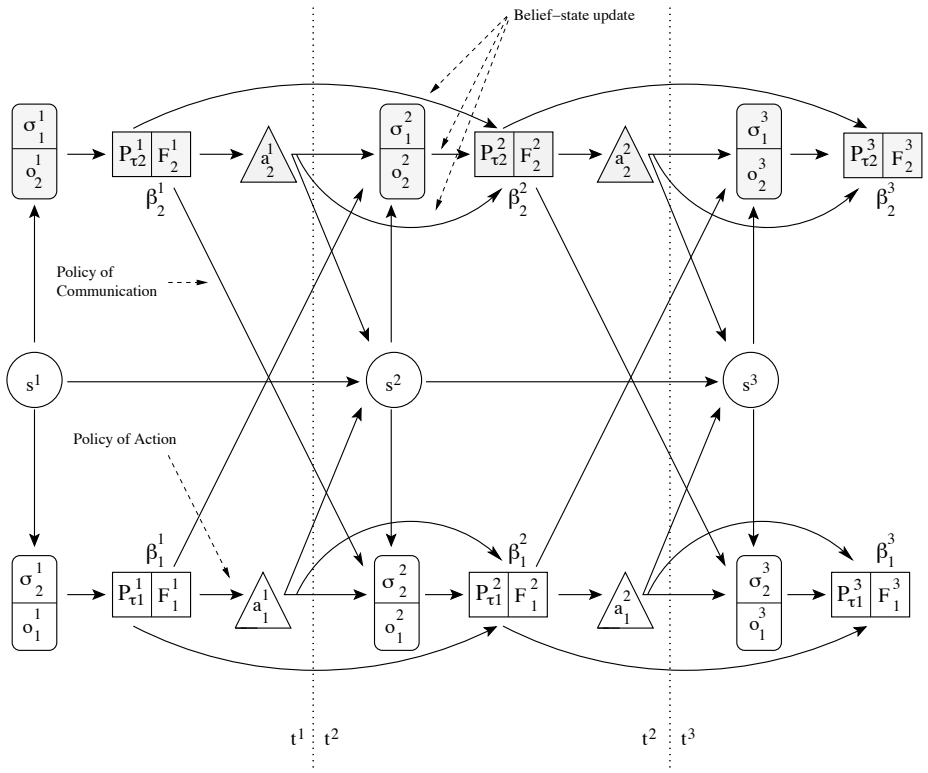
2 The general language-learning process

Figure 1 gives a graphical overview of the interrelations between the various components of the language-learning process as it occurs in the case of two inter-communicating agents. These elementary parts, which will be discussed in more detail later on, are as follows:

- *State*: The global state of the environment at any given time.
- *Observation:* How the global state actually appears to an individual agent.
- *Message*: The communications exchanged between agents.
- *Belief-state*: An agent's belief about the current state of affairs, consisting of:
  - *Translation*: An understanding of another agent's language, and of the most recent message in particular.
  - *Belief-features*: All other beliefs about the current state of the environment.

Such a process involves many elements that are familiar from the specification of problems involving agents acting under uncertainty. In particular, we will want to specify two important components:

State-action transitions: Given any global state s^i at some time t^i, and the actions taken by each agent, the environment will transition to some next state s^j.



s^i	State i	t^j	Time-step j
σ_i^j	Message j, sent by Agent i	o_i^j	Observation j of Agent i
a_i^j	Action j of Agent i	β_i^j	Belief-state j of Agent i
$P_{\tau i}^j$	Translation-distribution j of Agent i	F_i^j	Belief-features j of Agent i

Figure 1: A sketch of the language-learning process.

Observation functions: For each agent α, at any global state s^j, that agent makes some observation o^j.

Furthermore, an agent in such a situation will possess, or learn a behavior which includes the following three components:

Policy of action: Given a belief-state, the agent takes some action.

Policy of communication: Given a belief-state, the agent communicates something (maybe nothing) to some other agent.

Belief-state update: Given a belief-state, action taken, observation, and message received, the agent generates some new belief-state.

Language learning thus takes place within a general framework of action and communication between agents. While the process of updating belief-states involving the interpretation of what other agents say has certain special features arising from the fact that language is involved, it is a special case of a general problem, to do with finding successful policies that are based on one's beliefs. The main distinguishing feature of this work is the notion of a *translation*. While the usual approach to many problems of action and communication assumes that the language of communication is shared, we make no such presumption. Instead, we include an agent's

possibly imperfect understanding of another's language as part of the belief-state. 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. (Looking ahead a bit to the formal definitions found in Section 3, as far as α_i is concerned, the meaning of some received message forms a distribution over its own possible messages.)

Formally, we capture this model as a decentralized Markov decision process with direct communication (originally formulated in [25]). For simplicity of exposition we focus here on the case of two agents; the model can easily be extended to n agents.

Definition 1 (Dec-MDP-Com). *A decentralized MDP with direct communication is given by the tuple* $M = < S, A_1, A_2, P, R, \Sigma, C_\Sigma, \Omega_1, \Omega_2, O, T >$ *where:*

- S *is a finite set of world states with a distinguished initial state* s^0.
- A_1 *and* A_2 *are finite sets of control actions.* a_i *denotes an 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.* $R(s, a_1, \sigma_1, a_2, \sigma_2, s')$ *represents the reward obtained by the system as a whole, when agent* 1 *executes action* a_1 *and sends message* σ_1 *and agent* 2 *executes action* a_2 *and sends message* σ_2 *in state* s *resulting in a transition to state* s'.
- Σ *denotes the alphabet of messages and* $\sigma_i \in \Sigma$ *represents an atomic message sent by agent* i *(i.e.,* σ_i *is a letter in the language). When this language is not mutually shared by all the agents, we will denote by* Σ_i *agent* i*'s language of communication. A formal definition of a language appears in Definition 6.*
- C_Σ *is the cost of transmitting an atomic message:* $C_\Sigma : \Sigma \rightarrow \Re$. *The cost of transmitting a null message is zero.*
- 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'. *A Dec-MDP is jointly fully observable, i.e., 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* $J(o_1, o_2) = s'$.
- T *is the time horizon of the problem (finite or infinite).*

The framework we will propose is based on several important observations. First, learning to communicate occurs while the agents are involved in acting towards the maximization of some objective function. Second, agents share some context, in terms of their presumptions about the possible content of shared messages. While, this may not be a single context, agents still consider only a relatively small set of plausible candidates. Third, agents are assumed to base their understanding of one another upon probabilistic models of the relationship between language and the environment. A more philosophical question can be asked about the learning to communicate problem when such model is not accessible, but this is beyond the scope of this paper. Finally, we observe that obscuring the content of messages in the state features is not desirable. Solving the problem of learning to communicate will then become intractable given the complexity results and algorithms known to date to solve general Dec-POMDPs [4, 25]. Thus, we consider the use of communication separately from domain actions to allow us to

study techniques and features specific to language itself. We believe that this focus can enrich our understanding of the interactions and the capabilities of autonomous agents.

In a Dec-MDP-Com with a mutually shared communication language Σ, the local behaviors of the agents are given by local policies of action and communication, based on sequences of observations and messages as follows:

Definition 2 (Local Policy of Action with a Shared Language).

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

Definition 3 (Local Policy of Communication with a Shared Language).

$$\delta_i^\Sigma : \Omega^* \times \Sigma^* \rightarrow \Sigma.$$

Note that we assume here and throughout the paper that the communication-policy δ^Σ for an agent α returns some message given every possible observation; that is, α communicates something after every observation. We may wish that agents do not always communicate; this can be handled by either adding some special "null" message to the language Σ, or perhaps by re-defining δ^Σ as only a partial function on Ω.

In order to solve a decentralized process with a mutually shared communication language optimally, we evaluate the possible local policies of behavior and find one optimizing the expected value of the joint policy. This expected value is computed as the expected reward obtained while following the policies, equal to the summed reward for each state-action transition, weighted by the probabilities of those transitions. In previous research, we have computed this value for the case when communication lead to joint full observability [25]. In that case, whenever the agents exchange information, they both observe the global state. Thus, the local policies of action of these agents include the last synchronized state; that is: $\delta_i^A : S \times \Omega^* \times \Sigma^* \rightarrow A_i$. Here, we give the more general form of the relevant equations. The interactions among the agents are described as a process in which agents perform an action, then observe the environment, and then send a message that is instantaneously received by the other agent.

To compute the expected reward attained by a joint policy, we first define the probability of transitioning over a sequence of states. This is given by the one-step transition probability of reaching a destination state s' from any origin state q (which is part of the model), multiplied by the cumulative transition probability for any possible sequence suffix that leads to this state q, and weighted by the probability of sensing the given observation sequence. Formally:

Definition 4 (Transition Probability Over a Sequence of States). *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:*

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

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

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

Then, the value of the initial state s^0 of the Dec-POMDP-Com after following a joint policy δ for T steps can be defined as the expected reward attained over all possible sequences of states visited by δ starting in s^0:

Definition 5 (Value of an Initial State Given a Policy). *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:*

$$V_\delta^T(s^0) = \sum_{(\overline{\sigma_1}o_1,\overline{\sigma_2}o_2)} \sum_{q\in S} \sum_{s'\in S} \overline{P_\delta}(q|s^0,\overline{\sigma_1},\overline{\sigma_2},\overline{\sigma_2},\overline{\sigma_1}) * P(s'|q,\delta_1^A(\overline{\sigma_1},\overline{\sigma_2}),\delta_2^A(\overline{\sigma_2},\overline{\sigma_1}))* \\ R(q,\delta_1^A(\overline{\sigma_1},\overline{\sigma_2}),\delta_1^\Sigma(\overline{\sigma_1}o_1,\overline{\sigma_2}),\delta_2^A(\overline{\sigma_2},\overline{\sigma_1}),\delta_2^\Sigma(\overline{\sigma_2}o_2,\overline{\sigma_1}),s')$$

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.

Decentralized control with direct communication in a shared language is therefore the problem of finding an optimal joint policy δ^* for action and for communication such that $\delta^* = \arg\max_\delta V_\delta^T(s^0)$. We stress again that while it is straightforward to write the definition of policy-value as above, actually calculating the value of joint policies, and thus of finding the optimal policy, is generally very difficult.

In the problems studied in this paper, on the other hand, agents do not share a common language of communication. Following Figure 1, we notice that the local behaviors of the agents are mappings from belief-states to either domain actions or messages. That is, we wish to define the local policies of action and communication as functions from belief-state sequences, $\delta_i^A : \beta_i^* \to A_i$ and $\delta_i^\Sigma : \beta_i^* \to \Sigma_i$, respectively. These belief-states are composed of both translations of messages received and beliefs about the features of the states based on observations and message sequences. Section 4 discusses these policies and their value in full detail. Here, we simply note that our work is based on the idea that that agents learn to communicate on-line *while* acting towards the maximization of some global objective. We assume that each agent already has been assigned some policy of behavior that includes a local policy of action and a local policy of communication. For every possible belief-state of its own, an agent will know how to act and what message from its own language to send. All they lack, then, is a way to correctly interpret the messages received. Learning to act, to communicate and to interpret messages all at once on-line is beyond the scope of this paper. Here, we are interested in the process of updating these belief-states, resulting in updated interpretation of messages and consequently better coordination between agents.

3 Formal Definitions

The elementary items comprising the theory of language learning are *agents* taking *actions* and making *observations* in some environment, along with: (i) *languages* of communication between different agents; (ii) *translations* governing how messages in different languages are interpreted; (iii) *beliefs* about the environment, governing which messages are sent and which actions are taken; and (iv) *contexts* which govern how the interpretations of messages are generated and changed in response to feedback from the environment.

Definition 6 (Language). *A* language *is a pair $\Sigma = (W, \int)$, where:*

1. *$W = \{\sigma_o, \sigma_1, \ldots, \sigma_n\}$ is a set of primitive* words*, the smallest atomic units of which any message may be comprised.*

2. *$\int$ is a* syntax-function*, defined as follows:*

   - *$W^n = \underbrace{W \times \cdots \times W}_{n}$ is the set of all possible n-word messages; let $W^+ = \bigcup_{n\in\mathbb{N}} W^n$.*

- $\int : W^+ \rightarrow \{1, 0\}$.

A message $\sigma^+ \in W^+$ is *legal for language* Σ (written $\sigma^+ \in \Sigma$) iff $\Sigma = (W, \int)$ and $\int(\sigma^+) = 1$. For the sake of convenience we will have occasion to use the name of some language Σ to stand for the set of legal messages $\{\sigma^+ \in \Sigma\}$; such uses shall be clear from context.

As given, the definition of a language does not specify any particular structure for a language; in general, the syntax-function can be essentially arbitrary. For most purposes, of course, we would define such a syntax-function in some structured (perhaps compositional) way, so that the set of legal messages in our language was similarly structured.

Previous results [25] proved that agents controlling a Dec-MDP-Com benefit most by exchanging their own last observations when the cost of communication is constant over all messages. For more general cases, it may be worthwhile for the agent to send information about the actions it has performed. Since individual observations and actions are the components of a decentralized MDP that are only known locally, they are natural candidates to be exchanged as messages between the agents. For example, agents acting in a 2D grid may be modeled as observing their own local coordinates. A language of communication can then be defined as the set of pairs of possible coordinates: the language $\Sigma_{xy} = (\mathbb{N}, \int)$ of coordinates in a 2D grid of size $(MaxX \times MaxY)$ is given by the set of messages (x, y) where $x, y \in \mathbb{N}$ and $\int(x, y) = 1$ if and only if both $(0 \leq x \leq MaxX)$ and $(0 \leq y \leq MaxY)$. Of course, other languages could have yet more complex syntaxes, as for example messages that include the actions taken as well. So far, there has been no study of Dec-MDPs featuring communication of messages that are different from observations. Our work here considers just such cases.

Note that we do not specify a semantics as part of the definition of a language. In fact, no detailed treatment of the meaning or truth-conditions of messages is given. Rather, the semantics of any language Σ arises implicitly from elementary practices of actual communication. Agents communicate particular messages, based simply on their own observations, actions and resulting beliefs, according to the agent's local policy of communication, which can be presumed to depend upon the global goals. "Meaning," then, is simply a correlation between particular messages and the beliefs that cause their originator to send them; this is discussed in further detail below. Similarly, an agent establishes a *translation* between its own language and another by correlating the sets of messages in each (and so, indirectly, between messages in the other language and its own beliefs about the environment).

Definition 7 (Translation). *Let Σ and Σ' be two sets of messages. A* translation*, τ, between Σ and Σ' is a probability function over message pairs, i.e., for any messages σ, σ', $\tau(\sigma, \sigma')$ is the probability that σ and σ' have the same meaning. $\tau^+_{\Sigma,\Sigma'}$ is the set of all translations between Σ and Σ'.*

An agent translates between its own language and (some subset of) another by establishing a probabilistic correlation between the meanings of the messages in each.[1] Here we point out that "means the same thing" is given sense simply in terms of the agents' beliefs. For agent α_1 to say that message $\sigma_2 \in \Sigma_2$, sent by agent α_2, is likely to mean the same as message $\sigma_1 \in \Sigma_1$, is simply to say that it is likely that the situations in which α_2's beliefs cause it so send message σ_2 are the same as those in which α_1's beliefs would cause it to send σ_1.

For example, assume that the messages $\sigma_1 \in \Sigma_1$ and $\sigma_2 \in \Sigma_2$ name locations in a fully observable 2D grid, and that agent α_1 always sends out the name of its current location x, whenever it observes that it is in fact at x. Further, suppose that we have a completely specified

[1] In the most general case, the uncertainty of interpreting a message correctly can be captured by some general feasibility measure. In this paper, we focus on probabilities as an example of such measure. Other measures are beyond the scope of this paper.

translation τ between Σ_1 and Σ_2. Then, the probability $\tau(\sigma_1, \sigma_2)$ is simply the probability that the belief that causes α_2 to send σ_2 arises exactly when α_2 is in the same location that α_1 names using σ_1. Furthermore, in such a situation, any functional output $\tau(\sigma_1, \sigma_2) = 1$ will mean that α_1 is absolutely certain that σ_2 is to be translated as σ_1. This discussion can be made more precise by defining the notion of a belief-state.

Definition 8 (Belief-state). *For any agent α with language Σ in some state-space S, a belief-state β is a pair (P_τ, F), where:*

1. *P_τ is a probability distribution over translations.*
2. *F is a probability distribution over state-set S. That is, for any $s \in S$, $F(s)$ is the probability that the agent is in state s.*

An agent's belief-state thus consists of a probability distribution over states—a familiar notion from the literature on partially observable Markov decision processes (POMDPs)—along with a probability distribution over the set of translations between the agent's own language and the language of another with which the agent communicates. In the case that there exist more than one such other agent, the belief-state will require multiple such translation-distributions; in this work, we will treat only of the two-agent case, and so only provide a single such distribution. Furthermore, since each translation τ^* is in itself a probability distribution over message pairs, $(\sigma, \sigma') \in (\Sigma \times \Sigma')$, we can write $P_\tau(\sigma, \sigma')$ for the overall probability that some particular pair are equivalent, defined as:[2]

$$P_\tau(\sigma, \sigma') = \sum_{\tau^* \in \tau^+_{\Sigma,\Sigma'}} P_\tau(\tau^*)\tau^*(\sigma, \sigma'). \tag{1}$$

Thus, each agent may possess any number of distinct translations between its own language and some other, each assigned its own probability. The process of learning to communicate is thus the process of adjusting these sets of translations, replacing them with others that differ in the probabilities they assign to individual message pairs, or with respect to their own individual probabilities. In particular, we will assume that this process is governed by the observed outcomes of actions taken by the translator. That is, an agent α learns to communicate in some language Σ' as a result of taking actions based (at least partially) on messages received in that language, and then adjusting its current set of possible translations of Σ'.

In general, as given in Definition 7, a translation between language Σ and some other language Σ' involves a complete function over Σ. That is, the translation assigns values to translations for *every possible* message $\sigma \in \Sigma$. In real situations, of course, agents will not generally consider literally every possible meaning of a message σ received in some unknown language, for if they did, the process of interpretation and translation would never get off the ground. Rather, agents tend to restrict their attention to certain specific subsets of plausible meaning for the utterances of others, where such restricted subsets are given by contextual considerations of such things as relevance and expected intent.

Effectively, then, a context allows an agent to bound the possible interpretations of other agents, or to pre-assign translations to specific parts of the language, as for example in cases where two agents already share part, but not all, of their two languages. The models and the bounds involved in contexts of translation can also correspond to such things as assumptions

[2] As given in Definition 7, a translation $\tau^* \in \tau^+_{\Sigma,\Sigma'}$ is total over an agent's own language Σ, but may be partial over the other language, Σ', especially as all the particular contents of that second language may not yet be known. In general, if some incomplete translation τ^* is such that the particular pair (σ, σ') is not part of its domain, then we set $\tau^*(\sigma, \sigma') = 0$ for the purposes of the summation given in equation (1).

about the syntax of messages in the target language, or about the grammatical role of particular parts of certain messages. Such models, and their restrictions, are what makes language learning possible. Without some available context for translation—some model of the relationships between messages, actions, and outcomes—an agent α will be simply unable even to begin interpreting the language of another.

Our notion of context is thus in the same vein as the *local models semantics/MultiContext Systems* (LMS/MCS) model reviewed in [37]. This approach assumes that an agent has a local theory which contains the knowledge it needs to solve a problem. An agent can switch between contexts when it reveals that the current context is not adequate. Ghidini and Giunchiglia [20] argued that two principles govern contextual reasoning: the principle of locality (i.e., reasoning always happens in a context) and the principle of compatibility (i.e., reasoning processes can interact in different contexts). Each agent in our work associates a context-dependent vocabulary to the context in question. We assume that agents share the same context, although each agent may have a different vocabulary associated with it. This paper studies algorithms that map one vocabulary to the other so that agents can correctly interpret messages sent in the relevant context.

4 On Belief-State Updates

The problems studied in this paper involve agents that do not share one language of communication, and base their actions in part upon how they translate one another's messages. In our framework, an agent translates between its own set of messages and another by establishing a probabilistic correlation between them. Furthermore, in most cases agents need to consider multiple possible translations between messages; that is, agents possess beliefs regarding which translation to utilize in any given situation, and update those beliefs over time based on their experience. We now consider this update process, both in general and under some special simplifying assumptions. First, we discuss the way that the general process of updating translations, based on sequences of past observations, messages, and actions, interacts with policies of action and their value. Next, we look at the simplified special case, where translation updates are based solely upon information from immediately prior time-steps.

4.1 Belief Updates in General

As already mentioned, the local behaviors of agents are mappings from sequences of belief-states to either domain actions or messages, $\delta_i^A : \beta_i^* \rightarrow A_i$ and $\delta_i^\Sigma : \beta_i^* \rightarrow \Sigma_i$. As given by Definition 8, the belief-state $\beta_i = (P_{\tau i}, F_i)$ is composed of probability distributions over translations and system states, respectively. In particular, we are interested in updates of the translation portion, $P_{\tau i}$, of the belief-state (the portion F_i is itself dependent upon the current translation-distribution, along with the observation and message sequences), since the current translation of the foregoing sequence of messages will have direct effect upon an agent's current policy. That is, letting $P_{\tau i}^+$ be the space of all possible distributions over translations between languages Σ_i and Σ_j, we can write the local policies of each agent α_i in a manner analogous to those involving shared languages (Definitions 2 and 3).

Definition 9 (Local Policy of Action with Different Languages).

$$\delta_i^A : \Omega_i^* \times \Sigma_j^* \times P_{\tau i}^+ \rightarrow A_i.$$

Definition 10 (Local Policy of Communication with Different Languages).

$$\delta_i^\Sigma : \Omega_i^* \times \Sigma_j^* \times P_{\tau i}^+ \rightarrow \Sigma_i.$$

That is, the policies for each agent are functions (to either actions or messages) from observation and message sequences, along with translation-distributions.

We want to define transition probabilities over state sequences given such a policy, and define its expected value, as in the case of Dec-MDP-Coms with shared languages (Definitions 4 and 5). This task is complicated by the need to factor in updates to the translation-distributions. In general, agents will update their beliefs about translations (the distribution $P^+_{\tau_i}$) based on sequences of observations, messages, and actions.

Definition 11 (Translation-Update Function). *Translation updates are functions:*

$$U_{\tau i} : \Omega_i^* \times \Sigma_j^* \times A_i^* \times P^+_{\tau i} \to P^+_{\tau i}.$$

By convention, if any of these sequences is empty, then $U_{\tau i}$ returns some designated distribution, $\hat{P}_{\tau i} \in P^+_{\tau i}$, the default distribution*; that is:*

$$U_{\tau i}(\epsilon, \overline{\sigma_j}, \overline{a_i}, \centerdot) = U_{\tau i}(\overline{o_i}, \epsilon, \overline{a_i}, \centerdot) = U_{\tau i}(\overline{o_i}, \overline{\sigma_j}, \epsilon, \centerdot) = \hat{P}_{\tau i}.$$

Over time, then, these updates influence actions taken, and the outcomes of those actions in turn influence further updates. Relative to a given local action policy, which generates the action sequences, we define the update function based on observation or message sequences alone.

Definition 12 (Translation Updates for an Action Policy δ_i^A). *For a given local policy of action δ_i^A, and update function $U_{\tau i}$, we define the* translation-updates for δ_i^A*, written $\overline{U^{\delta_i^A}_{\tau i}}$, recursively on observation and message sequences as follows:*

$$\overline{U^{\delta_i^A}_{\tau i}}(\epsilon, \epsilon) = \overline{U^{\delta_i^A}_{\tau i}}(\overline{o_i}, \epsilon) = U^{\delta_i^A}_{\tau i}(\epsilon, \overline{\sigma_j}) = U_{\tau i}(\epsilon, \epsilon, \epsilon, \centerdot) = \hat{P}_{\tau i}. \tag{2}$$

$$\overline{U^{\delta_i^A}_{\tau i}}(\overline{o_i} o_i, \overline{\sigma_j}\sigma_j) = U_{\tau i}(\overline{o_i} o_i, \overline{\sigma_j}\sigma_j, \overline{\delta_i^A}(\overline{o_i}, \overline{\sigma_j}), \overline{U^{\delta_i^A}_{\tau i}}(\overline{o_i}, \overline{\sigma_j})), \tag{3}$$

where, for $\overline{o_i} = \langle o_i^1 o_i^2 \dots o_i^n \rangle$ and $\overline{\sigma_j} = \langle \sigma_j^1 \sigma_j^2 \dots \sigma_j^n \rangle$, we have action sequence, $\overline{\delta_i^A}(\overline{o_i}, \overline{\sigma_j})$ as:

$$\langle \delta_i^A(o_i^1, \sigma_j^1, \overline{U^{\delta_i^A}_{\tau i}}(o_i^1, \sigma_j^1))\, \delta_i^A(o_i^1 o_i^2, \sigma_j^1 \sigma_j^2, \overline{U^{\delta_i^A}_{\tau i}}(o_i^1 o_i^2, \sigma_j^1 \sigma_j^2)) \dots \delta_i^A(\overline{o_i}, \overline{\sigma_j}, \overline{U^{\delta_i^A}_{\tau i}}(\overline{o_i}, \overline{\sigma_j}))\rangle.$$

That is, whenever we have either an empty observation or message sequence, we return the default translation-distribution, $\hat{P}_{\tau i}$. Further, when we have full sequences of observations and messages, the update is based upon those sequences, along with the actions and updates previously generated by their prefix sub-sequences. (Note that the action sequence is given under the supposition that $\overline{o_i}$ and $\overline{\sigma_j}$ are of the same length; this is merely for convenience, and the definition can be rewritten easily for sequences of unequal length, given the proviso on $U_{\tau i}$ concerning empty sequences. Note also that by convention the action sequence for empty observation or message-sequences is also empty: $\overline{\delta^A}(\epsilon, \epsilon) = \epsilon$.)

Given such a translation-update function for the series of actions generated by a given local policy, it is straightforward to extend the definitions of transition probability and value for policies with a shared language (Definition 4 and 5) to the case of language learning. The transition probability $\overline{P_\delta}()$ with translations is similar to the sequence-transition probability as given in Definition 4. The difference is the inclusion of the processing of the messages received, i.e. the update function, $\overline{U^{\delta_A}_{\tau_i}}$. Thus, when we consider the actions taken under each individual policy, we take into account not only the most recent sequence of observations and actions, but also the latest state of the translation, given those sequences.

Definition 13 (Transition Probability Over a Sequence of States with Translations). *The probability of transitioning from a state s to a state s' following the joint policy $\delta = (\delta_1, \delta_2)$ in the presence of translations 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:*

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

$$\sum_{q \in S} \overline{P_\delta}(q|s, \overline{o_1}, \overline{\sigma_2}, \overline{o_2}, \overline{\sigma_1}) * P(s'|q, \delta_1^A(\overline{o_1}, \overline{\sigma_2}, \overline{U_{\tau 1}^{\delta A}}(\overline{o_1}, \overline{\sigma_2})), \delta_2^A(\overline{o_2}, \overline{\sigma_1}, \overline{U_{\tau 2}^{\delta A}}(\overline{o_2}, \overline{\sigma_1}))) *$$
$$O(o_1, o_2|q, \delta_1^A(\overline{o_1}, \overline{\sigma_2}, \overline{U_{\tau 1}^{\delta A}}(\overline{o_1}, \overline{\sigma_2})), \delta_2^A(\overline{o_2}, \overline{\sigma_1}, \overline{U_{\tau 2}^{\delta A}}(\overline{o_2}, \overline{\sigma_1})), s')$$

such that $\delta_1^\Sigma(\overline{o_1}o_1, \overline{\sigma_2}, \overline{U_{\tau 1}^{\delta A}}(\overline{o_1}o_1, \overline{\sigma_2})) = \sigma_1$ *and* $\delta_2^\Sigma(\overline{o_2}o_2, \overline{\sigma_1}, \overline{U_{\tau 2}^{\delta A}}(\overline{o_2}o_2, \overline{\sigma_1})) = \sigma_2$.

Similarly, we define the value of a joint policy with translations analogously to the case with a shared language (Definition 5). Again, the update function $\overline{U_{\tau_i}^{\delta A}}$ is taken into account to reflect how the sequence of past messages and actions is viewed, given the current state of the translation.

Definition 14 (Value of an Initial State Given a Policy with Translations). *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:*

$$V_\delta^T(s^0) = \sum_{(\overline{o_1}o_1, \overline{o_2}o_2)} \sum_{q \in S} \sum_{s' \in S} \overline{P_\delta}(q|s^0, \overline{o_1}, \overline{\sigma_2}, \overline{o_2}, \overline{\sigma_1}) *$$
$$P(s'|q, \delta_1^A(\overline{o_1}, \overline{\sigma_2}, \overline{U_{\tau 1}^{\delta A}}(\overline{o_1}, \overline{\sigma_2})), \delta_2^A(\overline{o_2}, \overline{\sigma_1}, \overline{U_{\tau 2}^{\delta A}}(\overline{o_2}, \overline{\sigma_1}))) *$$
$$R(q, \delta_1^A(\overline{o_1}, \overline{\sigma_2}, \overline{U_{\tau 1}^{\delta A}}(\overline{o_1}, \overline{\sigma_2})), \delta_1^\Sigma(\overline{o_1}o_1, \overline{\sigma_2}, \overline{U_{\tau 1}^{\delta A}}(\overline{o_1}o_1, \overline{\sigma_2})),$$
$$\delta_2^A(\overline{o_2}, \overline{\sigma_1}, \overline{U_{\tau 2}^{\delta A}}(\overline{o_2}, \overline{\sigma_1})), \delta_2^\Sigma(\overline{o_2}o_2, \overline{\sigma_1}, \overline{U_{\tau 2}^{\delta A}}(\overline{o_2}o_2, \overline{\sigma_1})), s')$$

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.

Once again, the optimal policy for a Dec-MDP-Com with given translation-update functions, $U_{\tau i}$ for each agent α_i, is that joint policy maximizing expected value. As before, we stress the difference in difficulty between stating the value-function, and actually calculating the value so that we can determine an optimal value-maximizing policy. Solving such a problem will be no easier in general than for general problems with shared languages. In any case, our work does not involve the generation of such policies. Rather, we are interested in the translation-update functions that produce behavior in concert with *given* policies. Therefore, we look now at ways in which these update functions can be simplified.

4.2 Belief Updates with Limited Information

In the processes we consider, belief-state updates need not be based upon possibly unbounded sequences of observations, messages, and actions. Figure 1 (in Section 2) has given a general overview of the relationships between various components of the language-learning process. As shown there, the belief-state of an agent at any time t, β^t, depends upon the following pieces of information:

1. The prior belief-state, β^{t-1}.

2. The prior action, a^{t-1}, taken on the basis of that belief-state.

3. The most recent observation of the environment, o^t, resulting from that action.

4. The most recent message received, σ^t.

It is to be noted that these points describe an ideal case for language learning. In particular, we are assuming here that our updates rely only upon information and action taken from the current and immediately prior time-step; as we shall see, the process may become quite complicated, if not infeasible, when updates rely upon information from further in the past. Furthermore, we cannot always guarantee that these sorts of updates are always possible, even given just the time-limited information specified here; we shall also examine cases in which updating the probabilities assigned to certain translation entries are simply infeasible, due to defects in the structure of the language in question.

As given by Definition 8, a belief-state is a two-part structure, consisting of an agent's current best belief about the meaning of the other language under consideration (the probability distribution over translations, P_τ), along with a belief concerning all other features of the environment (the probability distribution over states, F). The process of updating the belief-state therefore involves updating each of these components, either separately or together. As we model this process, it takes place sequentially, updating each part in turn. Let $\beta_i^t = (P_{\tau i}^t, F_i^t)$ be the belief-state of agent α_i, to be calculated at time t; ideally, we want the probabilities reflected in the two component distributions to be set correctly by way of a two-step update:

1. First, agent α_i updates its belief about language, setting the probability distribution over possible translations based upon its prior belief-state and action, its own current observation, and any message just received from agent α_j. This update should be such that for any pair of messages, it yields the probability that this pair has the same meaning (written $\sigma_i = \sigma_j$):

$$(\forall \sigma_i, \sigma_j)\ P_{\tau i}^t(\sigma_i, \sigma_j) = \sum_{\tau^* \in \tau^+_{\Sigma_i, \Sigma_j}} P_{\tau i}^t(\tau^*)\tau^*(\sigma_i, \sigma_j) = \mathbf{P}(\sigma_i = \sigma_j \,|\, P_{\tau i}^{t-1}, F_i^{t-1}, a_i^{t-1}, \sigma_j^t, o_i^t).$$

2. Second, the agent needs to update its belief about the state of the environment, setting the probability distribution over states according to the prior such belief and action, the most recent message and observation, and the current (just-updated) translation distribution:

$$(\forall s)\ F_i^t(s) = \mathbf{P}(s^t = s \,|\, P_{\tau i}^t, F_i^{t-1}, a_i^{t-1}, \sigma_j^t, o_i^t).$$

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^t, 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^{t+1}. This observation then leads to an update to a new belief-state, β_i^{t+1}, further affecting how later messages are translated, and thus influencing future actions.

The procedure governing the update from belief-state β_i^t to β_i^{t+1} comprises the agent's *language model*: a function from actions, messages, and observations, to distributions over translations and system states. (Definition 11 and what follows, above, gives a formal treatment of

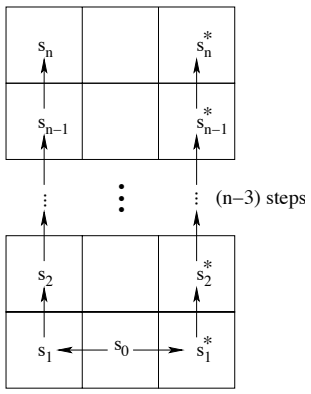


Figure 2: A process for which arbitrarily many messages and belief-states must be memorized.

the language portion of these updates.) Such probabilistic models can range from quite simple to highly complex, depending upon the nature of the languages and problem environment. For instance, in a Dec-MDP-Com in which agents already share a common language—so that translation is not an issue—and can freely communicate with one another, all that is required is that agents update their state-distributions. This, however, is elementary: since a decentralized MDP is *jointly fully observable* (Definition 1, clause 8), agents can compute the actual global state directly, and probabilities are strictly unnecessary. Where agents begin without a commonly understood language, things are obviously much more complicated, and the prescribed belief updates can be difficult to compute correctly. Indeed, it is not clear that it would be possible to generate a meaningful, let alone usefully tractable, probabilistic model of a full-blown natural language. Thus, our work has concentrated upon languages that are restricted to suit the needs of agents working in decentralized MDP environments, communicating specifically about their own observations and actions as they endeavor to achieve optimal performance in some set task. As it turns out, even such basic languages provide interesting challenges and difficulties.

4.2.1 Is the Update Markovian?

Much work in machine learning has focussed on first-order Markov processes, since the ability to usefully employ common learning algorithms often relies upon the fact that action transitions depend only upon the current state of the system. As the problems being solved become higher-order, and expected transitions depend upon longer and longer histories, state representations and necessary computations can quickly become intractable. For the same reasons, we would prefer that our agents' ability to generate probability distributions over message meanings also only depended upon their most immediate actions, observations, and communications. As it happens, however, this sort of update, which correctly sets the various probabilities based only on information from the current or prior time-step, is not always possible. In particular, even for very simple languages of communication, there are cases of the language-learning problem in which an agent's ability to correctly update the probability assigned to some state of the world or translation depends upon that agent remembering an arbitrarily long sequence of message, actions, and so on, from its past experience.

An example of such a process is given in Figure 2. Here, we imagine a case in which agent α_1 is trying to determine the location of agent α_2, based on what α_2 communicates at each time-step. At time t_0, we suppose that α_2 is known to occupy the location s_0; furthermore, agent α_1 understands all of the terms in the language of α_2, except for the terms for "Left" and

"Right". Now, agent α_2 communicates that it is moving either to the left or to the right, and we can suppose that α_1 has no way of determining exactly which, thinking either possibility equally likely. Thus, at time t_1, α_1 believes that α_2 is either in location s_1 or s_1^*, with equal probability. For the next $(n-1)$ time-steps, α_2 moves upwards, and communicates this fact. Since it understands this part of α_2's language, α_1 updates its belief-state at each step, so that at any time t_j, α_1 believes that α_2 is either in location s_j or s_j^*, with equal probability. Finally, suppose that at time t_n, agent α_1 makes some observation (other than the actual location of α_2) that allows it to finally translate all of α_2's language, including the terms for "Left" and "Right". At this point, it becomes possible for α_1 to determine the location of α_2, and update its belief-state about that location appropriately. However, that update is only possible if α_1 has remembered the sequence of things that α_2 communicated all the way back to time t_0, along with the original location at that time. Since the sequence of steps taken before α_1 comes to understand the required terms in the language may be arbitrarily long, it is not generally possible to bound how much information needs to be remembered in order to correctly update the belief-state over time.

Figure 3 shows the update relationships at work in the example just given. (Actions taken by agent α_1 are not shown, as we assume they do not affect the belief-state updates.) What we see in this figure is that at every time up to $(n-1)$, the belief-state update depends only upon information from the prior or current time-step. However, at time n, when new information allows the agent to correctly translate the other's language, the update process takes on a new character. Now, the belief distribution over world states at that later time, F_n, depends not only upon local information, but also about newly understood information from time-steps long past. Furthermore, that past information can only be understood if various components of the belief-state from those past time-steps are *updated again*, based now upon the translation available only at the *current* time-step. (Specifically, beliefs about α_2's location at those time-steps are updated based upon the new translation of terms for "Left" and "Right".)

In general, then, the presence of a language which is not completely understood adds potential complexity to the whole problem of belief-update. When the language of other agents is fully known, we may be able to base our beliefs about the current state of the world solely upon current or recent observations and communications; however, where we do not fully understand that language, we may need to revise our current beliefs based upon information about things long past, which we have only now come to understand.[3] The process of updating beliefs about the current state of the world can require that we remember every message, observation, and so on, in order that if and when we do come to understand another's language, we can reinterpret the various things we have seen based on our new understanding.

Such a situation is highly undesirable, if what we seek is tractable methods for updating translations and beliefs given uncertain communication. If agents must potentially remember unbounded amounts of past information, and attempt to update current belief-states based upon histories of arbitrary length, we cannot guarantee that they can perform these updates within acceptable bounds on computational resources. This is analogous to the difficulties faced in determining optimal courses of actions in *partially observable* MDPs (POMDPs), where an agent's belief-states over actual global states can rely upon observation histories of unbounded length. Thus, we would face the same sorts of intractability results [32] in our work, if we allowed agents to update their beliefs about communication in such a way as to guarantee

[3]Of course, such a situation can arise even for fully understood languages, especially as complex utterances may involve references to past situations. I can always say to you today, "You ought now to do that thing I told you about last Tuesday," and your being able to understand and follow that directive will require that you recall what I in fact said at that past time. What the example just given shows, however, is that the need to recall past events may arise even if we are dealing with simple languages, involving just present actions and observations.

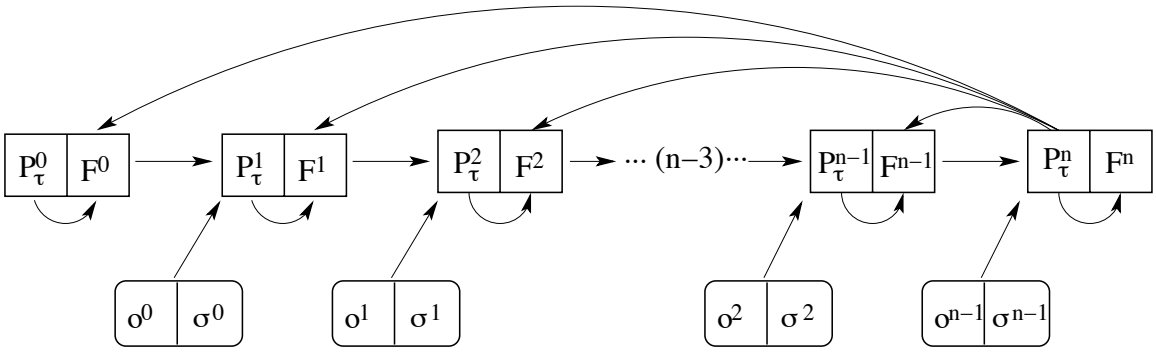


Figure 3: Belief-state update relationships for the process of Figure 2.

optimal behavior. We thus provide methods, described later, that do not guarantee optimal action at every step of a decision process. Rather, we focus on techniques that allow agents to converge eventually upon mutual understanding and optimal action in certain particular problem-environments, while basing belief updates solely upon their most recent observations, actions, and messages.

4.2.2 The Need to Restrict Languages

Unfortunately, the potential need to understand past communications in terms of current information can give rise to another potential problem, having nothing to do with the length of history an agent must consider, but rather concerning the very possibility of updating beliefs and interpretations coherently. On the one hand, an agent's current translation will depend upon what it believes about the world; on the other hand, its beliefs are affected directly by how it translates messages received. This then leads to a situation in which the update of a present belief about the language of another agent may affect how we are to understand our environment, and in turn, our new understanding of the environment affects our beliefs about language. Such a circular relationship can in fact turn vicious; it is possible, that is, to construct cases in which a "correct" process of updating beliefs is non-terminating.

For instance, suppose we have two agents in a gridworld environment. Agent α_1 is attempting to understand the language of agent α_2, and has learned to interpret all of that language with certainty, apart from a single expression, $\sigma^\star$. In our framework, this entails that the probability assigned any meaning of $\sigma^\star$ by α_1's current translation is strictly less than 1. We further suppose that α_1 is also uncertain about the location of α_2, and that α_1 attempts to infer that location given what α_2 communicates. Finally, suppose α_2 sends the following composite message:

At time $t-1$**:** "$\sigma^\star$, which means 'I am at location x', if you are not certain that I am at location x; otherwise, it means 'I am at location y'."

The difficulty here is relatively obvious: based on these messages, α_1's belief-update process at time t is thrown into confusion. In order to fix its belief about the present location of α_2, the interpreter needs to assign a meaning to $\sigma^\star$, which was part of the message just received. According to what it has now learned, $\sigma^\star$ should be translated as "I am at location x," since α_1 is uncertain as to α_2's location; that is to say, we have the translation:

$$P_\tau^t(\sigma^\star,\ \text{"I am at location } x\text{"}) = 1.$$

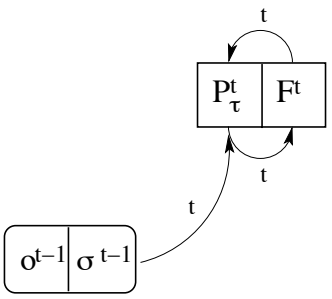


Figure 4: A vicious circle arises in the belief-update process.

Given this translation, then, α_1 can now assign location x to α_2, and so the environment-distribution gives us:

$$F^t(\alpha_2 \text{ is at location } x) = 1.$$

This, however, leads to our problem, namely that α_1 is now certain about the location of α_2, and according to its most recent message should interpret $\sigma^\star$ to mean "I am at location y." In turn, this should lead to a reassignment of y as the location of α_2, which means that α_1 is no longer certain that α_2 is at location x. This lack of certainty then leads to another re-translation of $\sigma^\star$, and so on in a circular fashion, as shown in Figure 4.

Obviously, the problem here is simply a variant of the Liar Paradox, as exploited in such as Gödel's famous theorems. Our point in bringing it up here is, as in the prior section, to illustrate the need for care in both language design and attempts to learn language. Given unrestricted linguistic resources it is easy to construct messages that will prove difficult, if not impossible, to translate correctly given what we know of our problem environment. We therefore concentrate in what follows upon simple languages, to do with the immediate actions and observations of agents in decentralized MDP environments. Furthermore, we restrict our update scheme to the elementary single time-step version already sketched. After having presented the main concepts relevant to the problem of learning to communicate, we continue with a formal analysis of the problem, learning algorithms and various properties. This work lays the groundwork to study other versions of this problem in more complex scenarios and under different characteristics.

5 The Language Learning Problem

As we have explained, the general problem of learning to communicate while acting in a decentralized environment can be captured as the process of updating belief-states about system-states and translations, together with policies that allow agents to act based upon these beliefs Such problems may arise in any number of contexts. One plausible use for such techniques arises in cases of automated systems coping with errors in their design or specification, as for instance the problem of disparate metric and imperial measures encountered in the Mars orbiter program [31]. Such systems, when communicating with one another, or with human operators, may need to learn to reinterpret instructions or state specifications, in light of new information. In our Dec-MDP-Com framework, agents would each possess a model of the overall problem domain, in terms of possible states of the system, along with a probabilistic model of action effects in terms of state transitions, and a model of expected reward for actions. When observations indicate discrepancies between the world and how received messages are understood, agents will then need to learn how to re-translate those messages. The Mars orbiter, for example, upon observing that following its given flight-path information was leading it too close to the surface,

might have been saved had it the capacity to learn how to adjust its understanding of those instructions by translating them into another system of measure.

Lacking any particular restriction on the updates and action choices, this general process can be very complex and is not guaranteed to converge to either a correct interpretation of the messages exchanged, or an optimal policy of action. In this section, we define the problem of learning to communicate precisely, giving criteria for what it means to solve it. Then, we identify desirable properties of algorithms for doing such learning, and give properties of Dec-MDP-Coms that allow us to give some guarantees on the performance of a system of probabilistic belief updates. The rest of the paper will concentrate on a particular implementation of an algorithm that solves the learning problem in such problem instances.

5.1 Solutions and Solution Algorithms

"Learning to communicate" comprises a very broad set of behaviors and capacities. We give the phrase specific meaning in terms of convergence to optimal action policies in a decentralized MDP. In this context, agents learn to communicate while acting towards a jointly optimal solution of the Dec-MDP-Com at hand, and thus we evaluate how successful learning has been according to its usefulness in achieving such a value-maximizing policy.

To make the problem studied here precise, we refer to the expected reward of the optimal joint policy, given some translations, as explained in Section 4.1. Then, we can tell that a system of agents has learned to communicate while acting if there exists some point in time such that the best plan based on translations from that point on is in fact optimal. We note here that we are not studying the optimality of the learning process itself, i.e., how much time the learning process takes until such a point in time is found. Instead, we are studying the performance of a decentralized system when miscoordination may arise as a result of discrepancies in the interpretation of messages exchanged.

We begin by defining what it is for a joint policy, in combination with agents' update functions, to converge. (In our formal definitions, we again restrict the presentation to a two-agent case. All definitions are directly extensible to the n-agents case.)

Definition 15 (Policy Convergence with Translation Updates). *Let M^1 be a Dec-MDP-Com process with multiple languages:*

$$M^1 = \langle S,\, A_1,\, A_2,\, \Sigma_1,\, \Sigma_2,\, C_\Sigma,\, P,\, R,\, \Omega_1,\, \Omega_2,\, O,\, T\rangle,$$

and let agents α_1 and α_2 have translation-update functions $U_{\tau 1}$ and $U_{\tau 2}$, and joint policy $\delta_{M^1} = [(\delta_1^{A1}, \delta_1^{\Sigma 1}), (\delta_2^{A2}, \delta_2^{\Sigma 2}))]$, where the local policies of each α_i are as follows:

$$\delta_i^{Ai} : \Omega_i^* \times \Sigma_j^* \times P_{\tau i}^+ \to A_i \qquad\qquad \delta_i^{\Sigma i} : \Omega_i^* \times \Sigma_j^* \times P_{\tau i}^+ \to \Sigma_i.$$

Further, let M^2 be a Dec-MDP-Com process that is identical to M^1, except that it has only the one shared language $\Sigma^+ = (\Sigma_1 \cup \Sigma_2)$:

$$M^2 = \langle S,\, A_1,\, A_2,\, \Sigma^+,\, C_\Sigma,\, P,\, R,\, \Omega_1,\, \Omega_2,\, O,\, T\rangle,$$

and for any state s at time t, let $\delta^*_{M^2}(s^t) = \arg\max_\delta V_\delta^{T-t}(s^t)$ be the optimal policy for M^2, starting from s at t. We say that the joint policy δ_{M^1}, with update functions $U_{\tau 1}$ and $U_{\tau 2}$, has converged *at some state s and time t if and only if:*

$$V_{\delta_{M^1}}(s^t) = V_{\delta^*_{M^2}(s^t)}(s^t).$$

That is, a policy converges, along with the associated update functions, when the agents reach a point in time after which their actions, in accord with their translations at that point and at all points afterwards, return the maximum value that could be expected if the agents did in fact share all their linguistic resources. This definition highlights the dual nature of convergence in the language-learning context, in that it depends upon both the policies of action and communication, as well as the translation-update functions in use. Given this notion of convergence, it is straightforward to give an exact statement of the learning problem.

Definition 16 (Learning to Communicate while Acting - The Problem). *Let M be a Dec-MDP-Com process with multiple languages: and let agents α_1 and α_2 have translation-update functions $U_{\tau 1}$ and $U_{\tau 2}$, and joint policy $\delta_M = [(\delta_1^{A1}, \delta_1^{\Sigma 1}), (\delta_2^{A2}, \delta_2^{\Sigma 2}))]$. We say that the agents have* solved the problem of learning to communicate while acting *at some state s at time t if and only if δ_M, with update functions $U_{\tau 1}$ and $U_{\tau 2}$,* has converged *at s^t.*

That is, the agents have solved the problem of learning to communicate whenever they arrive at a policy of action that, in combination with their given translation-update functions, leads to a course of action equal in value to one that would maximize return if the agents in fact shared all the same language. The definition therefore says nothing about the character of the translations at work. In particular, we are not committed to any view as to whether or not these translations are "correct" or not. Rather, agents are said to have solved the problem of learning to communicate whenever their actions, *however they translate* one another, are optimal. This opens the door to a number of possibilities, including translations that are only partially complete, or ones that conspicuously mistranslate certain language items. So long as these kinds of omissions or mistranslations do not affect the optimality of the associated policy, we accept that the agents have indeed learned to communicate.

5.1.1 Optimality of Converged Policies

A consequence of this manner of defining a solution to a communication problem is that it takes into account the possibility that the joint policy, along with the update functions, may converge to a pattern of behavior that is not *absolutely* optimal. That is, the optimality of that policy is relative to the point of convergence, s^t, and we do not guarantee optimal results if the agents were to "start over," implementing the policy from the initial Dec-MDP-Com start-state, s^0. This is a result of an unavoidable fact, namely that the time it takes to learn something in the setting of a decision problem may have negative consequences on the expected value in that environment. In general, we cannot guarantee that the time taken for learning does not cut agents off from some potential expected value. For instance, in some problem domains, there are states of the environment which are simply unreachable given the optimal joint policy of action. In such cases, an absolutely optimal policy need not prescribe actions that actually maximize value for such states; since they are never reached, any action whatsoever may be assigned to them without affecting overall optimality. When agents are learning, however, and taking sub-optimal actions along the way, they may very well find themselves in such states of the environment, and so their final policies may end up very different.

There is one important exception, however. In an *ergodic*, infinite-horizon Dec-MDP-Com, every state reachable with non-zero probability by following any action policy recurs infinitely often under that same policy in an aperiodic fashion; the set of such states visited is called the *recurrent set* [33]. If the recurrent set for every policy comprises the entire state space S, then the process is *irreducible*. It is well known that the expected value of policies in such environments is indifferent to the exact starting state, allowing us to establish the following.

Claim 1 (Optimality in Irreducible Dec-MDP-Coms). *Let M^1 be an irreducible two-agent and two-language Dec-MDP-Com, with an infinite time horizon. Let δ_{M^1}, with update functions $U_{\tau 1}$ and $U_{\tau 2}$, have converged at some s^t in M^1. Then δ_{M^1} is absolutely optimal for the shared-language version of the problem, M^2; that is, for initial state s^0 of M^2:*

$$\delta_{M^1} = \delta^*_{M^2} = \arg\max_{\delta_2} V_{\delta_2}(s^0).$$

Proof: δ_{M^1}, along with $U_{\tau 1}$ and $U_{\tau 2}$, have converged at s^t, and so by definition,

$$V_{\delta_{M^1}}(s^t) = V_{\delta^*_{M^2}(s^t)}(s^t), \tag{4}$$

where $\delta^*_{M^2}(s^t)$ is the policy that is optimal in the shared-language problem M^2 from point s^t onwards. Since M^1 is irreducible and infinite-horizon, so is M^2, as the problems are identical apart from the languages involved. Thus, starting at s^t in M^2, and following policy $\delta^*_{M^2}(s^t)$, we will visit original initial state s^0 infinitely often. Therefore, this policy is absolutely optimal:

$$\delta^*_{M^2}(s^t) = \arg\max_{\delta_2} V_{\delta_2}(s^0). \tag{5}$$

Equations (4) and (5) establish Claim 1. □

Our empirical work, as described in Section 7, has not in fact generally involved problems that are fully irreducible, nor even properly ergodic. On the other hand, because we wanted to impose natural stopping conditions on our learning algorithms, we dealt with problems in which agents eventually encountered every state in the environment often enough to drive their translations to a point at which those translations were in fact *complete* (in the sense of Definition 19, below). In such cases, then, and because starting conditions for our decision tasks repeated randomly (so that there are no "dead ends"), it is easy to show that the convergent policies of action with translation that we arrive at are in fact absolutely optimal for those problems. We do not prove this fact here, as it is only incidental to the goals of this paper. Rather, we note it only to point out that the choice was one of convenience: our proposed methods require neither the special properties just mentioned, nor full irreducibility, to succeed. For instance, we prove in Claim 6, Section 6, that a particular protocol for action and translation update converges to an optimal policy. While that claim requires an infinite time horizon in general, to allow enough time for learning, irreducibility is not assumed; the convergence proven is thus that of Definition 15.

5.1.2 τ-Learning Algorithms

The central feature of any attempt to solve the given problem is an algorithm for updating translations; here, we call such a communication-learning method a *τ-learning* algorithm. An example of such an algorithm is given in Section 6. In our work, we have concentrated upon instances in which policies of action and communication are given ahead of time. That is, the central problem facing an agent is that of updating its translations. Once this has been accomplished, these pre-selected policies dictate what is to be done, based upon the current interpretation. (More details of the policy-selection process are given below.) Of course, different τ-learning algorithms may result in different belief updates, and can thus have differing effects upon the values of the associated policies. For example, τ-learning algorithms that generate correct translations of only a proper subset of messages can result in an overall system utility that is lower than would be expected if agents were able to translate all messages properly. In

other instances, however, this may not be the case, as for instance when not all messages are actually essential to the successful performance of the system.

In addition to a τ-learning algorithm, agents may need a rule that instructs them how to choose one meaning for a message received out of their translations. We give an example of such a rule in the protocol presented in Section 6 where agents will act upon the meaning that maximizes the probability of a translation being the correct one. Another reasonable assumption to enable the agents to learn to communicate is that observations sensed after actions were taken upon interpretation of messages are informative. While this does not tell an agent the actual correct meaning of a message, it provides some guidance about whether it has acted correctly or not. There may be many ways to act right or wrong, and this observation is not required to distinguish among these. However, we avoid cases in which observations provide no useful information at all; clearly, such environments make language learning impossible, but this is a defect in the problems, not in the methods for solving them. Similarly, we are not interested in cases where any action taken upon any interpretation results in the same expected reward, since in such cases, there is clearly no more benefit to be had from learning language than from ignoring it altogether. This section focuses on some formal properties that τ-learning algorithms may have. These properties are helpful in understanding the guarantees of optimality of learning algorithms.

Definition 17 (A τ-learning algorithm is certain with respect to (σ_i, σ_j)). *A τ-learning algorithm is certain with respect to (σ_i, σ_j) if the algorithm is guaranteed to converge to a translation τ such that $\tau(\sigma_i, \sigma_j) = 1$ or $\tau(\sigma_i, \sigma_j) = 0$.*

Definition 18 (A τ-learning algorithm is ϵ-certain with respect to (σ_i, σ_j)). *A τ-learning algorithm is ϵ-certain with respect to (σ_i, σ_j) if the algorithm is guaranteed to converge to a translation τ such that $\tau(\sigma_i, \sigma_j) \geq \epsilon$, or $\tau(\sigma_i, \sigma_j) \leq (1 - \epsilon)$.*

This type of learning algorithms will enable us to test the levels of coordination in multiagent systems that have learned to communicate at different ϵ levels. This measure will serve us as a quantifier to compare the performances of such systems. Furthermore, we may be able to switch between contexts when the translations cannot be ϵ-certain for some ϵ. We leave this for future work.

Definition 19 (A τ-learning algorithm is complete). *A τ-learning algorithm is complete if it is certain with respect to all pairs of messages in Σ_i.*

That is, an agent employing a τ-learning algorithm that is complete to learn how to communicate in some language Σ will have converged upon a translation that is ultimately certain in all its assignments to message-pairs. In order to use such a translation to actually generate an optimal policy of action, it can also be necessary that agents are able to communicate about all relevant features of the environment.

Definition 20 (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 made in the system, and (*b*) any available action.*

The application of a complete τ-learning algorithm in a fully describable environment allows agents to solve Dec-MDP-Coms to optimality.

Claim 2. *Solving the problem of learning to communicate for a fully describable Dec-MDP-Com*

$$\overline{M}_2 =< S, A_1, A_2, \Sigma_1, \Sigma_2, C_\Sigma, P, R, \Omega_1, \Omega_2, O, T >$$

with a complete τ-learning algorithm is a sufficient condition for solving $\overline{M}_2$ optimally.

Proof. Since the τ-learning algorithm is complete, it is certain with respect to all pairs of messages. Then, there exists some time t for which the agents are guaranteed to know with certainty a mapping between messages in both languages Σ_1 and Σ_2. Furthermore, since the Dec-MDP-Com is assumed to be fully describable, given enough time, the agents know to map any possible observation and action in A_i and Ω_i to any action and observation in A_j and Ω_j. Therefore, given enough time for the learning algorithm to converge and without loss of generality, agent α_1 can exchange messages in Σ_2 instead of in its original language Σ_1. Hence, the expected value of the joint policy of action and communication that solves $\overline{M}_2$ will be equal to the expected value of the optimal solution to the corresponding problem with a mutually shared language. □

While this result establishes that we can in fact solve a Dec-MDP-Com optimally given a complete method, such a τ-learning algorithm is not a necessary condition for optimality. A τ-learning algorithm may not be complete—that is, it may not reach full certainty for every possible pair of messages—and nevertheless the agents' performance may be optimal. For example, this may occur when messages that do not get translated with certainty are not relevant for the optimal solution of the problem. Such uncertain translations cannot affect the value of the joint policy of action and therefore, agents may behave and communicate optimally even though their translations are not complete. For example, assume a set of two agents is assigned a task of revealing a hidden treasure. Agents need to move around a two-dimensional grid in order to find the treasure. Agents can exchange messages about locations, for example instructing the other agent to move towards a certain place. Discrepancies in the language of communication means in this example that agents do not have the same system of coordinates and therefore need to correlate pairs of locations in both languages. There may be some set of locations that is never used in the solution of the problem, so the fact that one agent may not know how to interpret the names of those locations correctly will have no effect on the quality of the solution. Similarly, agents may achieve optimal performance using an incomplete algorithm that does not assign complete certainty to translations, but is correct in terms of *relative* certainty (so that the correct translation of any message is always the most probable). In such a case, if agents always choose the most likely interpretation of any message as its actual interpretation, then the lack of absolute certainty will have no effect upon the value of those actions.

5.2 Suitability of Dec-MDP-Coms

The ability of agents to learn to communicate depends not only upon the algorithms employed to update belief-states, but also upon features of the problem instance at hand. In the previous section, we identified one such feature, full describability (Def. 20); here, we identify further properties of Dec-MDP-Coms, and discuss their usefulness and possible necessity in the learning process. The first of these properties has to do with the cost of communication in the system; while the general definition of a Dec-MDP-Com allows arbitrary costs for each message sent, we are particularly interested in cases where communication is cost-free.

Definition 21 (Freely describable). *A Dec-MDP-Com is* freely describable *if and only if the cost of communicating any message σ is 0.*

Free communication has the potential to considerably simplify a Dec-MDP-Com, since agents may communicate as much as they like without affecting overall system utility. Where the system is not freely describable, optimal control policies need to consider not only what to communicate, but when, according to a cost-benefit analysis of the potential value of the information shared as compared to the price of communicating it. Thus, as discussed in Section 5.2.1 below,

it may not be possible to generate optimal solutions to such problem instances in an effective manner. Detailed study of the communication-learning problem in a non-free decentralized environment is beyond the scope of this paper.

When combined with free communication, the ability to describe a Dec-MDP-Com in full can radically simplify the problem at hand. For example, we have shown [1] that when agents in fact share a common language, a Dec-MDP-Com that is both freely and fully describable is reducible to the much simpler case of a multiagent MDP (MMDP). MMDPs, in which each agent effectively observes the entire global system state, are known to allow for polynomial algorithms that generate optimal policies, and are thus tractable in ways that Dec-MDP-Coms are not [6]. As discussed below, such properties also affect the possibility of optimal solutions when agents must first learn to communicate before they can undertake their policies.

As we have pointed out, optimal action in a Dec-MDP-Com is importantly related to agents' ability to communicate effectively about their observations and actions. Similarly, the ability to update belief-states so that agents learn to communicate about such factors depends upon their capacity to discern more likely interpretations over time. We thus define what it is for a Dec-MDP-Com to be *suitable* for language learning, allowing agents in such a system to update their translation belief-states in a monotonic fashion, so that correct translations become ever more likely. Among other things, such monotonic updates allow agents to avoid the sorts of vicious-circle problems previously discussed in Section 4.2.2. To define the required property, we first give some additional 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, let*

$$P_i^\sigma(o_j \mid \sigma_j^o, \beta_i^t) \tag{6}$$

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

$$P_i^\sigma(a_j \mid \sigma_j^a, \beta_i^t) \tag{7}$$

is the probability that α_j will take action a_j, given message σ_j^a and α_i's current belief-state.

Let $\max_i^\sigma(o_j)^t$ and $\max_i^\sigma(a_j)^t$ be the observation and action maximizing expressions (6) *and* (7)*, 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 $\langle o'_1, \ldots, o'_n \rangle$ at time $t+1$. Then, for any agent α_i, let*

$$P_i^o(o_j \mid o'_i)^{t+1} \tag{8}$$

be the probability that agent α_j previously observed o_j, given that α_i now observes o'_i. Similarly,

$$P_i^a(a_j \mid o'_i)^{t+1} \tag{9}$$

is the probability that α_j took action a_j given α_i's current observation.

Two observations are worth making here. First, the probabilities indicated by each of the new notations are different in character. That is, the probability of Notation 1 is essentially *subjective*, whereas the probability of Notation 2 is *objective*. Notation 1 identifies a probability that each agent assigns to an event, given that agent's update scheme and language model. Notation 2, on the other hand, identifies a probability found in the Dec-MDP-Com model itself.

As we go on to show, in certain circumstances, these probabilities can be made to match, so that agents do in fact employ update schemes that cohere with the dynamics of their problem environment. The second important point concerns the nature of the observation o_i', as it appears in Notation 2. In much research into decentralized MDP models, system-wide reward is separated from an agent's observations, and as such is not generally available to each agent; while the goal may be to find policies that maximize this reward, agents do not act directly in response to these rewards. In other research, however, agents do in fact observe the system-wide reward; work using MDP models in reinforcement learning contexts, for instance, often uses the state-transition reward as the primary motive force driving the agents toward optimal action over time [36]. In some of the examples we will consider, we have chosen the latter path, and made the reward available to the agents at each step of the process. Nothing vital depends upon this choice, however, and readers who are used to the idea that reward is unobserved can simply choose other, analogous examples. Further, we sometimes distinguish between those components of the observation directly related to language-learning issues, and those related to the rest of the environment. Again, this is just for convenience. As given in the notation, we are concerned simply with the probabilities of certain actions and observations, given what is observed during a state transition; if only certain features of the observation are informative, this will not affect the relevant probabilities.

Definition 22 (Suitability). *Let M be any fully 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. We say that M is* suitable *just in case 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),*

$$P_i^o(o_j \mid o_i')^{t'+1} > P_i^o(\max_i^\sigma(o_j)^t \mid o_i')^{t'+1}, \tag{10}$$

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

$$P_i^a(a_j \mid o_i')^{t'+1} > P_i^a(\max_i^\sigma(a_j)^t \mid o_i')^{t'+1}. \tag{11}$$

That is, in a suitable Dec-MDP-Com, suppose agent α_j observes o_j and takes action a_j, communicating both to other agents using messages σ_j^o and σ_j^a. However, based upon its belief-state at that time, some other agent α_i incorrectly considers a different 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' "corrects" the situation. That is, at the next time-step, α_i will now consider it more likely that α_j observed o_j, rather than the incorrect observation. And, since this property holds at all future time-steps, any time that α_j happens to observe that same o_j again, α_i will still consider that correct observation more likely than the one previously thought most likely. (And similarly for the action a_j taken by α_j.)

This property is quite complicated to state, and may take a little time to digest. However, while suitability is somewhat difficult to formalize precisely, we do not consider it to be an overly strong or artificial condition. As we argue in the next section, lack of suitability may make it simply impossible to update the probabilities assigned to action- or observation-messages in any meaningful fashion, and in unsuitable environments a decision-theoretic agent may be incapable of achieving optimal performance. Furthermore, suitability is not necessarily an uncommon property of Dec-MDP-Coms. Note that the property as given does not require that the actual identity of the correct prior observation and action of other agents be determined exactly by individual observed outcomes. Rather, it is only needed that the correct such observation or action be *to some degree more likely* than ones that might have seemed most likely before. For

instance, domains in which agents have no idea what actions others are taking, but can at least positively eliminate some incorrectly chosen candidate—assigning it zero (0) probability given the immediate effects of the action taken—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. Lastly, we note that Section 7.2 contains an example implementation of a relatively complicated, realistic, and suitable problem instance, in which agents are able to learn to communicate in an effective fashion.

5.2.1 Importance of the Dec-MDP-Com Properties

We now establish that the three properties outlined in the previous sections—free communication, full describability, and suitability—are not overly strong or *ad hoc*. To do so, we argue for the claim that the lack of each in turn has the potential to make probabilistic updates of translations, and optimal action based on those updates, either intractable or impossible. In the absence of such properties, then, agents will either require additional restrictions upon the system if they are to generate optimal solutions to the associated decision problem, or must employ some completely different form of learning algorithm in order to converge to optimal behavior. Later, we will show an implementable technique for such convergence in the presence of all three properties, establishing their sufficiency with respect to communication learning. For now, we concentrate upon their necessity.

Claim 3 (Necessity of free communication). *Learning to communicate and act optimally in a Dec-MDP-Com that is not freely describable is generally intractable.*

Discussion: As already described, solving decentralized MDPs without communication is generally intractable [4]. While the presence of free communication can radically simplify such problems, prior work has shown that decentralized control in the presence of costly communication in a commonly understood language is also computationally hard in general [25, 24]. It is easy to see that the problem of learning to communicate in a Dec-MDP-Com without free message-passing mechanisms can be no easier. In the end, even if agents are able to converge upon a jointly understood language of communication, the task of calculating an optimal policy based upon sharing messages in that language can still remain too complex for solution in all cases. The value of a joint policy in such cases depends not only upon whether the language is interpreted correctly but also upon whether the agents should communicate at all due to its cost. If the cost of communication is too expensive, an optimal joint policy might in fact be to never communicate at all, thus making any learning of message-meanings useless at best, if not impossible. Therefore, we cannot guarantee any tractable means of convergence to mutual understanding and optimal action.

Claim 4 (Necessity of full describability). *Learning to communicate and act optimally in a Dec-MDP-Com that is not fully describable is generally intractable.*

Discussion: The same complexity results just discussed demonstrate the need to have full describability in order to solve a Dec-MDP-Com optimally in a tractable manner. The general hardness of decentralized MDPs without communication arises when agents only observe the state space in part, and cannot always know what other agents are observing, nor what actions they are taking. As mentioned, the ability to share such information freely can simplify these

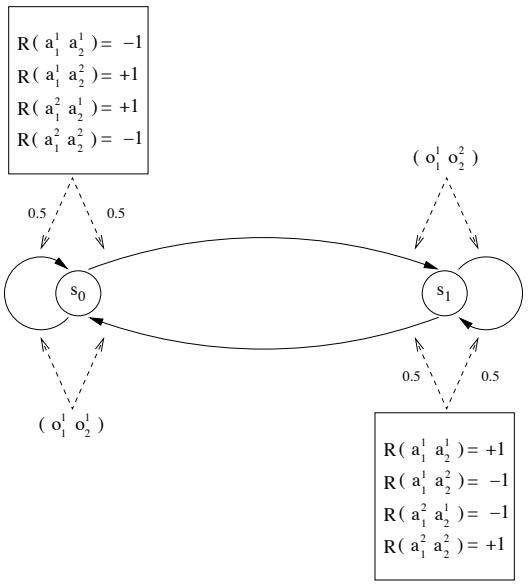


Figure 5: An unsuitable decentralized decision-problem for which optimal action cannot be achieved through probabilistic belief updates.

problems. However, even if communication is free, a simple lack of the linguistic resources to communicate all relevant information can once again make the decision problem too hard to solve in general. A simple example makes the point: a Dec-MDP-Com in which agents have no language for communicating their observations or actions at all, but can only communicate about the joint system reward, reduces to a simple decentralized MDP. Here, the limited ability to share only this particular information is tantamount to the case where no communication is available at all, since all that is being communicated between agents are facts about a reward that is already shared by all. Thus, the lack of full describability turns the problem of generating an optimal policy in the Dec-MDP-Com into the problem of solving a decentralized MDP without communication, which is generally intractable.

Claim 5 (Necessity of suitability). *In the absence of suitability, agents will be generally unable to update translation-belief probabilities so as to allow them to act optimally.*

Proof: We consider a counter-example instance of an unsuitable Dec-MDP-Com for which agents will be unable—given the available evidence of observations and rewards—to update translation probabilities in a fashion that eventually allows them to act optimally. Figure 5 diagrams this simple problem, consisting of two states, s_0 and s_1, and two agents α_1 and α_2. Each agent α_i possesses two possible actions, a_i^1 and a_i^2, and two possible observations o_i^1 and o_i^2; thus, there are four possible combinations of actions and four possible combinations of observations available at any point. System dynamics are as follows:

Actions: In each state, actions have the same possible transition effects; namely, in state s_i, any of the four action combinations cause a transition back to that same state s_i with probability 0.5, and cause a transition to the other state $s_{j\neq i}$ with the same 0.5 probability.

Observations: Only two possible observation combinations ever arise: whenever the state transitions to s_0, the pair (o_1^1, o_2^1) is observed, and whenever the state transitions to s_1, the pair (o_1^1, o_2^2) is observed, with certainty in either case. For purposes of this example, we assume that agents also observe the system-wide reward.

Rewards: These are as shown in the diagram. For instance, following any action transition originating in state s_0, the action pairs (a_1^1, a_2^2) and (a_1^2, a_2^1) each yield positive reward of $+1$ unit, whereas the other two action pairs yield negative reward of -1 unit. For action transitions originating in s_1, however, the situation is reversed, with (a_1^1, a_2^1) and (a_1^2, a_2^2) yielding positive reward this time.

We can assume that the problem is freely and fully describable and that each agent knows the full system description, with all relevant probabilities, ahead of time. On the one hand, if the agents shared a common language, then an optimal joint policy for this problem is obvious, and trivial: agents communicate their observations, and given a jointly observed transition into either state, simply communicate again to choose one of the pairs of actions that give a positive reward. Agents thus receive maximal reward at every time-step, for a total reward equal simply to the time horizon of the problem, T. However, where agents begin with no shared language a problem arises, namely that agent α_1 will be unable to update its probability assignments to messages in such a way that it can reliably choose appropriate actions, given what it can observe at each time-step.

To see this, consider the translation update possible for α_1 at initial time-step t_0. At this point, whatever state the system begins in, α_1 observes the same thing, o_1^1; thus, it must assign whatever observation-message σ_2^o it receives from α_2 the same probability of meaning either o_2^1 or o_2^2. Similarly, with no initial evidence, α_1 must consider whatever action-message σ_2^a it receives to be equally likely to mean either a_2^1 or a_2^2. Based on these indifferent probabilities, then, a decision-theoretic agent will be indifferent between either of its own actions a_1^1 or a_1^2, since the expected value of each is the same. Without loss of generality, suppose that α_1 chooses its own first action a_1^1. Then, a curious symmetry arises: no matter what happens following the ensuing joint action, α_1 will have no more reason to translate the action-messages it has received, or any further observation-messages, in any different fashion. As an example, suppose that following the joint action, the system receives a reward of -1, and agent α_2 now sends a different observation-message $\sigma_2^{o'}$. Now, from agent α_1's perspective, the negative reward, following its own action a_1^1 is equally likely to have arisen because either (a) the system was in state s_0, and transitioned to state s_1, when agent α_2 took its own first action, a_2^1; or (b) the system was in state s_1, and transitioned to state s_0, when agent α_2 took its second action, a_2^2.

Thus, following the state transition, observation, reward, and communication, agent α_1 will still be indifferent between all possible interpretations about how to translate any of the messages it has received from α_2. That is, from α_1's perspective, it is in exactly the same condition as when it began, and its new choice of action is again dependent upon a completely indifferent set of possible translations of the messages it has received. Furthermore, it is easy to show that this condition of persisting indifference holds no matter what action α_1 chooses, and no matter what observations, rewards, and communications are received at each time-step. Therefore, there is no point at which agent α_1 will have any more reason to choose one of its two possible actions over the other, and the expected reward of the resulting course of action is only half that of the optimum, or $\frac{1}{2}T$. □

Of course, this counter-example does not establish the strict logical necessity of the suitability property with respect to decision-theoretic solution techniques based upon probability updates, since some slightly weaker property may yet suffice. However, it does show that there are unsuitable problem instances for which meaningful probabilistic updates are impossible (later we show that suitability, when combined with free communication and a specific protocol for action and belief update, is sufficient for language learning, and for optimal action). In any case, interpreting this claim properly takes a little care. We are not claiming that in the absence of suitability, it is generally not possible for agents to learn to act optimally in a

decentralized MDP. Rather, we simply mean that if agents are basing their actions upon the probabilities assigned certain interpretations, in a straightforward utility-maximizing decision-theoretic manner, they will need to be able to make well-directed updates to those probabilities. This does not, then, militate against the application of completely different techniques to such a problem. For instance, the sample problem shown in Figure 5 can obviously be easily solved using reinforcement learning, ignoring the model of the environment entirely, and basing decisions upon observed rewards given the various different (small number) of combined rewards and observations. While we would argue that such simple and direct learning methods, which do not focus on specific and structured interpretation of messages, are of limited applicability to language learning in general, a full comparison of the advantages and possibilities of various policy-generation and learning algorithms is beyond the scope of this paper.

6 Particular Protocols and Algorithms for Language Learning

Learning to communicate while acting requires not only an update rule (the τ-learning algorithm), but also a protocol for coordinating communication, action and interpretation. Here, we present such a protocol, allowing agents in suitable and freely describable Dec-MDP-Coms to converge to optimal behavior. Following that, we describe the use of a Bayesian update method for calculating belief updates in this context, and describe how we have implemented that method in our tests and experiments (see Section 7).

Definition 23 (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*, $s^\star = J(o^\star)$. (Where J is a function from observations to global states, in accord with joint full observability, as in Definition 1.)*

(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).*

(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:

$$P_i^\sigma(o_j \mid \sigma_j^o, \beta_i^{t+1}) = P_i^o(o_j \mid o'_i)^{t+1}. \tag{12}$$

$$P_i^\sigma(a_j \mid \sigma_j^a, \beta_i^{t+1}) = P_i^a(a_j \mid o'_i)^{t+1}. \tag{13}$$

It is important to note three features of this protocol. First, 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 that already understand each other to coordinate optimally, avoiding the coordination problems Boutilier [6] sketches. Agents that are still learning the language act in the way they believe *most likely* to be coordinated.

Secondly, in the agent's new belief-state, the probability assigned in observation or action given the most recently received messages—in other words, 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 6.1 describes the use of a Bayesian Filtering algorithm to actually accomplish these sorts of updates in practice.

Finally, in general multiagent coordination, such a straightforward procedure is not necessarily optimal, nor even necessarily close to optimal. As Boutilier [6] 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.

Claim 2 showed that fully describable decentralized problems can be solved optimally when the translation updates are performed with a complete algorithm. The next claim shows that for fully describable problems which are also suitable, the elementary action protocol is sufficient to obtain optimal behavior. Intuitively, this protocol causes the agents to choose their actions in the *right* direction (because following the protocol, agents choose actions based on the *most likely* ones) and since the problem is suitable this behavior will lead to optimal results.

Claim 6. *Given an infinite time horizon, agents acting according to the elementary action protocol in a suitable and freely describable Dec-MDP-Com will converge upon a joint policy that is optimal for the states they encounter from then on.*

Proof: As agents act at some time-step t, they choose actions based always on the observations and actions of others that they consider most likely. That is,

1. Agent α_i translates the observation- and action-messages, σ_j^o and σ_j^a, for other agents α_j.
2. For each such message, α_i chooses the most likely translation, $\max_i^\sigma(o_j)^t$ or $\max_i^\sigma(a_j)^t$.
3. These interpretations are then used, along with α_i's own observation o_i to generate 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 action sub-sequence

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

Now suppose, without loss of generality, that one of these most likely observation-translations $\max_i^\sigma(o_j)^t$ is incorrect; that is, the prior observation of agent α_j is such that $o_j \neq \max_i^\sigma(o_j)^t$. Following the joint action a^+, α_i will receive its own observation o_i' and system-wide reward r'. Then, since the problem is suitable, we have that

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

(and similarly for all future time-steps). That is, the correct observation will be more likely, given the observation, than that one previously thought most likely. Furthermore, updates of messages proceed directly in accord with these probability assignments, since the protocol simply assigns

$$P_i^\sigma(o_j \mid \sigma_j^o, \beta_i^{t+1}) = P_i^o(o_j \mid o_i')^{t+1},$$

and therefore we have that

$$P_i^\sigma(o_j \mid \sigma_j^o, \beta_i^{t+1}) > P_i^\sigma(max_i^\sigma(o_j)^t \mid \sigma_j^o, \beta_i^{t+1}).$$

That is, at all future time-steps, α_i will assign the correct translation of message σ_j^o a higher probability than the prior, incorrect interpretation. Finally, once the correct translation of any message is actually 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 in a Dec-MDP-Com is finite by definition, agents will, when given enough time, choose the correct entries, since these will eventually become most probable, by process of elimination. After this point, the most likely observation and action sequences will in fact be the actual observation and action sequences for other agents, and α_i will implement a policy that is optimal from then on, since it is now acting based upon the actual states and next actions of the problem. (Note that the problem is freely describable, and so the utility of the policy from then on is not affected by the constant communication required by the protocol.) □

Usually, work on cooperation in decentralized environments has followed one of these approaches:

1. Agents can not communicate (e.g., social laws [22]).
2. Agents can communicate and understand each other correctly (e.g., SharedPlans [21], Partial Global Plans [12, 13], KQML [16]).
3. Agents can communicate and learn a language until it is completely understood (e.g., [43, 23, 40]).

Our work here enables a different perspective on cooperative multiagent systems, since in some cases decentralized systems can cooperate optimally even though the agents may not be able to interpret all messages exchanged fully and with certainty. As stated in the next corollary, cooperation may be achieved even though there may still be confusion and uncertainty about the meanings of some of the messages.

It follows from Claim 6 that at some point in the learning process of a suitable Dec-MDP-Com involving the Elementary Action Protocol, $max_i^\sigma(o_j)^t$ will become equal to o_j. This may occur when $P_i^o(max_i^\sigma(o_j)^t|o_i')^{t'+1} = \epsilon < 1$. The corresponding interpretation of o_j will be picked by agent i so long as it is still the most likely, even though it is not completely certain. Therefore, there exist cases when agents can completely understand each other even before their translations became complete. We can conclude the following.

Corollary 1. *Optimal cooperation can be viable with an ϵ-certain τ-learning algorithm, where we have $(0 < \epsilon < 1)$.*

Note that characteristics of the environment such as symmetries could also be exploited by the agents when interpreting messages in order to behave optimally. In such cases, even though the probability of interpreting a pair of messages is less than one, due to interactions between these interpretations and the actual effects of the agents' actions, their coordination can be optimal. This is left for future research.

6.1 Bayesian Filters

In order to calculate the probability updates required for our linguistic domains, we adopt the method of Bayesian filtering [14], used for example in robotics to handle localization [41]. As this method is usually employed, agents possess a set of beliefs, in the form of a probability distribution over possible states of the environment; the filtering algorithm updates this distribution over time. Updates occur under two basic circumstances. (1) An agent updates its belief function based on new *observations*: such observations may determine the state exactly, or may imply a probability distribution over the states, depending for instance on factors like noise. (2) Agents make belief updates *predictively*: before taking any action, the agent updates the probability distribution for each state in which it may end up.

In decentralized multiagent contexts without communication, however, these updates are not generally feasible, as Bayesian updating cannot be performed correctly. Recall that in Dec-MDPs, both observation and state-transition functions involve multiple distinct agents. That is, the observation function O gives the probability

$$O(o_1, \ldots, o_n \mid s, a_1, \ldots, a_n, s')$$

that each agent α_i observes o_i when actions $a_1, \ldots, a_n$ jointly cause the state transition from s to s'. Similarly, the transition function P gives the probability

$$P(s' \mid s, a_1, \ldots, a_n)$$

of moving from state s to state s', given joint action $a_1, \ldots, a_n$. On the other hand, an individual agent α_i, wanting to update its beliefs about the state of the system, needs to calculate both of these probabilities dependent solely upon its own observations and actions: $O(o_i \mid s, a_i, s')$ and $P(s' \mid s, a_i)$. Of course, if other agents chose actions according to known, probabilistic (or deterministic) strategies, these quantities could be derived using straightforward marginalization techniques. In general, however, such a presumption is invalid, and α_i can assign no meaningful prior probabilities to other actions a_j, especially in learning contexts where other agents are also adjusting their behaviors over time.[4] Therefore, unless special independence assumptions are made about the Dec-MDP in question, Bayesian filtering is not suitable for updating beliefs about system states (e.g., previous studies [25, 9] have looked at decentralized problems with independent transitions and observations). We note also that our use of this method relies in particular upon the ability to update beliefs based solely upon information from the current and immediately prior time-step; as discussed in Section 4.2.1, this is an important if not universally valid assumption.

In the context of communication, however, Bayesian filtering does in fact allow individual agents to update their belief function appropriately. If a Dec-MDP is freely and fully describable (Definitions 20, 21), then agents that already understand one another can simply share their observations at all times; joint full observability (explained in Definition 1) then makes the process of identifying global system states elementary, since each such state is determined by the collected observations of each agent.

Furthermore, even where agents do not fully understand one another, Bayesian filtering provides a method of updating *translations* over time. Figure 6 gives the **Bayes-Filter** algorithm, as used in our work on decentralized communication learning. The algorithm is presented for a pair of agents; for n agents, the basic structure of the technique is the same, although it is more complicated to present notationally and schematically. As shown in the pseudocode, the algorithm is used to update agent α_i's belief-state distribution over translations, $P_{\tau i}(\tau)$, given a

[4] For an approach in which agents can generate such priors by modelling other agents, see [19].

Bayes-Filter$(P_{\tau i}(\tau),\, \sigma_j,\, \mathbf{d})\{$
$NormFactor = 0$
If $\mathbf{d}$ is an observation o_i then
    For all translations τ do
        $P'_{\tau i}(\tau) = P(o_i \,|\, \tau,\, \sigma_j) P_{\tau i}(\tau)$
        $NormFactor = NormFactor + P'_{\tau i}(\tau)$
    For all translations τ do
        $P'_{\tau i}(\tau) = P'_{\tau i}(\tau) NormFactor^{-1}$
Else if $\mathbf{d}$ is an action a_i then
    For all translations τ do
        $P'_{\tau i}(\tau) = \sum_{\tau'} P(\tau \,|\, \tau',\, \sigma_j,\, a_i) P_{\tau i}(\tau')$
Return $P'_{\tau i}(\tau)$ $\}$

Figure 6: Bayes-Filter Algorithm

received message σ_j, and data $\mathbf{d}$, consisting of one of α_i's own observations o_i or own actions a_i. In the first case, where the algorithm receives a local observation o_i, it updates the distribution over translations based on the prior probability of that observation given each possible existing translation, normalizing as appropriate. In the second case, when an agent is about to take an action a_i, it updates the belief distribution by projecting the probabilities of possible next translations, based upon the existing ones, along with the message received and action to be taken. For details of how the basic algorithm can be implemented to deal with actual problem instances, see Section 6.3.

6.2 Bayesian Filtering and the Elementary Action Protocol

To perform the calculations described in the update algorithm, agents require two sets of prior probabilities:

(1) A **sensor model**, giving $P(o_i \,|\, \tau,\, \sigma_j)$, the probability of observation o_i, given any translation τ and message σ_j. This probability is used in the first, observation-update step of the filter algorithm, when calculating the probability distribution over translations, based upon observations and messages received.
(2) An **action model**, giving $P(\tau \,|\, \tau',\, \sigma_j,\, a_i)$, the probability of translation τ, following action a_i, and given translation τ' and message σ_j. This probability is used in the second update step, when predicting the next translation distribution based upon what is known now, and what action is intended.

The possibility of giving such models for a general Dec-MDP-Com clearly depends upon the content of the messages σ_j that are sent between agents. In particular, agents will generally need to share information about their own observations and actions in order to perform the correct updates.

We can understand these requirements better by focussing upon the Elementary Action Protocol (EAP), as given in Definition 23, concentrating upon the two-agent case for ease of explication. We note that in the EAP, agents swap messages at each step about their current observations and intended actions. Furthermore, an agent's translation of these messages induces probability distributions over those observations and actions of others. Thus, when agents swap information about their own local observations in the first step of the EAP, they

can update their belief-states about translations using the first part of the Bayesian filtering algorithm, based upon these observations, what is known about actions at the prior time-step, and the (marginalized) prior probabilities given by the Dec-MDP-Com model. Following this step, agent α_i can perform the second step of the EAP, calculating the most likely observation for α_j given the updated new translation, and the most likely global state of the environment based upon that information, in combination with its own local observation.

In the third step of the EAP, agents exchange their action-messages in order, using their current translations to choose most likely interpretations of those messages, and choosing actions that are optimal relative to those interpretations. Again, the current translation belief-state induces a probability distribution over the actions of the other agent, and so α_i can perform the second part of the Bayesian filtering algorithm, making its predictive update given the action it chooses. Following this update step, agents act jointly, a state transition occurs, and the process repeats, with new observations shared, and new Bayesian updates are performed.

6.3 An Example

One possible representation of a translation τ is a two dimensional table where each row in agent τ_i's translation table corresponds to some atomic component of the agent's own language Σ_i. Columns correspond to atomic components identified in messages received in language Σ_j. Entry (σ_i, σ_j) gives the probability that σ_j has the same meaning as σ_i. The overall structure of the table corresponds to bounds on the presumed structure of messages received, based on the assumption that cooperating agents communicate things related to the immediate task. This radically constrains the range of possible interpretations, in order to make communication generally feasible; such constraints correspond to the use of such considerations as relevance and context in real-world communication, in order to make interpretation easier, or even possible.

Probabilities then play two important roles here. First, the filtering algorithm assigns probabilities to various possible translation tables, taken as part of the local state of the agent. Second, agents choose actions based upon the probable meaning of recent messages, calculated based on the overall probability that some translation is correct, and the individual probabilities contained in the corresponding table. Language learning interleaves interpretation with action, in a joint process designed to narrow down possible translations, while still acting on current ones, however uncertain they may be.

To cope with particular features of this task, the basic filtering algorithm must be modified in two main ways. First, agents do not enumerate and update *all possible* beliefs: since belief-states themselves contain combinations of continuous probability distributions, there will be infinitely many available at any time. Instead, only those states necessary are generated and updated at each step; for many applications, the procedure is straightforward, and is sound so long as states not generated are properly taken to occur with zero probability. Second, the set of possible belief-states changes over time: since agents do not generally know all the elements of the language to be translated in advance, newly encountered components must be added to the translation-tables along the way.

As an example, consider a simple gridworld problem and a language of communication given by the agents' observations ($\Sigma = \Omega$): the environment is a 2×2 grid; to identify locations, agents use unambiguous proper names, meaning (1) each square in the grid has exactly one name, and (2) each name identifies exactly one square. Suppose agent α uses the integers $\{1, 2, 3, 4\}$ to name the four squares; the goal is to find a mapping between these names and those used by another agent that uses as names the letters $\{A, B, C, D\}$. Agents communicate grid locations using their own naming conventions; the other agent receives the message, attempts to interpret it, and proceeds to the most likely square. An observation follows, indicating whether or not

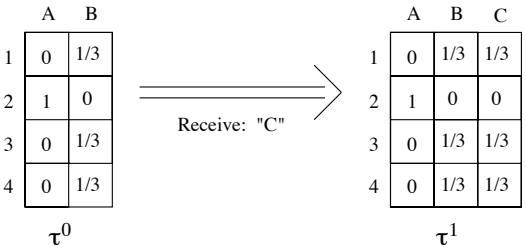


Figure 7: A new message is added.

the agent has successfully identified the correct location.

Figure 7 shows how α adds newly received messages into its translation table. We assume that α begins in the sole belief-state τ^0 (i.e., α assigns translation τ^0 unit probability). We see that α has previously received two messages, "A" and "B"; further, α has successfully translated the first of these, since $\tau^0(A, 2) = P(A \text{ means } 2) = 1$. Two more features of the table stand out. First, each column sums to 1; α knows the components of its own language, and knows that the distribution for any message must sum properly. Second, rows *do not* sum to 1; before α has seen all of the words in the other language, it cannot know how to fill out each row of its translation table. (Of course, in a simple example like this, α might be able to reason out that there were only four possible messages, and fill out the table with four entries per row, but we do not generally presume this to be the case.)

Suppose α now receives a new message, "C"; the belief-state is then updated from τ^0 to τ^1. A new column is added to the table: since α already knows that the name "2" corresponds to message "A", "C" receives a zero probability for this entry, and all remaining entries are uniformly distributed, reflecting the proper-name model of the language. Since "2" is translated as "A", it will not be translated as any other name, and so the 0 is inserted in the table; further, α has no reason to think "C" any more likely to name one remaining square in the grid than another, and so the remaining probability mass is distributed uniformly. Of course, for ambiguous languages, or ones in which locations can have more than one name, the first assumption would not hold. As well, in some cases an agent may have reason to favor one translation over another in advance, and probabilities need not be uniformly distributed.

Having expanded the table, α now chooses an action. Since the probability that "C" means "1", "3", or "4" is the same, it is indifferent which location it visits next. Of course, this need not be true; in general, the choice of an action is computed based on the most likely translation for the current message, weighted by the probability assigned to each possible translation table. Assume that α chooses "1" as the most likely translation: Figure 8 shows the *action model* update of α's belief function (the "else" clause of the filtering algorithm in Figure 6) before going to the square named by "1". We assume action outcomes are deterministic: after taking action $Go(1)$, α is sure to end up at the desired square. Furthermore, the observation sensed at the end of the action determines precisely whether or not α is correct in its translation (i.e., α perceives a particular observation if and only if it has correctly identified the chosen square). Thus, the update process replaces τ^1 with two new possible translations, τ^2 and τ^3.

Belief-state τ^2 reflects the outcome that the translation (C means 1) is *correct*. In this case, the entry $(1, C)$ in the table is set to unit probability, and all other entries in the row and column are set to 0. Again, this is a reflection of the unambiguous nature of the name language; more complex update scenarios are possible. Note also that the table in τ^2 also updates the probabilities contained in column "B"; since we normalize all columns, the probability mass for entry $(1, B)$ is distributed over the remaining possibilities. Belief-state τ^3, for its part, reflects

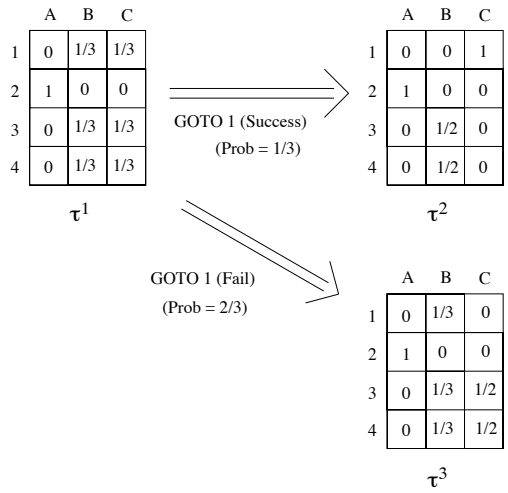


Figure 8: The two translations believed possible after $Go(1)$.

the outcome that the translation (C means 1) is *incorrect*. Here, the only column affected is "C"; the entry $(1, C)$ is set to 0, and its probability mass distributed over the remaining entries in that column. Lastly, α assigns predictive probabilities to these two new belief-states before it takes action $Go(1)$, based on what the outcome of that action can tell us about the chosen translation. In this case, $P(\tau^2|\tau^1, Go(1))$ is simply the original probability, 1/3, taken from table τ^1, that translation (C means 1) is correct. Similarly, the probability of the incorrect translation is $P(\tau^3|\tau^1, Go(1)) = (1-1/3) = 2/3$.

For this environment, the *sensor model* is also simple: α either senses appropriate observation for successfully translating "C" or not. We will use the notation OR when we refer to such an observation to distinguish from the usual observation of the environment. If α does receive OR, belief-state τ^2 is clearly correct and post-observation probability $P(\tau^2|OR) = 1$, while $P(\tau^3|OR) = 0$ and the latter belief-state can be discarded. If α does not receive OR, then the opposite holds. In either case, α is left with a single belief-state on which to base its next actions. If that message is one of "A", "B", or "C", then an action chosen based upon the existing probabilities, and the procedure shown in Figure 8 is repeated. If it is some new message, "D", then that message is added to the table of translations as in Figure 7, and the entire procedure repeats itself. Again, we stress that for more complex cases, the relevant updates can be more complicated. In particular, it need not be the case that α is left with but one translation at the end of the process of action and observation, since observed rewards may not determine single states, and actions may not have determinate outcomes. Still, the basic principles remain the same.

7 Experimental Results

We first present results from running experiments on two different scenarios with increasing complexity. Then, we discuss the possibility of extensions involving non-deterministic actions and multiple contexts of interpretation. The first scenario (first described in [18]) 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 gridworld 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 location whether or not the other agent was in fact there, the probability that the other observed that location is either 0 or 1, and the suitability of the Dec-MDP-Com follows immediately. Messages are chosen from the agents' observations, and were tested with different levels of structure. The local policies of action were assigned based on a locally goal-oriented mechanism [26]. That is, each agent acted towards a local goal that was computed as a function of the agent's own local observation and the interpretation of the message received.

In the second domain (first described in [1]), we test a network of pumps where agents can control the flow through different ducts. The objective is to maximize the outgoing flow while minimizing the usage of ducts. In this case the messages can be about both observations and actions. The local policies of actions are given to the agents based on a prior computation of the centralized solution assuming one mutual language. Translations are again updated based on the Bayes filtering algorithm (the agents collect all their messages, both observations and actions, and then do the same steps as in the algorithm). In both cases, the process is Markovian and therefore belief updates are based only on the last translation and last belief-state.

We note one important fact about these examples, before considering the details. In both sets of experiments, agents worked on updating translations until such a time as they had completely translated one another's languages. However, in neither case is the agent directly rewarded for a correct or complete interpretation. That is to say, while learning language facilitated the optimal solution of the relevant tasks, it was not in and of itself the main motive force. In the grid-location task, for instance, our agents happened to visit all squares of the grid with equal frequency, and so deriving a reward involved learning the vocabulary covering the entire grid. Nothing would have been changed, however, if one or both of the agents simply did not visit certain parts of the grid at all. In such a case, agents would work only on translating messages they actually received; portions of the vocabulary that covered unvisited grid-regions would not have been translated effectively, but this would make no difference to overall reward earned. Furthermore, as we show in consideration of the pumps-world domain, the average rate of accumulated reward is very nearly maximized long before the entire language of each agent is entirely translated, suggesting a role for approximate methods that forego complete translation where it no longer brings appreciably greater reward (see the discussion centered around Figure 14). In each case, then, agents are driven directly by the rewards in their environment, not by some special class of rewards specifically tied to the form of their translations—so far as language learning allows them to accumulate more such reward, they update their translations, and otherwise, they do not.

7.1 Gathering example

We consider a simple observation structure that can be interpreted as the values the agents perceive after reaching a location assumed to be their local goal. Each agent knows its optimal local policy composed of deterministic actions: for every possible local goal, each agent knows with certainty the observation it should receive if it correctly interpreted the message.

Our initial experimental results demonstrate the basic feasibility of the given approach to the language-learning problem, and illustrate how performance is affected by various features, such as the structure of the observations or of the language itself. We ran a number of tests involving the basic cooperative task for two agents in a simple gridworld environment, averaging results over one hundred runs with random starting locations. Work proceeds in turns; on even-numbered rounds:
(1) Agent 1 is the **actor**, randomly choosing a square and sending a message to agent 2 to that

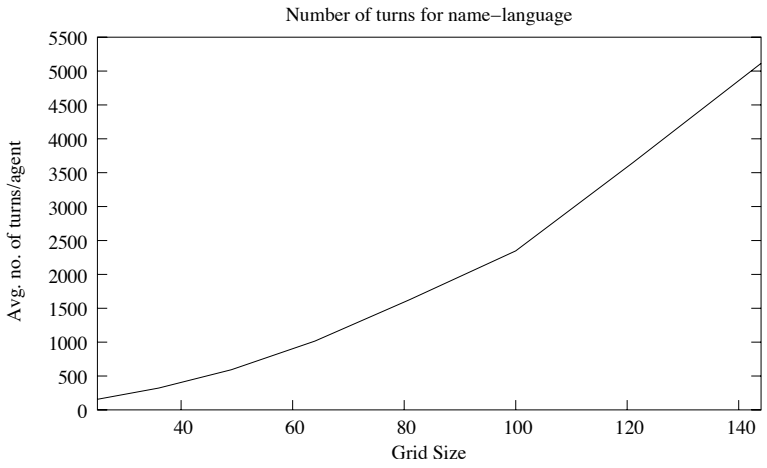


Figure 9: Results for simple-name language.

effect.[5]

(2) Agent 2 is the **translator**, choosing a square based on its current translation of agent 1's language.

(3) Agent 2 receives an observation, OR, depending upon its choice, and updates its beliefs accordingly.

On odd numbered-rounds, agents 1 and 2 switch roles; the task continues until one agent has successfully translated the other's language. Currently, this involves one agent assigning unit probability to a *complete* translation table, i.e., every row contains exactly one entry with probability 1 (and the rest 0). Obviously, this could be modified to involve other, perhaps more tolerant, criteria of success.

In step (2) of the process, the choice of a square is based on the current message and belief-state of the translator. This belief-state is a probability distribution over translation tables; for each such table τ_i, let P_{τ_i} be the probability that τ_i is the correct translation. Each table τ_i maps components of received messages to possible meanings; for any complete received message σ_j (in the actor's language) and meaning σ_k (in the translator's own language), let $\tau_i(\sigma_k, \sigma_j)$ be the probability computed from table τ_i that (σ_j means σ_k). The most likely meaning of the message σ_j is thus calculated as that σ_k satisfying: $\max_k \sum_i P_{\tau_i} \cdot \tau_i(\sigma_k, \sigma_j)$. The various possible tables τ_i are chosen based on the presumed structure of the two given languages; the probabilities P_{τ_i} are updated using the Bayes-Filter algorithm.

Our first study involves a simple language of *unambiguous names* (as in Section 6.3). That is, each agent assigns to each square in the grid a *fixed*, *unambiguous* name, meaning that each square has but one name, and each name corresponds to but one square. Each agent attempts to learn the other's mapping from names to squares. Observations are simple: translators sense 0 as the value of OR for correctly identifying the square chosen, and otherwise the observation's value is 1; translations are updated in either case. Figure 9 charts the average number of turns as translator an agent must take to arrive at a complete translation for this language. The ratio of grid size to number of turns is relatively stable: for grid size $G \times G$, the number of turns is roughly $G^2/4$. We can see that agents pursue a decision-theoretically optimal course of action here. Translators initially assign novel messages a uniformly distributed probability of naming

[5] Our ongoing research investigates cases in which agents can choose particular messages non-randomly, to guide the learning process. In particular, if the agents already learned some part of the language, then it is reasonable that they will choose locations corresponding to the remaining messages instead of choosing uniformly.

Language	Turns	Time (s)	Max. Beliefs
Names	156	0.5	1
Coordinates	27	119.3	13812

Table 1: Two languages on a 25×25 grid.

squares for which the name is not already known; further, after visiting any square, the translator knows the exact (0/1) probability that the latest message names it, based on the observation sensed. Once some message σ_i is known to correspond to the name of square x, the translator always visits x upon receiving σ_i. Conversely, the translator never visits x once it knows that current message σ_j cannot name it. Finally, for messages not yet translated with certainty, the most probable translation is chosen (breaking ties randomly). Since the translation-action strategy is thus optimal with respect to expected reward, it will not in general be possible to do better in terms of average number of attempts before achieving a complete translation. That the algorithm does not converge a little faster is explained by the fact that the observation information gained by the translating agent is not shared with the acting agent; the actor may thus randomly select a square multiple times, even after the translator already knows how to translate the name given that square.

Such a simple example shows the elementary feasibility of the filtering approach, but we are also interested in testing more complicated languages and observation structures. Within the same gridworld context, we also investigate messages in a language of (x, y)-coordinate pairs. Due to this structure, translation updates carry more information than when simple names are concerned. However, the number of possible translations may increase drastically after each update. In the name-language, the observation perceived after any action determined a single possible belief-state for the translator: either the translation of some name is known with certainty, or it is absolutely certain that one particular translation is incorrect. In a language of coordinate pairs, with a basic success/failure observation structure, things are not so simple. If selection of some square is successful, of course, the translation of some message (σ_x, σ_y) is known with certainty, and any other possibilities for that pair are eliminated. If selection is unsuccessful, however, there are three options: either both translations of σ_x and σ_y are incorrect, or only one of them is. In the worst case, then, the essentially uninformative (0/1) observation structure can cause a large increase in the number of belief-states an agent has to entertain, affecting overall performance. Indeed, this increase may be so great that while translation uses a smaller number of *turns* to achieve certainty, overall *time* spent is much worse, as agents have to update many more beliefs in any given turn (see Table 1). In general the problem becomes potentially intractable for the combination of the coordinate-language and an essentially uninformative observation function.

To improve the situation, we consider other possible observation structures, which give the translator more information about what may have gone wrong. These are: OR **0/1/2**: agents observe a 0 if translation is correct, 1 if translation fails but one coordinate is translated correctly, and 2 if failure results because both coordinates are translated incorrectly; OR **0/1/2/3**: agents receive 0 if translation is correct, 1 if it fails but σ_x is translated correctly, 2 if it fails but σ_y is translated correctly, and 3 if failure results because both are translated incorrectly. The more informative observations greatly improve performance in terms of all problem dimensions: number of turns before translation is successful, maximum number of beliefs that an agent must entertain at any time, and overall time of completion.

Figure 10 compares results for the name-language and for the different variations of the coordinate-language, in the context of a 25×25 grid. Scale here is logarithmic, with values

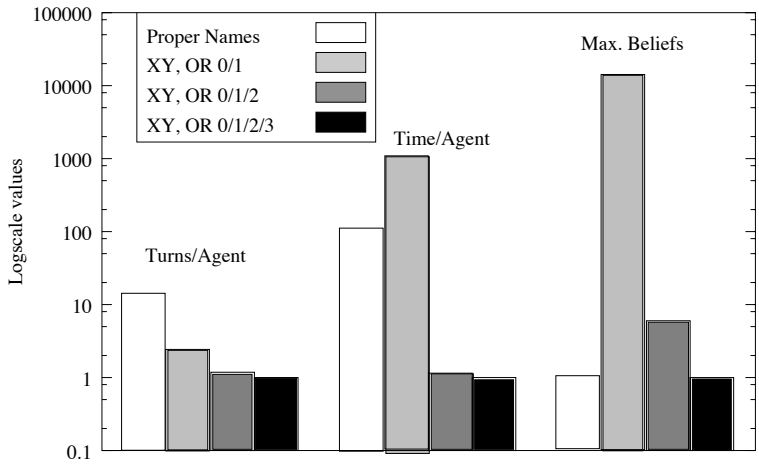


Figure 10: Different languages on a 25×25 grid.

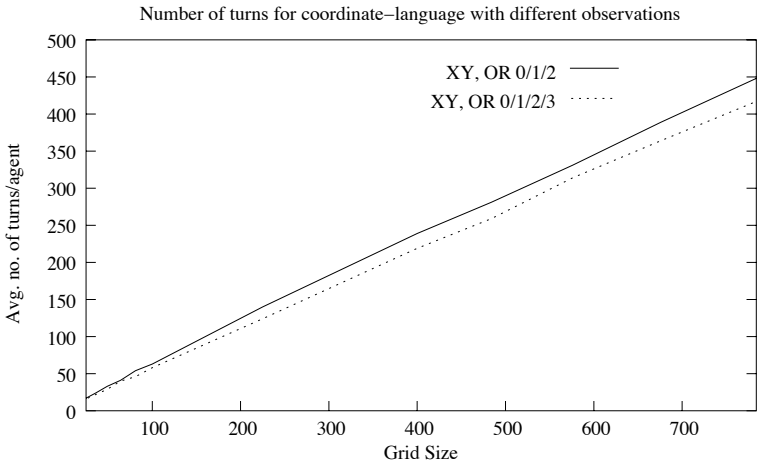


Figure 11: Comparing observation functions.

normalized to the OR 0/1/2/3 case; results average over 100 random runs. In general, the extra information provided by the coordinate-language as opposed to the name-language significantly reduces the number of turns taken. Further, with respect to the coordinate-language alone, the more informative observation functions lead to significantly better results in terms of maximum number of belief-states ever updated in one pass of the filtering algorithm, and thus in terms of overall time taken. Such relations are not absolute, however. For instance, although the maximum number of beliefs is somewhat larger for the OR 0/1/2 case than for the name-language, the increase in information gained by using coordinates still decreases the number of turns enough to lead to a decrease in overall time taken in the former case. In the single case shown, and in general, the performance of the two cases with more structured observations was essentially the same; while the 0/1/2/3 structure led to slightly improved performance, the differences were not significant (see Figure 11). Finally, Figure 12 shows the increase in the number of turns taken as the grid grows in size for both the name-language and for the coordinate-language with 0/1/2 observation structure. We did not test the name-language for very large grids, since the time taken quickly became unmanageable; in any case, the difference is evident for the limited range considered.

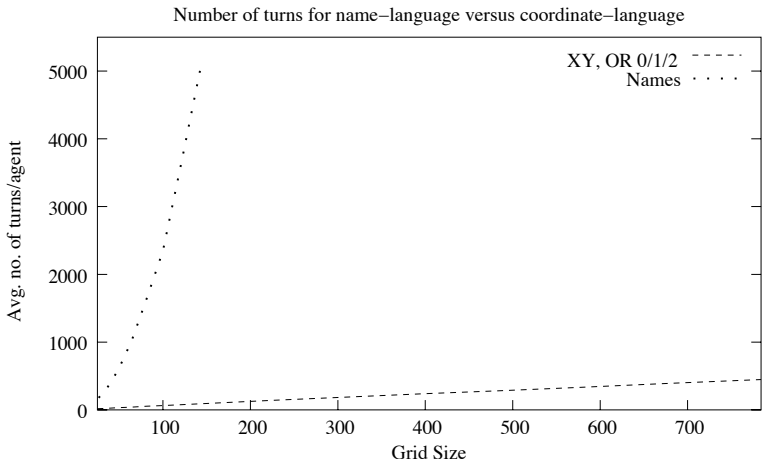


Figure 12: Names vs. coordinates.

7.2 A Network of Pumps Example

To explore the viability of our approach in more realistic settings, 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; further, agents observe the system-wide reward.

(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 methods.

Further, the environment is in fact an example of a suitable Dec-MDP-Com. The problem is both freely describable, by the cost function C_Σ, and (for the purposes of solving the problem) fully describable, as given by the set of messages Σ. 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 observed 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 and communicate actions based on their current beliefs, and then act, repeating the process to converge towards mutual understanding and optimal action. Thus, these experiments expand upon those of the previous section, both by including the language of actions where before agents could only speak about state observations, and by extending the method to a appreciably more complicated domain. Using their model of the environment, agents update belief-states using the two-step Bayesian Filtering algorithm as before, first projecting possible belief-states before acting, then updating those beliefs given the results of their actions (see Section 6.1). Note that there is no update between the two sets of observation- and action-messages. Rather, the agents collect all their messages, then do the steps of the algorithm as before. That is, they first do the predictive update, generating possible next belief-states before taking the action; then they act, and do the retroactive update after the next observation comes in. Actions are chosen based on the translation as it exists following the observation on the prior action step.

Agents interact until each learns the language of the other with certainty—achieved when each agent α_i reaches a belief-state β_i with distribution $P_{\tau i}$, and for any message σ_j received from the other agent, there exists exactly one message σ_i such that $P_{\tau i}(\sigma_i, \sigma_j) = 1$. 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.

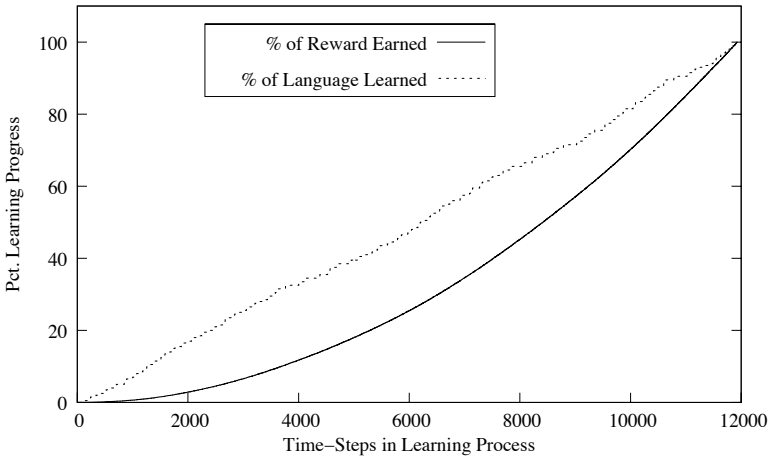


Figure 13: Reward accumulated as language is learned.

Our results show that the elementary protocol converges to optimal policies in each problem instance. 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 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 13 gives one example, for a problem 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 a problem of this size, 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 generally 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 of the 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.

Figure 14 plots normalized average rate of reward accumulation as a function of time. As can be clearly seen, by far the largest proportion of the overall reward is accumulated relatively early in the learning process. Such performance profiles reveal the possibility of employing approximate methods, in which we may decide to stop the learning process early without sacrificing much in overall value. Such approximations are open topics for future research.

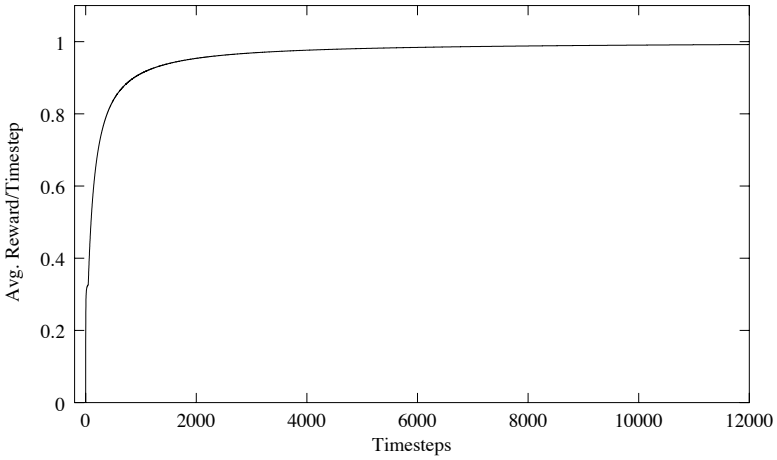


Figure 14: Normalized average rate of accumulated reward.

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-learning problem is somewhat new. Note also, that giving upper and lower baselines for these results is straightforward, but uninteresting. The lower bound on performance is given by agents who do not communicate at all. In such cases, since we are not working on solving the problem of learning how to act at the same time as learning how to translate, there is nothing more for the agents to learn. Given their model of initial starting conditions, they will simply choose actions maximizing expected value, and performance will be constant. (For instance, in the gridworld problems, agents will simply pick a square at random, guessing where the other agent is; for a grid of n squares in total, the expected average reward over time is then the constant function n^{-2}.) While it would be interesting to extend our work to cases in which agents still had learning to do even in the absence of learning about language, this would generally make the problem of translation updates prohibitively difficult; therefore, we leave that aside in the current work. Similar considerations apply to the case of upper bounds, given by agents who already share a language of communication. In such cases, as we have shown, the fully and freely describable nature of the problems makes generating optimal solutions straightforward, and agents will simply accumulate maximum expected average reward at all times. Again the plot of this reward will be a simple constant function. (For instance, the reward plotted for the pump-world case as shown in Figure 14 would simply be the unit line, which the learning case already shown would converge towards.) Again, if we added learning how to act into our problems, this would be more interesting, but for now it is not.

7.3 Learning to Communicate With Non-deterministic Actions

Here we note that the assumption that actions are deterministic, as in the gridworld problems, is of little consequence to the language-learning problem, and is simply for convenience. So long as agents can fully observe their local states following any action, the presence of stochastic actions does not fundamentally influence the applicability of the given methods. Suppose, for instance, that in the gathering example just described the action $Go(1)$ might in fact lead the agent to either squares 1 (with 70% probability) or 2 (30%). Then, the action model update step would simply produce two distinct sets of belief-states, $\{\tau_2^1, \tau_3^1\}$ and $\{\tau_2^2, \tau_3^2\}$, corresponding to the possible outcomes for the translations (C means 1) and (C means 2) respectively, with

each set weighted by the corresponding probability. In a fully observable environment, however, once the agent took the action $Go(1)$, and observed whether or not it was in either square 1 or 2, one of these sets would simply be discarded (i.e., its probability set to 0), and all else would proceed as usual. In the context of our empirical studies in the pump-world domain, the addition of stochastic actions therefore added nothing of import. Runs involving such actions suffered no notable effects under any of our measures. Of course, in environments in which an agent's state is only partially observable, the presence of stochastic actions can lead to quite different results; however, such environments generally make both online learning and offline solution techniques intractable even in the presence of deterministic actions [4], and lie outside the scope of our present research.

7.4 Switching Between Contexts

So far, we have assumed that agents share a single context. That is, agents each begin with a single model of the probabilistic structure of the other's language, based upon the fixed shared task at hand, and have been able to learn how to interpret messages within that model. A possible extension to our approach is to assume that there may be several contexts, reflecting different models of what the other agent is attempting to communicate. One way such a case could be handled would be to run the filtering algorithm in a sequential manner. That is, agents begin using one model of the relevant probabilities, and switch to the next context if the previous one is recognized to be invalid. Finding the conditions upon which a context switch must occur depends on the knowledge we have on the characteristics of the languages implemented by the agent. For example, assuming that the agents' languages are non-ambiguous and of the same size, a context switch must occur if either of the following two conditions becomes true. Assuming that $\sigma_j \in \Sigma_j$ are the messages sent by the actor, and $\sigma_i \in \Sigma_i$ are the messages in the translator's own language, we switch contexts if:

1. $\exists \sigma_{j_1}, \sigma_{j_2}, \sigma_i \; s.t. \; \tau_i(\sigma_i, \sigma_{j_1}) = \tau_i(\sigma_i, \sigma_{j_2}) = 1$. That is, two different messages in the actor's language are translated to the same message in i's language. This invalidates the fact that the languages are non-ambiguous.

2. $\exists \sigma_j \; \forall \sigma_i, \; \tau_i(\sigma_i, \sigma_j) = 0$. That is, there is a message in the actor's language that cannot be translated to any message in i's language. When languages have the same number of messages and these are non-ambiguous, this condition necessarily invalidates the current context being tested.

Further exploration is needed of the possibilities for this sort of multiple-context learning.

8 Related Work

Our approach to the problem of learning to communicate is to integrate this learning process into the decentralized control process which contains it. In such a framework, agents learn to interpret messages while acting towards some global objective. We are interested in a learning process that will improve the agents' interpretations of each other's messages, leading consequently to optimal choices of action. This is a different approach from the reinforcement learning approach taken by Yanco [45, 46], where a reinforcement signal is given explicitly for the interpretation of the messages. Moreover, her communication learning occurs in a stateless environment. We reinforce the agents only for their actions, which implicitly are chosen according to the agents' interpretation of the messages received. Our approach is thus more general and it can handle more complex language structures. We have also shown necessary and sufficient conditions under which learning to communicate can lead to optimal behavior. Yanco's

work does not provide any finite convergence proof. Her tests, based on the interval-estimation algorithm [27] consider the frequency of the positive reinforcement obtained for messages and the actions taken as a consequence.

Robotic communication is also studied by Balch and Arkin [3]. Their approach is inspired by biological models and refers to specific tasks such as foraging, consumption and grazing. The agents are assumed to share the meanings of the messages exchanged. Their empirical study was performed in the context of reactive systems and communication was free. Their aim was not to study the language learning problem as we do here.

Studies on the evolution of human language [5] have been recently pursued using computational models.[6] Komarova and Niyogi [28] study in particular how one language can be affected by other languages evolving in the same population. They also consider languages as probabilistic associations between form (the message structure) and meaning (i.e., the mapping from a message to an observation that expresses some feature of the environment). Although agents are embedded in some world, the process of learning to communicate is not related directly to the agents' actions. In our work, on the contrary, this is fundamental, as agents learn to interpret each others' messages with respect to goals or some other coordinated behavior they aim at achieving. Our approach is an engineering-based approach. Its purpose is to enable systems, composed of agents programmed to speak in different languages, to achieve coordination considering the context of their planning.

At the intersection between cognitive science and computational simulations, research has studied the evolution of lexicons and the problem of learning a language from positive examples [42, 17]. Here, we study how agents can learn to interpret each other's messages in order to improve global performance. Adaptive language games [40] and the anchoring problem ([10] and the cites therein) are also relevant areas of study. In particular, the latter work raises questions concerning formal models for the study of the language-learning problem. We present such a formal framework and a solution to the problem, formalizing the communication-learning problem as one of decentralized control. Our previous research has focused on the computation of optimal joint policies when a shared language of communication is assumed (in particular, *when* and *what* agents should communicate) [25]. Here, agents need to learn such a language in order to optimize their joint behavior.

The distributed artificial intelligence community has studied how autonomous agents can coordinate their actions while acting in the same environment (e.g., see work by [15, 21, 38]). A known and fixed language of communication was assumed when communication was allowed. KQML [16] is an example of one standard designed to pre-set the possible communication between the agents. We believe that robust decentralized systems require that agents adapt their communication language when new situations arise or when miscoordination occurs possibly due to misunderstandings. Such miscoordination can either be revealed in practice, or in simulation, and serves as a signal for reinterpretation of messages received.

Other work has proposed that rational and self-interested agents can negotiate to evolve a shared communication language [23]. In such a context, conflicts between these agents arise because each one prefers a distinct communication language, based on the cost of employing that language and the local utility it may yield. We are interested instead in communication that enables efficient coordination of agents towards a mutual goal. Communication serves to increase the overall utility of the system as a whole; the particular language learned will thus be directly related to this system-wide utility, rather than to the individual cost of using that language.

Importantly, our work relies on the idea that agents treat communicated messages as having some sort of *meaningful structure*. Initially, agents presume that others involved in a shared

[6] Evolution of communication among machines was also studied with an ALife approach, see for example [30]

cooperative task are communicating information relevant to that task. This has the effect of reducing the number of possible interpretations an agent has to consider, making the learning process more manageable. As well, treating messages as having meaningful structure speeds learning and allows for generalizations between various environments. Our empirical results confirm the advantages and complications of this approach. Treating messages as having some semantic structure can allow agents to learn their meanings more quickly; at the same time, the specification of this structure and the learning updates related to it can become more difficult. The concentration on semantics further distinguishes our approach from such prior work as [43], in which a generalization of the perceptron algorithm was proposed to allow a multiagent system to collectively learn a single shared concept. They show the convergence of a perceptron algorithm adapted to mutual learning of a concept by multiple agents. This process lacks any semantics, which we do consider to optimize the joint planning of the agents. We study the language-learning problem as part of the control process the agents are involved in. The agents can improve their utilities if they learn to communicate in a way that the messages are understood appropriately.

Game theorists [29, 2] have also looked at communication between players, although the focus is not usually on learning the communication language. For example, Wärneryd [44], and Blume and Sobel [7] study how the receiver of a message may alter its actions in games where only one agent can send a single message at no cost.

Finally, philosophically, this work stems from thought originating with the American philosophers Quine, Davidson, and Putnam. Quine [35] argues that interpreting speakers of foreign languages is essentially the same as interpreting speakers of our own. On this view, we are always constructing "translation manuals" between one another's utterances, and we understand others by relating what they say and do to what we ourselves would say and do in the context of our shared environment. Further, translation is always under-determined by available evidence, and there are often multiple competing possible translations of another's language. Davidson [11] and Putnam [34] extend these ideas. Davidson has written on what he calls "radical interpretation," exploring the idea that all understanding of others is a fundamentally indeterminate procedure of making and adjusting predictions about their behavior, based on very basic assumptions about the structure of their beliefs and intentions. Again, this process is essentially the same whether or not we share a common language with the one being interpreted. We update our interpretation of other agents constantly, based upon our success or failure in predicting how they will behave, hoping at best to converge on some generally successful set of expectations regarding that behavior. Putnam's famous essay, "The Meaning of Meaning" [34], examines, among other things, how changing context can alter the meaning of even apparently well-understood terms in a language. Together, these ideas suggest that designers of communicative agents must allow that even well-defined protocols and languages can lead to cases in which the interpretation of messages become ambiguous or error-laden, and must be adjusted in order to make coordination possible.

9 Conclusions and Future Work

The design of genuinely autonomous agents requires that those agents have the ability to learn how to interpret each other's messages, and consequently act adaptively. Multiagent systems can be made more robust if they can autonomously overcome problems of miscoordination, arising when they encounter new situations or receive messages that are not completely understood. This paper presents and analyzes a formal framework in which agents learn to communicate while they are acting to maximize some global objective. Specifically, we have combined the problem of learning to communicate with that of maximizing overall system utility in a de-

centralized decision problem. We have explained how agents can, in general, improve their coordination while improving their interpretation of messages exchanged in languages that they do not initially share. This model involves the notion of a translation between languages, and solves the problem of learning to communicate by adjusting these translations using probabilistic updating schemes. We lay the groundwork for further investigation by presenting and analyzing a formal decision-theoretic model of the problem, and by initiating empirical studies into algorithmic methods for solving it.

As shown, agents employing such update schemes can achieve value-maximizing policies so long as the decision problems they are working with have other important features. Prior work on decentralized Markov decision processes and communication has focused on messages that are composed simply of basic elements of the model, such as observations and actions. We have shown that even for such simple languages elements, the process of learning to interpret these messages in a desirable manner can become very difficult if our problem instances do not have the right structural properties. These features, such as free and full describability, along with our defined notion of suitability, are both necessary to, and sufficient for, the feasibility and success of the updates and learning methods we present, at least for the purposes of learning to communicate in languages composed of agents' observations and actions. We are interested in further research into more complex languages, allowing agents to convey other forms of information about the problems they are attempting to solve. While such richer languages may allow optimal policies to be enacted in a more efficient manner, they present additional difficulties for the learning process. We would like to investigate the possible benefits arising from more complex languages, and look at what new features of problem instances may be necessary if communication in such languages is to be learned.

Finally, our work so far has involved translation updates which converge to certainty, and lead to action policies that are optimal for all points encountered from the point of convergence onwards. This raises interesting questions. In particular, we are interested in the possibility of optimal behavior even in the absence of certainty about translation. Similarly, we are interested in the possibility of near-optimal behavior, and the trade-offs between optimality and certainty in the translation process. Our experimental results suggest that in many problems it will be possible to achieve very nearly optimal results without certainty about large portions of the language of communication. We are therefore interested in investigating approximate approaches to the problem of learning to communicate.

References

[1] M. Allen, C. V. Goldman, and S. Zilberstein, "Learning to communicate in decentralized systems," in *Proc. Workshop on Multiagent Learning, Twentieth Natl. Conf. on Artificial Intelligence* (AAAI Tech. Report WS-05-09), Pittsburgh, PA, 2005, pp. 1–8.

[2] R. J. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, vol. 2, Elsevier: North Holland, 1994.

[3] T. Balch and R. C. Arkin, "Communication in reactive multiagent robotic systems," *Autonomous Robots*, vol. 1 pp. 1—25, 1994.

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

[5] Y. Bhattacharjee, "From heofonum to heavens," *Science*, vol. 303 pp. 1326–1328, 2004.

[6] C. Boutilier, "Sequential optimality and coordination in multiagent systems," in *Proc. Sixteenth Intl. Joint Conf. on Artificial Intelligence*, Stockholm, Sweden, 1999, pp. 478–485.

[7] A. Blume and J. Sobel, "Communication-proof equilibria in cheap-talk games," *Journal of Economic Theory*, vol. 65 pp. 359—382, 1995.

[8] A. Bonarini and P. Sassaroli, "Opportunistic multimodel diagnosis with imperfect models," *Information Sciences*, vol. 103(1–4) pp. 161–185, 1997.

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

[10] S. Coradeschi and A. Saffiotti, "An introduction to the anchoring problem," *Robotics and Autonomous Systems*, vol. 43(2-3) pp. 85–96, 2003.

[11] D. Davidson, *Inquiries into Truth and Interpretation*, Oxford University Press: Oxford, England, 1984.

[12] E. H. Durfee and V. R. Lesser, "Using partial global plans to coordinate distributed problem solvers," in A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann Publishers, Inc.: San Mateo, California, 1988, pp. 285–293.

[13] K. S. Decker and V. R. Lesser, "Generalizing the partial global planning algorithm," *Intl. Journal of Intelligent Cooperative Information Systems*, vol. 1(2) pp. 319–346, 1992.

[14] A. Doucet, "On sequential simulation-based methods for bayesian filtering," Department of Engineering, University of Cambridge, Technical Report CUED/F-INFENG/TR.310, 1998.

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

[16] T. Finin, Y. Labrou, and J. Mayfield, "KQML as an agent communication language," in J. Bradshaw, editor, *Software Agents*, MIT Press: Cambridge, MA, 1997.

[17] L. Firoiu, T. Oates, and P. Cohen, "Learning regular languages from positive evidence," in *Proc.* 20^{th} *Annual Meeting of the Cognitive Science Society*, Madison, WI, 1998, pp. 350–355.

[18] C. V. Goldman, M. Allen, and S. Zilberstein, "Decentralized language learning through acting," in *Proc. Third Intl. Joint Conf. on Autonomous Agents and Multiagent Systems*, New York, NY, July 2004, pp. 1006–1013.

[19] P. Gmytrasiewicz and P. Doshi, "A framework for sequential planning in multiagent settings," *Journal of AI Research*, vol. 24 pp. 1–31, 2005.

[20] C. Ghidini and F. Giunchiglia, "Local models semantics, or contextual reasoning = locality + compatibility," *Artificial Intelligence*, vol. 127 pp. 221–259, 2001.

[21] B. J. Grosz and S. Kraus, "Collaborative plans for complex group action," *Artificial Intelligence*, vol. 86(2) pp. 269—357, 1996.

[22] C. V. Goldman and J. S. Rosenschein, "Emergent coordination through the use of cooperative state-changing rules," in *Proc. Twelfth Natl. Conf. on Artificial Intelligence*, Seattle, WA, 1994, pp. 408–413.

[23] P. J. Gmytrasiewicz, M. Summers, and D. Gopal, "Toward automated evolution of agent communication languages," in *Proc. 35th Hawaii Intl. Conf. on System Sciences*, Hawaii, 2002, p. 79.

[24] C. V. Goldman and S. Zilberstein, "Optimizing information exchange in cooperative multi-agent systems," in *Proc. Second Intl. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, 2003, pp. 137—144.

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

[26] C. V. Goldman and S. Zilberstein, "Goal-oriented Dec-MDPs with direct communication," Computer Science Dept., University of Massachusetts at Amherst, Technical Report 04–44, 2004.

[27] L. P. Kaelbling, *Learning in Embedded Systems*, MIT Press: Cambridge, MA, 1993.

[28] N. Komarova and P. Niyogi, "Optimizing the mutual intelligibility of linguistic agents in a shared world," *Artificial Intelligence*, vol. 154 pp. 1–42, 2004.

[29] R. D. Luce and H. Raiffa, *Games and Decisions*, John Wiley and Sons, Inc.: New York, NY, 1957.

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

[31] NASA, "Mars climate orbiter failure board report," *available at*: `ftp://ftp.hq.nasa.gov/pub/pao/reports/1999/MCO_report.pdf`, 1999.

[32] C. H. Papadimitriou and J. Tsitsiklis, "The complexity of Markov decision processes," *Mathematics of Operations Research*, vol. 12(3) pp. 441–450, 1987.

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

[34] H. Putnam, "The meaning of 'meaning'," in *Mind, Language and Reality: Philosophical Papers Volume 2*, Cambridge University Press: Cambridge, England, 1975, pp. 215–271.

[35] W. V. O. Quine, *Word and Object*, MIT Press: Cambridge, MA, 1960.

[36] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 2000.

[37] L. Serafini and P. Bouquet, "Comparing formal theories of context in AI," *Artificial Intelligence*, vol. 155 pp. 41–67, 2004.

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

[39] N. Sharygina and D. Peled, "A combined testing and verification approach for software reliability," *Formal Methods for Increasing Software Productivity, LNCS*, vol. 2021 pp. 611–628, 2001.

[40] L. Steels and P. Vogt, "Grounding adaptive language games in robotic agents," in *Proc. Fourth European Conf. on Artificial Life*, Brighton, UK, 1997, pp. 474–482.

[41] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128 pp. 99–141, 2001.

[42] P. Vogt and H. Coumans, "Investigating social interaction strategies for bootstrapping lexicon development," *Journal of Artificial Societies and Social Simulation*, vol. 6(1), 2003.

[43] J. Wang and L. Gasser, "Mutual online concept learning for multiple agents," in *Proc. First Intl. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, Bologna, Italy, 2002, pp. 362–369.

[44] K. Wärneryd, "Cheap talk, coordination, and evolutionary stability," *Games and Economic Behavior*, vol. 5 pp. 532—546, 1993.

[45] H. A. Yanco, *Robot communication: Issues and implementation,* Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1994.

[46] H. Yanco and L. A. Stein. "An adaptive communication protocol for cooperating mobile robots," in J. A. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *From Animals to Animats: Proc. Second Intl. Conf. on the simulation of adaptive behavior*, MIT Press: Cambridge, MA, 1993, pp. 478—485.