A Principled Information Valuation for Communications During Multi-Agent Coordination

Simon A. Williamson, Enrico H. Gerding and Nicholas R. Jennings
School of Electronics and Computer Science,
University of Southampton,
Southampton, SO17 1BJ, UK.
{saw06r,eg,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 information theory to value communications, and employ this valuation in a decision theoretic coordination mechanism. A preliminary empirical evaluation of the benefits of this approach is presented in the context of the RoboCupRescue simulator.

1 Introduction

Increasingly, complex real-world problems (including distributed sensing, air-traffic control, disaster response, network routing, space exploration and unmanned aerial vehicles) are being tackled by teams of software agents, rather than a traditional monolithic centralised system. Whilst this approach has many benefits in terms of creating robust solutions, it creates 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 a central 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* [4] to coordinate when communication is a restricted resource.

Against this background, this work presents a model of rational communication based on a principled formalisation for efficiently approximating the value of communications in a decentralised sequential decision making context. This approach allows the agents to attach a value

to the communication action, and so balance the possible value gained by the team with the costs associated with using the communication infrastructure. Whilst the current policy generation model for decentralised Partially Observable Markov Decision Processes (POMDPs) can already perform such trade-offs implicitly, these approaches must reason about all possible observation and communication histories of the team — leading to an intractable policy generation problem. We avoid this by introducing a novel principled valuation for communications based on information theory (specifically, the impact of any communication is measured using KL Divergence). This is an efficient calculation that does not require reasoning over team beliefs. This novel approach then allows decentralised POMDP models to be applied to much larger problems (for instance RoboCupRescue), whilst avoiding any domain specific knowledge to generate valuations for communication actions.

In the rest of this paper, Section 2 describes some of the related approaches for valuing communications in multi-agent coordination. Section 3 describes our general formalisation for valuing communications and Section 4 gives a specific instantiation in terms of RoboCupRescue, a large multi-agent disaster response simulation. Section 5 gives an empirical analysis of our model within the RoboCupRescue domain and shows the utility of our approach. Finally, in Section 6, we describe the future directions of our research.

2 Background and Related Work

In this section we first consider models of coordinating multi-agent systems since these will influence which communication valuations are possible and how they are utilised. Following this, we explicitly consider how to engender rational communication by the generation of valuations. Finally, it is important to consider action selection mechanisms which are capable of leveraging our coordination models and communication valuations.

2.1 Coordinating Multi-Agent Systems

In the current literature there are several general approaches to modelling coordination between teams of agents — including teamwork models, Bayesian Networks and Markov Decision Processes. All of these models can be distributed and utilise communication to manage inconsistencies between agents. Furthermore, they all allow a rational approach to communication by allowing the agents to assess the impact of communication actions in the future. We now describe each of these models in more detail.

Teamwork models such as *GRATE** [5], *STEAM* [14] and *Generalized Partial Global Planning* (GPGP/TÆMS) [2] allow agents to build models of the team problem and provide teamwork operators for perturbing these models. These models allow for complex team operations, but they also require a large communication overhead for their joint-planning stages — which is inappropriate in our problem domain where communication may be unavailable and agents need to perform reasonably well when out of contact with the team.

Bayesian Networks have also been used to model the distribution of knowledge in teams, allowing agents to reason about the uncertain state of the team for coordinated actions [4, 12]. Now, while this is a useful technique for modelling the impact of communication actions, it requires complex belief revision processes to implement, which makes it an inefficient approach with large teams.

Decentralised POMDPs have been introduced by a number of authors [17, 15, 8] 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 [17], 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 MDP to solve, with the other agents corresponding to a partially observable part of that MDP. Specifically, the *dec_POMDP_com* is defined by the tuple (for 2 agents) $DECPOM = \langle S, A_1, A_2, \Sigma, C_\Sigma, P, R, \Omega_1, \Omega_2, O, T \rangle$ where:

- S is the state space. The global state is defined as the joint state of both agents.
- A_1 and A_2 are the action spaces of each agent, with a_1 an element of A_i. An element $\langle a_1, a_2 \rangle$ of the joint action space $\mathcal{A} = \times A_i$, represents the concurrent execution of the actions a_i by each agent agent i.
- Σ is the alphabet of communications with $\sigma_i \in \Sigma$ a message sent by agent i. ε_σ is the null communication.
- C_Σ is the cost of communicating an atomic message. This cost is 0 for the null communication (sending an empty message).
- P is the transition probability function. The probability

$$P(s \in S, a_1 \in A_1, a_2 \in A_2, s' \in S) \in [0, 1] \tag{1}$$

  of moving from state s to state s' when the agents take actions a_1 and a_2.
- R is the reward function. Returns a real-valued reward

$$R(s \in S, a_1 \in A_1, \sigma_1 \in \Sigma_1, a_2 \in A_2, \sigma_2 \in \Sigma_2, s' \in S) \in \Re \tag{2}$$

  for executing actions a_1 and a_2 and sending communications σ_1 and σ_2 in state s, resulting in state s'.
- Ω_1 and Ω_2 are the observation spaces of each agent.
- O is the observation function. It is the probability

$$O(s \in S, a_1 \in A_1, a_2 \in A_2, s' \in S, \sigma_1 \in \Omega_1, \sigma_2 \in \Omega_2) \in [0, 1] \tag{3}$$

  of observing σ_1 and σ_2 when in state s and taking actions a_1 and a_2 resulting in state s'.
- T, the time horizon, whether infinite or if finite, a positive integer

The solution to the decentralised model consists of two policies: *i*) the normal action policy for the POMDP that associates belief states with actions and *ii*) the policy that associates belief states with communication acts. When the communication occurs the messages are typically broadcast to all agents and thus provide a means to synchronise the agent's knowledge of the global state.

2.2 Valuing Communication

To achieve rational communication, the key challenge is how the sender can estimate the value to the team of a particular communication. There are two main approaches to do this. The first is to measure the value of communication as the improvement in coordination that occurs. This involves modelling the coordination problem explicitly, and perturbing it to to see how it changes with communication. The second involves relating the content of communications to the reward structure of the problem, rather than the team. Both approaches will now be considered in more detail.

Considering the first approach, if we evaluate models of the coordination problem, such as *STEAM*, then we can predict the change in utility based on sending a communication. This is done in [16], which models the future stages of the team coordination in a MDP, where communication acts cause transitions in the model. A similar approach in [4] 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 good 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 which does not depend on a team model.

In the decentralised POMDP formalisation we have chosen to use in this work, the true impact of communications on expected reward could be calculated using the POMDP by considering the joint belief space during policy generation, but this is intractable since decentralised POMDPs have NEXP-time complexity [1]. Because of this, we propose a principled way to *approximate* this valuation using an information theoretic method. This makes the computation tractable by removing the need to consider the joint belief space in policy generation (more details are given Section 3).

Following this, information theory is a general model of valuing the information content of a particular message. Specifically, if we can relate the information content of communication to how useful it is to the team, then we have a simple local calculation of the value of a communication. Furthermore, this is not dependent on a team model and consequently scales with the number of agents. Examples of this approach are found in [9], where sensor networks must distribute only the most valuable observations because of power restrictions. Whilst we follow this broad approach, there are several additional problems to consider: *i*) how do we normalise the communication valuation with other rewards in the problem? and *ii*) how do we balance the costs for communicating with other actions? These are dealt with in Section 3.1 and 3.2 respectively.

2.3 Action Selection

Our decentralised POMDP affords us with some options for policy generation. Since we consider classes of problems where the transaction and observation functions are fully defined before acting in the problem, there are three relevant solution concepts:

- **Offline**: computation before the problem starts is used to generate an optimal or approximate policy.
- **Online**: agents select actions during the problem, rather than following a pre-computed policy.

- **Hybrid**: an approximate policy is generated offline, and online computation is used to improve the accuracy.

Usually, a large problem calls for offline or hybrid processing, but problems such as RoboCupRescue are too large to make these feasible approaches (for example, an instance of RoboCupRescue has roughly 2^{700} states). Online approaches such as [7] and [10], are more promising because they only generate an action for the current belief state of the agents — rather than all possible belief states. This means computation can be done at each action selection point during the simulation. These models then use heuristic search to return the action with highest expected reward after some search depth.

However, to date, most of these algorithms have been designed for single agent models. In order to apply these to the multi-agent case, the action selection mechanism needs to consider the other agents in order to coordinate (i.e. locally optimal action selection for each single agent may not lead to optimal team performance), and, as a result, the problem becomes much larger. However, models that explicitly consider the multi-agent case can reduce the size of the problem by exploiting interaction between the agents. This is seen in the solution detailed in [13] which finds optimal policies for decentralised POMDPs, but ignores communication and is only suitable for small problems. Similarly, [11] considers communication by assuming it is free in the offline planning stage, and then reasons about it online. Following this algorithm, at each step, agents calculate the joint action with and without sending their observation history. If the communication version results in a better outcome, then the observation history is communicated. However, this model relies on maintaining joint beliefs, which grow as no communication action is taken. As a consequence, this must be approximated to make the algorithm tractable for small problems, so it is very difficult to extend it to problems as large as RoboCupRescue. A multi-agent algorithm from [3] approximates the whole problem as a series of single step Bayesian games. This closely parallels the approaches taken in [7], but the algorithm is explicitly multi-agent. Unfortunately, there are no results with this algorithm in large problems. Thus, as a first step, this research will extend the work in [7] to the multi-agent case, since it has shown good results in RoboCupRescue, with no offline computation.

3 Decentralised Coordination with Valued Communications

In this section, we present our model — *dec_POMDP_Valued_Com* — a model of decentralised coordination which utilises an information theoretic communication valuation. We then proceed to describe an online policy generation algorithm which has been designed to leverage the communication valuations in our model.

3.1 *The dec_POMDP_Valued_Com* Model

Previous work in decentralised POMDPs considers communication to be a separate problem from other actions, which is always available in parallel. This assumes that it is possible to communicate and take other actions at the same time. However, we do not consider this to always be a realistic assumption because utilising the communication medium may stop other actions. Therefore we make communication an action like any other. This allows the model to plan actions that must be taken before communication is possible. Thus, for example, the model can evaluate the value to the team of a particular communication, but the agent may be in a state where communication is not possible. Given this, the agent can then estimate the cost of moving to a state which allows communication, and decide whether it is worth

performing this state change in order to send the communication. Consequently, a solution to our model is a single policy which includes all communications and domain actions.

Now, we would like to remove reasoning about the value of communications from the coordination model (because of the complexity of this method) and replace it with a principled approximation. We do this using a normalised (in terms of the concrete rewards available to the team) information theoretic valuation over possible communications. To this end, the other key feature which distinguishes our model is that we include a second reward function, which is used exclusively for the communication actions. Hence, our model has two reward functions that are weighted so that the communication reward function represents an approximation, using an information theoretic measure, of the true value (impact on expected reward) of the communication. This measure gives the amount of information in a subset of observations b_h relative to the communicating agent's current beliefs b_1. We use this second reward function to remove reasoning about the value of communications from the policy generation problem. Specifically, the weighting of this second reward function is intended to replace this reasoning with a principled approximation. The benefit of this approach is that policy generation is more scalable (because agents do not need to consider the possible beliefs of the other agents) and is explicitly concerned with choosing the most valuable action (and not analysing the impact of communication). At this time, the formal derivation of this weighting is ongoing work so here we use an empirical approach to show the utility of our approach (see Section 5 for more details).

The *dec_POMDP_Valued_Com* is defined by the tuple (for 2 agents) $DECPOMVALCOM = \langle S, A_1, A_2, \Sigma, C_\Sigma, P, R_p, R_c, R, \Omega_1, \Omega_2, O, T \rangle$ where all the symbols have same meaning as in Section 2.1, except:

- R_p is the *problem reward function*. It returns a real-valued reward

$$R_p(s \in S, a_1 \in A_1, a_2 \in A_2, s' \in S) \in \Re \tag{4}$$

  when executing actions a_1 and a_2 in state s, resulting in state s'. This is equivalent to R in the original formalisation, except that the communication substage has been removed.

- Σ is the alphabet of communications with σ_i a member sent by agent i. Here, we fix the communication alphabet to be the alphabet of observations so that we can employ generic metrics over it, hence $\Sigma = \Omega_1 = \Omega_2 = \Omega$ and ε_σ is the null communication.

- R_c is the *communication reward function*. $R_c(b_1, b_h)$ is the value of b_h (a subset of communication symbols) in the current belief state b_1. This value is information gained by the communicating agent from b_h. We approximate the influence of this value to the eventual reward gained, in order to give a rational value to communicating.

- Our empirical approach will aim to find an approximation for the relative importance of communicating compared with other actions. Thus we will assign reward using the function

$$R = \alpha R_p + (1 - \alpha) R_c \tag{5}$$

The intuition here is that a communication act allows the other agents to know the exact state of the communicating agent (in some sense this is a synchronisation point), and that we should be able to use the distance the beliefs of the agent moves from this point in the belief space as an approximation of the probability of mis-coordinating. The idea of representing the probability of mis-coordination is seen in [14], but there the probabilities are defined by the designer. Here on the other hand, we use a more general approach by approximating this

probability as a distance in the belief space. Specifically, in our work, the R_c valuation is performed using *KL Divergence* [6]— the difference in information in an agent's belief state b_1 with and without the communication b_h are compared:

$$R_c(b_1, b_h) = ND_{KL}(b_1 \| b_h) = N \sum_i b_1(i) log \frac{b_1(i)}{b_h(i)} \tag{6}$$

where N is a normalisation factor, b_1 is the agent's current belief state, b_h is the belief state at the time of the last communication and i represents any variable in these belief states (thus the difference is the information content of all observations since the last communication). KL Divergence is chosen because it evaluates all state variables in a single calculation, unlike Fisher Information, the other main information theoretic candidate, which evaluates a variable individually. This is useful because we need to evaluate the reduction in uncertainty given by a single observation, normalised by the uncertainty in the entire belief state. This is because knowing a single variable to a very high precision is not as useful, in our task, as having a rougher estimate of many variables. Essentially, we need to consider all variables at the same time. Furthermore, this calculation is closely related to the Bayesian updating of the POMDP model, making it computationally efficient. Finally, it can also be seen that this is a general valuation function as it is only expressed in terms of observations in a POMDP, thus making it straightforward to apply to a different problem domain.

3.2 Policy Generation

As described in the Section 2, *Real Time Belief Space Search (RTBSS)* represents a good starting point for the policy generation problem, as it has shown good performance in RoboCupRescue. The algorithm, in its original form, coordinates agents using a complex reward function, which rewards the coordinated actions. Now, while this approach is valid for achieving coordination in their particular scenario, it is still necessary to encode explicitly how the agents should coordinate and thus it is not very general. Consequently, we choose to augment their algorithm with the ability to consider joint actions — the actions taken by each team member at each decision point. In more detail, each agent must estimate the actions available to the other agent and consider rewards over these joint actions (see line 12 of Figure 1). This can be achieved by considering the state of the other agents, and does not require modelling their beliefs — which we aim to avoid. Rewards are still calculated in terms of the local interpretation of the state (see line 11 of Figure 1), which allows the agents to make coordinated actions when they have a good idea of the state of the other agent.

4 RoboCupRescue as a *dec_POMDP_Valued_Com*

This section instantiates the *dec_POMDP_Valued_Com* model from the previous section in terms of RoboCupRescue. We first describe the aspects of the RoboCupRescue domain we are interested in, followed by an instantiation of the model. Finally, we present a simple example of rational communication in this domain.

In more detail, RoboCupRescue is a multiagent simulator of the situation in an urban area in the immediate aftermath of an earthquake. 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. 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

```
1:  Function Modified_RTBSS(b, d, rAcc)
2:  b: The current belief state, d: The current time, rAcc: Accumulated rewards.
3:  Statics: D: Search time, bestValue: The best value found in the search, action: The best action.

4:  IF d = 0 THEN                                          if at the horizon of the search
5:    finalValue ← rAcc + γ^D × U(b)                       value of branch plus utility of belief state
6:    IF finalValue > bestValue THEN                       if this is better than the best so far
7:      bestValue ← finalValue                             set to the best so far
8:    END IF
9:    RETURN finalValue                                    return the value of this leaf
10: END IF
11: rAcc ← rAcc + γ^(D−d) × R(b)                           else add reward for Belief state to accumulator
12: JointActionList ← Sort(b, A)                           get next possible joint actions
13: max ← −∞                                               set smallest value
14: FOR ALL a ∈ JointActionList DO                         for each possible joint action
15:   expReward ← 0                                        accumulated reward is 0
16:   FOR ALL o ∈ Ω DO                                     for all observations
17:     b' ← τ(b, a, o)                                    calculate next belief state
18:     expReward ← expReward + γ^(D−d) × P(o|a, b)        current reward + value of subtree
19:       ×Modified_RTBSS(b', d − 1, rAcc)
20:   END FOR
21:   IF (d = D ∧ expReward > max) THEN                    if this is largest so far then
22:     max ← expReward
23:     action ← a                                         best action is current action
24:   END IF
25: END FOR
26: RETURN max                                             return the value for this subtree
```

Figure 1: Modified RTBSS

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 a highly efficient search strategy), and so we will constrain the problem. To this end, we will only consider the ambulance agents' task — that is, we will remove fires and road blocks, and consequently remove the fire brigade and police agents. 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). Several elements need to be defined from the point of view of the ambulance agents. Firstly, we model just two ambulance agents, a_1 and a_2, to keep the following example clear. The state S describes whether buildings contain trapped civilians or not, and also the position of the two ambulance agents, who can be in any buildings, 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). At each time step, agents select joint actions (an action is assigned to each team member) and implement their own part of that joint action. The rescue model has been altered from the standard seen in the competitions — the amount a civilian is dug out is now sub-linear with the number of ambulances digging. Consequently, a team of agents does much better than when the agents work individually. We did this 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 this communication C_Σ relates to the time required to

send the observation history. The reward function R_p gives a reward for each civilian rescued and building explored. R_c and R are defined as the general formula from Section 3.1. The observation function $\Omega_i = \Omega$ supplies each agent with the state of buildings nearby (i.e. whether these contain trapped civilians) and the location of the other agent if it close enough. The communication alphabet $\Sigma = \Omega_i = \Omega$, and so a message can be composed of any symbol in the observation alphabet. A summary of this formalisation is given in Figure 2.

Component	Representation	Example
S	Buildings can contain zero or more civilians and each of the 2 ambulances can be at any building, road or node. On typical maps there are approximately 700 buildings, 600 roads and 1000 nodes. This leads to a state space of approximately $2^{700} \times (700+600+1000)^2$ which is too large for offline computation	Any state is a complete enumeration of all variables $\langle a_1 \Rightarrow b_1, a_2 \Rightarrow n_8, b_0 \Rightarrow 0 \dots b_i \Rightarrow 1, n_0 \Rightarrow 0 \dots n_m \Rightarrow 1, r_0 \Rightarrow 0 \dots r_j \Rightarrow 1\rangle$ where i is the number of buildings b, m the number of nodes n and j the number of roads r
A_i	Each agent can move to an unexplored building, it can also load and unload civilians, and communicate	A move from Building b_k to Node n_o by agent a_1 will change the value of the variable $a_1 \Rightarrow n_o$
Σ_i	The alphabet of communications is the history of observations from the last communication	A communication can be null or any set of observations $\langle p(b_j = civ) = 0.0, p(b_k = civ) = 1.0, p(b_j = a_1) = 1.0\rangle, \langle p(b_k = a_2) = 1.0, p(b_j = a_2) = 1.0\rangle, \langle p(b_r = civ) = 1.0\rangle\rangle$
C_Σ	This cost is 0 for the null communication, and one timestep for all other communications	If the nearest non-blackout is n_0 then $C_\Sigma(a_1 \Rightarrow n_0) = 1$ timestep. If the nearest non-blackout is n_1 and it takes 2 timesteps to move $a_1 \Rightarrow n_1$ then $C_\Sigma(a_1 \Rightarrow n_0) = 3$ timesteps.
P	Defined by the simulator.	
R_p	$R_p = c \times r + e \times r/2$ (7) where r is a normalised reward (100), c is the number of civilians rescued, and e is the number of observed buildings (to encourage exploration)	If $c = 10$ and $e = 50$ then the reward is 3500 (from equation 7) but if $e = 40$ then the reward is 3000, giving a higher reward for exploring more
R_c	$R_c(b_1, b_h) = ND_{KL}(b_h \| b_1) = N \sum_i b_h(i) log \frac{b_h(i)}{b_1(i)}$	The belief state for a single building $b_1 = b_i \Rightarrow [0.5, 0.5]$ and the communication $b_h = b_i \Rightarrow 1$, with $N = 1000$ results in $R_c = 300$
R	$R = \alpha R_p + (1-\alpha) R_c$	If $R_p = 3000$, $R_c = 300$ and $\alpha = 0.8$ then $R = 2460$
Ω_i	In this case the ambulances can observe the state of any building within some range and the position of the other agent within that range. This is corrupted with some noise	Building b can be observed to contain civilians $p(b = civ) = 1.0$ or empty $p(b = civ) = 0.0$. Ambulance agent a_i is observed to be at some some Building b, Road r or Node n. Any observation is a set of these variables with values $\langle p(b_j = civ) = 0.0, p(b_k = civ) = 1.0, p(b_j = a_1) = 1.0\rangle$
O	Defined by the simulator	
T	5	
$\mathcal{A}$	There are 2 ambulance agents, with the joint action $J_i \in \mathcal{A}$	a_1 and a_2
b_i	The belief state for agent i is a probability distribution over the possible values of each state variable	$\langle p(b_j = civ) = 0.5, p(b_j = nociv) = 0.5, p(a_1 = b_k) = 1.0\rangle$

Figure 2: A dec_POMDP_Valued_com of the RoboCupRescue ambulance task

4.1 A Coordination Example

We will demonstrate this model with a simple coordination task. Two ambulances, a1 and a2, must rescue a civilian from a building b1, on a map composed of two buildings with a road r1 connecting them. Agent a1 is in b1 and has previously observed the civilian (*civ*) in b1, a2 is in b2 and does not observe any civilians. We consider the action selection for agent a1 as in Figure 3, and demonstrate its search to a depth of one. The agent must choose between attempting to rescue the civilian and communicating its existence. It is assumed that once both agents know about the civilian they will cooperate to save it. Other parameters are as described in Figure 2.

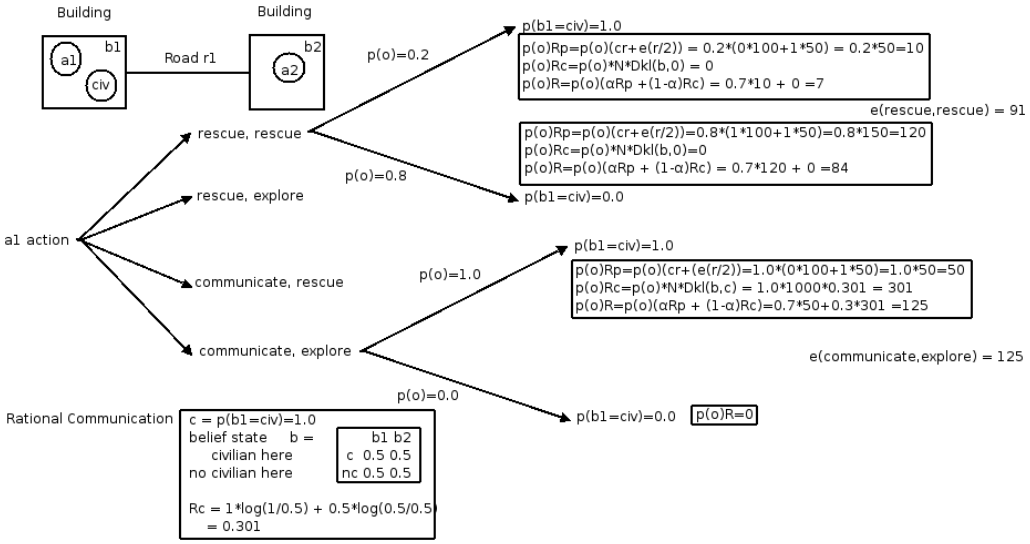

Figure 3: An execution example

Considering the example, the agent must choose between four joint actions:

$$\mathcal{A} = \langle\langle rescue, rescue\rangle, \langle rescue, explore\rangle, \langle communicate, rescue\rangle, \langle communicate, explore\rangle\rangle$$

The first action in each tuple represents the action taken by a1 and the second is taken by a2 concurrently. We demonstrate the calculations for $J_1 = \langle rescue, rescue\rangle$ and $J_2 = \langle communicate, explore\rangle$, as these are the most illuminative. After a1 implements its part of the joint action it will receive one or more observations according to equation 3. In the following, the probability of observation o is denoted by $p(o)$. In this example, this observation is always related to whether building b1 contains a trapped civilian ($p(b1 = civ) = 1.0$) or not ($p(b1 = civ) = 0.0$). In our example we do not need to utilise the full algorithm in Figure 1. Instead, we summarise the expected reward over observations and joint actions with the following equation:

$$e(J_i) = \sum_{o \in \Omega} p(o) R(J_i, o) \tag{8}$$

where we restrict the summation to only those observations o which satisfy $p(o) > 0$. This is valid because we are using a myopic example. Consequently, we must calculate R (by equation 5) for each joint action/observation pair. Calculating R requires values for R_p (by equation 4) and R_c (by equation 6), which we will describe in more detail for J_2 and the observation $p(b1 = civ) = 1.0$.

In this case, R_p uses the instantiation from Figure 2 which relates rewards to the number of civilians rescued and buildings explored (see equation 7 in Figure 2). Initially, one building has been explored and no civilians rescued, giving $R_p = (100 * 0 + 50 * 1) = 50$. Furthermore, $R_c = ND_{kl}(b, c)$ from equation 6, where b is the initial belief state for a1 which has no information about whether there are trapped civilians in each building — all probabilities are uniform. The communication c that we measure is the observation of a trapped civilian in b1 ($p(b1 = civ) = 1.0$) and in this example $N = 1000$. Thus the information gain (using KL Divergence) in that observation is 301 (which is scaled in line with R_p from equation 6). Since $R = \alpha R_p + (1 - \alpha) R_c$ (see equation 5), this gives $R = 0.7 * 50 + 0.3 * 301 = 125$ where $\alpha = 0.7$ in this example. Using equation 8, $e(J_2) = 1.0 * 125 = 125$ and $e(J_1) = 91$ (see Figure 3 for the calculations for this joint action) which means that a1 chooses to

communicate c. These calculations show that agent a1 expects to gain more reward (125) by communicating the existence of the civilian than assuming the other agent knows about it and attempting to rescue (91). Essentially, the normalised KL Divergence value of the communication represents the extra reward that is obtained by influencing the probability of mis-coordination inherent in assuming the other agent believes the same as the communicating agent (which is what our coordination algorithm does). If this value is lower than other actions or the cost of communicating, this indicates that it is not worth attempting to influence that probability.

If we consider our example further, the agent might choose to communicate the null observation (the probability of a civilian in any building is equal to the probability of no civilian, so $\langle p(b1 = civ) = 0.5, p(b2 = civ) = 0.5\rangle$). As per equation 6, this would have resulted in 0 for the same b. In this case communication would not have been selected. Furthermore, it also shows that on the next timestep, the value of communicating about the civilian will have dropped and the agent will rescue instead. This is because b will have changed to include knowledge of $p(b1 = civ) = 1.0$ and $R_c = 0$. These two examples show the rationality of our valuation — communicating zero information has no value and communicating previously communicated information also has zero value.

Furthermore, if the value of α was less than the value used here (0.7) then communication would not be used in this scenario, but ultimately the agents would take longer to save the civilian. Similarly, if α was greater then the agent would communicate too much and again the team would do less well. This shows the importance of setting the correct normalisation between the actual rewards for solving the problem R_p and the virtual rewards for communicating R_c. To this end, in Section 5, we experiment with a range of α values.

5 Experimental Evaluation

We now evaluate our coordination mechanism and communication valuation in the context of RoboCupRescue. We will first describe our experimental setup, including performance measures and experimental variables. Following this, we will establish an upper and lower bound on the performance of our coordination mechanism in this problem. Finally, we present results for our valuation mechanism and compare this to a benchmark. These results include a comparison of performance with varying communication availability — a common feature of many coordination scenarios that existing work ignores.

5.1 Experimental Setup

In these experiments, we compare four communication policies — two of these (**Zero** and **Full**) are designed to establish a lower and upper bound for the standard coordination problem, and between these we analyse our mechanism (**Valued**) for valuing communications and a simple benchmark solution (**Selective**):

- **Zero**: the agents do not communicate with each other, and essentially solve the problem in isolation.
- **Full**: the agents send a communication to each other containing their last observations at each time step (making communication effectively free). More formally, agent a_i receives observation o at timestep t. At timestep $t + 1$, a_i chooses an action and communicates o to all other agents, who receive it at timestep $t + 2$. This is equivalent to a centralised solution, because the agents have full knowledge of the state of the other agents and so they are all calculating the solution to the same single-agent POMDP.

- **Selective**: the agents can choose to communicate all observations since their last communication action at each time step, but doing this has a cost. Specifically, this cost is incurred because the agents cannot take any other actions whilst communicating. Here communication is an option, and at the same time a simple static domain valuation is used to estimate the reward that communication represents. This value increases with a constant each time the agent does not communicate, and resets to 0 when communication is employed. More formally, initially $R_c = 0$ and at each timestep t, $R_{c_{t+1}} = R_{c_t} + c/10$ where c is the value of rescuing a civilian from R_p, and $\alpha = 0.5$ giving equal weighting to each reward function. If communication is used then $R_{c_{t+1}} = 0$. We use these values for c and α as these result in the greatest performance after empirical analysis (which we do not present here due to space constraints). Furthermore, agent a_i has a history of observations since the last time it communicated b_h. At timestep t, a_i receives observation o and appends this to its history $b_h \Rightarrow b_h + o$. At timestep $t+1$, a_i chooses an action, including the option of communicating b_h. If communication is taken then $b_h \Rightarrow \varnothing$ and $R_c = 0$.

- **Valued**: the agents can select to communicate as in Selective communication, but the information content of the message is used to value the communication actions and a parameter controls the mixing of this with problem rewards. Thus, communication is an option, but now the agents can evaluate whether it will be helpful. More formally, agent a_i has a history of observations since the last time it communicated b_h. At timestep t, a_i receives observation o and appends this to its history $b_h \Rightarrow b_h + o$. At timestep $t+1$, a_i chooses an action, including the option of communicating b_h and which will have a value $R_c = N \times Value(b_i, b_h) = N \times D_{KL}(b_i \| b_h) = N \sum_i b_i(i) log \frac{b_i(i)}{b_h(i)}$ where b_i is the belief state of agent a_i and N is a normalisation factor. If communication is taken then $b_i \Rightarrow \varnothing$. We define the relationship between the two reward functions as $R = \alpha R_p$ and $(1-\alpha)R_c$. Intuitively α controls the relative importance of communicating to problem solving, and we explore its value empirically.

To summarise, communication is completely free in **Full** — hence it is used all the time; the agents never communicate in **Zero**; **Selective** and **Valued** both use the model of communication valuations but **Selective** uses a constant reward per timestep, whereas **Valued** uses an information valuation over the agent's knowledge and possible communications.

In these experiments we measure performance as the number of civilians moved to refuges by the end of the simulation run. Each test run starts on the same map with random placement and status of civilians. The ambulance agents always start in the same place. Maps could be generated randomly, but we hold that this does not add any validity to our method, since the map used represents a standard competition map which has not been altered to favour our approach. Furthermore, generating random maps can add noise to the process as ambulance agents can start off trapped in collapsed buildings and we have not considered this scenario at this stage.

When considering **Full**, **Zero** and **Selective**, the dependent variables are graphed with respect to simulation timestep and mean behaviour is compared directly. When considering **Valued**, the mean summaries of the experimental variables are graphed with respect to the α values employed. In more detail, for the dependent variables it is useful to obtain a figure summarising the entire simulation performance. This is useful when evaluating the impact of α on the other simulation variables, as we need to compare performance across the entire space of α. This takes the form of the dependent variable values after 300 time steps in each run. In general 30 runs are performed for statistical significance, which is computed using a standard t test for the 95% confidence interval — ensuring the error statistic is less than 0.005. α is explored between 0 and 1 with increments of 0.1, interpolating in-between.

5.2 Results

Initially, we compare civilians saved by the end of the simulation in **Full**, **Zero** and **Selective**, in order to investigate the upper and lower bounds on our coordination method. As Figure 4 shows, **Full** performs the best because the agents model each other at all times (they have a centralised view of the problem) — this means that the agents actions are always coordinated. By way of contrast, the agents do not coordinate well in **Zero** because they do not model each other accurately. Hence they duplicate the areas of the map they have searched and do not dig co-operatively. In **Selective**, the agents do a little better because communication happens periodically. Still, the agents do not communicate efficiently, since this algorithm assumes the agents gather information at a constant rate — which is clearly not true because an agent is not compelled to gather new information.

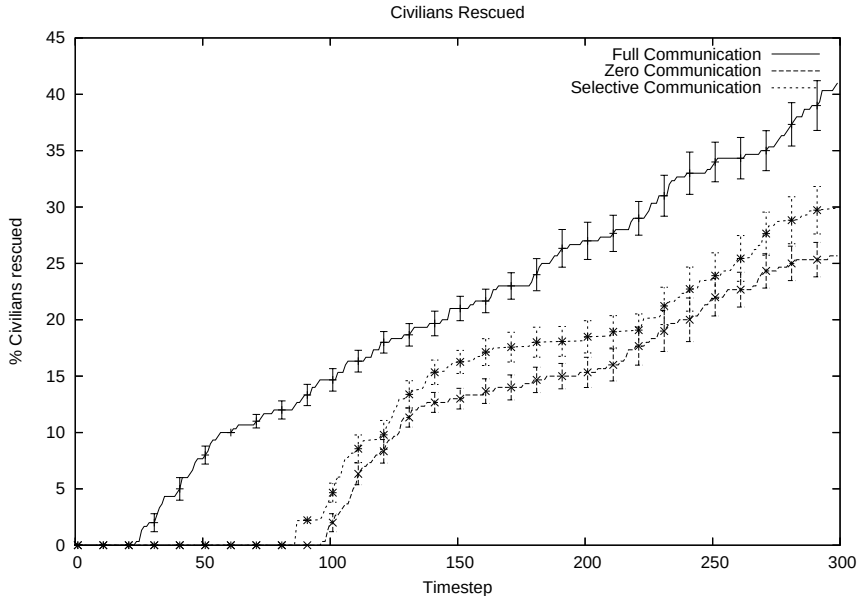

Figure 4: Percentage of civilians rescued during the simulation averaged over 30 runs

Given these bounds on performance, we now investigate the utility of valuing communications in our **Valued** model. In more detail, our model requires us to mix the rewards from acting in the problem with rewards in communicating — denoted by R_p and R_c respectively. It is interesting to consider how these should be mixed, to find where maximal performance occurs. To this end, α controls the relative importance of solving the problem R_p, against information dispersal R_c and will vary from 1 (only assign reward to solving the problem) to 0 (only assign reward to dispersing information). It can be seen in Figure 5 that for a range of α values, the performance of **Valued** with no restrictions on communication availability (*Valued 0% Blackout*) approaches **Full** (the communication time requirements make this an unrealistic assumption). When $\alpha = 0$ the agents communicate all the time (leading to very low performance), and when $\alpha = 1$, the agents never communicate, reducing it to **Zero** (although the points do not meet exactly due to the noise introduced by extra actions — communication). When comparing **Valued** with **Selective**, it is clear that both can be used to value communications appropriately but **Valued** is more efficient and leads to a higher team utility. This is because **Selective** assumes a constant information gain with time which is not the case — **Valued** measures the information gain before deciding whether to communicate.

With the utility of our method established in the simple case, we now consider the impact

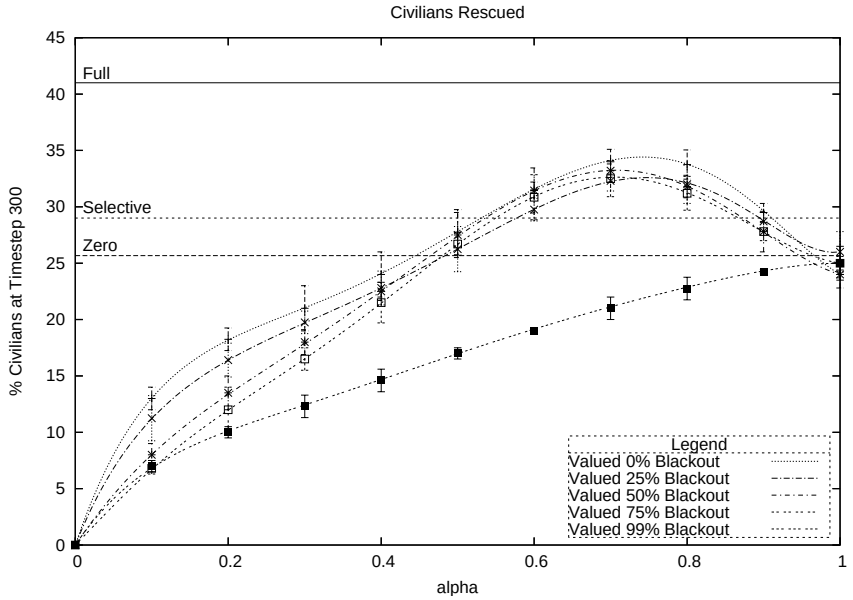

Figure 5: Percentage of civilians rescued at the end of the simulation averaged over 30 runs

of communication restrictions. The communication restrictions we describe here more realistically model the sorts of communication conditions found in many real problems, and we are interested to see if our mechanism is robust to these restrictions. Here we define 'blackouts' over some areas of the RoboCupRescue maps, where it is impossible for the agents to communicate. If an agent chooses to communicate within a blackout area, the agent first moves to the nearest point where communication is available. This area is defined randomly as a number of points on the map, and within a small radius of these points the blackout exists. We experiment with a range of blackout volume (25%, 50%, 75% and 99%). We perform the same experiments as with unrestricted communication and present the results in Figure 5.

For blackouts ranging from 0-75%, the change in response to the α parameter is not statistically separable. It is clear that whilst the overall performance hardly changes, the shape does — reflecting a change in the value of communication because of the higher cost when there are restrictions. With a blackout covering 99% of the map, performance is drastically impacted because of the increased time involved in travelling to an area where communication is possible. Consequently, performance never exceeds **Zero**. This suggests that when communication is very expensive, it is better to try to solve the problem in isolation.

6 Conclusions

We develop a model of *rational communication* that can evaluate the usefulness of communicating to the team using an information theoretic measure. This is combined with a decentralised decision theoretic coordination mechanism to balance the cost of communicating with the benefit of communicating. We then implement this in terms of RoboCupRescue and compare our approach with a centralised version, a non-cooperative team, and communicating selectively with no information about the importance of that communication.

In more detail, 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. We also demonstrate that our model

is robust to severe communication restrictions, which existing work assumes will not occur. This represents a first step towards a robust, scalable coordination mechanism which is able to employ communication rationally.

Our experiments form a useful empirical validation of this technique, and in future work we intend to extend the model and analyse its theoretical properties. In more detail, we need to demonstrate that a parameterised information theoretic reward function for communications is a valid approximation to analysing joint beliefs in the decentralised POMDP model. This includes proving the rationality of our valuation function. Furthermore, we want to be able to calculate the value of communications as a function of the problem specification and the communication language employed. This would then enable us to analytically define the optimal α rather than obtaining this value empirically. Both of these aspects would prove the utility of this technique for dramatically reducing the complexity of decentralised POMDPs and increasing their applicability beyond the RoboCupRescue domain. Finally, we envisage this work being useful in domains where communication costs are dynamic. Consequently, it would be interesting to consider how to learn the approximation online.

Acknowledgements

We would like to thank Dr Jeremy Baxter for his insightful comments. This work was supported by an industrial CASE studentship funded by EPSRC and QinetiQ.

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] K. Decker and V. Lesser. Generalizing the partial global planning algorithm. *International Journal on Intelligent Cooperative Information Systems*, 1(2):319–346, June 1992.

[3] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun. Approximate solutions for partially observable stochastic games with common payoffs. In *AAMAS '04: Proceedings of the 3rd international joint conference on Autonomous agents and multiagent systems*, pages 136–143, New York, USA, 2004.

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

[5] N. R. Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, 75(2):195–240, 1995.

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

[7] S. Paquet, L. Tobin, and B. Chaib-draa. An online pomdp algorithm for complex multiagent environments. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 970–977, The Netherlands, 2005. ACM Press.

[8] D. V. Pynadath and M. Tambe. Multiagent teamwork: Analyzing the optimality and complexity of key theories and models. In *AAMAS'02: Proceedings of the First Autonomous Agents and Multiagent Systems Conference (AAMAS)*, pages 873–880, Bologna, Italy, 2002.

[9] A. Rogers, N. R. Jennings, and E. David. Self-organized routing for wireless micro-sensor networks. *IEEE Transactions on Systems, Man and Cybernetics - Part A*, 35(3):349–359, 2005.

[10] S. Ross and B. Chaib-draa. Aems: An anytime online search algorithm for approximate policy refinement in large pomdps. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 2592–2598, Hyderabad, India, 2007.

[11] M. Roth, R. Simmons, and M. Veloso. Reasoning about joint beliefs for execution-time communication decisions. In *AAMAS '05: Proceedings of the 4th international joint conference on Autonomous agents and multiagent systems*, pages 786–793, The Netherlands, 2005.

[12] J. Shen, V. Lesser, and N. Carver. Minimizing communication cost in a distributed bayesian network using a decentralized mdp. In *AAMAS'03: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 678–685, Melbourne, Australia, 2003.

[13] D. Szer and F. Charpillet. An optimal best-first search algorithm for solving infinite horizon dec-pomdps. In *ECML'2005:Proceedings of the 16th European Conference on Machine Learning*, pages 389–399, Porto, Portugal, 2005.

[14] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

[15] P. Xuan, V. Lesser, and S. Zilberstein. Communication decisions in multi-agent cooperation: model and experiments. In J. P. Müller, E. Andre, S. Sen, and C. Frasson, editors, *AAMAS '05: Proceedings of the 4th international joint conference on Autonomous agents and multiagent systems*, pages 616–623, The Netherlands, 2001. ACM Press.

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

[17] S. Zilberstein and C. V. Goldman. Optimizing information exchange in cooperative muti-agent systems. In *AAMAS'03*, pages 137–144, Melbourne, Australia, 2003.