Hierarchical Reinforcement Learning in Communication-Mediated Multiagent Coordination

Felix Fischer* Michael Rovatsos Gerhard Weiss
Department of Informatics
Technical University of Munich
85748 Garching, Germany
{fischerf,rovatsos,weissg}@cs.tum.edu

Abstract

This paper proposes hierarchical reinforcement learning methods for multiagent coordination problems modelled as Markov Decision Processes (MDP).

Starting from the observation that communication can aid in predicting others' behaviours, we suggest the use of inter-agent messages to mediate between state transitions in the original MDP. Since message exchange has little effect on the MDP (both consequence- and utility-wise), we are able to reduce the problem of learning an optimal policy for the multiagent MDP to learning an optimal communication policy.

To solve this problem for realistic domains, we utilise interaction frames *as powerful, knowledge-level policy abstractions that can be combined with case-based reasoning techniques. The approach is validated through experiments in a complex application domain which prove that it is capable of heuristically handling significantly larger state and action spaces than exact MDP solution methods.*

1. Introduction

Over the past decade, reinforcement learning (RL; e.g., see [15]) has been an active area of AI research in general and agent and multiagent systems (MASs) research in particular. According to the original *Markov decision process* (MDP; e.g., see [10]) formulation of RL, other agents the agent interacts with are treated as part of its environment. [9] identified the inability of MDPs to model multiple adaptive agents as the main drawback of this approach. As a consequence, recent years have seen a growing interest in extending the RL framework to explicitly take into account other agents as autonomous and self-interested entities. In this respect, research has concentrated almost exclusively on the theory of Markov games [3, 6, 9].

Although Markov games constitute the most natural and technically appropriate way to formalise the multiagent RL (MARL) problem, and despite the fact that research in this field has lead to very interesting results, the approach suffers from a number of problems:

- Until now, results are only available for special classes of games like *zero-sum* or *common-payoff*, and only for environments with a small number of agents. This seriously limits the practical applicability of the framework.
- The strong focus on equilibria inherent in Markov games leads to a variety of practical problems, most notably the lack of prescriptive force. In particular, this is a problem in the presence of multiple equilibria.

In addition to these observations, [14] remarks that there is a considerable lack of clarity as to which problem actually is to be *solved* in the context of MARL. This leads the authors to call for a return to the "AI agenda" in MARL research, maintaining the "optimal agent design" stance of classical AI. This means finding the best learning strategy for a given environment, which in the context of MARL is also characterised by specific classes of peer agents.

In this paper, we follow this intuition while focusing on *communication-mediated* multiagent coordination problems. These can be described in the traditional Markov game framework, but are additionally characterised by the fact that "physical" state transitions may be preceded by "communicative" actions that allow for a prediction of further physical actions. By assuming that communicative actions do not manipulate the environment (i.e. hardly affect the states agents find themselves in) and have (almost) no utility effects, we can view the exchanged messages as symbols that "encode" anticipated courses of physical action.

For this class of problems, the agent design problem can be reduced to designing agents that are capable of learn-

[*] The first author of this paper is a student.

ing the communication strategies of others and devising an optimal counter-strategy. So, under the assumption that the communicative behaviour of agents can indeed be learned, others' reactions become entirely predictable and there is no longer a need to learn optimal behaviour for the original Markov game. With this respect, the contribution of this paper is twofold:

1. We apply hierarchical RL methods [2] to the problem of communication learning. As we show, these methods are well-suited for communication-mediated multiagent MDPs, and this intuition is confirmed by experimental results in a complex domain.
2. We propose a new kind of powerful policy abstractions called *interaction frames* that allow for a generalisation over communication strategies. Interaction frames are able to handle speech-act-based [1] agent communication languages (ACLs) with propositional content and hence bridge the gap between ACL and MARL research.

The remainder of this paper is structured as follows. Section 2 describes our assumptions regarding communication and coordination in multiagent systems. In section 3 we discuss RL and the hierarchical RL *options* framework. 4 introduces interaction frames and presents how this data structure can be combined with the RL techniques of section 3 to learn the effective use of a set of communication patterns. Experimental results in a complex application domain are reported on in section 5. Section 6 concludes.

2. Communication and Coordination

Before we can describe our framework, we first have to explain what role is precisely played by communication in the MASs we consider. In fact, our view of communication is inspired by two independent aspects.

One of them is the model of communication suggested in [12] that is based on a *consequentialist*, *empirical* and *constructivist* outlook on multiagent communication. According to this model, the *meaning* of communicative actions or messages (which differ from so-called "physical" actions or "non-messages" in that they have no utility-relevant impact on the physical environment and very little cost) lies in the expected consequences as envisioned by agents that participate in or observe the respective interaction and have derived their expectations from previous experience. The *purpose* of communication is to be used by agents so as to evoke desirable courses of joint (physical) action and to reduce the contingency regarding other agents' future behaviour. In other words, the meaning of messages is constantly re-shaped by they way in which they are used by communicating agents, and is employed strategically to manage interactions effectively, i.e. to coordinate the activities of autonomous agents.

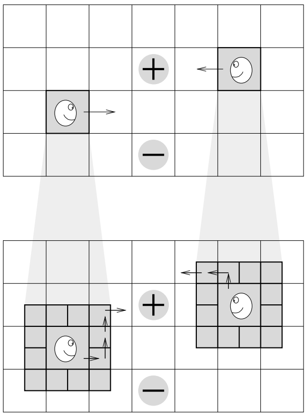

Figure 1. Traditional and communication-mediated view of an MDP

The second aspect concerns the extreme complexity of multiagent coordination problems in the Markov games formalism. For realistic domains, it is highly unlikely that agents can learn the complete behavioural structure of all of their opponents within reasonable amounts of time. Thus, while the framework is theoretically appealing, it is necessary to think of more practical ways of attacking coordination problems heuristically. Figure 1 illustrates how the model of communication sketched above can be used to simplify the MARL problem. If messages have no (or just very little) impact on agents' welfare and no real effect on the environment (apart from being observable by others), additional, communicative actions (which are usable regardless of the physical state) can be added to the MDP view of the system without changing the overall structure of the decision (and learning) problem.

In such a *communication-mediated* multiagent MDP, the transitions between "communication states" have virtually no impact on agent utilities (as suggested by the small shaded carets in the lower part of figure 1), but – according to the consequentialist model of communication – they can be *causally associated* with subsequent "physical" transitions that do matter (for reaching/avoiding the positive/negative state marked with $\oplus$/$\ominus$). Thus, if agent A learns the transition model of these communication states, it can predict those next actions of agent B that really matter for A. Over time, communicative actions will then have utility values associated with them that indicate which physical actions will result if a particular communication strategy is followed.

It should be noted, however, that by suggesting methods to learn strategic behaviour in communication we do *not* claim that this solves the *overall* (i.e. physical+communicational) MARL problem. Quite the opposite is the case: in learning how communication relates to subsequent physical behaviour, we deliberately *ignore* behaviour that does not result from communicative encounters.

3. Reinforcement learning and the *options* framework

Standard RL is based on the MDP model of sequential decision processes. An MDP is defined by a finite set $\mathcal{S}$ of *states* and finite sets $\mathcal{A}_s$ of *admissible actions* for each state $s \in \mathcal{S}$. *Transition probabilities*

$$p^a_{ss'} = P(s_{t+1} = s' | s_t = s, a_t = a) \qquad (1)$$

and *expected rewards*

$$r^a_s = E(r_{t+1} | s_t = s, a_t = a). \qquad (2)$$

specify the system's behaviour if action $a \in \mathcal{A}_s$ is taken in state $s \in \mathcal{S}$ and time step t. In multiagent settings, the environment dynamics p includes the behaviour of other agents.

Agent behaviour is represented by means of a (stochastic) *policy* $\pi : \mathcal{S} \times \bigcup_{s \in \mathcal{S}} \mathcal{A}_s \rightarrow [0, 1]$, meaning that action a is executed with probability $\pi(s, a)$ whenever state s is perceived. According to the *expected discounted infinite-horizon reward maximisation* criterion which we follow here, an optimal policy π^* is one that maximises the expected sum $E(\sum_{j=0}^{\infty} \gamma^j r_{t+j})$ of discounted future rewards, where r_{t+j} is the reward obtained j steps in the future and $0 \leq \gamma < 1$ is a geometric discount factor.

Based on that, the objective of RL is to learn an optimal policy by sampling state transitions and rewards. Q-learning [17] solves this problem by learning the value $Q^*(s, a)$ of taking action a in state s and following π^* thereafter. For this, an approximation Q is updated according to

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha \left[r + \gamma \max_{a' \in \mathcal{A}_{s'}} Q(s', a') \right]$$

for a sampled transition from s to s' due to action a and with associated reward r. For a learning rate α appropriately decaying over time and using an exploration strategy which ensures that in the limit each action is executed infinitely often in each state, Q-learning can be shown to converge to Q^*. An optimal policy is then given by

$$\pi^*(s, a) = \begin{cases} 1 & \text{if } a = \arg\max_{a' \in \mathcal{A}_s} Q^*(s, a') \\ 0 & \text{otherwise.} \end{cases}$$

To allow for *temporal abstraction* in RL, we use the *options* framework [2] which is based on augmenting the sets $\mathcal{A}_s$ of admissible "primitive" actions by sets of so-called "options". An option is a triple $o = (\mathcal{I}, \pi, \beta)$ consisting of an input set $\mathcal{I} \subseteq \mathcal{S}$, a (stationary, stochastic) policy π over primitive actions, and a termination condition $\beta : \mathcal{S} \rightarrow [0, 1]$. Option o is admissible in a state s iff $s \in \mathcal{I}$. If invoked it will behave according to π until it terminates stochastically according to β (we assume that $\{s | \beta(s) < 1\} \subseteq \mathcal{I}$). $\mathcal{O}_s$ is used to denote the set of admissible options in state s, which may or may not include some or all of the primitive actions in $\mathcal{A}_s$. If an option's policy π is Markov, i.e. action probabilities depend solely on the state of the core MDP, the option itself is called a *Markov option*.

For greater flexibility with respect to action selection and to consider policies $\mu : \mathcal{S} \times \bigcup_{s \in \mathcal{S}} \mathcal{O}_s \rightarrow [0, 1]$ over options, however, so-called *semi-Markov options* are required. These build on the theory of *semi-Markov decision processes* (SMDPs; e.g., see [10]). In contrast to MDPs, the duration τ of an action within an SMDP is a random variable, such that temporally extended courses of action can be modelled. In the case of options, τ is the number of time steps from the invocation of an option to its termination.

Since the core MDP together with a set $\mathcal{O}$ of options constitutes an SMDP, SMDP learning methods can be used to learn an optimal policy over $\mathcal{O}$. In turn, with the core MDP serving as an explicit representation of the SMDP, intra-option policies can be evaluated and learned. The SMDP version of Q-learning [4] applies the update equation

$$Q(s, o) \leftarrow (1 - \alpha)Q(s, o) + \alpha \left[r + \gamma^{\tau} \max_{o' \in \mathcal{O}_{s'}} Q(s', o') \right]$$

after option o has been running for τ time steps between s and s'. r denotes the cumulative discounted reward over this time, which could be computed as $r = r_1 + \gamma r_2 + \cdots + \gamma^{\tau-1} r_{\tau}$ from the individual rewards r_i. However, in an SMDP only the complete reward R obtained from executing o in s is known. Assuming an equal distribution of R over the τ steps yields

$$r = \sum_{i=1}^{\tau} \gamma^{i-1} \frac{R}{\tau} = \frac{\gamma^{\tau} - 1}{\gamma - 1} \cdot \frac{R}{\tau}.$$

$Q(s, o)$ can be shown to converge to $Q^*(s, o)$ for all $s \in \mathcal{S}$ and $o \in \mathcal{O}$ under conditions similar to those for conventional Q-learning.

4. Interaction frames

Interaction frames are a key concept of the abstract social reasoning architecture InFFrA proposed in [13]. There, they describe patterns of interactions that can be used strategically by knowledge-based agents to guide their communicative behaviour based on a reasoning process called *framing*.

For the scope of this paper, it suffices to look at (interaction) frames as *policy abstractions* (in the sense of MDP policies). This interpretation forms the basis of a formal model of InFFrA called m^2InFFrA (where the m^2 stands for "Markov-square" and hints at the underlying hierarchical two-level MDP view), details of which can be found in [5].

In m^2InFFrA, a frame describes a set of two-party, discrete, turn-taking interaction *encounters* which can be thought of as conversations between two agents. A sequence of message patterns called *trajectory* specifies the surface structure of the encounters described by the frame, while a list of *substitutions* captures the values of variables in the trajectory in previously experienced interactions. Each substitution also corresponds to a set of logical *conditions* that were required for and/or precipitated by execution of the trajectory in the respective encounter. Finally, *trajectory occurrence* and *substitution occurrence* counters record the frequency with which the frame has occurred in the past. This leads to the following formal definition of m^2InFFrA frames:

Definition 1 *A* frame *is a tuple* $F = (T, \Theta, C, h_T, h_\Theta)$*, where*

- $T = \langle p_1, p_2, \ldots, p_n \rangle$ *is a sequence of message patterns* $p_i \in \mathcal{M}$*, the* trajectory *of the frame,*
- $\Theta = \langle \vartheta_1, \ldots, \vartheta_m \rangle$ *is an ordered list of* variable substitutions*,*
- $C = \langle c_1, \ldots, c_m \rangle$ *is an ordered list of* condition sets*, such that* $c_j \in 2^{\mathcal{L}}$ *is the condition set relevant under substitution* ϑ_j*,*
- $h_T \in \mathbb{N}^{|T|}$ *is a* trajectory occurrence counter *list counting the occurrence of each prefix of the trajectory* T *in previous encounters, and*
- $h_\Theta \in \mathbb{N}^{|\Theta|}$ *is a* substitution occurrence counter *list counting the occurrence of each member of the substitution list* Θ *in previous encounters.*

Thereby, $\mathcal{M}$ is a language of speech-act like message and action patterns of the form $\texttt{perf}(A, B, X)$ or $\texttt{do}(A, Ac)$. In the case of messages, `perf` is a performative symbol (`request`, `inform` etc.), *A*/*B* are agent identifiers or agent variables and X is the propositional content of the message taken from a logical language $\mathcal{L}$. In the case of physical actions with the special "performative" `do`, Ac is the action executed by A (an action has no recipient as it is assumed to be observable by any agent in the system). Both X and Ac may contain non-logical *substitution* variables that are used for generalisation purposes. We further use $\mathcal{M}_c \subset \mathcal{M}$ to denote the language of actual (ground) messages that agents use in communication (i.e. messages that do not contain variables other than "content variables" used in a logical sense).

Writing $T(F)$, $\Theta(F)$, etc. for functions that return the respective elements of a frame F, its semantics can informally be summed up as follows: The agent "owning" F has experienced $h_T(F)[1]$ encounters which started with a message matching the first element $m_1 = T(F)[1]$ of the trajectory. $h_T(F)[2]$ of these encounters continued with a message matching $m_2 = T(F)[2]$, and so on. Θ, h_Θ and C provide more information about specific past encounters: For $i \le |\Theta|$, F represents $h_\Theta[i]$ past encounters matching $T(F)\Theta(F)[i]$, and $C(F)\Theta(F)[i]$ held during each of these encounters. Agents are assumed to maintain a knowledge base KB encoded in the same propositional language $\mathcal{L}$ that is used as a content language for messages.

From the standpoint of RL, ground instances of a frame can be seen as temporally extended policies that range over sequences of actions (i.e. over options). Moreover, by virtue of generalisation over possible variable substitutions, frames capture a whole set of such policies.

4.1. Frame-based options

We will now describe how interaction frames can be integrated with the options framework. For this, we view agent communication as an MDP with a set $\mathcal{S}$ of states, which is derived using some kind of *state abstraction* that partitions the current knowledge base content KB and the perceived *encounter prefix* $w \in \mathcal{M}_c^*$ (the sequence of messages exchanged so far during an encounter) into equivalence classes denoted by $s_{(w,KB)}$. $\mathcal{A} = \mathcal{M}_c$ is used as the set of primitive actions.

For a frame F to induce an option $o_F \in \mathcal{O}$, $o = (\mathcal{I}_F, \pi_F, \beta_F)$, over the core MDP given by $\mathcal{S}$ and $\mathcal{A}$, we need to define $\mathcal{I}_F$, π_F and β_F appropriately based on F. Obviously, a frame can be selected iff there exists a substitution to enact it under. Thus, we have

$$\mathcal{I}_F = \left\{ s_{(w,KB)} \in \mathcal{S} \mid \Theta_{poss}(F, w, KB) \neq \emptyset \right\},$$

where $\Theta_{poss}(F, w, KB)$ is the set of substitutions F can be enacted under given encounter prefix w and knowledge base KB. Θ_{poss} can effectively be computed by unifying w with the appropriate prefix of $T(F)$ (which yields a "fixed" substitution $\vartheta_f(F, w)$) and restricting the variable bindings for the corresponding postfix $post(T(F), w)$ (to which ϑ_f has already been applied) to those executable under KB.

As for the termination of o_F, several reasons are imaginable:

1. o_F will definitely terminate if the end of the trajectory $T(F)$ has been reached.
2. o_F will also terminate if $T(F)$ no longer matches the encounter prefix w.
3. Changes to the knowledge base may prohibit the execution of the remaining steps of F. As above, this is verified by checking whether $\Theta_{poss}(F, w, KB) = \emptyset$.

4. The remaining steps of F may be rendered undesirable due to changes to the agent's knowledge base.

5. Actions by the peer may prohibit execution of the frame under the most desirable substitution.

Items 2, 3 and 4 concern the *validity*, *adequacy* and *desirability* of F, respectively [13]. Item 5 can be viewed as concerning the desirability of F as well as the validity of the most desirable substitution.

If we assume that desirability (conditions 4 and 5) corresponds to the profit the agent will obtain from executing a message/action sequence w under knowledge base KB which it can estimate using a utility function $u : \mathcal{M}_c^* \times KB \rightarrow \mathbb{R}$, then a boolean desirability criterion $\delta(F, w, KB)$ can be defined which determines desirability-based option termination:

$$\beta_F(s_{(w,KB)}) = \begin{cases} 1 & \text{if } \Theta_{poss}(F, w, KB) = \emptyset \\ & \text{or } \delta(F, w, KB) \\ 0 & \text{otherwise.} \end{cases}$$

As [2] points out, optimal policies over the set of available options are in general suboptimal policies of the core MDP, if not all primitive actions $a \in \mathcal{A}_s$ remain admissible in s. This is obvious for the special case of frame-based options, since a frame's expected reward may change during the enactment, rendering another frame more desirable. We are, however, willing to accept this drawback for the sake of complexity reduction, all the more in the domain of multiagent interaction another benefit may arise: If agents accept frames as an established means of interaction and follow them normatively to a certain extent rather than constantly driving for optimal actions, this will make them more comprehensible and dependable, thereby reducing the contingency inherent in interactions. As a result, agents should adhere to a frame as long as possible.

What now remains to be specified is the intra-option policy π_F corresponding to a frame F. From a rational actor's point of view, the agent should take the best possible action according to its utility function u, considering the restrictions imposed by the active frame. This yields the (greedy, deterministic) *enactment policy*

$$\pi_F(s_{(w,KB)}, m) = \begin{cases} 1 & \text{if } m = m^*(F, w, KB) \\ 0 & \text{otherwise,} \end{cases}$$

where $m^*(F, w, KB)$ is the optimal action given encounter prefix w and knowledge base KB and restricted by frame F. A concrete definition of m^* will be given in the following section.

To ensure convergence of Q-learning, we can add *Boltzmann exploration* to obtain a stochastic frame selection criterion with a temperature T that decreases over time:

$$P(F|w, KB) = \frac{e^{Q(s_{(w,KB)}, o_F)/T}}{\sum_{\Theta_{poss}(F',w,KB) \neq \emptyset} e^{Q(s_{(w,KB)}, o_{F'})/T}}$$

4.2. Frame enactment

To determine the optimal action m^* we should select *within* a frame, we apply expected utility maximisation within the temporal scope of the remaining trajectory steps, since, in general, the postfix of $T(F)$ with respect to w can contain unbound variables so that the utility of its execution is not ex ante deterministic.

From the framing view, both the agent itself and the peer it is interacting with have the freedom to substitute concrete values for free variables that occur first in one of their trajectory steps. We will write Θ_s and Θ_p for the sets of possible substitutions the agent and the peer can apply respectively and ϑ_s and ϑ_p for specific elements of these sets. Then, the expected utility of executing the remaining steps of $T(F)$ is given by

$$E[u(\vartheta_s|F, w, KB)] = \\ \sum_{\vartheta_p \in \Theta_p} u_\gamma(post(T(F), w)\vartheta_s\vartheta_p, KB) \cdot P(\vartheta_p|\vartheta_s, F, w),$$

where $post(T(F), w)$ again denotes the postfix of $T(F)$ with respect to w, $u_\gamma(w, KB)$ is the discounted utility of executing a message sequence w with initial knowledge base KB and $P(\vartheta_p|\vartheta_s, F, w)$ is the probability with which the peer will conditionally choose a substitution ϑ_p depending on the agent's own choice ϑ_s. Based on that, the optimal action is given by

$$m^*(F, w, KB) = T(F)[|w| + 1]\vartheta^*(F, w, KB),$$

where

$$\vartheta^*(F, w, KB) = \arg\max_{\vartheta_s \in \Theta_s} E[u(\vartheta_s|F, w, KB)].$$

To compute the probability $P(\vartheta_p|\vartheta_s, F, w)$ in accordance with the model provided by the frame F, we will compare the (projected) message sequence of the present encounter with those of the (past) encounters stored in F.

According to the consequentialist and empirical view of communication, the future probability for the occurrence of any message sequence should be equal to the frequency with which it has been observed in the past. However, $T(F)$ can be very abstract, an it is unlikely that all the past cases stored in F are equally relevant for every new encounter prefix that matches $T(F)$. Intuition suggests that this relevance should be expressed using some notion of *similarity* between message patterns in the vein of *case-based reasoning* [8]. To formally capture this notion, we introduce a real-valued *similarity measure* $\sigma : \mathcal{M}^* \times \mathcal{M}^* \rightarrow [0,1]$ on sequences of messages, allowing us to compare the (perceived) encounter prefix with the past cases stored in a frame. In general, the definition of σ will be domain-dependant. A very simple default choice that proves viable in many cases is to define sequence similarity recursively as the average pairwise similarity of sequence elements and their arguments. At the term/operator level, a strict equality criterion can be applied while assigning a similarity of 1 to term/variable and variable/variable pairs.

Based on this similarity measure on message sequences, the similarity of a substitution ϑ to a frame F can be defined as

$$\sigma(\vartheta, F) = \sum_{i=1}^{|\Theta(F)|} \sigma\big(T(F)\vartheta, T(F)\Theta(F)[i]\big) \cdot h_{\Theta}(F)[i] \cdot r\big(C(F)[i], \vartheta, KB\big).$$

where $r(C, \vartheta, KB)$ is 1 if $C\vartheta$ holds under KB and 0 else (where obvious from the context, we omit KB for readability). The probability that a frame F is enacted under a specific substitution ϑ is then computed as the similarity of ϑ to F relative to all substitutions in Θ_{poss}, i.e.

$$P(\vartheta|F, w) = \begin{cases} \lambda \cdot \sigma(\vartheta, F) & \text{if } \vartheta \in \Theta_{poss}(F, w, KB) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

for a normalisation constant λ.

To determine $P(\vartheta_p|\vartheta_s, F, w)$ we can use the Bayesian product rule

$$P(\vartheta_p \wedge \vartheta_s|F, w) = P(\vartheta_p|\vartheta_s, F, w) \cdot P(\vartheta_s|F, w),$$

where $\vartheta_p \wedge \vartheta_s$ denotes the event of the peer selecting $\vartheta_p \in \Theta_p$ and the agent selecting $\vartheta_s \in \Theta_s$, such that F is enacted under the complete substitution that results from combining ϑ_f with ϑ_p and ϑ_s.

On the other hand, the probability that the agent has previously chosen substitution ϑ_s is given by the sum of the probabilities for the occurrence of complete substitutions that ϑ_s is part of, such that

$$\begin{aligned} P(\vartheta_p|\vartheta_s, F, w) &= \frac{P(\vartheta_p \wedge \vartheta_s|F, w)}{P(\vartheta_s|F, w)} \\ &= \frac{P(\vartheta_f(F, w)\vartheta_s\vartheta_p|F, w)}{\sum_{\vartheta} P(\vartheta_f(F, w)\vartheta_s\vartheta|F, w))}. \end{aligned}$$

Applying equation 3 to both numerator and denominator finally yields

$$P(\vartheta_p|\vartheta_s, F, w) = \frac{\sigma(\vartheta_f(F, w)\vartheta_s\vartheta_p, F)}{\sum_{\vartheta} \sigma(\vartheta_f(F, w)\vartheta_s\vartheta, F)},$$

provided that $\vartheta_f(F, w)\vartheta_s\vartheta_p \in \Theta_{poss}(F, w, KB)$ (observe that the denominator is constant in ϑ_p and does not need to be computed to determine $\vartheta^*(F, w, KB)$).

To sum up, a frame F is enacted by executing the next step of the trajectory $T(F)$ under the substitution that promises the highest expected utility for the complete trajectory suffix, while computation of the occurrence probability for each substitution is based solely on its similarity to the past cases stored in F.

5. Experimental results

The frame-based learning approach has been tested in the multiagent-based link exchange system LIESON. In this system, agents representing Web sites engage in communication to negotiate over mutual linkage with the end of increasing the popularity of one's own site and that of other preferred sites.

Available physical actions in this domain are the addition and deletion of numerically rated links originating from one's own site and the modification of ratings (where the probability of attracting more traffic through a link depends on the rating value).

LIESON provides a highly dynamic and complex interaction testbed for the following reasons:

- Agents only have a partial and incomplete view of the link network. In particular, agents engage in non-communicative goal-oriented action in between encounters, so that the link network (and hence the agents' utility situation) may change while a conversation is unfolding.
- The number of possible link configurations is vast, and agents can only predict possible utilities for a very limited number of hypothetical future layouts.
- There is no notion of commitment – agents choose frames in a self-interested way and may or may not execute the physical actions that result from them. Also, they may undo their effects later on.

LIESON agents consist of a non-social BDI [11] reasoning kernel that projects future link network configurations and prioritises goals according to utility considerations. If these goals involve actions that have to be executed by other agents, the m^{2}InFFrA component starts a framing process which runs until the goal of communication has achieved or no adequate frame can be found. We report on experiments in which these agents were equipped with frames with the following six trajectories:

```
req(A,B,X)→acc(B,A,X)→conf(A,B,X)→do(B,X)
req(A,B,X)→prop(B,A,Y)→acc(A,B,Y)→do(B,Y)
req(A,B,X)→prop-also(B,A,Y)
                  →acc(A,B,Y)→do(B,X)→do(A,Y)
req(A,B,X)→reject(B,A,X)
req(A,B,X)→prop(B,A,Y)→reject(B,A,Y)
req(A,B,X)→prop-also(B,A,Y)→reject(B,A,Y)
```

The first three frames allow for `acc`epting to perform a `req`uested action X, making a counter-`prop`osal in which

Y is suggested instead of X, or using `prop-also` to suggest that B executes X if A agrees to execute Y. The last three frames can be used to explicitly `reject` a request or proposal. In that, X and Y are link modification actions; each message is available in every state and incurs a cost that is almost negligible compared to the utilities gained or lost through linkage actions (yet high enough to ensure no conversation goes on forever). Also, agents can always send a `stop` action to indicate that they terminate an encounter if they cannot find a suitable frame.

After their termination, encounters are stored in the frame from which they have originated. For example, agent a_1 would store the encounter $req(a_1, a_2, add(a_2, a_1, 2)) \rightarrow reject(a_1, a_2, add(a_2, a_1, 2))$ by adding a substitution $[A/a_1, B/a_2, X/add(a_2, a_1, 2)]$ to the respective frame together with an automatically generated list of conditions that were required for physical action execution.

As state abstraction, we use generalised lists of statements of the form $\{\uparrow|\downarrow\}(\{I,R\},\{I,R,T\},\{+,-,?\})$ representing the physical actions talked about in an encounter. $\uparrow$ and $\downarrow$ stand for a positive/negative link modification (i.e. addition/deletion of a link or an increase/decrease of its rating value), I/R for the initiator/responder of the encounter, T for a third party; $+/-/?$ indicates whether the (learning) agent likes/dislikes/doesn't know the target site of the link modification. For example, if a_1 and a_2 talk about $do(a_1, deleteLink(a_1, a_3))$ in an encounter initiated by a_1 (while the learning agent a_2 is the responder and likes a_3's site) this is abstracted to $\downarrow(I, T, +)$. If in the same conversation a_2 suggests to modify his own link toward a_1 (whom he does not like) from a rating value of 1 to 3, the state (viz *subject*) of the encounter becomes $\{\downarrow(I, T, +), \uparrow(R, I, -)\}$. The intuition behind this state abstraction method is to capture, in a generalised form, the *goal* of the conversation that can currently be realised while at the same time reducing the state space to a reasonable size.

Figure 2 shows a comparison for a system with ten agents with an identical profile of private ratings (preferences) towards other agents (both plots show the performance of the best and the worst agent in the group as well as the average utility over all agents). In the first plot, agents employ BDI reasoning and additionally send requests to others whenever they favour execution of someone else's action according to their BDI queue. These requests are then enqueued by the recipient as if he had "thought of" executing the respective action himself. Thus, it depends on the recipient's goal queue and on his utility considerations whether the request will be honoured or not. As one can see, after a certain amount of time agents do no longer execute any of the actions requested by others, and cannot find any profitable action to execute themselves, either. The system converges to a stable state.

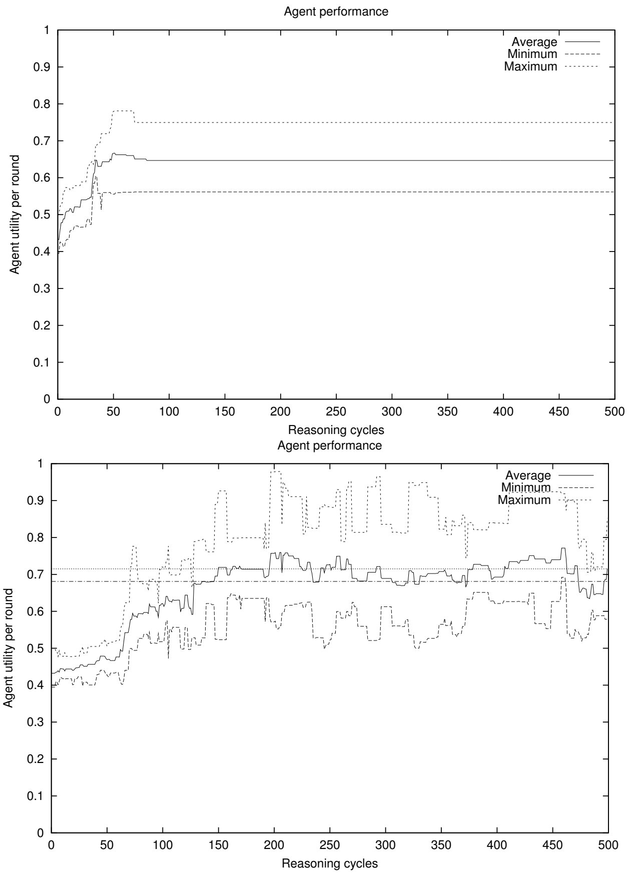

Figure 2. Performance plots.

The second plot shows the results of a simulation with the same setup as above but using m^{2}InFFrA agents. Again, agents issue requests whenever they identify that someone else could do something useful. After this initial message, the framing procedure takes over. Quite clearly, despite the fact that there is a greater variation in maximal/minimal/average agent utility, the average and the best agent perform significantly better than in the BDI case, while the weakest agent performs just as good as in the BDI case on the average.

More interesting still is that the average utility lies within the range of the two horizontal lines in the plot. These denote the average utilities for two very interesting linkage configurations: the lower of the two corresponds to a fully connected linkage graph, in which each agent (honestly) displays the ratings of his out-links, i.e. reveals his true opinions about others. The slightly higher utility shown by the upper line is attained if agents do not lay any links toward agents they dislike. It is an interesting property of the utility function used in LIESON, that being "politically correct" is slightly better than being honest. The fact that agent utilities evolve around these benchmarks indicates that they truly strive to make strategic communication moves and to exploit the advantages of concealing certain beliefs.

6. Conclusions

In this paper, we have proposed hierarchical RL methods for learning in communication-mediated multiagent coordination problems. Rather than attempting to solve Markovian multiagent games which suffer from the "curse of dimensionality" directly, we rely on learning communication strategies using a rich representation for policy abstractions called *interaction frames*.

We have formally defined frames in the m^{2}InFFrA framework as sets of encounter patterns supplemented with logical conditions, variable substitutions and occurrence counters. By virtue of the options framework, frames can be reinterpreted as temporal abstractions in the sense of hierarchical RL. Also, by applying similarity criteria they can be seen as case abstractions in terms of case-based reasoning. We have defined a two-level hierarchical decision-making apparatus for learning and reasoning with frames and underlined its usefulness through experiments in a complex multiagent domain.

Essentially, frame-based learning follows the intuition that crucial interaction processes usually unfold within communicative "episodes", so that learning a complete model of other agents' behaviours overshoots the mark. Like in human societies, "private" action may indeed have effects on other parties, but if these effects are substantial, its execution will usually be preceded by communication to ensure a coordinated flow of interaction (especially in the case of cooperation and collaboration). We claim that learning optimal *communication* strategies and using communication to predict immanent physical action is paramount to reducing the complexity of interaction processes in realistic application domains.

In this respect, a major advantage of our approach is that it combines the decision-theoretic power of RL models with the knowledge-based aspects of symbolic agent communication, interaction protocols and ACL research in general. It is this aspect that makes rational action and learning possible for high-level agent architectures that employ logical reasoning.

An extensive treatment of the additional components required to use frame-based learning as part of a complete agent architecture can be found in [5]. This particularly includes a generalisation method for frame trajectories, which uses cluster validation techniques [7] on the (possibly fuzzy) clustering a set of frames induces on the space of possible message sequences, thereby enabling agents to create frames for encounters not matching any existing frame and to extend the use of these frames to similar encounters in the future.

Future work includes investigations into how interaction frames can be constructed from scratch (first steps in this direction concerning frame concatenation have already been described in [5]). Developing the theory of *hierarchical options* (built around a policy over options) [2] into "meta-frames" that allow for an online combination of different interaction patterns and subgoals seems to be a promising idea for the construction of frames for longer-term interactions. Also, the issue of some general form of state abstraction is still largely unresolved and deserves our attention in the future.

References

[1] J. L. Austin. *How to do things with Words*. Clarendon Press, 1962.

[2] A. G. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(4):41–77, 2003.

[3] M. H. Bowling and M. M. Veloso. Rational and convergent learning in stochastic games. In *Procs. IJCAI'01*, 2001.

[4] S. J. Bradtke and M. O. Duff. Reinforcement learning methods for continuous-time Markov decision problems. *Procs. NIPS-7*, 1995.

[5] F. Fischer. Frame-based learning and generalisation for multiagent communication. Diploma Thesis. Department of Informatics, Technical University of Munich, 2003.

[6] J. Hu and M. P. Wellman. Multiagent reinforcement learning: theoretical framework and an algorithm. In *Procs. ICML-98*, 1998.

[7] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice Hall, Upper Saddle River, NJ, 1988.

[8] J. L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Francisco, CA, 1993.

[9] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Procs. ICML-94*, 1994.

[10] M. L. Puterman. *Markov Decision Processes*. John Wiley & Sons, New York, NY, 1994.

[11] A. S. Rao, M. P. Georgeff. An abstract architecture for rational agents. In *Procs. KR-92*, 1992.

[12] M. Rovatsos, M. Nickles, and G. Weiß. Interaction is Meaning: A New Model for Communication in Open Systems. *Procs. AAMAS'03*, Melbourne, Australia, 2003.

[13] M. Rovatsos, G. Weiß, and M. Wolf. An Approach to the Analysis and Design of Multiagent Systems based on Interaction Frames. *Procs. AAMAS'02*, Bologna, Italy, 2002.

[14] Y. Shoham, R. Powers, and T. Grenager. Multi-agent reinforcement learning: a critical survey. Technical report, Stanford University, 2003.

[15] R. S. Sutton and A. G. Barto. *Reinforcement Learning*. MIT Press, Cambridge, MA, 1998.

[16] R. S. Sutton, D. Precup, and S. P. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999.

[17] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.