Reward Shaping for Valuing Communications During Multi-Agent Coordination

Simon A. Williamson
School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, UK
saw06r@ecs.soton.ac.uk

Enrico H. Gerding
School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, UK
eg@ecs.soton.ac.uk

Nicholas R. Jennings
School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, UK
nrj@ecs.soton.ac.uk

ABSTRACT

Decentralised coordination in multi-agent systems is typically achieved using communication. However, in many cases, communication is expensive to utilise because there is limited bandwidth, it may be dangerous to communicate, or communication may simply be unavailable at times. In this context, we argue for a rational approach to communication — if it has a *cost*, the agents should be able to calculate a *value* of communicating. By doing this, the agents can balance the need to communicate with the cost of doing so. In this research, we present a novel model of rational communication, that uses *reward shaping* to value communications, and employ this valuation in decentralised POMDP policy generation. In this context, reward shaping is the process by which expectations over joint actions are adjusted based on how coordinated the agent team is. An empirical evaluation of the benefits of this approach is presented in two domains. First, in the context of an idealised benchmark problem, the multiagent Tiger problem, our method is shown to require significantly less communication (up to 30% fewer messages) and still achieves a 30% performance improvement over the current state of the art. Second, in the context of a larger-scale problem, RoboCupRescue, our method is shown to scale well, and operate without recourse to significant amounts of domain knowledge.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Algorithms, Experimentation, Theory

Keywords

Agents, Decentralised POMDPs, Communication

1. INTRODUCTION

Increasingly, complex real-world problems are being tackled by teams of software agents. Whilst this approach has many benefits in terms of creating robust solutions, it also poses a new challenge — how to coordinate the actions of the agent teams to solve the problem efficiently. In this context, coordination involves managing the interactions of the autonomous entities so that they do not disrupt each other, can take proactive actions to help each other, and can take multiple actions at the same time when this is required to solve the problem.

Now, in almost all existing work, communication is an integral component of the coordination problem. That is, the agents communicate their state and intentions to each other in order to reach agreements and an understanding about how to coordinate their actions. However, in many real-world problems, communication is a scarce resource. Specifically, communication is typically limited in bandwidth, is not always available, and may be expensive to utilise. In such circumstances, many coordination mechanisms break down because the agents can no longer accurately model the state of the other agents. Given this, in our research, we consider how to utilise *rational communication* [3] to coordinate when communication is a restricted resource.

Against this background, this work presents a model of rational communication using a novel, principled valuation for communications based on belief divergence. Specifically, we demonstrate that belief divergence, a measure of how coordinated the beliefs of a distributed team are, can be used to model the likelihood of coordination on the decentralised computation of joint actions, and furthermore, generate new expectations over joint actions that account for the probability that team members will coordinate. This new approach efficiently approximates the value of communications in a decentralised sequential decision making context — where the generation of the exact value of communicating is an intractable problem because agents must reason about all possible team observations and action histories. In particular, our approach allows the agents to attach a principled value to the communication action, and so balance the possible value gained by the team with the costs associated with using the communication infrastructure. Consequently, our formalisation extends the state of the art in two main ways. First, it allows decentralised POMDP models to be applied to larger problems (for instance RoboCupRescue), than has hitherto been possible, whilst avoiding any domain-specific knowledge to generate valuations for communication actions. Second, it generates more accurate online valuations of communication than the previous state of the art, and by doing so, allows an expensive communication to medium to be used more efficiently.

In the rest of this paper, Section 2 describes the main related approaches for valuing communications in multi-agent

Cite as: Reward Shaping for Valuing Communications During Multi-Agent Coordination, Simon A. Williamson, Enrico H. Gerding, Nicholas R. Jennings, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 641–648

coordination. Section 3 describes our general formalisation for valuing communications and Section 4 gives a specific instantiation in terms of the multiagent Tiger problem and RoboCupRescue. The former problem allows us to compare our approach with the state of the art and theoretically optimal solutions, whilst the latter allows us to show how our model works in larger-scale problems. Section 5 gives an empirical analysis of our model within these two domains, showing the utility of our approach. Finally, Section 6 concludes.

2. BACKGROUND AND RELATED WORK

In this section we consider work that has attempted to create a rational value of communicating and the coordination mechanisms they are embedded within. To achieve rational communication, the key challenge is how the sender can estimate the value to the team of a particular communication. Now, this is often done by measuring the value of communication as the improvement in coordination that occurs. This involves modelling the coordination problem explicitly, and perturbing it to see how it changes with communication.

Considering this approach, if we evaluate models of the coordination problem, such as in $STEAM$ [12] where agents build models of the team and have teamwork operators to perturb this model, then we can predict the change in utility based on sending a communication. Indeed, this is carried out in [12], which models the future stages of the team coordination in a Markov Decision process (MDP), where communication acts cause transitions in the model. However, the communication semantics are dictated by the teamwork model — making the solution less general.

A similar approach in [3] models the state of the team knowledge using a Bayesian Network, and values communication based on how it changes the expected utility of possible actions. Both methods rely on agents maintaining complex models to generate coordinated actions, rather than explicitly modelling coordination. Whilst this general approach is very powerful, and generates an accurate value of the impact of communication, it requires an estimation of the state of each team member, which is not realistic for larger teams where the agents can be in many different states. In essence, the computational complexity of this approach does not scale well with the number of agents, and it would be better if we could derive a valuation that does not depend on a complex team model as this is difficult to maintain and grows with the size of the team.

Another alternative, decentralised Partially Observable Markov Decision Processes (Dec-POMDPs), have been introduced by a number of authors including [13], in order to model the team decision problem in a sequential domain. Such approaches are good at representing partially observable, stochastic problems with a more general communication framework than teamwork models. Unfortunately, these models do not scale well because of the classic curse of dimensionality problem. Nevertheless, this still forms the point of departure of our work because it allows us to model our problem with communication restrictions and our communication valuations can be combined with existing work on efficient policy generation to make for a more scalable solution. To give more details, consider the *dec_POMDP_com* from Zilberstein and Goldman [13], which is a decentralised POMDP with an added alphabet of possible communications. In this context, the difference between centralised and decentralised POMDPs is that the former is a single POMDP that can be solved by each agent or a central authority — since the state of each agent is known to all others. In a decentralised version, however, each agent has its own POMDP to solve, with the other agents corresponding to a part of that POMDP. The *dec_POMDP_com* is the tuple $DECPOM = \langle n, S, \mathcal{A}, \Sigma, C_\Sigma, P, R, \Omega, O, T\rangle$ where:

- n is the number of agents.
- S is the global state space.
- $\mathcal{A} = \times A_i$ is the joint action space, with A_i the action space for agent i. An element $a = \langle a_1, \ldots, a_n\rangle$ of the joint action space represents the concurrent execution of the actions a_i by each agent i.
- Σ is the alphabet of communications with $\sigma_i \in \Sigma$ a message sent by agent i. σ is a joint communication from set Σ^n. ε_σ is the null communication, i.e. sending an empty message.
- C_Σ is the cost of communicating an atomic message. This cost is 0 for ε_σ.
- P is the transition probability function. That is, the probability
$$P(s \in S, a \in \mathcal{A}, s' \in S) \in [0, 1] \tag{1}$$
of moving from state s to state s' when the agents take joint action a.
- R is the reward function. Returns a real-valued reward
$$R(s \in S, a \in \mathcal{A}, \sigma \in \Sigma^n, s' \in S) \in \mathbb{R} \tag{2}$$
for executing joint action a and sending joint communication σ in state s, resulting in state s'.
- $\Omega = \times\Omega_i$ is the joint observation space, with Ω_i the observation space for agent i. An element $\omega = \langle\omega_1, \ldots, \omega_n\rangle$ of the joint observation space, represents the concurrent observation ω_i by each agent i.
- O is the observation function. It is the probability
$$O(s \in S, a \in \mathcal{A}, s' \in S, \omega \in \Omega) \in [0, 1] \tag{3}$$
of joint observation ω when in state s and taking joint action a resulting in state s'.
- $T \in \mathbb{N}^+$ is the (possibly infinite) time horizon in which the agents take their actions.

The solution to the decentralised model consists of two policies: *i*) the action policy that associates belief states with actions and *ii*) the policy that associates belief states with communication acts. Now, general offline solvers can typically approximate solutions for problems with around 2^{16} states for two agents [2]. However, this is not sufficient for problems as large as RoboCupRescue (which has about 2^{700} states). To operate at this scale, previous work typically uses an online approach to solve the decentralised POMDP [8]. However, this model relies on extensive domain knowledge in order to generate efficient decentralised polices. Furthermore, this model assumes that communication is some parallel activity to other actions. However, we believe this is an unreasonable assumption since, in many problems, communication is only available at specific instances or shares resources with the other actions. Furthermore, the cost of communication in this model is represented by some arbitrary negative utility over the communication alphabet Σ.

This is a considerable weakness, because it does not represent the opportunity costs of using communication when communication shares some resource with the other actions (i.e. power, bandwidth or time).

Now, [11] considers how to generate a communication policy, but starts by analysing an offline centralised MDP solution rather than the decentralised POMDP. Against this, the true impact of communications on expected reward can be calculated using a POMDP by considering the joint belief space during policy generation, but this is intractable since decentralised POMDPs have NEXP-time complexity [1]. In this context, [9] presents the ACE-PJB-Comm algorithm which represents the state of the art in generating communication valuations by making communication decisions an online issue and solving the centralised POMDP offline. In this method, agents maintain a distribution of the joint belief space and use this to decide whether to send a communication. Nevertheless, this remains infeasible for large problems since the joint belief space still has NEXP-time complexity. To combat this, [10] proposed the *dec_POMDP_Valued_Com* model, which uses online policy generation over local observations. However, this model has a parameter, α, that values communication as a weighting of the information-theoretic value of the content of the message and the original reward function. Unfortunately, α needs to be hand-tuned for a specific domain. To overcome this, we propose a principled way to *approximate* this valuation using a measure of belief divergence in the team. This makes the computation tractable by removing the need to consider the joint belief space in policy generation (more details are given Section 3), and at the same time, does not require significant amounts of domain information.

3. REWARD SHAPING FOR COMMUNICATION VALUATIONS

This section introduces the *RS_dec_POMDP*, a modification of the decentralised POMDP formalisations presented in Section 2 which utilises a novel reward shaping mechanism to compute decentralised policies using only local observations. We first describe the intuition of reward shaping and detail how the model aligns with previous decentralised POMDPs. Then we describe how the reward shaping is calculated and, finally, how this is used with communication in an online policy generation algorithm.

3.1 Reward Shaping

We would like our agents to be able to calculate policies for decentralised POMDPs using only local observations and communication histories. To this end, we describe a model of reward shaping that uses the concept of *belief divergence* to estimate the need for communication in a principled fashion. Now, the standard model of rational communication models the exact beliefs of other agents and analyses how communication would change their actions. However, it is cheaper to maintain an estimate of how coordinated the beliefs of the agents are and use this to decide when to communicate. The intuition here is that agents that have a very small difference in their beliefs can calculate the impact of joint actions independently and arrive at the same answer. Since this same answer relates to the joint action space, then they will be coordinated if they follow their own part of the joint action. However, if the difference in beliefs is greater, then some communication may be needed in order to resynchronise their beliefs and allow them to make independent coordinated actions again. Within this setting, reward shaping is the process by which independent estimations of the expected reward of joint actions are modulated by the agent's perception of the belief divergence in the team. Low divergences mean the beliefs are coordinated and so all agents can independently calculate the same expected reward for each joint action. Conversely, large divergences mean that agents cannot independently value joint actions and in this case can only estimate the value of local actions and assume the other agents act randomly (or according to some predetermined distribution).

Therefore, we first extend the *dec_POMDP_com* model by including communication actions into the standard action selection policy problem. This is a non-trivial change. In particular, by doing so, the reward function for the problem needs to be changed to consider just joint actions (an action for each agent), unlike previous models which assign rewards for a joint domain action and joint communication together. The benefit of this modification is that communication penalties can now be represented by opportunity costs — a more general representation. Furthermore, each agent now has to maintain parameters about the rest of the team — this is the estimation of the belief of the rest of the team about the state of the problem. This is compared with an agent's own belief to give an estimation of belief divergence, which is then used to modulate the reward function for all actions in order to approximate the value of communication (see Section 3.2). As a result, this extended model allows the communication valuation problem to be extracted from the policy computation problem, resulting in complexity reduction benefits compared to the *dec_POMDP_com* which requires that the full joint experiences of the team are analysed to value communications. Finally, we restrict the communication alphabet to the observation alphabet, to maintain the generality of the communication valuation because we do not introduce any problem-specific communications.

More formally, *RS_dec_POMDP* is the tuple $RSDPM = \langle n, S, \mathcal{A}, P, \Omega, O, T, R, R_{rs}, \Sigma, C_\Sigma, B, B_d, \pi \rangle$ where the definitions are the same as those for the *dec_POMDP_com* except:

- R is the reward function for all actions (including communication where $R(C) = C_\Sigma$). It returns a real-valued reward:

$$R(s \in S, a \in \mathcal{A}, s' \in S) \in \mathbb{R} \tag{4}$$

  when executing joint action a in state s, resulting in state s'. This is equivalent to R in the original formalisation, except that the communication substage has been removed.

- $R_{rs} \in \mathbb{R}$ is the reward signal supplied to the policy generation problem. Here, we introduce a principled shaping function over the original R which uses belief divergence to modulate the reward based on how coordinated the team's beliefs are.

- B represents the agent's estimation of the current state. Specifically, $B(s) \in [0, 1]$ is the probability that the problem is in state s.

- $B_d \in \mathbb{R}$ represents the agent's current estimation of the divergence in the beliefs of the agents. Section 3.2 will explore this parameter further.

- π is a policy that relates joint actions (including communications) to beliefs and belief divergences, $\pi : B \times B_d \to \mathcal{A}$.

Now, we need to detail how an R_{rs} can be constructed without considering the full joint observation space, and which approximates the policy constructed over the original R when we do.

3.2 Expected Rewards using Belief Divergence

In this section we discuss how to measure belief divergence, and then how to use this to shape expected rewards.

We would like a principled metric that indicates the distance between two different beliefs B and B' in a general fashion. Hence we need to consider: *i)* how to measure a distance in belief; and *ii)* how to estimate the difference in beliefs for a team of distributed agents.

Considering the first problem, since we are measuring the distance between probability distributions, it is appropriate to use information theory to measure the difference. The actual measure is dependent on the problem, so simple domains might use an absolute difference ($B_d(B_i, B_j) = \sum_{s \in S}(B_i(s) - B_j(s))$) in belief variables or relative entropy. In contrast, more complex belief spaces might use an aggregate measure like KL Divergence [5]

$$B_d(B_i, B_j) = D_{KL}(B_i||B_j) = \sum_{s \in S} B_i(s) log \frac{B_i(s)}{B_j(s)}$$

In the second problem of estimating a distributed belief, we can use a simple estimation of information propagation. Specifically, we assume that the other agents will not have independently received any of the observations the communicating agent is deliberating over, and that its beliefs have remained static since the last communication action. To this end, we establish a reference point, B_{ref}, which is the belief of the agent when it last synchronised its knowledge. We then compare the current belief state B_i with this point. More formally, the approximate divergence B_{da} is:

$$B_{da}(B_i) = B_d(B_i, B_{ref}) \tag{5}$$

Considering this assumption could result, on the one hand, in an over-estimate of the divergence due to assuming that the other agents will not have gained any of the new information that the deliberating agent has received since the last communication point. On the other hand, this assumption does not account for information the other agents have received which the communicating agent has not — causing an under-estimate in the divergence. Consequently, it is hard to place bounds on the approximation of the divergence using this assumption but we believe it is still a useful departure point due to its ease of implementation in a decentralised fashion. Furthermore, it may be possible to make the approximation more accurate using the observation function to obtain probabilities of features being commonly known. With the divergence measure established, we now consider how to use it to shape rewards.

Each agent calculates the expected value for each joint action over its impression of the belief state B and divergence B_d. If each agent has the same beliefs ($B_d = 0$) then they will all calculate the expected reward $E(a)_u$ of a joint action $a = \langle a_i, a_{-i} \rangle$, where a_i is the action taken by agent i and a_{-i} are the actions taken by the other agents, as:

$$E(a)_u = \sum_{s \in S} \sum_{s' \in S} B(s).P(s, a, s').R(s, a, s') \tag{6}$$

If the divergence is maximum (normalised, $B_d = 1$), then the agents cannot assume they will each generate the same value for the joint actions, and so they will mis-coordinate. Consequently, an agent locally calculates the expected reward of a joint action assuming the other agents act randomly:

$$E(a)_r = \frac{1}{|A_{-i}|} \sum_{a_{-i} \in \mathcal{A}_{-i}} \sum_{s \in S} \sum_{s' \in S} [B(s). \\ P(s, \langle a_i, a_{-i} \rangle, s').R(s, \langle a_i, a_{-i} \rangle, s')] \tag{7}$$

where $\mathcal{A}_{-i}$ is the joint action space of all agents except i. In general $0 < B_d < 1$, so in this case the shaped expected reward for a joint action is given by the following function (which is problem-dependent):

$$R_{rs}(a, B_d) = f(B_d, E(a)_u, E(a)_r) \tag{8}$$

Section 4.gives examples for our two exemplar domains.

3.3 Communication within Policy Generation

We want our model to be applicable to large problems and consequently it is important to only generate solutions to parts of the belief space that the agents encounter. With this in mind, we take inspiration from online policy generation algorithms, such as [8], but create a new model of policy generation that incorporates communication and reward shaping. A new model is necessary because previous online algorithms do not consider the joint action space or communications explicitly (as per Section 2). In more detail, agents are allowed to communicate their history of observations from the last time they communicated (at this stage we only consider synchronisation communication) and the information in these observations represents the belief divergence. The expected reward for each action is calculated using Equation 8 and we assume we know f. In this model, communication causes the divergence to be reset to zero. Using this mechanism, we expect the agents to either employ actions with a lower penalty (where the average expectation is high) for mis-coordination or to communicate when the divergence is high, and to perform actions that have large rewards for coordinated behaviour when it is low.

More formally, each agent performs a search in the local belief space b, using joint actions and local observations to generate new belief states. Essentially, nodes are belief states, and branches are composed of joint actions and local observations. The search is pursued to depth D (this again is dependent on the problem). We define a new function $P(\omega_i|b, a)$ which is the probability of an observation ω_i in a belief state b given a joint action a. An action a is given by:

$$\pi(b, D) = argmax_a \sum_{\omega_i \in \Omega_i} P(\omega_i|b, a)\delta(\rho(b, a, \omega_i), D - 1) \tag{9}$$

where $\delta(b, d)$ calculates the recursive payoff in the search tree and is defined by:

$$\delta(b, d) = \begin{cases} 0 & \text{, if } d = 0 \\ f(B_d, E(a)_u, E(a)_r) + \gamma \max_a \sum_{\omega_i \in \Omega_i} & \\ \quad [P(\omega_i|b, a) \times \delta(\rho(b, a, \omega_i), d - 1)] & \text{, if } d > 0 \end{cases} \tag{10}$$

where γ is the discount factor and $\rho(b, a, \omega_i)$ gives a new belief state b' when action $\rho(b, a, \omega_i)$ is performed in b and ω_i is received. Incidentally, we must calculate a new belief divergence B'_d after each action:

$$B'_d = \begin{cases} 0 & \text{, if } a_i = \texttt{COM} \\ B_d \cup \omega_i & \text{, if } a_i \neq \texttt{COM} \end{cases} \tag{11}$$

where COM is a communication action (a message composed of elements of Σ). This is the only place where communication actions are treated differently to other actions. In effect this says that, if the agent communicates, its belief divergence is reset to 0, else the observation that has just been received must be integrated into the divergence estimate.

Using this technique, our approach can handle problems as large as RoboCupRescue, yet does not require extensive domain knowledge for solving decentralised POMDPs. We can do this because we only use local observations, making the search tree significantly smaller (the branching factor is of the order of the number of possible observations, rather than a combination of number of agents and observations).

4. EXAMPLE IMPLEMENTATIONS

This section describes the multiagent Tiger problem [6] — a well known coordination problem which allows us to compare our method with the state of the art, and then demonstrate how our coordination mechanism can be used to facilitate agent teams in this domain. Following this, we describe a large problem — RoboCupRescue which, unlike the Tiger problem, requires an approximate solution for the optimal policy due to its size and so is included to illustrate the scalability of our method.

4.1 The Multiagent Tiger Domain

The multiagent Tiger domain is a multiagent extension to the classic Tiger problem, which we describe here along with the modifications we have made to allow communication. We describe the problem for two agents, since this is the case considered by previous work, but the problem can be extended trivially to more agents.

In more detail, two agents must each open one of two doors. Behind one door is a treasure and behind the other is a penalty in the form of a tiger. The agents do not know which door contains the tiger. This gives two states: SL where the tiger is behind the left door, and SR when it is behind the right door. If both agents open the door containing the treasure then they receive a large reward. If one agent opens the door with the tiger then they both receive a large penalty. If both agents open the tiger door then they receive a smaller penalty. Consequently, the agents should coordinate on the location of the tiger. In order to do this the agents can request independent, noisy observations of where the tiger is. An observation has a probability of being correct equal to $1-w$ where w is the noise in the observation function. Furthermore, they can communicate to the other agent their belief about the location of the tiger. The problem is sequential in nature and each action (opening a door, listening for where the tiger is and communicating) takes the same length of time. The problem is reset to a random state whenever a door is opened ($p(SL) = p(SR) = 0.5$). The full details of this problem are in [6] with the modification that we have introduced of a communication action that takes the same length of time as other actions and costs the same amount as listening for the location of the tiger.

The aim of the problem is to maximise, over a potentially infinite horizon, the cumulative reward for the agents as a team. That is, all agents should aim to open the door with the reward. Consequently they should open the correct door as often as possible, whilst minimising the amount of time spent listening or communicating.

4.1.1 As a *RS_dec_POMDP*

Basic aspects of the decentralised POMDP components of this model are already defined for this problem in [6]. Therefore we focus on the parts that are specific to *RS_dec_POMDP*.

Belief: Since the tiger problem has only two states, SL or SR, the belief space can simply be represented as the probability that the instantiation is in state SL. More formally, for agent i, the belief is defined as $B_i = Pr(SL)$, where $B_i \in [0, 1]$.

Belief Divergence: In this simple belief space we can use the absolute difference as a divergence measure (as described in Section 3.2). More formally, $B_d(B_i, B_j) = |B_i - B_j|$. We do not worry about direction since our reward shaping function will be insensitive to it.

Expected Rewards: We need to derive E_u and E_r for the joint actions available to the agents. These are defined using Equations 6 and 7.

Table 1: Expected Reward bounds

Joint Action	E_u	E_r
$E(\langle \texttt{OL}, \texttt{OL} \rangle)$	$30 - 80B$	$\frac{-155B-21}{2}$
$E(\langle \texttt{OR}, \texttt{OR} \rangle)$	$80B - 50$	$\frac{155B-176}{2}$
$E(\langle \texttt{LISTEN}, \texttt{LISTEN} \rangle)$	0	-23
$E(\langle \texttt{COM}, \texttt{COM} \rangle)$	-5	$\frac{-51}{2}$

Reward Shaping Function: Now, we can construct f (from Equation 8) by considering the impact of the likelihood of coordination based on the divergence in beliefs for two or more decision makers. This likelihood modulates between the average E_r and coordinated expectation E_u. We calculate this likelihood by considering the simple policy that assumes all agents have the same beliefs. This is easy to calculate in general by reducing the decentralised POMDP to a centralised multiagent POMDP (which has a lower complexity class). For the Tiger problem this policy can be represented by the alpha vectors for each action to a horizon of one, as shown in Figure 1. It is important to note that we only consider the dominating joint actions $\langle \texttt{OL}, \texttt{OL} \rangle$, $\langle \texttt{OR}, \texttt{OR} \rangle$, $\langle \texttt{LISTEN}, \texttt{LISTEN} \rangle$, $\langle \texttt{COM}, \texttt{COM} \rangle$ since a centralised policy would not consider any other combination for this problem because their expected value is less than the dominating actions for all belief states. This policy shows

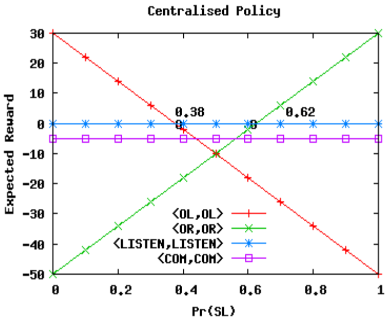

Figure 1: Centralised policy which assumes all agents have consistent beliefs

that, if both agents have a belief $B \in [0, 0.38]$, then the best action for both agents is to open the left door. If both agents have a belief $B \in [0.62, 1.0]$ then the best action for both agents to open the right door. Finally, for other beliefs, the best action is to request an observation. Here, it is interesting to note that, since the agents always assume the other agents have the same beliefs, it is never the best action to communicate since this would be redundant and only has a negative reward.

Using this policy we can derive a function of distance between two points in the belief space, which returns the proportion of B in which the two decision makers would select the same vector. This gives us a probability for a given divergence that the two agents will successfully coordinate. More formally, we introduce the probability of coordination, PC, for a belief divergence B_d,

$$PC(B_d) = \sum_{i=1}^{n} max(d_i - d_{i-1} - B_d, 0) \qquad (12)$$

for a set of intersections in alpha vectors $D = \{d_1, d_2, ..., d_n\}$. It is then simple to use our belief divergence metric to calculate the distance between the intersections.

Policy Computation: Together these components make up the full R_{rs} function which uses an estimation of the belief divergence to estimate the value of communication in the Tiger problem. The following equation is now used for the expected value of each joint action:

$$R_{rs}(a, B_d) = E(a)_r + PC(B_d)(E(a)_u - E(a)_r) \qquad (13)$$

This equation uses the probability of coordination, PC, based on belief divergence to weigh two expected rewards — the fully coordinated reward for a joint action and the value of an action when other agents act randomly. Consequently, if belief divergence is low, policy computation uses an expected reward that assumes coordination in the team, and if not, it assumes the agent must act alone. The `COM` action is used to alleviate belief divergence during policy computation.

4.2 The RoboCupRescue Domain

RoboCupRescue is a multiagent simulator of the situation in an urban area in the immediate aftermath of an earthquake [4]. Here, heterogeneous intelligent agents such as fire fighters, the police and ambulance crews conduct search and rescue activities in this virtual disaster world. Specifically, they search for civilian agents trapped in damaged and burning buildings. The aim of the agents in this domain is to rescue as many civilians as possible. Ambulance agents are responsible for freeing trapped and hurt civilians and moving them to a refuge; Fire Brigade agents must fight the spread of the fire; and the Police agents must unblock roads. In still more detail, the environment consists of buildings connected by roads. Nodes connect different roads and buildings together, thus the map can be seen as a graph. Agents have limited sensing capabilities; they can only tell the state of buildings that are very close, with some amount of noise. They have knowledge of the layout of the map, but do not initially know which roads are blocked, where civilians are trapped and which buildings are on fire. All agents can move along roads and into buildings, if those roads are not blocked. Agents are hurt if they move into burning buildings. Communication is peer to peer and has a cost which we can define for our problem.

In this context, the full RoboCupRescue problem requires several components not relevant to this research (such as an estimation of how fire spreads and an efficient search strategy), and so we constrain the problem. To this end, we only consider the ambulance agents' task. We do this because the police task does not require teamwork to unblock roads and the fire brigade task requires a complex model of the spread of the fire to do well (thus it is less about coordination).

4.2.1 As a RS_dec_POMDP

In order to specify the basic decentralised POMDP, several elements need to be defined from the point of view of the ambulance agents. Specifically, the state S describes a set of state variables for whether each building contains trapped civilians or not, and also the position of the ambulance agents, who can be in any building, or on any road or node (but only one of them at any one time). The actions A_i available to the agents are complex behaviours to move to unexplored buildings, rescue civilians, move civilians to refuges, and finally, communicate their observation history since the last time they communicated (b_h). Agents select joint actions (an action assigned to each agent from the set $\langle$`EXPLORE`, `LOAD`, `RESCUE`, `UNLOAD`, `COM`$\rangle$) and implement their own part of that joint action. In this case, a joint action might involve multiple ambulances digging out one civilian and in this case the extent to which a civilian is dug out is sub-linear with the number of ambulances digging. Consequently, a team of agents does much better than when the agents work individually. This is so that tight coordination on rescue actions is desirable, and the problem is not dominated by the need to search the entire map in order to do well. The cost of communication C_Σ relates to the time required to send the observation history. The reward function R gives a reward for each civilian rescued and building explored. The observation function O supplies each agent with the state of buildings nearby (i.e. whether these contain trapped civilians) and the location of any agents close enough. The communication alphabet $\Sigma = \Omega_1 = \Omega_2 = \ldots \Omega_n$, and so a message can be composed of any symbol in the observation alphabet.

Belief: We use a factored state space consisting of the probability that the state variable is true or false. Considering all state variables together gives the full belief space.

Belief Divergence: This is measured using KL Divergence (as discussed in Section 3.2). We use this aggregate measure because, unlike the Tiger problem, there are many belief variables to consider (including one for each building and location of the ambulances). We can obtain the increase in KL Divergence during the belief revision process after new observations very efficiently. More formally:

$$B_d(b_{ref}, b_h) = D_{KL}(b_h \| b_{ref}) \qquad (14)$$

Expected Rewards: $E_{u,r}$ can be calculated trivially and so is not presented here.

Reward Shaping Function: In the Tiger problem we were able to define an exact reward shaping function (Equation 8), but this is not possible here since we need a probability of coordination and that involves solving the centralised POMDP (which has PSPACE complexity [7]). However we can estimate the function as a linear function of the belief divergence measure giving:

$$PC(B_d) = \frac{B_{dM} - B_d}{B_{dM}} \qquad (15)$$

where B_{dM} is the maximum KL Divergence in this belief space.

Policy Computation: Given this, R_{rs} is:

$$R_{rs}(a, B_d) = E(a)_r + PC(B_d)(E(a)_u - E(a)_r) \qquad (16)$$

The intuition here is that we assume there is a linear relationship between belief divergence and the chance of coordinating. We believe this is valid because the belief space is large and small differences should not cause a mis-coordination.

5. EMPIRICAL EVALUATION

This section compares our approach for generating decentralised coordinated agents in the Tiger problem to several benchmarks, including theoretical optimal solutions and the state-of-the-art from the literature. We then go on to show its utility in the RoboCupRescue ambulance problem.

5.1 The Tiger Problem

We compare our *RS_dec_POMDP* model with ACE-PJB-Comm because this represents the state of the art in computing communication valuations for decentralised POMDPs (see Section 2). Specifically, we compare the performance in the two-agent Tiger problem described in Section 4.1 where we run 20000 simulations of 6 timesteps (more are not needed because the game is generally iterated twice or more in this duration). We also present results for a *Full* communication model, where the agents communicate all observations to each other at each timestep with no cost, which means each agent is solving an identical POMDP with a global (though still partially observable) view. Finally, as a lower bound, we present results for a model (here called *Zero*) that never communicates.

In this setting, we compare the average reward per timestep obtained for different values of the noise parameter w in the observation function, and the number of messages sent (each model communicates in the same alphabet and the messages are generally of the same size across the experiments). We use average reward per timestep as it allows us to compare our model (which requires a timestep to communicate) with one which communicates in parallel directly, and we are interested in performance when the problem is increasingly difficult to observe directly. Consequently, we vary w from exact ($w = 0$) to random observations ($w = 0.5$). We aim to test the hypothesis that our *RS_dec_POMDP* model can reduce the communication cost, compared to the state of the art, without any noticeable drop in team utility.

To this end, Figure 2 shows that, on average, our model achieves 84% of the utility of the *Full* model. This is compared to ACE-PJB-Comm which achieves only 53% of the *Full* utility. This unexpected improvement is because there is an inherent weakness in considering the non-communication policy in isolation — that it does not allow for efficient exploitation of communication. Furthermore, for $w > 0.35$ our model avoids dropping below zero reward and remaining there, whilst ACE-PJB-Comm does not. This is because our model identifies when door opening has potentially disastrous results when the agents might have the wrong coordinated impression, even if communication aids in maintaining consistent beliefs. Also, our model always does better than the *Zero* communication model and stays close to *Full* for all w. In *Full* the agents never mis-coordinate, but when observations are noisy it is risky to open a door, hence the average reward tends towards zero. Similarly, *Zero* mis-coordinates more and more as the noise increases, until the agents estimate that opening a door is too risky based on the noisy observations and hence, it tends back towards zero.

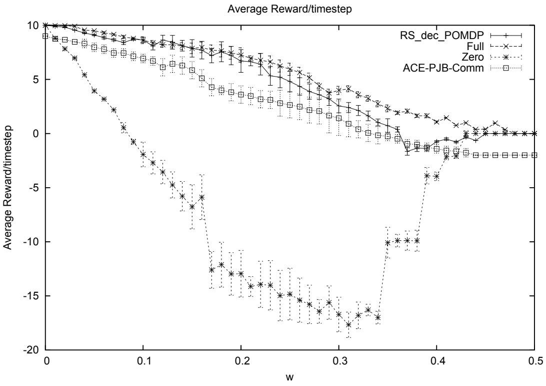

Figure 2: Performance of coordination models against noise, error bars are at 95%.

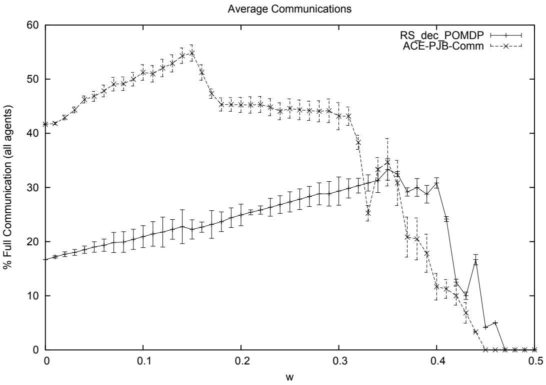

Figure 3: Communication in coordination models against noise, error bars are at 95%.

Figure 3 compares the number of messages the team of agents send during the simulation. As can be seen, for $w < 0.35$, our model sends fewer messages — communicating up to 30% less and, on average, 20% less. This is important since in our model communication is a more costly medium (in that communication takes a timestep). Specifically, the agents must be much more careful with their use of the communication medium. ACE-PJB-Comm is not equipped to deal with communication that shares resources with other actions — making our algorithm more generally applicable. Considering Figure 3 further, both models initially increase communication as noise increases, and then start to drop as the information communicated becomes less informative — making communication redundant. Consequently, our mechanism does better than the state of the art whilst communicating less, because it is able to explicitly reason about the benefit gained from communication versus the cost. The ACE-PJB-Comm algorithm over-communicates to achieve the same result because it pays no penalty to do so (except the somewhat artificial communication reward penalty, which subtracts some utility for using communication when a more general cost is in terms of lost opportunity whilst using the communication medium). Furthermore, these re-

Table 2: Results for the RoboCupRescue ambulance task, averaged over 30 runs with the 95% confidence interval in brackets.

	Average Reward	Comms
Full	41 (2)	300 (0)
dec_POMDP_Valued_Com	25 (2)	108 (5)
RS_dec_POMDP	32 (3)	35 (10)
Zero	26 (5)	0 (0)

sults demonstrate the utility of embedding *rational communication* within policy generation — the policy explicitly accounts for the cost of communication in deciding whether it is a useful action. In contrast, assuming communication is free during policy computation (as in ACE-PJB-Comm) means that the policy does not consider the *cost* of communicating, and consequently, exploits it inefficiently.

5.2 The RoboCupRescue Problem

In this problem we compare our model with benchmarks based on *Zero* and *Full* communication (as in Section 5.1). Furthermore, we compare this model with a heuristic model *dec_POMDP_Valued_Com* (see Section 2), which values communications using a learned parameter α, which we set randomly since we are comparing it with our model (which has no offline learning stage). These simulations use two agents, however our model can operate with larger teams [1]. There is no optimal solution for this problem, as the decentralised POMDP cannot be solved exactly by current techniques.

In this case, we compare the percentage of civilians saved by the end of the simulation and average the results over 30 runs. The results in Table 2 show that *Full* does the best because agents never duplicate search and always assist each other in digging out civilians, yet pay no penalty (in terms of time) for communicating. Furthermore, our new model outperforms both *Zero* and *dec_POMDP_Valued_Com* in terms of average reward. This is because the latter communicates too much (108 messages on average), which represents a third of the duration of the simulation. In contrast, our model only communicates 35 times on average, which leaves substantially more time to rescue civilians and, consequently, the approach does better. Finally, we also see that some communication is useful (because agents avoid duplicating search and help each other in digging out civilians) — which is why our model does better than *Zero*.

6. CONCLUSIONS

We have developed a model of *rational communication* that can evaluate the usefulness of communicating to an agent team using an information-theoretic measure of the belief divergence. This is combined with a decentralised decision-theoretic coordination mechanism that utilises reward shaping to balance the cost of communicating with the benefit it accrues. We then implement this in terms of the multiagent Tiger and RoboCupRescue problems. The results show that our approach can provide a principled, domain-independent valuation function for communication actions that allows for agent coordination, without the complexity of considering all agent beliefs. By doing so, we extend the state of the art in online communication valuations by providing a technique that outperforms existing work, whilst employing a more realistic and costly communication medium (specifically that communication takes time like any other action).

In future work we intend to extend the model and analyse its theoretical properties. In particular, we want to, using reward shaping, place bounds on the approximation of a joint belief-based coordination mechanism whilst allowing for complexity reductions. With this established, we want to improve scalability still further by developing an online mechanism that can learn the reward shaping function whilst the agent team is acting on the problem.

[1] To illustrate, using our technique a plan in RoboCupRescue with a horizon of 5 timesteps, and with 6 agents, takes about one minute to process on a normal desktop machine.

7. REFERENCES

[1] D. S. Bernstein, S. Zilberstein, and N. Immerman. The complexity of decentralized control of markov decision processes. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 32–37, Stanford, USA, 2000.

[2] A. Carlin and S. Zilberstein. Value-based observation compression for dec-pomdps. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 501–508, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[3] P. Gmytrasiewicz and E. Durfee. Rational communication in multi-agent environments. *Autonomous Agents and Multi-Agent Systems*, 4(3):233–272, 2000.

[4] K. Hiroaki. Robocup rescue: A grand challenge for multi-agent systems. In *Proceedings of the 4th International Conference on MultiAgent Systems*, pages 5–12, Boston, MA, 2000.

[5] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[6] R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 705–711, 2003.

[7] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[8] S. Paquet, L. Tobin, and B. Chaib-draa. An online pomdp algorithm for complex multiagent environments. In *Proceedings of the 4th international joint conference on Autonomous agents and multiagent systems*, pages 970–977, The Netherlands, 2005. ACM Press.

[9] M. Roth, R. Simmons, and M. Veloso. Reasoning about joint beliefs for execution-time communication decisions. In *Proceedings of the 4th international joint conference on Autonomous agents and multiagent systems*, pages 786–793, The Netherlands, 2005.

[10] S. A. Williamson, E. H. Gerding, and N. R. Jennings. A principled information valuation for communications during multi-agent coordination. In *Proceedings of the AAMAS Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains*, pages 137–151, 2007.

[11] P. Xuan and V. Lesser. Multi-agent policies: From centralized ones to decentralized ones. *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, Part 3:1098–1105, 2002.

[12] W. Zhang and M. Tambe. Towards flexible teamwork in persistent teams: Extended report. *Autonomous Agents and Multi-Agent Systems*, 3(2):159–183, 2000.

[13] S. Zilberstein and C. V. Goldman. Optimizing information exchange in cooperative muti-agent systems. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 137–144, Melbourne, Australia, 2003.