UNIVERSITY OF SOUTHAMPTON
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

Rational Communication for the Coordination of Multi–Agent Systems

by Simon A. Williamson

Supervisors: Prof. Nicholas R. Jennings and Dr. Enrico H. Gerding
Examiners: Dr Jeremy Pitt and Dr. Jason Noble

A thesis submitted in partial fulfillment for the degree of
Doctor of Philosophy

December 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Simon A. Williamson

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 the more traditional centralised systems. Whilst this approach has many benefits in terms of creating robust solutions, it creates a new challenge — how to flexibly coordinate the actions of the agent teams to solve the problem efficiently. In more detail, coordination is here viewed as the problem of managing the interactions of these autonomous entities so that they do not disrupt each other, can take proactive actions to help each other, and take multiple actions at the same time when this is required to solve the problem.

In this context, communication underpins most solutions to the coordination problem. That is, if the agents communicate their state and intentions to each other then they can coordinate their actions. Unfortunately, however, in many real-world problems, communication is a scarce resource. Specifically, communication has limited bandwidth, is not always available and may be expensive to utilise. In such circumstances, typical coordination mechanisms break down because the agents can no longer accurately model the state of the other agents. Given this, in this thesis, we consider how to coordinate when communication is a restricted resource. Specifically, we argue for a rational approach to communication. Since communication has a *cost* then, similarly, we should be able to calculate a *value* of sending any given communication. Once we have these costs and values, we can use standard decision theoretic models to choose whether to send a communication, and in fact, generate a plan which utilises communications and other actions efficiently.

In this research we explore ways to value communications in several contexts. Within the framework of decentralised Partially Observable Markov Decision Process (POMDP) we develop a simple information theoretic valuation function (based on Kullback–Leibler (KL) Divergence). This techniques allows agents to coordinate in large problems such as RoboCupRescue, where teams of ambulances must save as many civilians as possible after an earthquake. We found that, in this task, valuing communications before deciding whether to send them results in a level of performance which is higher than not communicating, and close to a model which utilises a free communication medium to communicate all the time. Furthermore, this model is robust to increasing communication restrictions, whereas simple communication policies are not.

We then extend this framework to value communications based on a technique from the field of Machine Learning, namely *Reward Shaping*, which allows the decentralised POMDP to be transformed into individual agent POMDPs that can be solved more easily. This approach can use a heuristic transformation to allow the approach to work in large problems like RobocupRescue or the Multi–Agent Tiger problem, where it outperforms the current state of the art. Further to this, this approach can also use an exact reward shaping function in order to generate a bounded approximation of the intractable optimal decentralised solution in slightly smaller problems.

Finally, we show how, if we restrict our attention to relatively static (i.e. the problem does not change without an agent doing something) problems than those which the reward shaping technique was designed for, we can generate an optimal solution to decentralised control based on communication valuations. In more detail, we extend the class of *Bayesian coordination games* to include explicit observation and communication actions. By so doing, the value of observation and exchange can be derived using the concept of opportunity costs. This is a natural way of measuring the relationship between communication and information gathering on an agent's utility, and removes the need to introduce arbitrary penalties for communicating (which is what most existing approaches do). Furthermore, this approach allows us to show that the optimal communication policy is a Nash equilibrium, and to exploit the fact that there exist many efficient algorithms for finding such equilibria in a local fashion. Specifically, we provide a complete analysis of this model for two–state problems, and illustrate how the analysis can be carried out for larger domains making use of explicit information gathering strategies. Finally, we develop a procedure for finding the optimal communication and search policy as a function of the partial observability of the state and payoffs of the underlying game (which we demonstrate in the canonical Multi–Agent Tiger problem).

In performing all of this work, we demonstrate how communication can be managed locally by accurately placing a value on the cost and benefit of using a restricted communication resource. This allows agents to coordinate efficiently in many interesting problem domains, where existing approaches perform badly. We contribute to the field of rational communication by providing several algorithms for utilising costly communication under different domain conditions. Our reward shaping approaches are highly scalable in problems with large state spaces and come with sound theoretical guarantees on the optimality of the solution they find.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisors, Nick Jennings and Enrico Gerding for all of their guidance, help, discussions and corrections. I always felt that I was on the right course, whilst still being independent. Your supervision has really equipped me with the skills needed to carry on without you. My thanks also go to my industrial sponsor Jeremy Baxter for all his thoughts during my time on placement (and for offering me the job that got me to Southampton in the first place!).

I would also like to thank all of my friends and colleagues in Southampton and at other institutions for providing a excellent research environment and nice place to spend too many hours. I would especially like to thank my generation: Thanasis; Rama and Archie for always making sure there was somebody who had done the bureaucracy first, so that I could copy. It was always good to know that there were other people going through the same process.

Finally, I would like to thank my parents, sister, grandfather and the rest of my family for their unending support. I would not be writing this now without their belief that I could go as far as I wanted.

For my parents

Chapter 1

Introduction

In this thesis we consider how to solve decentralised control problems where independent decision makers must *coordinate* to solve a joint problem in domains where they cannot communicate with each other freely. In particular, we explore ways to coordinate teams efficiently when communicating is a restricted resource. Specifically, we consider cases where communication is costly, in terms of time or opportunity, and argue for *rational communication — the ability to dynamically evaluate the benefit of coordination (through communication) against the cost of achieving it*. For example, consider the case where a rescue robot must decide whether or not to pull together a team of robots to search in a burning building to find civilians. Its chances of succeeding are increased if it has assistance from the other robots, but it must leave the building to ask for such help because the fire may have destroyed the building's communication infrastructure. Given this, the coordinating robot must evaluate the probability that the other robots already know the building needs searching, and so will arrive at the scene based on local information, or whether they explicitly need to be told of this fact, in order to gauge the benefit of communicating or not.

In particular, we cast this problem as one of multi–agent coordination. Here, agents are defined as autonomous, goal-directed, reactive problem solvers capable of communicating and interacting with one another (Wooldridge, 2002). Multi–Agent Systems (MAS) research is thus concerned with managing the interactions of several such autonomous software agents (Wooldridge, 2002). These agents may represent different stakeholders in the environment, in which case they may compete to maximise their individual utility. Alternatively, the agents may act cooperatively and try to solve a problem as a team. The latter is the view of MAS taken in this research. Now, this definition of a multi–agent system captures many problems in real-world decentralised control and in this research we are particularly interested in developing local algorithms for agents which allow them to efficiently solve such decentralised control problems using local information and communication.

Against this background, in many real-world domains, teams of software agents must attempt to solve some global problem together. Relevant examples include distributed sensing (Lesser

and Erman, 1980), network routing (Dutta et al., 2005), air-traffic control (Ljungberg and Lucas, 1992), disaster response (Hiroaki, 2000), space exploration (Estlin et al., 2005) and unmanned vehicles (Karim and Heinze, 2005). In many of these cases, the agents have heterogeneous capabilities and, in general, the problem cannot be solved by just one of them because it requires some interaction of their disparate abilities. Given this, the coordination problem — how to make a team of agents act together in a coherent goal-directed manner — is a central concern in the field of MAS. Specifically, effective coordination requires agents to anticipate the needs of teammates in terms of actions and information and manage the interdependencies between their various activities. Relevant examples from Wooldridge (2002) include:

- You and I both want to leave the room, and so we independently walk towards the door, which can only fit one of us. I graciously permit you to leave first.
- I intend to submit a grant proposal, but in order to do this, I need your signature.
- I obtain a soft copy of a paper from a Web page. I know that this report will be of interest to you as well. Knowing this, I proactively photocopy the report, and give you a copy.

Such coordination can, in some cases, be managed by a central controller (see Gerkey and Mataric (2001) or Khoshnevis and Bekey (1998) for example), but this would require a reliable communication medium and represents a single point of failure for the entire system. Given these shortcomings, an alternative approach, that will be adopted here, is based on decentralised control, where the interactions between the agents are managed locally (Jennings, 2001). In such systems, agents make independent decisions based on their histories of observations, communications and utilities (this is in contrast to centralised systems that attempt to assess the global state of the problem and assign an action to each agent). In general, a decentralised system is more robust because single points of failure are avoided and agents can be removed or added more easily (in particular, the computational complexity of naïve action selection algorithms will increase exponentially with the size of the team (Papadimitriou and Tsitsiklis, 1987)). However, decentralised control systems are more complex to design. This complexity is introduced because the agents do not have a global view of the system (due to limited sensing, computational and communication capabilities) unlike a central coordinator which can ascertain the state of all agents and whether their tasks were completed successfully. To help alleviate this partial perspective, the agents can communicate state information and intentions (Tambe, 1997; Gmytrasiewicz and Durfee, 2000; Shen et al., 2003), but as we will show, this process must be done efficiently if the system is to perform well.

To this end, our aim in this work is to produce agent teams that are capable of coordinating in domains where the communication resource is limited or may vary in its availability and cost during the course of the team's activity. In particular, we believe that accounting for the communication medium is important because, in most real world cases, it is never completely reliable and many traditional coordination techniques break down in the face of communication failures (see Chapter 2 for a more detailed review). Moreover, many problem domains have

inherently limited communication resources and so it is important these are used effectively. For example:

- In physical agents, communication may require drawing on a finite power source which limits the number of times it can be utilised, a problem exacerbated if the agent is using that same source for its other activities.
- In disaster recovery scenarios, the communication medium may become saturated by a sudden increase in demand as many civilians make calls on the cellular network or emergency workers use the same radio frequencies.
- In military applications, communication may be inherently dangerous because it reveals the location of the agent to the enemy.

Finally, communication may simply be expensive because the agent cannot take any other action whilst it is communicating (e.g. unmanned water vehicles can only communicate at the surface, whilst their goal can only be completed underwater) or it may have had to take several actions to make communication possible (e.g. the same unmanned water vehicles going to the surface).

Thus, under all of the above conditions, we believe coordination should account for the cost of communicating and the explicit actions taken in the course of communicating. Specifically, we believe an agent needs to be able to understand and compare the value, to both itself and the team, of the information that it is considering sending. In order to do so, however, it needs to have an estimate of the effect that the information will have on the team's mission and the likelihood that other team members will independently discover the same information. By way of illustration, consider the following example from the disaster rescue domain seen in RoboCupRescue (Hiroaki, 2000):

> A team of search and rescue robots must search an urban area in the immediate aftermath of an earthquake and rescue trapped civilians. They must work together to search the area efficiently and must also cooperate to remove trapped civilians to safety. These robots communicate using the local mobile phone network, but in an earthquake many of the area's inhabitants immediately call for help. This saturates the network and introduces communication blackspots (where no communication is possible). Rescue robots enter these blackspots and assess how many civilians need rescuing. They must leave the blackspot to report this to the rest of the team but this requires a substantial amount of time, during which the agent is not rescuing civilians or performing more search.

We now give a second example from the Multi–Agent Tiger problem (Nair et al., 2003):

> A team of agents have the choice to open one of two doors — a left door and a right door. Behind one door is a large treasure and behind the other is a tiger. Agents

> want to obtain the maximum reward by both opening the door containing the treasure. Further to this, if an agent opens the door containing the tiger then both agents incur a large penalty — the penalty is reduced if both agents open the door with the tiger. The agents do not know which door contains which outcome and so they can request noisy observations of where the tiger is — they can independently hear that the tiger is on the left or right. Now, agents choose to open either door or listen independently, but they should agree on which action to take. Communication can be used to tell each other about their beliefs of where the tiger is, however communication itself has a cost and takes as much time as listening or opening a door. Consequently, agents can choose to individually ascertain where the tiger is before opening a door (but this could lead to mis-coordination with noisy observations) or they can communicate to achieve a global view. Depending on the cost of observations verses communicating, different strategies are optimal, and agents should be able to choose correctly when employing rational communication.

The Multi–Agent Tiger problem is the canonical form of the choice between acting with local information or communicating to achieve a synchronised view of the world. This decision is found in all problems and the Tiger problem is a useful abstraction for studying that choice. We now give a third example from a military application (Baxter, 2006):

> A team of unmanned air vehicles is carrying out a mission to clear a corridor through hostile air defences to allow a pair of manned aircraft to attack a fixed target. Once inside hostile airspace, communication is to be inhibited since transmitting may alert the air defences to the vehicles' presence. One vehicle detects a previously unknown air defence system. It needs to be able to decide whether, and how, to communicate this to the other members of its team. Thus, the vehicle may transmit a warning on high power immediately, with a high probability of detection, or it may divert from its planned route to get closer to another vehicle and transmit a warning on low power, which is less likely to be detected. Finally, it may decide not to transmit the information because it thinks the other aircraft already know about the defence, or it decides that this information will not affect the success of the mission.

Finally, we can see this problem in one of the coordination examples from Wooldridge (2002):

> I obtain a soft copy of a paper from a Web page. I know that this report will be of interest to you as well. Knowing this, I proactively photocopy the report, and give you a copy. However, I would not need to make this communication if I knew that you had already viewed that Web page.

To make the decision between the different forms of communication, or not communicating at all, the agent needs to be able to understand and compare the value, to both itself and the team, of the information. In order to do so, it needs to have an estimate of the effect the information will have on the team's mission and the likelihood that other team members will independently discover the same information. In other words, it needs to understand what effect all of the options will have on both its own ability to complete its task and for the team's task as a whole. For example, in the UAV task, diverting to a safe location to communicate may prevent the vehicle from completing its assigned task in the agreed time frame, but not communicating may lead to another vehicle being detected and attacked by the air defences.

Following this, the other common elements of these examples are that the domains are inherently partially observable. This means that the agents (together or individually) cannot observe the full current state or the impact of taking an action. This feature is intimately linked with the value of communication since agents' will, generally, have different views of the problem. Further to this, parts of the problem may change outside of the agents' control indicating that there is an element of dynamism to contend with. Finally, actions may not be successful or may have unexpected consequences (which may only be partially observable). As a result, the environment is inherently stochastic. Now, all of these features influence the coordination problem and furthermore, the value of communication. Thus any solution should take them into account.

Indeed, although we address the problem within the specific realms of multi-agent systems, the problem itself has a much broader interpretation. For example, within the realm of social psychology the influence of communication is seen in the notion of mental models of other actors (such as the recursive models seen in Goodie et al. (2009)). Here the issue of how the semantics of communication changes the model of the other actor is a key question. Further to this, there is the notion of whether communication limits the need to recursively model other actors since it forms a commonly known information set. In such cases, if the impact of sending a communication is properly understood then this would influence how much information about about the other agent and its reasoning it is required to know.

Following this, communication between humans is also closely related to how communication can be treated formally in the open multi-agent systems domain. Specifically, in the examples given thus far, we have only considered closed systems where agents have common stakeholders and share a common understanding of the problem (because they have been developed together or according to an agreed standard such as FIPA-ACL[1]). However many multi-agent systems do not share this property, and agents have very different architectures. In this case, the problem of sending a communication must be expanded to include some understanding of how the other agent will use the communicated information. In our solution we will leverage the fact that we know what each agent will do with a communication, although we cannot do this in the general case.

[1] http://www.fipa.org

Further to this, examples of our problem can also be found in economic theory where communication expands the set of Bayes-Nash equilibria (but perhaps at a cost) (Krishna, 2007). This property is also found in distributed constraint optimisation where there is a minimum amount of messages required for an optimal solution (Petcu and Faltings, 2005). We can relate our problem to this one by considering what solution could be achieved if the available communication was less than this minimum amount. Similarly, in biologically inspired artificial intelligence such as Bonabeau et al. (1999), communication is often implicitly utilised (e.g. with pheromones) but there exists the question of how well such a system would operate if the amount of pheromone was limited in some way.

However the area in which we can most fully expand this line of enquiry is in the area of bounded rationality. In more detail, bounded rationality (Simon, 1955) is the concept that an agent's ability to make a decision is limited by its processing power, time or information. In this sense, it is clear that the value of communication is closely related to the information constraint of bounded rationality. Even more than this, as the amount of communication is allowed to increase, agents must reason less about each other and consequently processing power or time can be reduced. As a result, in any distributed system with limited communication between the nodes, whether an optimal solution can be found is bounded by the limits on communication. Understanding this very fundamental problem is vital to the development of general artificial intelligence.

Put in more general terms, in all of these examples and areas, the agents can make their communication decisions based on pre-determined rules, made before the team begins the task, or they may need to be able to plan for each possible action and evaluate its short and long term consequences. In either case, however, the agents need a way to compare what has to be done to achieve communication and the benefit to itself and the team of doing so. Essentially then, the question considered in this work is:

> *how to dynamically evaluate the benefit of coordination, through communication, against the cost of achieving coordination.*

1.1 Research Aims and Challenges

This work aims to show the benefit of a *rational approach to communication* and, further to this, we also propose a way to deal with the challenge of a costly communication medium efficiently. That is, a model is developed which assigns a value to possible communications, and this value is balanced against the cost of communicating. In particular, this section discusses, in high level terms, what such a model might look like and how it would help in the problems described previously.

In more detail, rational communication is the process by which agents attempt to ascertain beforehand the value of sending a particular communication. If this value is an estimation of

future improved utility by coordination then the *cost of communication can be balanced with the estimated added utility*. In this case, the value can be derived from the information content of the communication, the likelihood of another agent already knowing its contents and more domain specific features (e.g. a constant value for communicating about a civilian to save in RoboCupRescue or the location of the tiger in the Multi–Agent Tiger problem). We can then measure the information content of communications and calculate the expected decrease in global uncertainty as a means of valuing communicating. Following this, information theory is a standard model for measuring how much 'knowledge' is added by new observations (Shannon, 1948), and so this can be used to measure how much 'knowledge' is transmitted in a communication. This increase in team-wide knowledge can then be seen as the utility of the communication.

Now, once a value is known for a communication, then it can be used in a decision problem. This decision problem might simply be to select the action or communication with the greatest expected utility, or a more complex model may be utilised which models the future outcome of possible sequences of actions to find the long term optimal action. The interesting question here is how to balance the somewhat abstract value of a communication action with the more concrete rewards for solving a problem completely or partially with some level of efficiency. Communicating may have a very real and situated cost (e.g. energy usage or time), but this can, perhaps, be offset by estimating how much better the team will perform in the future as a result of sending the communication. Thus, this decision problem should explicitly manage the trade-offs of communicating — communication allows greater levels of coordination, but it may be expensive and take time. Furthermore, in some domains, calculating this trade-off optimally may be computationally intractable, and so the decision problem may be approximate. In this case, it should be possible to understand the impact of using an approximation on a global level. In other words, any solution should either give the optimal global solution (as if the team was controlled centrally) or approximate this global solution in a principled way.

Against this background, if each team member is utilising a rational approach to communication then, at the level of the team, communication is managed effectively and contributes positively to the global utility. A team model that can manage the trade-offs between communicating and not communicating (by comparing both the cost of using the communication medium and the benefit of coordinating) can thus be used to operate in domains where the communication medium is uncertain and expensive. In particular, such a model can recognise how much communication will help coordination and from this evaluate the best time to communicate when compared with the other rewards and penalties in the problem. Such a system should be able to recognise when communication is cheap and in that case increase the level of coordination, but also find the most useful communications when the medium is expensive.

With this in mind, it is important to consider the scale of the agent team we are interested in. This is because different scales of agent teams typically require different coordination mechanisms (Tambe et al., 2005). For example, teams with hundreds of agents are far too large for explicit coordination and perform far better using simple rules and unsophisticated agents, for

example in Swarm Intelligence (Bonabeau et al., 1999), flocking UAVs (Watson et al., 2003) or in social learning systems (Noble and Franks, 2004). Conversely, teams with tens of agents perform much better if the coordination decisions are explicitly made by more sophisticated agents (for example, Belief, Desire, Intention agents (Bratman et al., 1991) or robotic box-pushing (Mataric et al., 1995)). Essentially, as the scale of the agent team increases, their sophistication decreases towards rule based behaviour. It should be highlighted that problems which exhibit some structure can allow for larger numbers of agents to coordinate in a sophisticated fashion. However, we consider this to be a different problem to the one tackled in this thesis where we are concerned with the general coordination problem with domain information.

The other factor influencing scalability is the size of the problem (not the number of agents). The RoboCupRescue problem highlighted in the previous section has a large state space (number of buildings etc). We are interested solving problems of this scale. With this in mind, in this work we consider teams with up to ten agents — allowing us to employ sophisticated agent coordination mechanisms and rational communication.

Against this background, there are three research challenges in creating a rational communication mechanism for coordinating agents in stochastic, partially observable, dynamic environments. Specifically, our coordination mechanism should:

- be able to accurately place a cost on the use of communication.
- be able to place a value (exactly or approximately) on the use of communication.
- allow communication to be used to control a decentralised system locally and efficiently

All of these issues will now be described in more detail.

Challenge 1: Communication Cost

As discussed earlier, agents often coordinate in domains where the communication medium cannot be employed without *cost*. Now, this cost is often domain specific — consider UAVs which cannot communicate in certain areas of the environment or network routing agents which use the same bandwidth to route packets and to send coordination messages. As a consequence, coordination mechanisms often use domain specific representations of this cost. In these cases, often the real costs are difficult to assess and arbitrary penalties are used instead. By contrast, our goal is to create a general rational communication coordination mechanism. To do so, we need to specify a general framework that captures the cost of communicating which allows agents to reason about whether or not to communicate. Specially, this mechanism should capture all of the disparate communication costs described earlier and allow agents to reason about the opportunity cost (i.e. the value of taking an alternative action compared to communicating) when using a communication medium.

In summary, the first research challenge is to specify a general problem solving and coordination framework which captures the cost of communication across several domains in terms of the opportunity cost whilst communicating — allowing agents to employ general algorithms when solving problems with communication costs.

Challenge 2: Valuing Communication

Whilst being able to accurately and generally establish the cost for sending a message is important for using communication rationally, it is just as important to be able to derive the *value* of sending that message. Now, the value of sending a particular message is that, in the future, and as a result of communicating, the team of agents will be able to perform a task better. Consequently, an agent should be able to establish, before sending a message, the likely benefit of sending that message. This benefit should be expressed generally in terms of the other rewards and penalties in the system. Together with an accurate cost of communicating, the benefit of communicating provides a rational communication mechanism. Unfortunately, however, in some domains this may be an intractable calculation. This is because of the size of the problem (the number of agents, the the number of possible communications etc). Consequently, it is also desirable to be able to *approximate* the value of communication.

Thus, the second research challenge is to be able to specify a general mechanism for deriving the value of sending a message. As before, this will allow agents to use general decision making algorithms over the costs and benefits of sending a message to give a general rational communication framework.

Challenge 3: Decentralised Coordination

Once a rational communication mechanism is developed, it needs to be deployed in a decentralised control setting. Ideally, agents employing the mechanism to make decisions in a decentralised fashion using local information should arrive at a solution which has the same performance as if the team was controlled centrally. This would correspond to an optimal decentralised solution. Now, in some situations, the complexity of the problem may make this infeasible because of the computational power of the agents, resulting in a lower solution quality when employing our rational communication mechanism. However, when we know that it is not possible to generate an optimal decentralised control solution, we should at least be able to place bounds on the loss of solution quality.

Against this background, the third research challenge is to show that our rational communication mechanism provides optimal solutions, or failing that, solutions in which the quality loss is bounded.

1.2 Research Contributions

In addressing the above-mentioned research challenges, we make a number of contributions to the state-of-the-art in cooperative MAS for decentralised control. First of all we propose a new framework based on models from sequential decision making that enables the formal expression of communication costs in an agent team. In this way we address the first research challenge, placing an accurate cost on communication. Now, this framework can be used for deriving the *true* benefit of communicating, however, this derivation is often intractable. Because of this, as per Challenge 2, we develop approaches for *approximating* the value of communication.

In more detail, we show how the amount of communication required to solve a team problem can be learnt offline using a simple procedure in our decision making framework. Now, this technique lacks the ability to estimate how close the generated solution is to the optimal one, and furthermore, the offline learning phase is computationally expensive. Consequently, we show how *Reward Shaping*, a technique from machine learning, can be leveraged to provide a sound theoretical basis for deriving communication valuations and, furthermore, reducing the computational complexity of the agent's decision making process. Hence we tackle the second research challenge where we require the value of communication to be generated.

Following this, we show how, using this technique, we can place a theoretical bound on the loss of utility compared to the setting where we can optimally calculate the true value of communication using a centralised approach. In this spirit, we also show analytically how an appropriate level of communication can be an optimal equilibrium strategy in the team problem expressed as a Bayesian Game (which is a formalism from game theory). Further to this, we show that if the problem is static (in contrast to the dynamic problems which the previous approaches were designed for), then we can specify an optimal communication valuation mechanism, which allow agents to derive an equilibrium strategy before acting in the problem. In more detail, we show how the value of communication can be expressed analytically in certain types of games which allows agents to make local decisions in order to coordinate globally. This shows how our valuations and costs can be used to control a decentralised system optimally as desired in the third research challenge.

With this in mind, there are several axes we can classify our contributions against. Scale — small (small state space) or large (large state space). Environment — static (only agents' actions cause the environment to change) or dynamic (some parts are outside of the agents' control). Solution — optimal (the best possible solution), bounded (has an error but this is defined) or approximate (has an error which may be unbounded).

We summarise our contributions against these axes in the following way:

- *Dynamic, small scale, approximate*: offline learning.
- *Dynamic, large scale, approximate*: heuristic reward shaping.

- *Dynamic, small scale, bounded*: exact reward shaping.
- *Static, large scale, optimal*: Bayesian game valuations.

In this way, we believe we provide a technique for valuing communications across the full spectrum of problems relevant in Multi–Agent Systems. Other points in the space are known to be intractable (dynamic, large scale, optimal) (Bernstein et al., 2000) or not of interest for real problems (static, small scale). Thus we focus on the most interesting parts of the spectrum. We now discuss these contributions in more detail.

1.2.1 Learning A Communication Valuation Offline

Two of the research challenges identified above were to provide a framework which can represent (in general terms) the cost and the value of communication. To this end, in Chapter 3 we specify such a framework. In particular, sequential decision making provides a standard formalism for representing such problem characteristics, and this is indeed where we will base our work. Specifically, we embed our work in the context of Decentralised Partially Observable Markov Decision Processes (dec-POMDPs), as this provides a general formalisation for making these kinds of decisions. Furthermore, within this context, there even exists several proposed mechanisms for calculating the value of communication (see Chapter 2 for more details). However, it has been shown that the generation of exact values for communication is an intractable problem for most practical situations because of the underlying complexity of reasoning about all possible team observations and action histories (Bernstein et al., 2000). Consequently, we argue for the need for an approximation to the true value of communication that is fast enough to compute every time an agent considers communicating. Such approximations are especially effective because the inherent uncertainty and dynamism of the target environments we consider means that excessive attempts to achieve accuracy will only contain inherent inaccuracies in any case.

Now, this work presents a novel model of rational communication, *dec_POMDP_Valued_com*, based on a principled formalisation for efficiently approximating the value of communications in a decentralised sequential decision making context. This new 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 (Dec-POMDPs) can already perform such trade-offs implicitly, it is 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 Kullback–Leibler (KL) Divergence). This is an efficient calculation that does not require reasoning over team beliefs. We choose information theory because it is a standard method for measuring how much 'knowledge' is added by new observations, and so this can be used to measure how much 'knowledge' is transmitted in a communication. This increase in team-wide

knowledge is seen as the utility of the communication. This novel approach then allows for wider applicability of decentralised POMDP models (for instance problems such as RoboCupRescue or UAV tasking are too large for the current state of the art), whilst avoiding any domain specific knowledge to generate valuations for communication actions.

With this established, the model we develop shows some interesting empirical results in a domain with a constrained communication medium, where it is compared with a model that does not communicate, a model that communicates constantly at no cost and a model that considers the value of a communication to be a linear function of the time since the last communication. Our domain, in this case, is RoboCupRescue — a large multi–agent simulator. This is used as an example and for our experiments because it is a hard problem requiring complex coordination utilising a communication medium in which we can specify the sorts of constraints we are interested in. In particular, we show that our model can approximate the performance of the full (free) communication model quite closely (within 5%), whilst utilising a costly communication medium. Consequently, our model is capable of balancing the costs of communicating with the benefits it brings in an efficient manner which greatly reduces the complexity of decentralised POMDP problems.

To date, this work has produced the following publication:

> S. A. Williamson, E. H. Gerding and N. R. Jennings (2008) A principled information valuation for communications during multi–agent coordination *Proc AAMAS Workshop on Multi–Agent Sequential Decision Making in Uncertain Domains*, Estoril, Portugal.

1.2.2 Heuristic and Exact Reward Shaping for Valuing Communication

While making some headway with some of the aims, the previous strand of work failed to account for research challenge 3 — being able to coordinate efficiently (either optimally or with a bounded error in dynamic environments) using our communication mechanism. In more detail, we would like to know how close to optimal our approximations are. However, the previous approach using offline learning lacked this feature and so, in Chapter 4, we propose a mechanism to explicitly capture the relationship between the optimal and approximate approaches. To do so, we transform a high complexity decentralised POMDP (in which agents reason about the complete experiences of all the other agents) to a lower complexity POMDP (in which an agent just reasons about itself), and in that transformation account for the coordination problem (so that the problem contains an automatic method of coordinating). In other words, we transform the full problem into several smaller ones whilst including new information (how to coordinate and when to use communication). Now, this could be achieved using relatively standard predefined decision rules that separate the problem and dictate what happens when coordination is needed (as discussed in Section 2.5), but this model lacks a theoretical basis and so we cannot

establish how close to optimal this approach really is. Furthermore, such rules require extensive domain knowledge to implement (again see Chapter 2 for more details).

In contrast, a more principled technique is that of *reward shaping* (Ng et al., 1999), an approach originally from the field of reinforcement learning, that supplies artificial reward signals to agents in order to speed convergence towards the optimal policy. Now, in our problem we are not concerned with learning, but with acting in a coordinated fashion using only local information. Thus, reward shaping can be recast as the process by which an agent's expected utilities for actions are modified (based on information other than that which is in the reward function — we discuss using *belief divergence* to perform this task next) during execution in order to elicit a more preferred behaviour. This important insight means that we can use reward shaping to transform a decentralised POMDP into individual agent POMDPs and, furthermore, to determine when it becomes important to communicate. By so doing, we implicitly account for the coordination problem. Moreover, reward shaping can be approximate in the sense that it biases learning towards a good (but not necessarily optimal) policy in the original problem (Laud and DeJong, 2002). Thus, in Chapter 5, we will use this idea to produce a heuristic reward shaping function that scales well.

One downside of this approach, however, is that it introduces an error into the transformation. Given this, we would like to find a means of bounding the error (in line with research challenge 3). To do so, in Chapter 6, we consider *exact* reward shaping in that the policy learned is guaranteed to be the optimal one in the original problem (Wiewiora et al., 2003). Although this approach scales less well (but still better than existing techniques), it allows us to find a theoretical bound on the solution error of the transformed problem given the observation dynamics of the underlying problem. This is particularly useful since it allows us to ascertain beforehand whether our technique is appropriate for a given problem, and what performance, in the worst case, we will lose by applying this approximation. This is currently lacking from all existing decentralised POMDP models that are based on valuing communication in order to coordinate.

Now, in-order to apply reward shaping (both heuristic and exact) to achieve coordination, we need to supply the transformation process with information about the likelihood of coordination given the team beliefs (i.e. how much in agreement the agent's are about the state of the world). Crucially, this information should be easy to maintain and estimate in a distributed fashion because we would like a solution which can operate in restricted communication domains. To do this, we choose to base reward shaping upon *belief divergence* (a measure of how similar the beliefs of a distributed team are) [2]. We could equally use full team models (such as Bayesian Networks (Gmytrasiewicz and Durfee, 2000) or Particle Filters (Roth et al., 2005)) (but these are large and difficult to maintain) or coordination statistics such as the likelihood of independently knowing about a feature of the environment (such as *STEAM* (Tambe, 1997) or Zhang et al. (2004)) (which are hard to compute), however, the advantages of belief divergence, are that it is cheap to maintain and there are several existing techniques for establishing it in a decentralised

[2] This is not central to our method, and we could use other information in other cases if that would give better results.

way. This means we can efficiently approximate the value of communications in a decentralised sequential decision making context. 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.

In undertaking this work we extend the state of the art in three main ways. First, by using heuristic reward shaping, we allow decentralised POMDP models to be applied to larger problems (such as RoboCupRescue), than has hitherto been possible, whilst avoiding using significant amounts of domain-specific knowledge to generate valuations for communication actions. Second, with exact reward shaping, our approach generates more accurate online valuations of communication than the previous state of the art, and by doing so, allows an expensive communication medium to be used more efficiently. Third, both of the previous advances are made in a principled fashion which allows us, in the case of exact reward shaping, to present the first bound on making communication decisions using local information. Such a bound is important in this context because it highlights the features of the environment that make the error grow (in this case it is the accuracy of the observation function and individual agent's planning horizon). Furthermore, it is also useful for comparing different approximate approaches which is particularly vital in this domain as optimal solutions are not really viable.

To date, this work has produced the following publications and submissions:

> S. A. Williamson, E. H. Gerding and N. R. Jennings (2009) Reward shaping for valuing communications during multi–agent coordination *Proc. 8th Int. Conf on Autonomous Agents and Multi–Agent Systems*, Budapest, Hungary.

> S. A. Williamson, E. H. Gerding and N. R. Jennings (Submitted) Rational Communication in Multi–Agent Coordination using Reward Shaping *Journal of Artificial Intelligence*, Elsevier.

1.2.3 Communication as an Equilibrium Strategy

Further to the work in Chapter 6, we continue to address the question of whether communication valuations can be used to provide an optimal decentralised control paradigm using Bayesian Games in Chapter 7. Specifically, Chapter 6 addresses research challenge 3 by proposing a bounded transformation from the decentralised problem to individual local problems. This approach works well in dynamic environments, however, if the problem is static in nature then we can do better and make an optimal coordination mechanism which exploits an analytical form of the communication valuation. To this end, in this chapter we give a complementary approach that solves the decentralised problem optimally by considering how the strategy space can be reduced to allow for an easily computable analytic form of the true value of communicating. In essence, in this strand of work we restrict the generality of the game played in order to arrive at

an optimal valuation, in contrast to the more general previous work which can only arrive at a bounded approximation, but can do so for a wider range of situations.

In more detail, we propose a game-theoretic model that explicitly optimises costs and benefits of communication in a team information-gathering problem (specifically, the Multi–Door, Multi–Agent Tiger problem). Using this model, we characterise, analytically, the value (and cost) of communicating the expected information gathered given a specific search strategy — for example a uniform search of domain features. This allows us to specify a general equilibrium analysis of search and communication strategies which shows that given the properties of the underlying game (the noise in the observations and the pay-offs) a communication profile is an optimal equilibrium. Furthermore, this analysis holds for several different search strategies if their expected behaviour can be expressed in our framework. This is an important contribution in the context of deriving communication valuations since it shows the expected valuation on communication can be used to control a decentralised system optimally using local information.

To date, this work has produced the following publications and submissions:

> S. A. Williamson, A. C. Chapman and N. R. Jennings (Submitted) Information gathering and exchange for coordination of multi–agent teams *Proc. 9th Int. Conf on Autonomous Agents and Multi–Agent Systems*, Toronto, Canada.

1.3 Thesis Structure

The rest of this thesis is structured as follows:

- Chapter 2 analyses the most relevant previous research in using cooperative MAS to control decentralised systems. In this review, we focus in particular on sequential decision making and communication valuations.
- Chapter 3 develops a sequential decision making framework which can capture the costs and benefits of communication. In this chapter we also instantiate an offline learning technique which can arrive at a level of communication for good performance and we demonstrate how this is useful in a large coordination problem.
- Chapter 4 takes the sequential decision making framework from Chapter 3 and develops a theoretical basis for valuing communication using reward shaping. Here we show how this mechanism can separate the team problem into individual agent problems.
- Chapter 5 uses the framework in Chapter 4 to extend the state-of-the-art in large decentralised control problems using a heuristic function within the context of reward shaping to give an approximate, but highly scalable, solution.

- Chapter 6 shows how this approximation can be bounded in terms of the solution error compared to the intractable team sequential decision making problem at the sacrifice of some scalability — which is still better than existing techniques.
- Chapter 7 shows how communication of the expected results of specific information-gathering strategies can be an optimal equilibrium in a decentralised control problem — showing (as we also did in Chapter 6) that reasoning about communication can be the basis for a theoretically sound separation of a decentralised problem into a local problem for each agent.
- Chapter 8 gives the conclusions and presents the future directions of this work.

Chapter 2

Related Work

This chapter introduces the relevant work in coordinating MAS in our kind of problem, and how communication is used and valued in this task. Initially, we concentrate on coordination. In particular, the first section details methods of coordination based on modelling the intentions of other agents. Section 2.2 describes methods of coordination based on message passing without maintaining team models. Both of these approaches rely on intensive models of the other agents in order to compute solutions. We then move onto approaches which aim to avoid this problem. Specifically, Section 2.3 describes game theory, a principled way for modelling agent interactions. In Section 2.4 we describe Sequential Decision Theory for both single and multiple agents, showing how this model closely relates to our problem. In Section 2.5, the problem of how to value communication is considered. Then, in Section 2.6, we will describe the test domains used throughout this work as benchmarks. Finally, a summary is given which details what existing work will be used as the point of departure for this research, and how it will need to be extended to meet the research challenges identified in Chapter 1.

2.1 Intentional Coordination

We first consider coordination based on modelling the 'intentions' of other agents — that is, if an agent knows what a second agent will do next then it can select a complementary action itself. Initially, we describe the Belief, Desire and Intention (BDI) agent theory as this forms the basis for modelling intentions and then show how this has been extended to multiple agents and coordination.

2.1.1 BDI Agents

The BDI model of practical reasoning (Bratman et al., 1991) was proposed as a theoretical method for limiting (by removing reasoning about choices inconsistent with current intentions)

an agent's reasoning based on its intentions, in a bid to combat the bounded resources of an agent. Specifically, BDI agents are composed of:

- **Beliefs** represent an agent's view of the world, and what it believes to be true,
- **Desires** represent states of the world the agent wishes to bring about,
- **Intentions** are what the agent has committed to doing after some deliberation

Deliberation is limited to actions that achieve desires, and subsequent action planning to those that achieve intentions. In particular, the process by which a BDI agent decides upon its next action is summarised by the pseudo-code in Figure 2.1.

```
function action(p: P) : A
begin
    B := brf(B,p)
    D := options(B,I)
    I := filter(B,D,I)
    return execute(I)
end function action
```

FIGURE 2.1: BDI action selection taken from Weiss (1999)

B, D, and I represent the agent's beliefs, desires and intentions respectively. The belief revision function *brf* takes a precept and updates the agent's set of beliefs. The *options* function maps beliefs and intentions onto a set of desires, and *filter* represents the deliberation stage, updating intentions from the current B, D, and I sets. The *execute* function creates a plan to carry out the current intentions. Here coordination is achieved by the agents reasoning about interactions with each other — closely modelling human interactions (Tambe et al., 2005). However, BDI agents are less able to work in uncertain stochastic environments such as those considered in this research, which, unfortunately, are difficult to approach using formal planning (Brooks, 1999) (because they are hard to model in logical propositions). For this reason, we will now consider more explicit models of agent coordination which use the intentional ideas of BDI when reasoning about other agents and not just themselves.

2.1.2 Teamwork Models

The intentional model of agency can be extended to explicitly consider the set of agents as a team. Teamwork models such as *GRATE** (Jennings, 1995) represent a way to structure global information about the team and perform local reasoning about coordinated actions. At their heart, teamwork models attempt to follow human reasoning about how to work in teams. Joint Intentions (Cohen and Levesque, 1991) are one such attempt, where mental states called *intentions* inform other agents about what that agent will do. The intentions here are different to those

in BDI, where an intention is what the reasoning agent intends to do, but here an intention is a notion of what the other agents will do. In this context, teamwork models encode rules about how teams should coordinate to achieve goals and what they should do when goals are found to be unobtainable. This model contrasts with general action selection, which does not attempt to encode how the team should work together. As a result, general action selection needs to infer how to work as a team, whilst this is made explicit in teamwork models. This potentially makes teamwork models powerful joint problem solvers, and some examples will be given next.

A number of prominent models of teamwork have been introduced in recent years. In particular, Durfee and Lesser (1991), Decker and Lesser (1992), and Lesser et al. (2002) describe the *Generalized Partial Global Planning* (GPGP/TÆMS) coordination framework and Tambe (1997) and Zhang and Tambe (2000) develop the *STEAM* teamwork model.

In more detail, the *GPGP* coordination framework aims to schedule the activities of a small group of agents in a team with a joint goal, and increase the overall utility accrued by the group. In order to achieve their joint goals, a distributed algorithm is used which schedules the activities of each agent (each of which has a partial view of the activities of the other agents). The subtasks of each agent can be dependent on the activities of the other team members, and the framework aims to increase the coherence of the agent activities. In this case, communication is used to manage the partial models of the other agents that each one maintains. However, whilst the algorithm does have the high-level aim of restricting unnecessary communication, it does not explicitly consider the balance between coordination actions and domain-level task achieving behaviour when a combination of both is required to both coordinate and solve the problem. That is, communication is considered in isolation of problem solving, and this research postulates that they need to be considered together to coordinate efficiently.

In contrast to *GPGP* which performs partial planning at each agent, *STEAM* implements a rule set to manage Joint Intentions. *STEAM* agents can then exploit these rules to reason about coordination and communication. By doing this, agents build a partial hierarchy of joint intentions, and have actions to maintain coherence across the team. Teams using *STEAM* can be dynamically reorganised as the environment changes (or is interpreted by individual team members), allowing the agents to opportunistically complete goals for other team members or distribute important information (such as the completion, failure or unattainability of a team goal). Moreover, this model also employs a decision-theoretic approach to communication which follows research challenge 2 from Section 1.1. This balances the cost of communication with the cost of mis-coordination. This aims to reduce the communication overhead (which is still large because the agents must maintain models of how the team goals are progressing).

Now, in both of these models communication is used to aid coordination and there is no central coordinator. However, in terms of considering the *value* of a particular communication, only *STEAM* attempts this as per research challenge 2. This valuation is calculated based on the cost of mis-coordination, and the probability of whether some domain feature is commonly known across the team. In order to decide whether to communicate, the value of communication is

balanced with the cost of communicating. Unfortunately, however, both the cost and value of communicating is specified in terms of the semantics of the joint intentions used within STEAM, making it difficult to generalise. In more detail, depending on the joint plan being executed, costs are attached to mis-coordinating and probabilities of mutual knowledge are attached to specific domain features — all these values are defined by the system designer. Also, communication is assumed to be myopic and so the future impact of communication is not taken into account. In this context, Becker et al. (2005) show that this myopia leads to inefficient coordination and, hence, inefficient problem solving. A second problem with *STEAM* is the distributed planning nature of the protocol which leads to a team that is very dependent on communication, and also obscures the communication decision problem that is studied here. A more desirable system would be able to perform reasonably well with zero communication, and communication should only improve this level of performance, which, unfortunately, is not the case with *STEAM*.

In conclusion, teamwork models allow the agents to explicitly reason about the coordination problem, which has potential uses for valuing communication, and consequently, tackling research challenge 2, but they have a large communication overhead because they perform distributed planning. When communication is highly restricted, such as in the domains considered in this work, then these mechanisms are not robust as the agents need to perform well in isolation for much of the time, and as a result these models do not meet research challenge 3. Furthermore, the cost of communication is expressed as an arbitrary penalty and so this does not meet research challenge 1. Thus, for the purposes of this work, these models will only be considered as an approach for calculating the communication valuation from now on (see Section 2.5 for more details). The next section introduces two different ways to coordinate based on messages passing, without modelling the intentions of the other agents.

2.2 Message Passing Coordination

This section introduces two coordination mechanisms where message passing is the explicit method of coordination — these messages are assumed to alter the state of the other agents in the team. This in contrast to intentional coordination where communications take the form of commitments whereas here messages are problem information and partial solutions. We look at these models because communication is the explicit coordination mechanism making it appropriate for our problem. Furthermore, understanding how a message alters other agents' behaviour is key to rational communication and research challenge 2. Agents in both mechanisms model the other agents in different ways. The first approach uses Distributed Constraint Satisfaction Problems (DCSPs) to model agents as sets of constraints between the values the agents can take, and uses messages to inform the other agents of the valuations they might use. The second approach, Bayesian Networks, allows agents to model the information the other agents know and uses messages to perturb this.

2.2.1 Distributed Constraint Satisfaction Problems

DCSPs model the coordination problem as a constraint satisfaction problem (CSP), but where constraints and variables are distributed amongst multiple agents (Yokoo et al., 1998). Here, constraints are rules relating the value of one variable to another (perhaps held by a different agent). A solution is an assignment of values to variables which satisfies all constraints. In order to reach a solution, communication is used to distribute possible assignments. In this context, coordination is needed in the generation and distribution of possible assignments, in order to speed convergence towards a solution, and reach a solution at all for many problems. More formally, a CSP consists of n variables $x_1, x_2, \ldots, x_n$, whose values are taken from finite, discrete domains $D_1, D_2, \ldots, D_n$, respectively, and a set of constraints $p_k(x_{k1}, \ldots, x_{kj})$ on their values. A constraint is a predicate that is defined on the Cartesian product $D_{k1} \times \cdots \times D_{kj}$. This predicate is true iff the value assignment of these variables satisfies this constraint.

Given the interest in this area, there are several algorithms for solving these problems such as Asynchronous Backtracking (Yokoo and Hirayama, 2000), Asynchronous Partial Overlay (Mailler and Lesser, 2004), Dynamic Programming Optimisation (Petcu and Faltings, 2005), and Adopt (Modi et al., 2005). These algorithms use a communication network between agents to distribute partial instantiations and information about whether these instantiations are consistent.

In more detail, a DCSP is an explicit crystallisation of the coordination problem with communication. Specifically, constraints represent interdependencies between the actions (instantiations) the agents can take, and communication is used to manage these interdependencies. They are most suited to studying global behaviour from repeated local interactions, in domains where not all agents interact with each other (Tambe et al., 2005). This is because it is simple to assess whether coordinated global behaviour is emerging from agents (simply count the number of unsatisfied constraints) who can only interact with a few (not all) other agents. Unfortunately, it is hard to model uncertain stochastic domains with this formalisation, because the constraints represent interdependencies between agents — not agents and the world. As a result, the problem must be represented very abstractly (i.e. in terms of task allocations) and so cannot easily capture properties of the domain which influence communication (research challenge 1 and 2). Therefore, this model will not be considered in this work. Other coordination mechanisms (introduced later) will match our problem more closely. At the same time, however, there are several interesting questions related to communication in this domain, particularly concerning the balance between distributing partial solutions and the cost of so doing. Whilst we do not pursue this mechanism further, we foresee that techniques for valuing communications will be equally applicable in DCSPs. We now consider an approach which uses messages in a similar way, but does not represent the problem in such an abstract way — Bayesian Networks.

2.2.2 Bayesian Networks

In general, Bayesian Networks model the coordination problem as the distribution of knowledge in the problem; that is, the probability of agents knowing about features of the problem and intentions of other agents are modelled in a Bayesian Network. Here, coordination is seen as the problem of maintaining good models of the knowledge of the other team members and, from this, predicting what they will do. This is then used to estimate the impact of communication on the other agents' knowledge and hence gives a value that can be used in a decision problem — a technique for solving research challenge 2.

An example of this method is in the work of Gmytrasiewicz and Durfee (2000). Here, self-interested agents are studied, but the ideas they develop carry over to teams of co-operating agents if they have an identical utility function. In more detail, an agent models the knowledge of other agents in a Bayesian Network. Consequently, the modelled agent is also modelling the first agent, so there is inherent recursion in this approach. As a result, a Recursive Modelling Method (RMM) is used and, in this case, the recursion is limited to two levels because otherwise the model would quickly become intractable, and, moreover, the authors maintain that this is a good approximation in any case. From this, the value of communication is calculated by an agent as the expected gain by that agent if it modifies the knowledge of the other agent so that it takes some expected action. This value is then used to decide what to communicate.

The idea of modelling knowledge for coordination can be taken forward to include planning for the entire problem. To this end, Shen et al. (2001, 2003) develop a hybrid approach of Bayesian Networks and Markov Decision Processes (MDPs are discussed in the next section). Here, a Bayesian Network is used to represent the distribution of evidence between the agents. This evidence is essentially the current state of the knowledge space. Based on this representation a communication policy is derived, which distributes the agents' individual knowledge in order to solve some joint goal (this is done using a MDP). This technique follows Gmytrasiewicz and Durfee and Tambe in that communication has some value based on how it transforms the knowledge space of the agents in the team. Obvious problems with this approach, however, include how to dynamically construct the Bayesian Networks and how the communication policy transforms the Bayesian Network for continued interactions.

From the above description, it is clear that Bayesian Networks are a useful model for representing knowledge in a team, and thus can be used to value the manipulation of knowledge in a team. However, they say less about the general coordination problem with communication, since they do not explicitly consider both actions and communications (making the link between research challenge 1,2 and 3 difficult), and it will be seen that decentralised POMDPs fit the domain more closely. Furthermore, Bayesian Networks are inherently myopic (hence the requirement of an MDP to plan ahead) and it is clear that a valuation of communication should consider the future impact in order to be accurate. Therefore, Bayesian Networks will only be considered as a valuation for communicating from now on (see Section 2.5 for more details). We now consider agent coordination from the perspective of game theory — a more formal theory of agent interaction.

2.3 Game Theoretic Coordination

This section will describe some of the game theoretic paradigms used for coordination in agent teams. Game theory is a well studied field covering many of the problem characteristics we consider. It is especially concerned with deriving the optimal strategy given another agents strategy — which is part of the rational communication problem and key to achieving research challenge 3. Specifically we detail Bayesian games of incomplete information which aim to specify strategic actions based on imperfect knowledge of the other agents and stochastic games which capture many of the domain features we are interested in. Before this, however, it is useful to present some basic game theoretic ideas including the concept of games as a representation of agent interactions and solution concepts within these games.

2.3.1 Games and Strategies

There are many dimensions to games. Games can be presented in *strategic* (or *normal form*), where agents decide upon a strategy once and for all, and all decisions are made simultaneously, or in *extensive form* which explicitly models the timing of decisions and actions. Games can be repeated in the future, allowing players to adapt their play or learn about their opponents. In such cases, the game played in each round is referred to as the *stage game*, while the complete sequence of games is typically called the *repeated game*. In cases where the stage game varies (possibly probabilistically) in each round, the sequence of games is called a *stochastic game*. Furthermore, games can be differentiated by their information structure. In games with *perfect information*, players are completely informed about other player's utilities or payoffs, while under *imperfect information* they are not: games with imperfect information are known as *Bayesian games*. In this thesis we focus on stochastic games since the problem can be in many states and actions cause uncertain transitions between these states. Furthermore, these are states of incomplete information because the agents do not know about the complete state of the problem or other agents. For a good discussion of these distinctions and their importance in game theory see Fudenberg and Tirole (1991).

In more detail, a non-cooperative game in strategic form is a tuple $\Gamma = \langle N, (S_i, u_i)_{i \in N} \rangle$, comprising:

- a set of **players**, $N = \{1, \ldots, n\}$,

  and for each $i \in N$:

  - a set of **strategies** S_i,
  - a **utility function** $u_i : S \to \mathbb{R}$.

We use the terms action and pure strategy interchangeably throughout the review. The set of joint strategy profiles S is the Cartesian production of all S_i:

$$S = \prod_{i \in N} S_i,$$

and a particular joint strategy profile $s = \{s_1, \ldots, s_n\} \in S$ is referred to as an *outcome* of the game. The complimentary set of s_i is denoted s_{-i}, and when discussing a particular player's choice of strategy, the notation $s = \{s_i, s_{-i}\}$ will be used. A utility function specifies a player's preferences over outcomes by the condition that, if and only if the player prefers outcome s' to outcome s'', then $u_i(s') > u_i(s'')$.

Against this background, the Nash equilibrium is a widely applicable solution concept, and the most important solution concept in game theory.

Nash Equilibrium

A strategy profile s^* is a **Nash equilibrium** if each player's strategy is a best response to the other players' equilibrium strategies:

$$u_i(s_i^*, s_{-i}^*) - u_i(s_i, s_{-i}^*) \geq 0 \quad \forall\, s_i,\ \forall\, i. \tag{2.1}$$

Intuitively, in a Nash equilibrium, no individual player has an incentive to deviate to a different strategy.

The Nash equilibrium in definition 2.3.1 is a *pure strategy* Nash equilibrium, as in equilibrium each player selects a strategy with a probability of one. However, the equilibrium concept can be extended to include the case when a player's best response is to randomise its selection of strategies. It is assumed that each player has preferences defined by lotteries or a set of probability distributions over strategy, $\Sigma(S_i) \in \Sigma(S)$, called "von Neumann-Morgenstern preferences". A probability distribution over mixtures of pure strategies $\sigma_i(S_i) \in \Sigma(S_i)$ is called a *mixed strategy*, and the support of a mixed strategy is the elements of S_i to which σ_i assigns a positive probability. The utility function of the mixed extension of the game is given by the expected value under u_i of all players' joint, independent lottery $\sigma \in \Sigma$ over S:

$$u_i(\sigma_i, \sigma_{-i}) = \sum_{s \in S} \left(\prod_{j \in N} \sigma_j(s_j) \right) u_i(s). \tag{2.2}$$

Following this, the concept of Nash equilibria can be extended to mixed strategies, $B_i(\sigma_{-i})$. A Nash equilibrium corresponding to a joint profile of mixed strategies is called a mixed strategy Nash equilibrium, and is defined as:

$$u_i(\sigma_i^*, \sigma_{-i}^*) - u_i(\sigma_i, \sigma_{-i}^*) \geq 0 \quad \forall\, \sigma_i,\ \forall\, i, \tag{2.3}$$

or equivalently:

$$\sum_{s\in S}\left(\prod_{j\in N}\sigma_j^*(s_j)\right)u_i(s)-\sum_{s\in S}\left(\prod_{j\neq i\in N}^{n}\sigma_j^*(s_j)\right)\sigma_i(s_i)\,u_i(s)\geq 0\quad\forall\,\sigma_i,\ i. \tag{2.4}$$

The set of mixed strategy Nash equilibria include any pure strategy Nash equilibria, furthermore, Nash proved that any strategic game with a finite number of strategies has a mixed strategy Nash equilibrium point (Nash, 1950). It can be seen that proscribing a method which achieves a Nash Equilibrium would solve research challenge 3. Now these games are perfectly observable, which is not appropriate in our setting. In the next sections we will consider games with a partially observable nature.

2.3.2 Bayesian Games

In some situations, the agents in a system may not be able to observe the actions of other agents in the system. As such, they may have to act without knowing the true state of the world. These situations are known as games of *imperfect* information. In other situations, agents may not know the characteristics of other agents in the system. This is known as *incomplete* information, and usually refers to situations where players do not know the preferences (or utility functions) of other players. Harsanyi (1967) introduced the idea that in such games "nature" moves first to allocate "types" to players. In this way, *incomplete* information about a player's preferences is treated as *imperfect* information about nature's move. Both situations are modelled by a Bayesian game, which is an extension to the simple normal form introduced earlier.

Formally, a Bayesian game is a tuple $\Gamma=\langle N,\Omega,(S_i,\Theta_i,\zeta_i,p_i,u_i)_{i\in N}\rangle$, comprising:

- a set of **players** $N=\{1,\ldots,n\}$,
- a **state space** Ω,

  and for each player $i\in N$:

  - a set of **strategies** S_i ($S=S_1\times\ldots\times S_n$),
  - a set of possible **types** Θ_i,
  - a **signal function** $\zeta_i:\Omega\to\Theta_i$,
  - a **prior belief**, a probability measure $p_i:\Omega\to[0,1]$, for which $p_i(\zeta_i^{-1}(\theta_i))>0$ holds for all $\theta_i\in\Theta_i$, and
  - a **utility function** $u_i:S\times\Omega\to\mathbb{R}$.

In this model $\omega\in\Omega$ is interpreted as a particular "state of nature" (i.e. a particular type profile of all players). The signal function maps from states to types, such that $\zeta_i(\omega)=\theta_i$ is the type of player i in state ω. Then the conditional probability $p_i(\omega|\theta_i)$ summarises what i believes about

the state of nature (other players' types) given its own type. There exists a solution concept, the *Bayes-Nash Equilibrium*, where on average no individual player can deviate to a better strategy, given their prior belief distribution over other players' types or strategies.

Against this background, the use of communication in games of complete and incomplete information to implement Nash and correlated equilibria has been studied extensively (see Gerardi (2004) for a review). In particular, Krishna (2007) shows that communication can extend the set of Bayes-Nash equilibria in 2 player games. However, the communication medium employed in this line of work makes it inappropriate for our problem. Specifically, communication is often *mediated* (it involves a disinterested third party), goes in only one direction (from a more informed sender to a less informed receiver) or involves *cheap-talk* (free communication before the game is played). Crucially, communication is always assumed to be free in these models. This makes these techniques lacking when it comes to dealing with research challenge 1 and 2. Finally, these games are repeated with the same characteristics. They lack a representation of how time or action changes the game being played. The next section will address this flaw.

2.3.3 Stochastic Games

In these games, the world can be in one of a set of states (in each state the game being played is different — different strategies are available and or different utilities are available). The strategies taken in one stage game change which stage game is played next. This captures many of our problem characteristics. More formally, a stochastic game (Owen, 1982) is a tuple $(n, S, A_{1\ldots n}, T, R_{1\ldots n})$, where n is the number of agents, S is a set of states, A_i is the set of strategies available to agent i (and $\mathcal{A}$ is the joint action space $A_1 \times \cdots \times A_n$), T is a transition function $S \times \mathcal{A} \times S \in [0,1]$, and R_i is a reward function for the ith agent $S \times \mathcal{A} \in \mathbb{R}$. The solution concept here is to determine a course of action for an agent in this environment. Specifically, we want to learn a stationary, though possibly stochastic policy, ρ, that maps states to a probability distribution over its actions. The goal is to find such a policy that maximises the agent's discounted future reward with discount factor γ. Now, this game is suitable for domains where the agents can observe the state, but this is not always the case. So, there is an extension to environments where the results of strategies are not fully observable, namely the Partially Observable Stochastic Game (POSG) (Oliehoek and Vlassis, 2006). We will see later that this formalisation is analogous to decentralised POMDPs (see Section 2.4.2) and so we will not consider it here.

To date, most research in this area has focussed on Nash equilibria as a solution concept. However, Binmore (1990) and Kadane and Larkey (1982) have indicated this is only useful for describing a system which has reached a stable state — and is less useful as a general control paradigm needed for the sorts of domains we consider here. This is because there may be multiple non-unique equilibria to choose between and these equilibria do not specify a strategy when other agents may not be acting according to their own equilibrium strategies (incompleteness). In fact, in our work we will use the concept of best response to the current state rather than the

other agents' strategy because, as we will see later, it is a standard solution concept in sequential decision making — which suits our problem domains very well.

Against this background, we have introduced the game theoretic paradigms which most closely resembles our problem (Stochastic and Bayesian games), but it is clear that Nash and Bayes—Nash Equilibria are not appropriate as solution concepts in this domain. Specifically, the sorts of problems we consider in this work are unlikely to ever reach a stable state because of the degree of uncertainty involved and consequently the problem of incomplete strategies is particularly difficult. Consequently, we will consider optimising the best response to a given state. Also, they do not consider communication and its costs explicitly (thus ignoring research challenges 1 and 2), unlike the Decision Theoretic models of coordination we will introduce next. As a result, we will address some of the flaws relating to communication and stability in later work (see Chapter 7 where we have a static domain making Game Theory an attractive concept because of research challenge 3) but focus most of our work in the more general and expressive sequential decision making formalism.

2.4 Sequential Decision Theory

This section introduces a general model of sequential decision making for single agents and then for the coordination of teams in the partially observable, stochastic domains we consider in this research. First the basic, fully observable, single agent model will be described — the Markov Decision Process. Then, following that, an extension to partially observable domains will be given. Recall that partial observability is a feature of the domains considered in this research. After this, the centralised multi–agent version is introduced and, finally, the decentralised multi–agent formalisation is given. This final model matches our problem and requirements most closely, and forms the main point of departure for our work. We will consider decentralised models which explicitly model the team and those which do not, highlighting the advantages and disadvantages of both approaches.

2.4.1 Single Agent Decision Making

A Markov Decision Process (MDP) is a formal model of control, defined by the tuple $M = \langle S, A, P, R \rangle$ where:

- S is the state space,
- A is the action space,
- P is the transition probability function. $P(s \in S, a \in A, s' \in S) \in [0,1]$ is the probability of moving from state s to state s' when the agent takes action a,

- R is the reward function. $R(s \in S, a \in A, s' \in S) \in \mathbb{R}$ returns a real-valued reward for executing action a in state s, resulting in state s'.

The solution to an MDP is a *policy* π, a mapping from states to actions. The optimal policy gives the largest cumulative reward over an infinite horizon $\sum_{t=0}^{\infty} \gamma^t R(s_t)$, where t is the timestep of the process and γ is the discount factor for future reward. It is the optimal policy that we aim for in research challenge 3.

In more detail, this model is suitable for modelling a single agent inhabiting a fully observable domain. However, the main features of our domain are partial observability and several interacting agents. As a consequence, the model must be developed further. The extension of the MDP to domains where the complete state of the problem cannot be observed reliably is the Partially Observable Markov Decision Process (POMDP) (Kaelbling et al., 1998). This is defined by the tuple $POM = \langle S, A, \Omega, O, P, R \rangle$ where:

- S is the state space (as above),
- A is the action space (as above),
- Ω is the observation space,
- P is the transition probability function which now accounts for possible observations. $P(s \in S, a \in A, \omega \in \Omega, s' \in S) \in [0,1]$ is the probability of moving from state s to state s' when the agent takes action a and receives observation ω,
- O is the observation function. $O(s \in S, a \in A, s' \in S, \omega \in \Omega) \in [0,1]$ is the probability of observing ω when in state s and taking action a resulting in state s',
- R is the reward function which also now accounts for possible observations. $R(s \in S, a \in A, \omega \in \Omega, s' \in S) \in \mathbb{R}$ returns a real-valued reward for executing action a in state s, resulting in state s' and receiving observation ω.

This formalisation is used in domains where the state cannot be observed directly, but the agents receive observations, which is true of the domains considered in our work. In this case, a policy is a mapping from *belief states* to actions. Here belief states are probability distributions over the actual state and are updated using Bayesian reasoning about observations and prior knowledge.

In summary, this model moves closer to our requirements than the MDP, because it models the uncertainty and partial observability in the environment, but it still does not explicitly consider multiple agents. The next models presented will address this challenge.

2.4.2 Decentralised Decision Making

MDPs and POMDPs are only appropriate for single agent problems, or MAS problems where the other agents are considered to be part of the environment. If the agent needs to explicitly

consider the actions of the other agents (i.e. to coordinate) then a model is needed which considers each possible combination of actions that the team could take (joint actions). Any such model needs to be able to identify individually coordinated joint actions (i.e. to ensure actions taken by one agent are not cancelled out by another's action) and sequences of coordinated joint actions (i.e. two agents co-operatively pushing a box towards a goal past an obstacle). In this context, the first MAS extension to the MDP is the *Multi–Agent Markov Decision Process (MMDP)* introduced by Boutilier (1999), which is a normal MDP but with joint actions and a global state based on the state of each agent. Now, the MMDP consists of a set of agents, each with their own action and observation spaces, defined by the tuple $MM = \langle \alpha, \{A_i\}_{i \in \alpha} S, P, R \rangle$ where the symbols are as for the MDP except:

- α is a finite collection of n agents,
- each agent $i \in \alpha$ has at its disposal a finite set A_i of *individual actions*. An element $\langle a_1, \ldots, a_n \rangle$ of the joint action space, $\mathcal{A} = \times A_i$, represents the concurrent execution of the actions a_i by each agent i.
- P is the transition probability function which is the same as for the MDP but is defined over joint actions. $P(s \in S, a \in \mathcal{A}, s' \in S) \in [0,1]$ is the probability of moving from state s to state s' when the agents take action a,
- R is the reward function which is the same as for the MDP but is defined over joint actions. $R(s \in S, a \in \mathcal{A}, s' \in S) \in \mathbb{R}$ returns a real-valued reward for executing action a in state s, resulting in state s'.

The MMDP is centralised because all agents know the global state (and hence can be reduced to an MDP where the joint actions are represented as primitive actions). Typically, the extension to the decentralised domain models the problem as several interacting MDPs and includes the possibility of communication. In this case, communication is used to share information between the MDPs and can represent something as simple as the local perception of the global state, or it can be more complex and cover items like intentions and plans (see Section 2.1). We now deal with such decentralised models.

In more detail, an alternative to the teamwork models of Section 2.1.2 is decentralised Partially Observable Markov Decision Processes (Dec-POMDPs), which have been introduced by a number of authors including Zilberstein and Goldman (2003), Xuan et al. (2001), Peshkin et al. (2000), and Pynadath and Tambe (2002). These all describe decentralised versions of the POMDP. In this context, a good representative example is the *dec_POMDP_com* model from Zilberstein and Goldman, which is a decentralised POMDP with a separate alphabet describing the possible communications. Many other models, such as Xuan et al. and Pynadath and Tambe, choose to restrict the communication alphabet to be the same as the observation alphabet. Finally, Peshkin et al. is derived from a Partially Observable Stochastic Game (see Section 2.3.3) but makes the payoff function for each agent identical and consequently can be combined with

these models. Apart from these relatively minor differences, the models are largely equivalent. In particular, the difference between centralised and decentralised MDPs is that the former is a single MDP 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. We will describe the *dec_POMDP_com* since it is the most general.

In more detail then, the *dec_POMDP_com* is defined by the tuple (for 2 agents) $DECPOM = \langle n, S, \mathcal{A}, \Sigma, C_\Sigma, P, R, \Omega, O, T \rangle$ 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:

$$P(s \in S, a \in \mathcal{A}, s' \in S) \in [0,1] \tag{2.5}$$

  of moving from state s to state s' when the agents take joint action a.

- R is the reward function. This returns a real-valued reward:

$$R(s \in S, a \in \mathcal{A}, \sigma \in \Sigma^n, s' \in S) \in \mathbb{R} \tag{2.6}$$

  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:

$$O(s \in S, a \in \mathcal{A}, s' \in S, \omega \in \Omega) \in [0,1] \tag{2.7}$$

  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: one is the normal action policy for the POMDP associating belief states with actions and the other policy associates belief states with communication acts. Before we analyse this model further, it is clear that it represents a good fit for many of our requirements. Specifically, it considers uncertain, stochastic, partially observable domains and explicitly considers the other agents in the environment. Furthermore, a solution to this model is appropriate for research challenge 3.

Now that both centralised and decentralised models have been introduced, it is interesting to consider the link between them. To this end, Xuan and Lesser (2002) consider how to convert a centralised MDP formalisation into a decentralised version with a communication strategy. The motivation for this work comes from the fact that solving centralised MDPs has PSPACE complexity (Papadimitriou and Tsitsiklis, 1987), whilst decentralised MDPs has NEXP-time complexity (Bernstein et al., 2000). Because of this, it is sometimes desirable to generate plans using centralised MDPs and then convert them to decentralised versions. An interesting aspect of this work is that the conversion can be modified to balance global utility with communication overhead. This transformation is, in spirit, similar to the one we propose using reward shaping. However, specific communications are not evaluated individually (only a frequency of communicating is defined) and, furthermore, communication is not valued directly in any way as per research challenge 2. This model represents a different approach to managing communication, but it is hard to apply in large problems because even generating multi–agent MDP policies is computationally challenging (Boutilier, 1999). As a result, this does not represent a sustainable approach. Following this, the rest of the section will focus on decentralised models and their uses.

It is interesting to note that Decentralised POMDPs do not explicitly model the beliefs of the other agents — they have a flat belief space concerning only the physical state of the problem. A different extension to the single agent POMDP model *Interactive POMDPs* (I-POMDPs) (Gmytrasiewicz and Doshi, 2004*a*,*b*, 2005) does consider the beliefs of the other agents within each agent's own belief space. This follows from the work of Gmytrasiewicz and Durfee with the RMM (see Section 2.2) being replaced by a more general POMDP structure. Consequently, similar problems and benefits are manifest. The benefits of this approach are clear — sophisticated models of other agents allows a more refined analysis of their behaviour and better predictions of their actions. Conversely, these beliefs can be nested to infinite levels and the necessary increase in the belief space means that, in general, only approximate solutions are computable. The increase in complexity can be seen in the formal definition $IPOM = \langle IS_i, A, T_i, \Omega_i, O_i, R_i \rangle$ where IS_i is the set of interactive states defined by $IS_i = S \times_{j=1}^{N-1} M_j$ for N agents, S physical states and M_j models of other agents. M_j can be composed of subintentional models (history independent or no-information) or intentional models (modelling beliefs and assuming rationality). The intuition in our research is that good performance can be achieved without modelling other agents — but by an efficient communication policy which follows research challenges 1 and 2. Furthermore, an assumption of I-POMDPs is model non-manipulability (MNM) which says that agents do not have the ability to modify the models of other agents directly. The ability

to communicate breaks this assumption. Consequently, we will only consider I-POMDPs as a benchmark for comparison.

Considering Decentralised POMDPs further, they can be classified in several ways. The models described above can exhibit *Transition Independence* and *Reward Independence* (Becker et al., 2003). The use of communication influences whether a particular decentralised POMDP has these properties. In more detail:

Transition Independence. A 2-agent DEC-MDP is said to be **transition independent** if there exists a P_1 and P_2 such that

$$P(s_1'|(s_1,s_2),(a_1,a_2),s_2') = P_1(s_1'|s_1,a_1)P(s_2'|(s_1,s_2),(a_1,a_2),s_1') = P_1(s_2'|s_2,a_2) \tag{2.8}$$

That is, the new local state of each agent depends only on its previous local state and the action taken by that agent.

Reward Independence. A 2-agent DEC-MDP is said to be **reward independent** if there exists an R_1 and R_2 such that

$$R((s_1,s_2),(a_1,a_2),(s_1',s_2')) = R_1(a_1,a_1,s_1') + R_2(s_2,a_2,s_2') \tag{2.9}$$

That is, the overall reward is composed of the sum of the two local reward functions, each of which depends only on the local state and action of one of the agents.

Against this background, it can be seen that if both of these properties hold, then the decentralised POMDP can be trivially reduced to two separate POMDPs. Unfortunately our problem domains do not exhibit either of these properties — rewards depend on the actions of the agents in some non-additive way and communication represents an action of one agent which influences the other agents in some explicit or non-explicit way (transition independence is equivalent to MNM from I-POMDPs and communication breaks this property). Consequently, the focus of this work is the general communication model and so we need the most general decentralised POMDP formalisation to provide a setting in which rational communication is need. However, we will see how reward shaping can be used to achieve this separation.

In more detail, Decentralised POMDP models allow an interesting study of rational or decision theoretic communication. In particular, Becker et al. (2009) defines the *Value of Communication* as the difference in expected reward if communication happens or not, in the decentralised MDP case with transition and observation independence. These assumptions are relaxed in the decentralised POMDP case in Carlin and Zilberstein (2009). Whilst these models tackle research challenge 2, however, in both cases communication is assumed to be myopic and furthermore these models do not consider a number of key issues. Specifically, the cost model for messages is dependent on the message being sent, but not on the state of the agent (research challenge 1). As a result, this does not consider the case that in some states (e.g. due to the geographical location of the agent) communication is more expensive than in others. Moreover, whilst the model

does consider unreliable communication in that the communication act may fail, it assumes that the act will either succeed for all agents in the broadcast or for none of them. This is not a realistic assumption in this research because we are considering domains where communication availability is not consistent across the state of the problem (e.g. some agents may be out of communication range). More subtly, communication has a cost, but the model does not account for the benefits that communication may bring except implicitly in terms of group synchronisation. In this work a different approach is taken which models the benefits of communicating explicitly. In particular, valuing communications explicitly should allow the model to only communicate when it is the most beneficial thing to do (in terms of the future reward that can be obtained after communicating) — the key requirement of this work in research challenges 2 and 3.

To sum up then, decentralised POMDPs are a well studied formal model for the coordination problem, and there are many reasons to use them in this research. They allow the separation of the communication and acting problem, although at this time there is no result indicating whether the two policy solution is better than treating communications as primitive actions and generating a single policy. The intuition is that the agents should always be able to act and communicate at the same time, but restricting this can lead to quicker policy computation, because there is only one policy to find. Furthermore, Decentralised POMDPs and POSG (considered to be equivalent) are well equipped to deal with stochastic, partially observable domains. They can model both the problem and the team, making them very flexible. Finally, more complex planning can be achieved by casting the model into the future. This makes Decentralised POMDPs and POSG the most appropriate formalisation to use in this research. As a result, Decentralised POMDP models will form the point of departure for this research, and so it is important to investigate policy computation algorithms for these models since this is needed to provide a solution to research challenge 3. This is because we need to compare the policies generated with and without communication valuation, and their respective performance. This will be done in the next section, where the solution classes will be defined with respect to the challenges defined earlier.

2.4.3 Policy Generation

Having decided to adopt a decentralised POMDP solution, we must now consider the algorithms which can solve such models and the influence of communication within these solutions. The solution to the decentralised model consists of two policies: one is the normal action policy for the POMDP associating belief states with actions and the other policy associates belief states with communication acts. When a 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. In more detail, a policy is typically represented as a set of α vectors (each representing the expected reward for following a specific action). These vectors are defined across the belief

space, and the agent simply selects the action corresponding to the α vector which is maximal at the belief of that agent. This gives the optimal action at that point.

With this established, there are several ways to solve decentralised POMDPs and it is important to consider what POMDP solution classes are appropriate for the sizes of problems considered in this research. In particular, POMDP solvers can be divided into the following three classes; each of which will be discussed in turn in the remainder of this section:

- **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.

Offline algorithms can be divided into optimal and approximate solutions. The former process the model offline to generate an optimal plan for each agent to follow. There are optimal solutions for finite and infinite horizon policies (Hansen, 1998). However, they require a fully specified model and, for anything except the smallest problems, an intractable amount of time and memory. Because of this intractability, several approximate methods have been proposed (Poupart and Boutilier, 2003), which can solve slightly larger problems, but still not the sorts of sizes seen in typical multi–agent domains. There are also error bounded approximate algorithms such as Point-Based Value Iteration (Pineau et al., 2003) where the problem is solved for a number of belief points rather than the entire belief space and the error is bounded by the density of the belief points considered. However, as a result, these problems are too large for exact or approximate offline policy computation (Hauskrecht, 2000).

Instead of generating policies offline before acting on the problem, the agents can use online algorithms to generate policies during the problem. Here, the agents generate single or multi-step plans whilst acting on the problem. These solutions have the advantage of not requiring large amounts of offline computation beforehand (although some algorithms, such as *BI-POMDP* (Washington, 1997), use offline computation to improve the accuracy of the policies they generate), but, in general, they still require a fully specified model. An example of this is Paquet et al. (2005), who details an algorithm for online search in a problem formulated as a POMDP — *Real Time Belief Space Search (RTBSS)*. Following this algorithm, at each time step the agent searches a factored belief space of reachable states in order to find the action with the highest expected reward. In this algorithm, a heuristic function is required to give an estimate of the utility of each belief state. This algorithm has shown good results in domains as large as RoboCupRescue, which is too large for offline techniques, although an obvious difficulty is how to generate the heuristic function used to prune the search space. This represents a closer fit to our requirements since large problems can now be considered.

There are also hybrid approaches that utilise online learning of the POMDP model. In general, these methods learn something about the underlying model, such as an approximate solution, and then use online algorithms to improve this approximate solution in real-time. This is a useful approach since online approaches are often easier to compute but have a lower accuracy — and this technique aims to achieve the benefits of both. In more detail, AEMS (Ross and Chaib-draa, 2007) is an example of this which provides better results than Paquet et al, at the expense of solving the underlying MDP offline. Unfortunately, the same limitations as in the case of online algorithms are evident in this approach, but also with the same limitations as offline algorithms. Consequently, they are of interest but not useful in our problem domain.

Up to this point, all of the above algorithms were originally designed for single agent models. But, in order to apply these to the MAS 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 MAS case can reduce the size of the problem by exploiting interaction between the agents. In more detail, in problems with suitable structure, agents can be optimised independently most of the time and only jointly at certain pre-defined points. This reduces the computation needed but is only useful for problems exhibiting a suitable structure. This is seen in the solution detailed by Szer and Charpillet (2005) which finds optimal policies for Decentralised POMDPs, but ignores communication and is only suitable for problems where agents interact only at well defined points. Similarly, in the ACE-PJB-Comm algorithm, Roth et al. (2005) consider communication by assuming it is free in the offline planning stage, and then reason about it online. Following this algorithm, at each step, agents calculate the joint action with and without sending its observation history. If the communication version results in a better outcome, then the observation history is communicated. We will dicuss this algorithm in more detail in Section 2.5.1.1, however for now this approaches research challenge 2 and we view this algorithm as representing the state-of-the-art in online valuations for communications in decentralised POMDPs, and, as such, we will compare the performance of our mechanism against it in later chapters. However, this model relies on maintaining joint beliefs, which grow as no communication action is taken. As a consequence, the joint beliefs 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. The online search method is also seen in POSGs. Specifically, Emery-Montemerlo et al. (2004, 2005), approximate the whole problem as a series of single step Bayesian games. This closely parallels the approaches taken in Paquet et al., but the algorithm is explicitly multi–agent. Unfortunately, however, it can only be used myopically, or it suffers from the same intractable complexity in the planning case, making it inappropriate for our problem.

To sum up, the requirements for an action selection mechanism in decentralised POMDPs are that it must be tractable for large problems and consider the MAS case explicitly. From this review, it is clear that online algorithms are the only methods applicable for large problems, however, in order to meet research challenge 3 we must understand the loss in utility that comes

from using an online algorithm. Thus, in this research we will extend *RTBSS* from Paquet et al. (2005) to the MAS case, since it has shown good results in large problems, with no offline computation.

2.5 Rational Communication

This section considers the issue of how inter-agent communication can be valued as per research challenge 2. Specifically, the question is how the sender can estimate the value to the team of a particular communication. This is the core question of this research. Once that value is known, then reasonably standard decision theoretic models can select the most appropriate action or communication. There are two clear models for valuing communications in coordination — mis-coordination and information. We also consider formal agent communication and there are a number of other approaches that are less easy to classify, that is, they are not specified in terms of the general coordination mechanisms described previously, but instead are part of more ad-hoc coordination models. Nonetheless they still represent attempts at rational communication and should be considered.

Consequently, we first cast 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. This makes it a good approach for tackling all the research challenges from Section 1.1. After this a valuation based on the information content of a message is considered. This uses information theory to measure communications — which, as we will see, makes it difficult to reason about research challenge 3. We then consider formal agent communication (which is a distinct problem to the one we are tackling, however it is worth looking at why that is). Finally, some heuristic valuations of communication are analysed to see if they offer any inspiration in terms of what makes communication valuable for the general problem.

2.5.1 Valuing Coordination

This section considers how to reason about the coordination problem explicitly. If the problem can be reasoned about, then a value for improving coordination though communication can be derived. This is referred to as the *value of communication*.

In this context, Section 2.1 showed that teamwork models are capable of reasoning about the coordination problem. With this in mind, an extension to STEAM in Zhang and Tambe (2000), *STEAM-L*, uses an MDP to analyse the future impact of coordination activities. Essentially the model attempts to ascertain the cost of attaining coherence when the agent is not making task-achieving actions in the meantime. This is an important advance in the context of this review as our ultimate aim is to build a model which can calculate the value of coordination in environments with restricted communication. Unfortunately, however, this approach is embedded in the

STEAM model, and consequently has all of the drawbacks mentioned earlier. Other techniques (discussed below) also attempt to model the team in order to value communication, but do not perform any distributed planning, making the communication overhead associated with them much less.

In more detail, Gmytrasiewicz and Durfee (2000) value communication by how it changes the knowledge of the other agent. If the agent has an estimate of the knowledge of the other agents, then a coordinated action can be selected. Thus the valuation is based on the difference in coordination that is achieved. This method is also seen in Roth et al. (2005) and we will discuss this algorithm in some detail next. However, the main difficulty with these approaches is that each agent must maintain an explicit model of the team in order to analyse how coordination is influenced. Although this is a powerful technique, for medium sized teams (tens of agents) this is not practical because of the sheer size of this team model.

In conclusion, coordination modelling is clearly a powerful method for deriving the value of communication, but it is also clear that it requires an extensive team model. Nevertheless, valuations based on the improvement in coordination are perhaps the closest to the "true" value (although how to measure this is a hard question). However, there exist several problems — the agents must model the coordination problem, which may be impractical for medium or large teams, and this model must remain consistent with the actual state for the valuation to be accurate. For these reasons, coordination modelling with heavy–duty models will not be pursued as a valuation in this work to meet research challenge 2. However, first we will describe in detail one algorithm for valuing coordination when deciding whether to communicate or not.

2.5.1.1 ACE-PJB-Comm

This section describes the ACE-PJB-Comm algorithm in some detail since it represents the state-of-the-art in execution time communication decisions and is the main benchmark in our work. This algorithm out performs all other current approaches to this problem such as decision rules or heuristic valuations. The algorithm consists of two parts - a heuristic for executing a centralised policy in a decentralised fashion that avoids mis-coordination and a decision rule for whether or not to communicate. We will describe these parts next.

The input to the algorithm is a centralised policy defined over joint beliefs b^t at time t. The algorithm then models the distribution of possible joint beliefs that could have been observed by the team as a tree with the set of leaves at depth t denoted by $\mathcal{L}^t$. Each $\mathcal{L}_i^t$ is a tuple consisting of $\langle b^t, p^t, \overrightarrow{\omega}^t \rangle$ where $\overrightarrow{\omega}^t$ is the joint observation history that would lead to this leaf, b^t is the joint belief and p^t is the probability of this joint observation history. If each agent avoids including its local observations then this tree is computed identically by all the agents. Consequently, if actions are selected according to this tree then the agents are guaranteed to be coordinated (although not necessarily taking the optimal action). The leaves in this tree are expanded according to the algorithm in Figure 2.2.

GROWTREE($\mathcal{L}_i^t, a$)
  $\mathcal{L}^{t+1} \leftarrow 0$
  $b^t \leftarrow b(\mathcal{L}_i^t)$
  FOR ALL $\omega \in \Omega$ **DO**
    $b^{t+1} \leftarrow 0$
    $Pr(\omega|a,b^t) \leftarrow \sum_{s' \in S} O(s',a,\omega) \sum_{s \in S} T(s,a,s') b^t(s)$
    FOR ALL $s' \in S$ **DO**
      $b^{t+1}(s') \leftarrow \frac{O(s',a,\omega)\sum_{s \in S} T(s,a,s') b^t(s)}{Pr(\omega^j|a,b^t)}$
    END FOR
    $p^{t+1} \leftarrow p(\mathcal{L}_i^t) \times Pr(\omega|a,b^t)$
    $\overrightarrow{\omega}^{t+1} \leftarrow \overrightarrow{\omega}(\mathcal{L}_i^t) : \langle \omega \rangle$
    $\mathcal{L}^{t+1} \leftarrow \mathcal{L}^{t+1} \cup \left[b^{t+1}, p^{t+1}, \overrightarrow{\omega}^{t+1} \right]$
  END FOR
  RETURN $\mathcal{L}^{t+1}$

FIGURE 2.2: Algorithm to grow the children of one leaf in a tree of possible beliefs (Roth et al., 2005)

Now, an action is selected according to the following Q-POMDP heuristic using the tree generated previously:

$$Q - POMDP(\mathcal{L}^t) = argmax_a \sum_{\mathcal{L}_i^t \in \mathcal{L}^t} p(\mathcal{L}_i^t) \times Q(b(\mathcal{L}_i), a) \tag{2.10}$$

where:

$$Q(b^t, a) = \sum_{s \in S} R(s,a) b^t(s) + \gamma \sum_{\omega \in \Omega} Pr(\omega|a,b^t) V^{\pi}(b^{t+1}) \tag{2.11}$$

This approximates the decentralised POMDP using the centralised policy generated for the underlying POMDP and maximising expected reward over possible joint beliefs. This heuristic is conservative since it ignores local observations, and the communication substage is used to decide whether communicating these observations would improve the action that can be selected.

In more detail, the ACE-PJB-Comm algorithm compares the expected reward for the action that would be selected if it communicated its information to its teammates, a_C, to the reward for the action if it did not, a_{NC}. If communication would result in a better expected reward then the message is sent and the algorithm is run again with the new belief state. This is seen in Figure 2.3. It should be noted that receiving a communication causes the agent to run this algorithm, and possibly communicate again — which may cause multiple instances of communication and so agents must wait a period of time for the system to reach a stable state. Our mechanism will aim to address this flaw (although we could simply assume broadcast synchronisation communication).

DEC-COMM($\mathcal{L}^t, \overrightarrow{\omega}_j^t$)
  $a_{NC} \leftarrow Q - POMDP(\mathcal{L}^t)$
  $\mathcal{L}' \leftarrow$ prune leafs inconsistant with $\overrightarrow{\omega}_j^t$ from $\mathcal{L}^t$
  $a_C \leftarrow Q - POMDP(\mathcal{L}')$
  IF $a_{NC} \neq a_C$ **THEN**
    communicate $\overrightarrow{\omega}_j^t$) to the other agents
    RETURN $DEC - COMM(\mathcal{L}', 0)$
  ELSE
    IF communication $\overrightarrow{\omega}_k^t$) was received from k **THEN**
      $\mathcal{L}^t \leftarrow$ prune leafs inconsistant with $\overrightarrow{\omega}_k^t$ from $\mathcal{L}^t$
      RETURN $DEC - COMM(\mathcal{L}^t, \overrightarrow{\omega}_j^t)$
    ELSE
      take action a_{NC}
      receive observation $\overrightarrow{\omega}_j^{t+1}$
      $\overrightarrow{\omega}_j^{t+1} \leftarrow \overrightarrow{\omega}_j^t : \langle \overrightarrow{\omega}_j^{t+1} \rangle$
      $\mathcal{L}^{t+1} \leftarrow 0$
      FOR ALL $\mathcal{L}_i^t \in \mathcal{L}^t$ **DO**
        $\mathcal{L}^{t+1} \leftarrow \mathcal{L}^{t+1} \cup GROWTREE(\mathcal{L}_i^t, a_{NC})$
      END FOR
    END IF
  END IF
  RETURN $\left[\mathcal{L}^{t+1}, \overrightarrow{\omega}_j^{t+1}\right]$

FIGURE 2.3: One time step of the Dec-Comm algorithm for an agent j (Roth et al., 2005)

2.5.2 Valuing Information

Rather than valuing communication by how it alters the other agents' knowledge, another approach is to value communication based on the amount of information it conveys. The latter can be achieved using information theory. This has the advantage of valuing communication on local calculations (as opposed to team models), but, on the other hand, information theoretic methods only produce a heuristic estimate, unlike methods which value coordination explicitly, and consequently, we cannot say if research challenge 3 is achieved. As a result, this method represents an approximation, whose usefulness is heavily dependent on the information measure employed. Still, it is interesting to pursue, with the aim of making it more general.

To this end, Rogers et al. (2005, 2006) and Padhy et al. (2006) develop economics-inspired approaches to the problem of balancing acting and coordinating within the domain of sensor networks, which calls for a value of communication. In this domain, nodes can transmit their own observations to a central station or to each other in order to minimise communication cost by hopping. Thus, nodes closer to the centre can relay the observations from nodes further out. These nodes have limited battery life and communication is therefore a very expensive act. With this in mind, communications are valued based on their information content. In Rogers et al. (2006) the interesting aspect is that observations are valued in terms of how much the observation reduces the uncertainty about some variable (using Fisher Information (Schervish,

1995)). Essentially, the mechanism aims to distribute the most useful observations between the agents in order to make the entire network more efficient. A similar approach is detailed in Dash et al. (2005) which values the information based on the reduction of uncertainty using a Kalman Filter. Following this, it is clear that this work evaluates how valuable a communication is before sending it. Whilst the value here is in terms of maximising information gain for sensor networks, it is not difficult to transfer that value to Decentralised POMDPs that must share observation histories to remain synchronised — which represents a useful direction for this research and one we will consider further in later chapters whilst tackling the research challenges.

To sum up, if the coordination problem can be cast as each team member having a similar impression of the global state of the problem, then information theory is a simple and general way to value how much a communication will impact this impression. Therefore, this model relies on agents taking coordinated actions if they share the same knowledge. Specifically, communications with a large information value are more useful to the team than communications with a low information value, and this measure can be calculated locally based on the agents' own knowledge. The difficulty in this approach, however, is that the information content of a particular communication is not necessarily the same to every member of the team (especially if there has been a long time since the last synchronisation of team knowledge). In this case, the information content of a communication must be assessed with respect to some level of team knowledge. Also, the issue of how to normalise the information value with rewards from the problem domain is a challenge and may require some form of calibration (thereby making any solution less general). Despite this, as a first step, this research will utilise information theory to value communication, since it is easy to integrate with decentralised POMDPs and requires no explicit coordination framework, unlike methods based on valuing coordination.

2.5.3 Formal Agent Communication

Formal agent communication is concerned with specifying the impact and use of communication in a very high-level way compared to what we have discussed up until this point. In more detail, Speech act theory (Austin, 1962), much like rational communication, is concerned with treating communication as an action which has the effect of influencing the activity of the other agent by changing their mental state. This approach is similar to Gmytrasiewicz and Durfee (2000), although it is expressed formally, rather than in a Bayesian fashion, which makes it less useful for our aims (research challenge 3). Now, Speech act theory informed the development of formal agent communication such as KQML (Finin et al., 1994). These languages specify the semantics and syntax of communications between agents. Work such as Pitt and Mamdani (1999) attempt to make these semantics verifiable so that agent communication systems can be engineered with provable results. This allows open agent systems to interact such as in argumentation systems (Artikis et al., 2007). Whilst this is a useful direction for open agent systems, it is distinct to the problem we are considering which is valuing information exchange between relatively similar agents. Consequently, we will not consider them further.

2.5.4 Heuristic Valuations

This section describes a number of heuristic and domain specific communication valuations and policies for agent teams. They do not provide a general solution, and are simply considered for inspiration about what makes a communication valuable (research challenge 2). First, we discuss approaches that attempt to identify who needs information in a team and generate a policy based on that. After this, we consider approaches which attempt to parameterise the level of communication and coordination in robotic and network routing domains — which is useful in research challenge 3.

In the context of defining the information needs in a team, an interesting approach by Zhang et al. (2004) attempts to define information "needers" and "providers" in a team based on analysing the plans the agents are about to execute. It then gives a communication strategy selection algorithm, called *DTPC (Decision-Theoretic Proactive Communication)*, which each agent uses, based on its role, to select whether to communicate. The expected utility of these strategies is calculated by estimating the information production of providers and the information rate necessary for the needers. This is interesting in that it makes communication an online issue, rather than integrating it with the offline planning stage. Unfortunately, there are problems in that it requires repeated interactions to estimate the communication requirements, and does not consider the need to distribute exceptional information. Other issues arise from analysing the plans to determine team roles. Also, in many situations it may be that all agents are both providers and needers, and this algorithm relies on information going one way for its efficiency. Furthermore, the assumption is made that communication is always available and uniform in cost. Because of this, it is difficult to see how the need to communicate can be integrated with action taking. Also, in the theme of reasoning about the information needs of team members, Yen et al. (2004) develop a similar logic based approach. However, in both of these approaches, no explicit value or cost is placed on the communication acts (as desired in research challenges 1 and 2), but this represents another way to determine a value of communication — the needs of the team members. Although this will not be used at this stage, it is something to consider for the future.

Instead of attempting to define communication pathways, the level of communication can be parameterised by a characteristic of the problem state (i.e. the distance between agents). For example, Rosenfeld et al. (2006) use a neighbourhood parameter to decide on levels of communication in a robotic foraging task. In their work, a robot has a local neighbourhood and as more robots enter this area, the level of communication is increased from none, to locally direct, to a centralised server. In more detail, the task of coordinating is transferred from the robot to a central server as robots get closer together. Although this approach produced good results in robotic foraging (a partially observable, stochastic problem), it is difficult to see how it can be applied more generally, especially as the top level of communication amounts to a centralised solution which is impractical in the domains considered in our work.

Following the approach of parameterising a level of communication, in Dutta et al. (2006, 2007) the problem of co-ordination and acting in a mesh network routing domain is considered. Here, calls must be routed from one node in the network to another via some path of nodes in-between. Typically, there is limited bandwidth between the nodes which can be used for routing calls and for coordination messages. Following this, the problem here is that increasing the degree of coordination (communication) reduces the available bandwidth for routing calls and hence the efficiency of the network. To address this problem, a protocol is developed which models each node as an agent and the agents themselves decide which neighbour to route a call through based on their observed history of bandwidth availability through each neighbour. In addition to routing calls, nodes (agents) are also responsible for propagating their available bandwidth information. The decision problem of when to communicate this information is based on whether it has changed beyond a threshold from the last time it was communicated. This threshold is set based on some offline network usage statistics and is found to give varying performance. Clearly a better solution would adapt the threshold online based on how the network is being used. As a consequence, this indicates another communication valuation — the change in state since the last communication. Dutta et al. (2006, 2007) presented this in a very domain specific way, but the idea could be generalised using information theory to measure the change in state, as was discussed in the last section. We will attempt to generalise this idea in Chapter 4 to meet the research challenges.

In the context of adaptive communication valuations, learning has also been applied to the problem of rational communication for co-ordination. An example is Ghavamzadeh and Mahadevan (2004) where the *COM_cooperative_HRL* algorithm is developed. This algorithm is based on Hierarchical Reinforcement Learning where the agents learn a task decomposition of the problem modelled as a Semi-Markov Decision Process (SMDP). This is similar to the MDP formalisation but actions are allowed to take different lengths of time. The technique is used in the hope of making reinforcement learning tractable in real domains. In more detail, agents learn to balance communication cost with the need to coordinate to maximise the team goal. Learning has also been applied to the problem of who to communicate to. In particular, Ohko et al. (1997) adjust the communication load adaptively by learning proper addressees for Task Announcements messages on Contract Net Protocol (CNP) with Case-Based Reasoning (CBR) with the aim of avoiding broadcasting. Whilst this is an example of learning applied to rational communication it is not clear how it could be adapted to the problem domain considered here. Following the previous section, Kinney and Tsatsoulis (1998) attempt to learn the information needs of neighbouring nodes so that efficient and responsive information routing is achieved. This is enacted using a feedback control system and a classification of types of information. The types of information are associated with a usefulness to neighbouring nodes who communicate back feedback indicating that usefulness. However, this is a very static and domain specific learning regime, which is heard to generalise.

To sum up, this section has introduced another source of communication value — the information needs of team members. However, the approaches described here are very domain specific

and offer only static solutions. Furthermore, we have also introduced a model to control the amount of communication — parameterising it based on some environmental characteristics. Again, the approaches here are very specific, and do not consider the question of how to dynamically adjust the thresholds they introduce. Learning has also been considered, this is a useful technique if the question of how to value the communication is answered. Agents would be able to send only high value communications if it is expensive to communicate, and balance this with other rewards in the system. In this context, learning could be used to achieve this dynamic behaviour, although it is clear that none of the work here does this. Unfortunately, the difficulty with all these approaches is that their valuations are heavily integrated with the problem they consider. As a consequence, this research will not consider them any further.

2.6 Test Domains

In this section we will describe the two test domains from the literature which we use to evaluate the effectiveness of our techniques. The first is the Multi–Agent Tiger problem (Nair et al., 2003) with an extension to restricted communication, which is a standard problem in sequential decision making and allows us to test our algorithms against theoretical optimal results and the current state-of-the-art. There are, of course, other smaller test domains with similar characteristics, such as cooperative box pushing. However, we felt the Tiger domain offered the correct mix of sequential planning and partial observability - for instance the box problem becomes a standard planning problem once both agents have initially agreed on where the box is. We also test our formalism in RoboCupRescue (Hiroaki, 2000) which represents a very large problem that existing techniques cannot solve without extensive amounts of domain knowledge. We use this domain to demonstrate the scalability of our approach. This is a good problem to show the real world applicability of our approaches since it combines a large scale search problem and tightly coupled tasking — making it unique in the literature.

2.6.1 The Multi–Agent Tiger Domain

This section describes the Multi–Agent Tiger problem — 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. The Multi–Agent Tiger domain is a multi–agent extension to the classic Tiger problem, which we describe here along with the modifications we have made to incorporate 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. Each agent can open either door ⟨OL⟩ or ⟨OR⟩. 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 — a ⟨LISTEN⟩ action. 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 — a ⟨COM⟩ action. 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 Nair et al. (2003) with the modification that we have introduced 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. Following this, the Tiger problem is used because it meets all of our requirements for a scenario (see Chapter 1):

- **Coordination**: Agents need to agree on which door to open to maximise reward.
- **Communication**: Communication can be used (or not) to make this agreement.
- **Decentralised**: Agents can only make local observations and communication is limited — so a centralised controller cannot be employed.
- **Partial Observability**: The location of the tiger is unknown, however agents can make noisy observations of this information.
- **Stochastic domain**: Interactions with the simulation world fail with some small probability. In more detail, communications may fail, incorrect doors could be opened for example.

2.6.2 Rescuing Civilians

RoboCupRescue (Hiroaki, 2000), shown in Figure 2.4, represents a widely studied problem, that has many existing tools for developing agents and is easily extended to our specific problem. Specifically, there is an existing simulator which requires very little modification for our purposes. Further to this, it is a problem which requires a high degree of scalability — the problem can have many agents and each instance has many state variables. Consequently it is a good benchmark for showing that approaches can deal with a very large state and observation space. However, as we described in Section 2.4.3, there exist few algorithms which are scalable in this

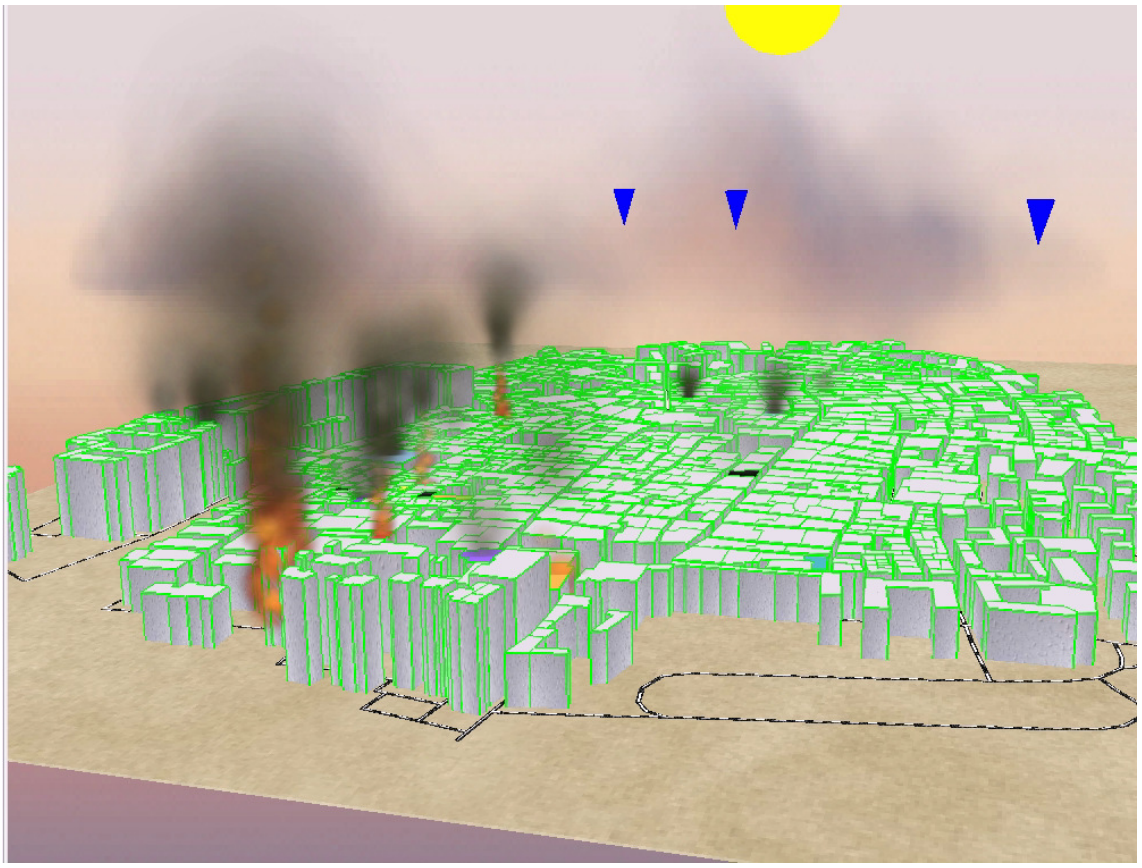
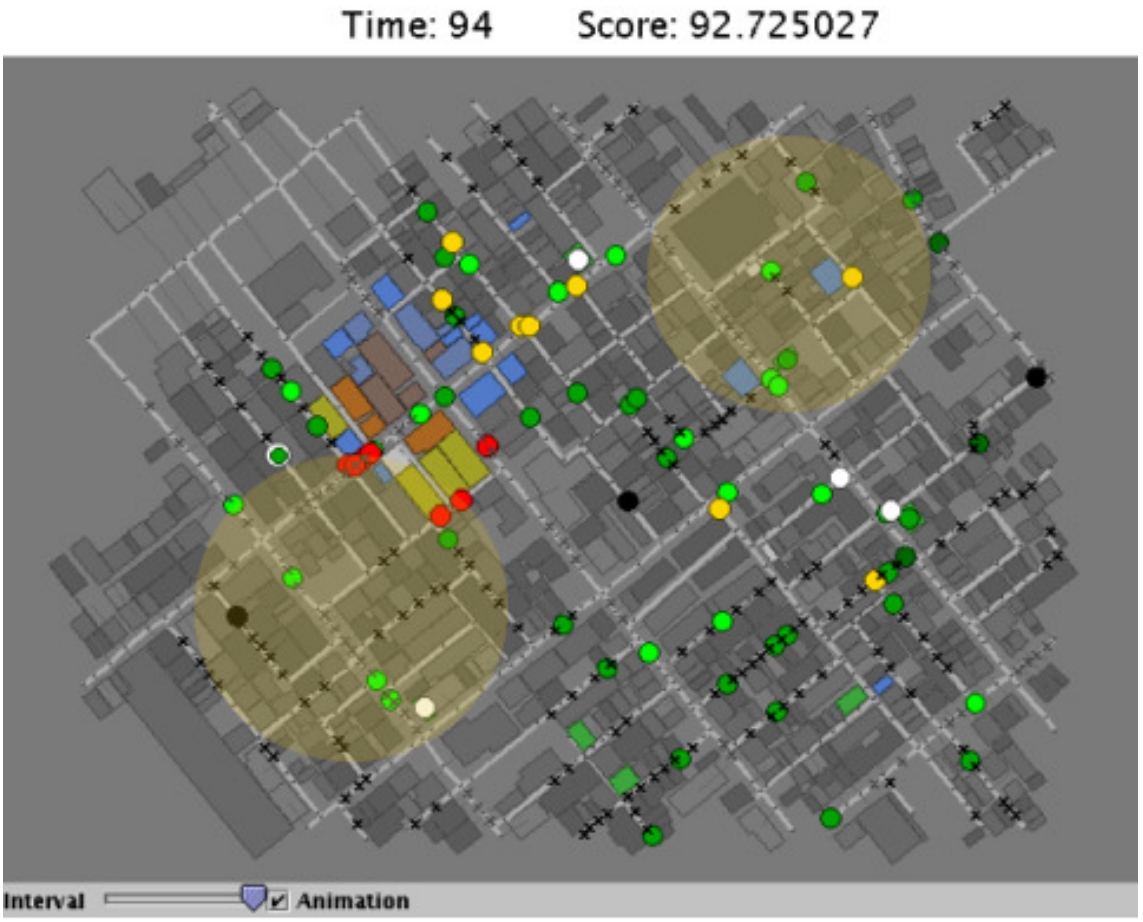


FIGURE 2.4: RoboCupRescue in 3D

respect and so our benchmarks in this problem take the form of rule based heuristics. This is still a useful comparison, since general techniques should be developed for large approaches rather than relying on domain dependent algorithms.

RoboCupRescue, in more detail in Figure 2.5, is a multi–agent 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. 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 Brigades must fight the spread of the fire; and the Police 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, as seen in Figure 2.5. Agents have limited sensing capabilities, specifically, 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 about 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 and will die after some time at a rate defined by the damage incurred. Communication is peer to peer (i.e. messages are sent to and from any agents without going through a central controller) and has a cost which we



- Red Circle : Fire Brigade
- White Circle: Ambulance
- Yellow Circle: Police
- Green Circle: Healthy Civilian (darker means less healthy, black is dead)
- Red, Orange, Yellow Rectangles: Burning buildings (red is burning a lot, orange is some burning and yellow means burning a little)
- Grey Rectangles: Normal Buildings (no fires)
- Blue Rectangles: Extinguished fires
- Crosses: Road blocks
- White and Green Circle: Ambulance carrying a civilian
- Yellow areas: Blackout zones

FIGURE 2.5: RoboCupRescue

can define for our problem. Communication is prohibited in 'blackout' zones (cannot send or receive) and the agent must move out of that area in order to make communication possible.

In this context, the full RoboCupRescue problem requires several components not relevant to this research (such as an estimation of how fire spreads and very efficient path planners), 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). Following this, the simulator is used because it meets all of our requirements for a scenario (see Section 1.1):

- **Coordination**: This scenario will concentrate on the Ambulance agents which need to coordinate to rescue as many civilian agents as possible from damaged buildings and deliver them to shelters. These civilian agents are only alive for a certain length of time so the ambulances work under a time constraint. Specifically, trapped civilians can be saved more quickly if ambulances work together on them. Furthermore, the disaster area is large and an efficient search strategy is required. As a result, the ambulance agents need to coordinate to free trapped civilians together (before the civilians die), and they need to organise how they search the map to avoid repeated effort.

- **Communication**: There is a communication medium which utilises a single time step to send a message, when the agent cannot do anything else — thus it is a restricted resource because each agent can only send a single message per timestep and cannot do anything else. In some areas communication is not possible because the communication medium is saturated — this is defined probabilistically for each simulation.

- **Decentralised**: Agents can only make local observations and communication is limited — so a centralised controller cannot be employed.

- **Partial Observability**: The initial distribution of civilians is not known in advance by the ambulance agents and they continue to appear at random as emergency calls are made. Ambulance agents can make observations of where there are civilians that need rescuing, but only within a small radius of that ambulance.

- **Stochastic domain**: Interactions with the simulation world fail with some small probability. In more detail, messages may be lost, move actions may not result in the desired location and load/unload actions are not guaranteed to succeed.

To sum up, RoboCupRescue represents a scenario which fulfils all of our requirements. Because of this, we will use it in our evaluation, and specify our models in terms of RoboCupRescue as an example.

2.7 Summary

This review has analysed several coordination frameworks, and how they address the communication problem. Some, such as teamwork models, allow future coordination to be considered and provide a way to tackle research challenge 2, but are necessarily more complex than approaches which do not model the team. Others allow team knowledge to be specifically encoded and reasoned about. As a result, the Decentralised POMDP and POSG formalisations were chosen because of their general nature, and the ease with which a problem containing domain actions and communications can be modelled. They also allow simple integration of many different types of communication valuation. Consequently, the work presented next will modify these formalisms so that they model the cost of communication generally (not a parallel activity) and meet research challenge 1. Following this, information theory is one such valuation, which was chosen because it does not place any demands on the coordination mechanism, making it a general method of valuing communication. The only issue with following this approach is that we need a way to make the link between information and coordination in order to solve research challenges 2 and 3.

However, traditional approaches based on Decentralised POMDPs do not explicitly value the communication acts that they allow; they rely on the value being inferred during policy computation — which does not meet research challenge 2. Our work aims to value communication directly so that it can be reasoned about explicitly and used to solve research challenge 3. If this is done, then large amounts of policy computation are not required to derive an implicit value. Also, if a general valuation can be defined, then it can be used in domains outside of completely specified models, and form the basis of a dynamic rational communication model. The next chapters will go some way towards achieving this model and showing its utility.

Chapter 3

Offline Learning of Communication Valuations

This chapter describes a general model of decentralised coordination that is suitable for small scale agent teams (up to ten agents) using rational communication based on offline learning. Our model will attempt both research challenges 1 and 2 from Chapter 1 by specifying a general framework for capturing the cost of communication, and using offline learning to find the value of communication. This generic model is instantiated in the RoboCupRescue problem. To this end, in Section 3.1 we describe the architecture of our model, *dec_POMDP_Valued_com*, for coordinating with restricted communication and detail a modified online action selection algorithm (based on Paquet et al. (2005), see Section 3.1.2). Then, in Section 3.2 we cast the RoboCupRescue problem (see Chapter 2 for more details) as a *dec_POMDP_Valued_com* and give a worked example of rational coordination using our mechanism in this scenario. Then we empirically evaluate our approach in Section 3.3. Finally, Section 3.4 concludes.

3.1 A Model of Coordination with Communication Valuation

In this section we first introduce our model *dec_POMDP_Valued_Com* — a model of decentralised coordination which utilises an information theoretic communication valuation and offline learning, and then proceed to describe our online policy generation algorithm which has been designed to leverage the communication valuations our model calculates during the offline learning phase.

3.1.1 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 (see Section 2.4.2 for more details).

This therefore 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 prevent an agent from taking other actions — this distinction is core to tackling research challenge 1 from Chapter 1. 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 communication and domain actions.

Now, models such as I-POMDP (see Section 2.4.2) address the problem of valuing communications by considering the beliefs of other agents as part of the state space, and allowing communication actions to perturb this part of the space. However, this results in a huge model which cannot be solved by existing techniques for even small problems (see Chapter 2 for more details). Hence, we would like to remove reasoning about the value of communications from the coordination model and replace it with a principled approximation, as per research challenge 2.

In our work, we do this using a normalised (in terms of the concrete rewards available to the team) information theoretic valuation over possible communications. In more detail, in addition to the reward function defined for the problem, we include a second reward function, which is used exclusively for the communication actions. This measure gives the information gain from a particular communication relative to the communicating agent's current belief state. Consequently, our model has two reward functions that are weighted appropriately 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. 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). Furthermore, we can see that this approach goes some way to solving research challenge 2 (valuing communication either exactly or approximately). We will see later that this model does not allow for research challenge 3 to be easily solved since the error in approximation cannot be easily bounded, however work in Chapter 4 will significantly change this model (but following its spirit) to allow for this and furthermore, tackle larger scale problems.

Now, our model, the *dec_POMDP_Valued_Com*, is an extension of the *dec_POMDP_Com* model (see Section 2.4.2 for more details). As described above, we include a second reward signal for communication actions (and restrict the original to non-communication actions) and make the original reward function use a weighted combination of these two reward signals. We also include communication actions into the standard action selection policy problem. This model allows the communication valuation problem to be separated from the policy computation problem, because the agent does not need to compare the value of policies with and without

communication to get a valuation (resulting in complexity reduction benefits compared to the *dec_POMDP_Com*). Finally, we restrict the communication alphabet to the observation alphabet, in order to maintain the generality of the communication valuation. Specifically, we do this to preclude high level communications such as plans or intentions, since we are more interested in communication as information exchange. More formally, this is defined by the tuple (for 2 agents) $DECPOMVALCOM = \langle n, S, \mathcal{A}, P, \Omega, O, \Sigma, C_\Sigma, R_p, R_c, R, T \rangle$ where the definitions are the same as those for the *dec_POMDP_com* (see Section 2.4.2 for more details) except:

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

$$R_p(s \in S, a \in \mathcal{A}, s' \in S) \in \mathcal{R} \tag{3.1}$$

  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 as this represents an unrealistic assumption about the availability of communication. Furthermore, it is convenient to generate a single policy, as agents can now plan a sequence of actions to make communication available (and hence calculate a cost).

- R_c is the *communication reward function*. $R_c(b(\overrightarrow{L\omega}), \overrightarrow{H\omega})$ is the value of $\overrightarrow{H\omega}$ in the current belief state $b(\overrightarrow{L\omega})$. $\overrightarrow{L\omega}$ is the local history of observations $\omega \in \Omega$ and actions $a \in \mathcal{A}$ that the agent has experienced itself and received in communications so far. $\overrightarrow{H\omega}$ represents the history of observations since the last communication point. $b(\overrightarrow{L\omega})$ is the current belief state — a probability distribution over states which represents the agent's estimation of the current state. In more detail, $b(\overrightarrow{L\omega})$ is a probability for each state s and $Pr(s|\overrightarrow{L\omega}) \in [0,1]$ is the probability that the problem is in state s and is computed by:

$$Pr(s|\overrightarrow{L\omega}) = \frac{Pr(s) \cdot \prod_i Pr(\overrightarrow{L\omega_i}|s)}{\sum_{s' \in S} Pr(s') \cdot \prod_i Pr(\overrightarrow{L\omega_i}|s')} \tag{3.2}$$

  The communication valuation is made explicit compared to the *dec_POMDP_Com* in order to allow us to specify generic metrics over possible communications and to reduce the complexity of the basic decentralised POMDP.

- R is the reward signal supplied to the policy generation problem. Our empirical approach will aim to find an approximation for the relative importance of communicating compared with other actions. Thus we assign reward using the function:

$$R = \alpha R_p + (1 - \alpha) R_c \tag{3.3}$$

  This formalisation is used so that the approximation of the value of communicating can be changed in a principled manner over different problems and communication costs (by changing α).

The basic premise of our valuation is that, when an agent communicates, the beliefs of the other agent are synchronised with the communicator. This is not bidirectional and so all an

agent can assume is that the other agent knows what is in the communication. We then use the information available at the synchronisation point as a reference and measure the value of communicating after this point as the distance, in terms of the belief relative to this synchronisation point. Thus, we communicate the history of observations since the last communication and measure the importance of this communication in terms of that history. Essentially, at that synchronisation point, the probability of taking an uncoordinated action is small (although not zero since the synchronisation is not two-way). As the time since that point increases this probability will grow, but it is not time-dependent since the agent may learn nothing new in that time — it is *information* dependent. Consequently, we can approximate this increase in probability of mis-coordinating as an information theoretic measure. A similar idea is seen in the decision theoretic communication of *STEAM* (see Section 2.1), but in that work the probability of mis-coordination is defined for each state feature by the system designer — a very domain dependent solution which we attempt to make more general. That work also included a pre-defined cost of mis-coordination (to balance with other utilities available) which we attempt to generalise using our reward functions R, R_p and R_c. It is clear that the main difficulty with this approach is that we have created a reward function R that must be optimised offline (currently) for each problem the model is instantiated in. We will change this model in later chapters so that this offline learning phase is not required and furthermore, we can relate the approximation of the communication valuation in research challenge 2 to the utility of the global solution and hence solve research challenge 3.

Against this background, *KL Divergence* (Kullback and Leibler, 1951) is used to determine the value of R_c. In more detail, given an agent's belief state $b(\overrightarrow{L\omega})$, the difference in information between communicating $\overrightarrow{H\omega}$ and not communicating is given by:

$$R_c(\overrightarrow{L\omega}, \overrightarrow{H\omega}) = ND_{KL}(b(\overrightarrow{H\omega}) \| b(\overrightarrow{L\omega})) = N \sum_{s \in S} Pr(s|\overrightarrow{H\omega}) \cdot log \frac{Pr(s|\overrightarrow{H\omega})}{Pr(s|\overrightarrow{L\omega})} \tag{3.4}$$

where N is a normalisation factor, $b(\overrightarrow{L\omega})$ is the agent's current belief state, $b(\overrightarrow{H\omega})$ is the belief state at the time of the last communication. Furthermore, $Pr(s|\overrightarrow{L\omega}) \in [0,1]$ is the probability that the problem is in state s and is computed by:

$$Pr(s|\overrightarrow{L\omega}) = \frac{Pr(s) \cdot \prod_i Pr(\overrightarrow{L\omega}_i|s)}{\sum_{s' \in S} Pr(s') \cdot \prod_j Pr(\overrightarrow{L\omega}_j|s')} \tag{3.5}$$

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 each variable individually (recall discussion in Section 2.5.2). This is useful because we need to evaluate the reduction in uncertainty given by a set of observations, 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 (since there are many important state variables and not just a few very valuable ones). 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.

3.1.2 Policy Generation

Now that we have specified a new model of decentralised coordination, we need to describe a policy generation algorithm for that model which can leverage the communication valuations rather than modelling other agents' beliefs. As described in Chapter 2, *Real Time Belief Space Search (RTBSS)*, introduced by Paquet et al, represents a good starting point for the policy generation problem, as it has previously shown good performance in RoboCupRescue. The algorithm, in its original form, coordinates agents using a complex reward function, which rewards the coordinated actions. However, while this approach is valid for achieving coordination in their particular scenario, it is still necessary to encode explicitly how the agents should coordinate (e.g. giving more reward for having more than one agent attend a task or for having the agents attend different tasks). Thus this approach 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. With this established, if each agent has similar knowledge then they will each choose the same joint action and coordination is achieved. In more detail, each agent must estimate the actions available to the other agent and consider rewards over these joint actions. 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, which allows the agents to make coordinated actions when they have a good idea of the state of the other agent.

More formally, each agent performs a search in the belief state space $b(\overrightarrow{L\omega}) \in B$, using joint actions and local observations to generate new belief states or histories. 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). In order to predict the likelihood of each new observation, we define a new function $P(\omega|\overrightarrow{L\omega},a)$ which is the probability of an observation ω in a belief state $b(\overrightarrow{L\omega})$ given a joint action a. An action is given by:

$$\pi(\overrightarrow{L\omega},D) = argmax_{a\in\mathcal{A}} \sum_{\omega\in\Omega} P(\omega|\overrightarrow{L\omega},a)\cdot\delta(\rho(\overrightarrow{L\omega},a,\omega),D-1) \tag{3.6}$$

where $\delta(\overrightarrow{L\omega},d)$ is given by:

$$\delta(\overrightarrow{L\omega},d) = \begin{cases} 0 & \text{, if } d=0 \\ R(\overrightarrow{L\omega}) + \gamma \max_a \sum_{\omega\in\Omega} & \\ \quad [P(\omega|\overrightarrow{L\omega},a)\cdot\delta(\rho(\overrightarrow{L\omega},a,\omega),d-1)] & \text{, if } d>0 \end{cases} \tag{3.7}$$

1: **Function** Modified_RTBSS($b(\overrightarrow{L\omega})$, d, $rAcc$)
2: $b(\overrightarrow{L\omega})$: The belief state, d: The time, $rAcc$: Accumulated rewards.
3: **Statics:** D: Search time, $bestValue$: The best value found, $action$: The best action.

4: **IF** $d = 0$ **THEN** — if at the horizon of the search
5: $finalValue \leftarrow rAcc + \gamma^D \times 0$ — branch plus belief state
6: **IF** $finalValue > bestValue$ **THEN** — if better than the best so far
7: $bestValue \leftarrow finalValue$ — set to the best so far
8: **END IF**
9: **RETURN** $finalValue$ — return the value of this leaf
10: **END IF**
11: $rAcc \leftarrow rAcc + \gamma^{D-d} \times R(\overrightarrow{L\omega})$ — else add reward for Belief state to accumulator
12: $JointActionList \leftarrow Sort(\overrightarrow{L\omega}, A)$ — get next possible joint actions
13: $max \leftarrow -\infty$ — set smallest value
14: **FOR ALL** $a \in JointActionList$ **DO** — for each possible joint action
15: $expReward \leftarrow 0$ — accumulated reward is 0
16: **FOR ALL** $\omega \in \Omega$ **DO** — for all observations
17: $b(\overrightarrow{L\omega'}) \leftarrow \rho(\overrightarrow{L\omega}, a, \omega)$ — calculate next belief state
18: $expReward \leftarrow expReward + \gamma^{D-d} \times P(\omega|a, \overrightarrow{L\omega})$ — current reward + value of subtree
19: $\times Modified\_RTBSS(\overrightarrow{L\omega'}, d-1, rAcc)$
20: **END FOR**
21: **IF** $(d = D \wedge expReward > max)$ **THEN** — if this is largest so far then
22: $max \leftarrow expReward$
23: $action \leftarrow a$ — best action is current action
24: **END IF**
25: **END FOR**
26: **RETURN** max — return the value for this sub-tree

FIGURE 3.1: Modified RTBSS

where $R(\overrightarrow{L\omega})$, $P(\omega|\overrightarrow{L\omega}, a)$ and $\rho(\overrightarrow{L\omega}, a, \omega)$ are computed according to the POMDP dynamics (see Paquet et al. (2005) for more details). The algorithm which computes Equation 3.6 is presented in Figure 3.1.

In this algorithm the agent searches in a depth first fashion (line 17) through the points in the belief space that can be reached by possible joint actions (line 12) and local observations (line 16). It searches until a depth d (line 4), at which point it returns up the search tree the value of being in that belief state (line 7), and this accumulated for that branch (line 11). In this way, an agent can find the best action to take at the root of the tree — the action which leads to the best belief state weighted by the likelihood of reaching that state.

In summary, we have taken an online POMDP policy generation algorithm which considers other agents to be part of the environment and extended it to explicitly consider the agent team. We have also modified it to utilise our decentralised POMDP coordination mechanism with communication valuations in a way that does not significantly increase the complexity of the decision problem (unlike existing decentralised POMDPs). This is because the only increase in the search space over the single-agent model is in considering joint actions — we avoid considering joint observations. This gives a solution to research challenge 2 but because the online algorithm is approximate, we cannot (at this point) say whether we have solved research challenge 3. Work in Chapter 6 will address this point. Now, we use the online policy generation algorithm to power an offline learning phase for deriving the best value for α (see Section 3.3 for more details).

3.2 Coordination in RoboCupRescue

In this section we instantiate the modified RoboCupRescue problem domain, described in Chapter 2, in our model. Initially, we use this instantiation to provide a step-by-step example of our algorithm for rational communication. Then, in the next section we will use this instantiation for an empirical evaluation of this technique.

3.2.1 RoboCupRescue as a *dec_POMDP_Valued_Com*

This section instantiates the *dec_POMDP_Valued_Com* model from the previous section in terms of RoboCupRescue. As mentioned in Chapter 2, we only consider the ambulance problem in this work. Thus, 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 ($\overrightarrow{H\omega}$).

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 $\langle a_1 \Rightarrow b_1, a_2 \Rightarrow n_8, b_0 \Rightarrow 0 \ldots b_i \Rightarrow 1, n_0 \Rightarrow 0 \ldots n_m \Rightarrow 1, r_0 \Rightarrow 0 \ldots r_j \Rightarrow 1 \rangle$ 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	$R_p = c \times r + e \times r/2$ (3.8) 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 3.8) but if $e = 40$ then the reward is 3000, giving a higher reward for exploring more
R_c	$R_c(\overrightarrow{L\omega}, \overrightarrow{H\omega}) = ND_{KL}(b(\overrightarrow{H\omega}) \| b(\overrightarrow{L\omega})) = N\sum_{s \in S} Pr(s\|\overrightarrow{H\omega}) \cdot log\frac{Pr(s\|\overrightarrow{H\omega})}{Pr(s\|\overrightarrow{L\omega})}$	The belief state for a single building $b_1 = \overrightarrow{L\omega}_1 \Rightarrow []$ and the communication $\overrightarrow{H\omega} = \overrightarrow{L\omega}_1 \Rightarrow 1$, 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	
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 3.2: A *dec_POMDP_Valued_com* of the RoboCupRescue ambulance task

At each time step, agents select joint actions (an action assigned to each team member) and implement their own part of that joint action. The cost of communication C_Σ relates to the time required to send the observation history. This time is dependent on the location of the agent and the volume of 'blackout' in the city. 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 consequently a message can be composed of any symbol in the observation alphabet. Finally, our algorithm does not consider communication noise (although this would be a possible extension). In this case a communication is either received as it was sent (if not in a communication blackspot) or not at all. A summary of this formalisation is given in Figure 3.2.

There are some alternatives to the model outlined above — specifically, the state space could record the position of each civilian explicitly, but this would require knowing the number of civilians to be rescued in advance, and it would grow with the number of civilians and buildings. Another alternative is for agents to have static movement actions. In more detail, instead of dynamic actions (i.e. moving to any neighbour), the agents could have north, south, east and west movements but these are hard to interpret in terms of RobocupRescue, because it is not a uniform domain. Furthermore the communication alphabet could be local interpretations of the global state (e.g. an agent's current belief state), but we believe instantiations of state variables is a more general approach.

3.2.2 A Coordination Example

We will demonstrate this model with a simple coordination task. Two ambulances, a_1 and a_2, must rescue a civilian from a building b_1, on a map composed of two buildings with a road r_1 connecting them. Agent a_1 is in b_1 and has previously observed the civilian (*civ*) in b_1, a_2 is in b_2 and does not observe any civilians. Figure 3.3 shows this scenario.

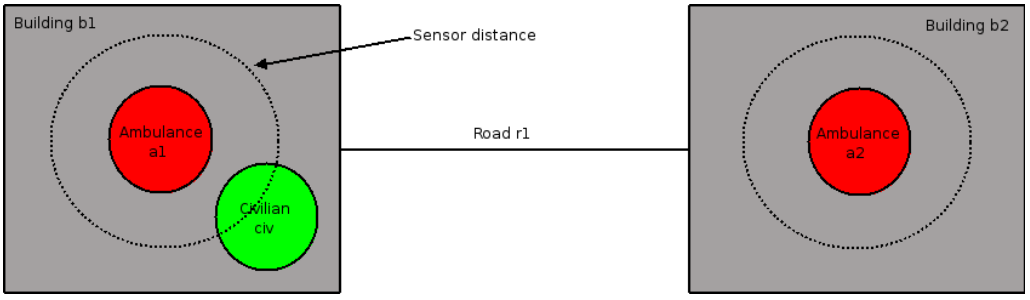


FIGURE 3.3: A simple rescue scenario

In the first example there are no communication restrictions, meaning the agents can send and receive a communication at any point on the map. This problem is chosen to show how communication may or may not be useful even when it is free. In the second example there is a blackout on the building b_1 and consequently agent a_1 must move to building b_2 if it wants to send a communication to agent a_2. This shows our mechanism using a costly communication resource. Our example will show how a change in the cost of communicating changes the agents behaviour.

3.2.2.1 Rational communication with no restrictions

We consider the action selection for agent a_1, shown in Figure 3.4, 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 3.2.

Considering the example in more detail, 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 a_1 and the second is taken by a_2 concurrently. We show the calculations for $J_1 = \langle rescue, rescue \rangle$ and $J_2 = \langle communicate, explore \rangle$, as these are the most illuminative. After a_1 implements its part of the joint action it will receive one or more observations according to the probabilities defined by Equation 2.7. In the following, the probability of observation ω is denoted by $p(\omega)$. In this example, this observation is always related to whether building b_1 contains a trapped civilian ($p(b_1 = civ) = 1.0$) or not ($p(b_1 = civ) = 0.0$). In our example, we do not need to utilise the full algorithm in Figure 3.1 because the problem is so small that future planning is not required. Instead we summarise the expected reward over observations and joint actions with the following equation:

$$e(J_i) = \gamma^t \sum_{\omega \in \Omega} p(\omega) R(J_i, \omega) \tag{3.9}$$

where we restrict the summation to only those observations ω which satisfy $p(\omega) > 0$ and t is the time of the action. This is valid because we are using a myopic example. Consequently, we must calculate R (by equation 3.3) for each joint action/observation pair. Calculating R requires values for R_p (by Definition 3.1) and R_c (by equation 3.4), which we will describe in more detail for J_2 and the observation $p(b_1 = civ) = 1.0$.

In this case, R_p uses the instantiation from Figure 3.2 which relates rewards to the number of civilians rescued and buildings explored (see equation 3.8 in Figure 3.2). Initially, one building has been explored and no civilians rescued, giving $R_p = (100 * 0 + 50 * 1) = 50$. Furthermore, $R_c = ND_{kl}(c, cb)$ from equation 3.4, where b is the initial belief state for a_1 which has no information about whether there are trapped civilians in each building — all states are equally

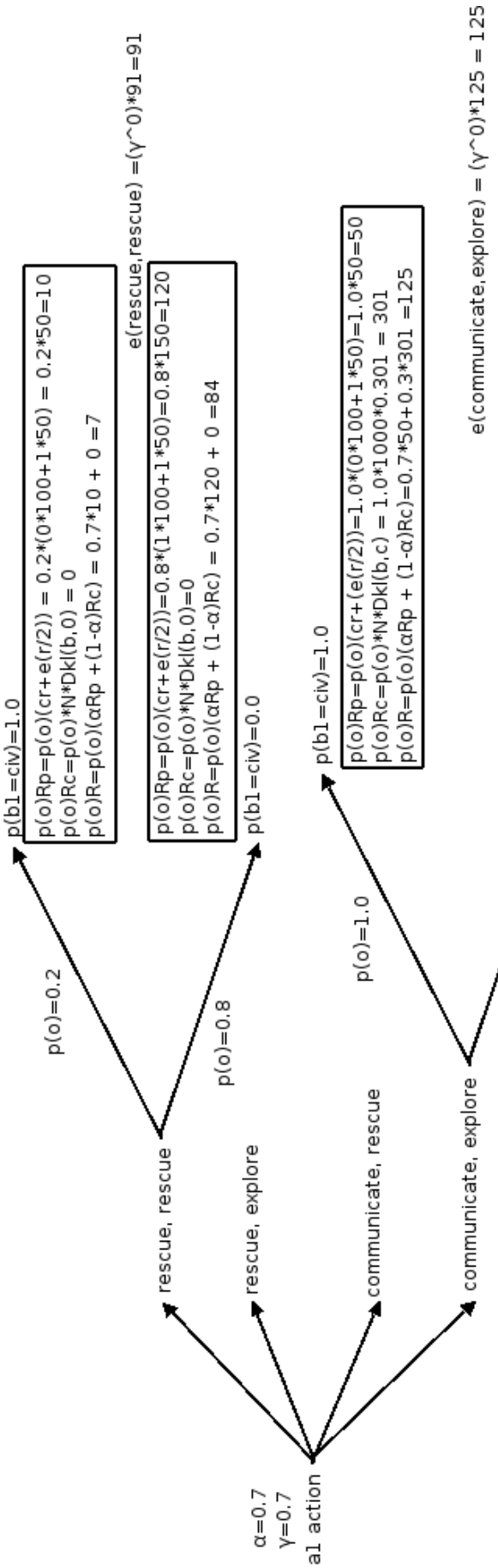


FIGURE 3.4: An execution example

likely. The communication c that we measure is the observation of a trapped civilian in b_1 ($p(b_1 = civ) = 1.0$) and in this example $N = 1000$. This communication is seen in Figure 3.5.

Thus the information gain (using KL Divergence) of that observation is $1000 * 0.301$ (which is scaled as denoted by equation 3.4). Since $R = \alpha R_p + (1-\alpha)R_c$ (see equation 3.3), and $\alpha = 0.7$ in this example, this gives $R = 0.7 * 50 + 0.3 * 301 = 125$. Using equation 3.9, $e(J_2) = 1.0 * 125 = 125$ and $e(J_1) = 91$ (see Figure 3.4 for the calculations for this joint action) which means that a_1 chooses to communicate c. In both these cases, the discount factor is γ^0 since the horizon is 1. These calculations show that agent a_1 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).

a1's belief state b =

	b1	b2
civilian here	0.5	0.5
no civilian here	0.5	0.5

c = p(b1=civ)=1.0

Rc = 1*log(1/0.5) + 0.5*log(0.5/0.5)
= 0.301

FIGURE 3.5: The value of communicating the existence of the civilian initially

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 $\langle p(b_1 = civ) = 0.5, p(b_2 = civ) = 0.5 \rangle$). As per equation 3.4, this would have resulted in 0 for the same b. In this case communication would not have been selected. Figure 3.6 demonstrates this.

a1's belief state b =

	b1	b2
civilian here	0.5	0.5
no civilian here	0.5	0.5

c = <>

Rc = 0.0

FIGURE 3.6: The value of communicating the null observation

Furthermore, this example also shows that on the next timestep, the value of communicating about the same civilian will have dropped and the agent will rescue instead. This is because

b will have changed to include knowledge of $p(b_1 = civ) = 1.0$ and $R_c = 0$. This is seen in Figure 3.7. These two examples show the rationality of our valuation — communicating zero

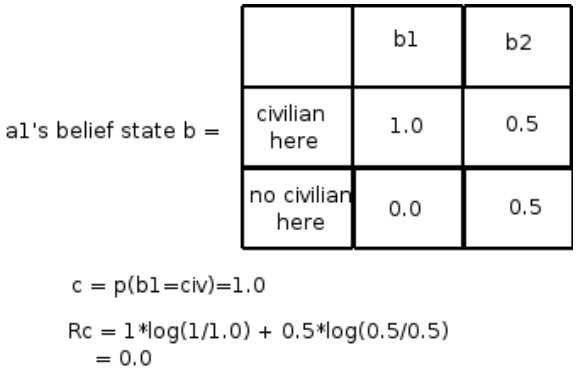


FIGURE 3.7: The value of communicating the same observation twice

information has no value and communicating previously communicated information also has zero value.

Finally, 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. This is because agent a1 would attempt to dig and fail, whilst a2 would have to explore to find the civilian — and not dig for at least 2 timesteps. Similarly, if α was greater then the agent would communicate too much and again the team would do less well. This extra communication would occur because a_2 would choose to communicate the existence of no civilian in b_2 and the value of communicating the same information would not drop as much. Consequently, 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.

3.2.2.2 Rational communication with restrictions

In this example the communication medium is restricted meaning that agent a_1 must move to building b_2 to communicate with agent a_2. In our example this journey takes 1 timestep and so there is a larger discount on that action compared to digging. This is shown in Figure 3.8.

In this case the benefit of communicating must be discounted by the time it takes to communicate (hence communication has a cost which does not have to be artificially defined but is captured automatically by the model). This is seen in the larger discount factor of the communication action (because it takes a timestep longer than other actions). Consequently the expected reward of communicating is now less than attempting to dig and so digging is chosen ($91 > 87.5$). Intuitively, it can be seen that in this example communication has little benefit since both agents will explore the map in the same length of time it takes to communicate.

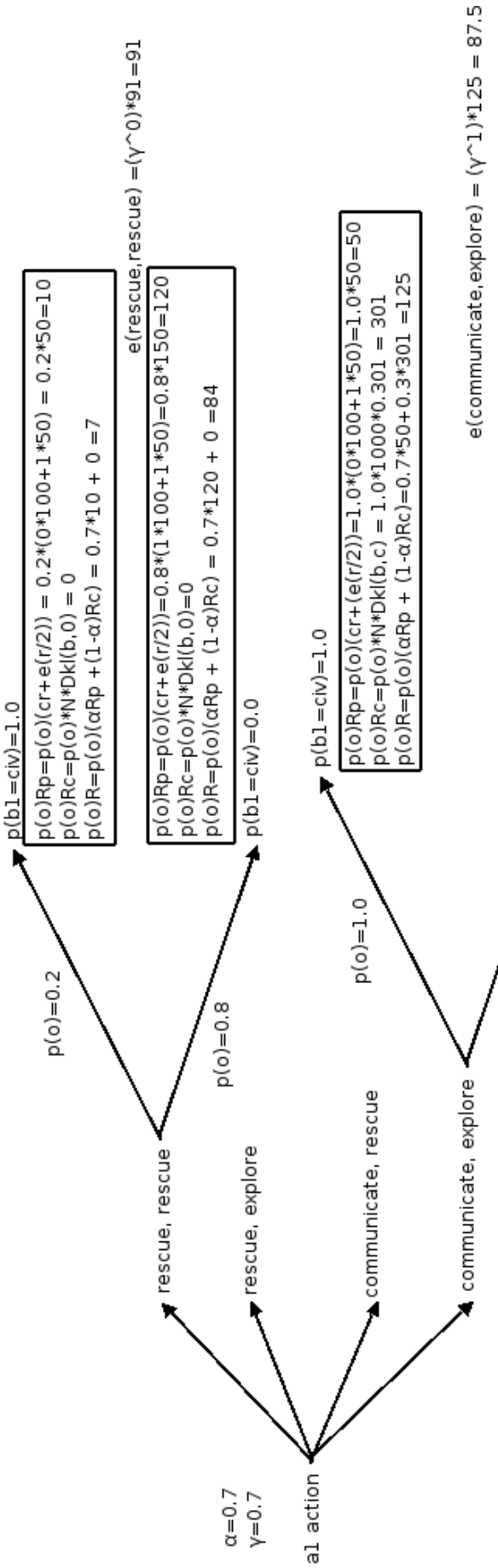


FIGURE 3.8: An execution example with communication restrictions

3.3 Empirical Evaluation

In this section we evaluate our model in the RoboCupRescue domain described in Chapter 2. Initially we specify the hypotheses we consider in this analysis. Following this, we describe the experimental methodology we use in these tests, and in fact throughout the thesis. Finally, we give the results of the experiments against these hypotheses.

3.3.1 Hypotheses

We propose a set of experiments that allow us to measure the influence of communication on the coordination problem. With this in mind, we then turn to the problem of balancing actions and communication restrictions in the problem to achieve coordination. In more detail, the following hypotheses are to be tested in this preliminary set of experiments:

> *Hypothesis* 3.1. Our information theoretic valuation mechanism for selective communication results in better coordination than models which block all communication, and those which assume a static value of communicating.
>
> *Hypothesis* 3.2. Our information theoretic valuation for selective communication is more effective in domains with a costly communication medium than models which assume a static value of communicating or only communicate when it is free.

The first hypothesis establishes the value of communication in multi–agent coordination; that is, the difference in the utility obtained between models which communicate and those which do not is the value of communication in the coordination task. This is key to solving both research challenge 1 and 2 from Chapter 1. Secondly, it establishes that our communication valuation is rational in this domain — that is, that for some value of α, communications lead to an improvement in performance over not communicating, and consequently approaches research challenge 3 (although later chapters will do this better). Finally, it shows that a dynamic evaluation of communication is better than assuming a static value to communicating.

The second hypothesis is designed to show the utility of our valuation mechanism in balancing the use of an increasingly expensive communication medium. It will show that a dynamic valuation based on information theory is better than managing communication using simple rules (i.e. communicate when free) or assuming a static value to communicating. Essentially, we demonstrate that it is better to plan communication actions based on their content than relying a rule-based approach commonly seen in the literature. This also tests our solution to research challenge 1 — namely that we can capture different sorts of communication cost generally.

3.3.2 Methodology

This section describes the experimental methodology employed in this set of experiments. Firstly, we describe the algorithms we experiment with — including some simple benchmarks and bounds. Next, we describe the control variables that we influence, and then we detail the dependent variables which we measure from the simulations. Finally, we describe the general methodology of the experiments, including how we achieve statistical significance.

3.3.2.1 Experimental Policies

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**). In more detail:

- **Zero**: the agents do not communicate with each other, and essentially solve the problem in isolation. This is equivalent to having each agent solve an individual POMDP with all other agents treated as part of the environment.
- **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 ω at timestep t. At timestep $t+1$, a_i chooses an action and communicates ω 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 centralised multi–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 $R_{c_{t+1}} = 0$. This valuation function requires some optimisation (we experimented with several increments) but it is not interesting to present that here since the method is ad–hoc. It is worthwhile to consider that our approach gives a finite space in which to optimise α, but assuming an incremental model like this essentially gives an infinite optimisation space. With this established, agent a_i has a history of observations since the last time it communicated $\overrightarrow{H\omega}$. At timestep t, a_i receives observation ω and appends this to its history $\overrightarrow{H\omega} \Rightarrow \overrightarrow{H\omega} : \omega$. At timestep $t+1$, a_i

chooses an action, including the option of communicating $\overrightarrow{H\omega}$. If communication is taken then $\overrightarrow{H\omega} \Rightarrow 0$ and $R_c = 0$.

- **Valued**: this is the *dec_POMDP_Valued_com* model as introduced in this chapter. The message passing semantics are the same as in **Selective**.

It is important to note that **Selective** and **Valued** can both choose to communicate whilst the agent is in a blackout area — in this case, the agent will move to the nearest point where communication is possible and then send the message. Similarly, **Full** will only communicate in parallel when the agent is outside the blackout areas, and will take no action to make communication possible — a standard rule-based response to communication restrictions.

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 agents' knowledge and possible communications. **Full** represents an upper bound on performance because the agents are solving a simpler centralised POMDP (and intuitively agents should do better when they know everything the other agent does). **Zero** represents a lower bound since each treats the other agents as part of the environment in a single agent POMDP (a notoriously inefficient approach for sophisticated coordination) and intuitively, coordination cannot be achieved without some idea of what the other agent knows.

3.3.2.2 Control Variables

The major control variables that remain static during these experiments are:

1. The number of ambulance agents who must act together n. We set $n = 2$ in our experiments in order to maintain tractable computation for the chosen test domain.
2. The number of goals to achieve in the problem G. Specifically, G is the number of civilians to save in any experiment. It is dependent on the size of the map considered in that experiment, and so for the size of the map described in Chapter 2, it would be about 30 (following the RoboCupRescue competition guidelines (Hiroaki, 2000)).
3. The time period of the simulation. A simulation time of 300 steps is standard in this test domain (Hiroaki, 2000).
4. The reward function for the problem R_p, as defined in equation 3.8 with in Figure 3.2. This is given by $R_p = (b \times r) + (e \times r/2)$ where r is a normalised reward (100), b is the number civilians rescued and e is the number of observed buildings
5. The reward function for communication actions R_c. The communication reward function R_c is the information content (which is measured using KL Divergence, see also Section 3.1) of the communication $\overrightarrow{H\omega}$. This is normalised to match the range of reward offered by R_p.

6. The cost of communication C_Σ. This is a function of the state of the communicating agent; that is, in some states (blackouts) communication is not possible and the agent must move to make it available. Hence, the cost is dependent on the location of the agent. There is no noise in the communication medium. The communication act itself requires a single timestep.

The following control variables are varied during the experiments:

1. The relative weighting between the reward functions for problem solving and communication α. In both hypotheses, we attempt to show the utility of valuing communications in policy computation. In more detail, this involves mixing two reward functions R_p and R_c. 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).
2. V, the percentage of the map where it is impossible to communicate — represents an increasing time cost to communicating because communication is unavailable in more of the map and so the agents must travel further to be able to communicate.

3.3.2.3 Dependent Variables

Dependent variables are determined by simulation runs. The interesting variable at each time step of the simulation is the percentage of total civilians saved at that timestep. This represents a measure of how well the agents solve the problem.

3.3.2.4 Initial Configuration and Statistical Significance

Each test run starts on the same map of Kobe (a real map of Kobe used in RoboCupRescue competitions) 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.

We run these simulations on standard desktop Linux system with a dual core 2.13GHz and 2GBit memory. This will be the case for all simulations in this thesis. The RoboCupRescue simulator is available from www.robocuprescue.org and Java source code for our basic algorithm is provided in Appendix A.

When considering **Full**, **Zero** and **Selective**, the dependent variable is graphed for each simulation timestep. When considering **Valued**, however, we have an additional control variable α. In order to visualise the results with respect to the values of α we give the mean summaries of the experimental variable at the end of the simulation. This takes the form of the dependent variable value 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. α is explored between 0 and 1 with increments of 0.1, interpolating in-between.

3.3.3 Results

In this section we present the experimental results. Firstly, the hypotheses are presented again, followed by the algorithms that will be evaluated and the variables that will be measured. Values for control variables will be detailed. After this, results are presented and a discussion given.

3.3.3.1 Hypothesis 3.1: No communication restrictions

> *Our information theoretic valuation mechanism for selective communication results in better coordination than models which block all communication, and those which assume a static value of communicating.*

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 3.9 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 the agents may not explore any new part of the map or encounter any new civilians.

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 3.10 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

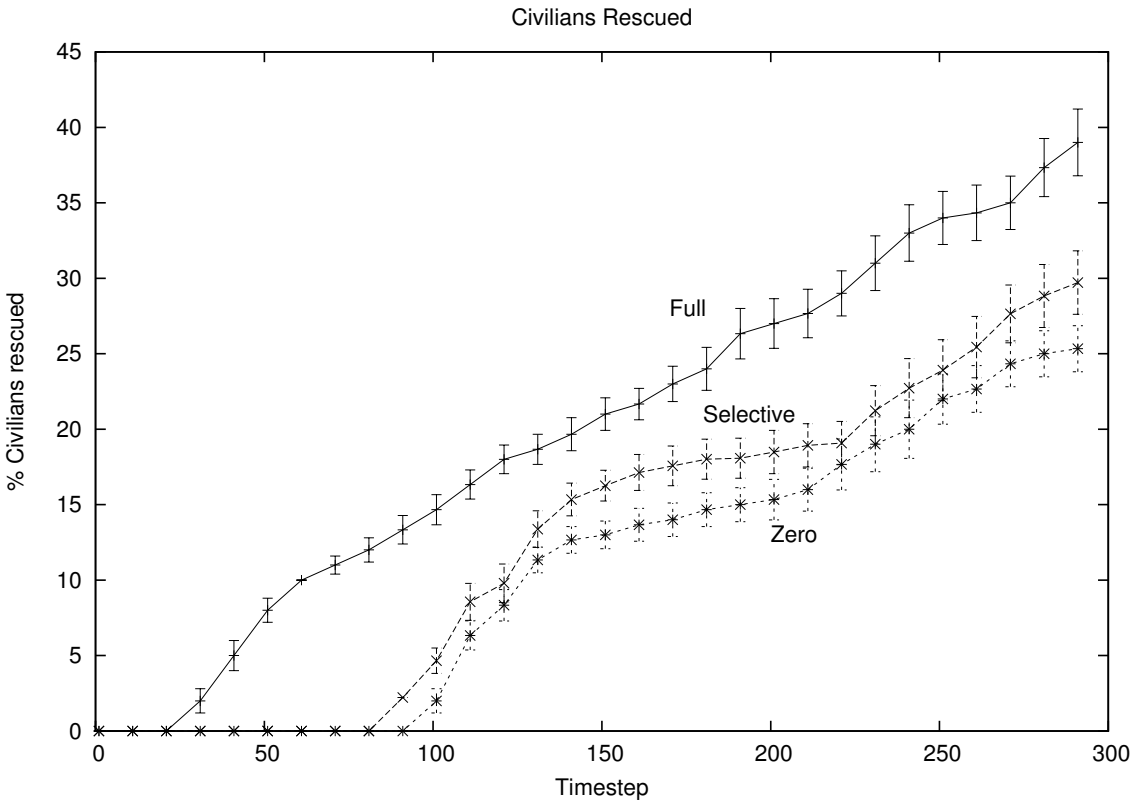


FIGURE 3.9: Percentage of civilians rescued during the simulation averaged over 30 runs

$\alpha = 1$, the agents never communicate, reducing it to **Zero**. It might seem that the lines should meet, but the agents are solving different sized POMDPs in both models, so we can accept some noise. 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 in reality — **Valued** measures the information gain before deciding whether to communicate.

To conclude, the evidence presented here supports hypothesis 1. Specifically, for some α values $0.6 < \alpha < 0.9$, performance is better than not communicating or using a static valuation.

3.3.3.2 Hypothesis 3.2: Communication restrictions

We now turn to the second hypothesis:

> *Our information theoretic valuation for selective communication is more effective in domains with a costly communication medium than models which assume a static value of communicating or only communicate when it is free.*

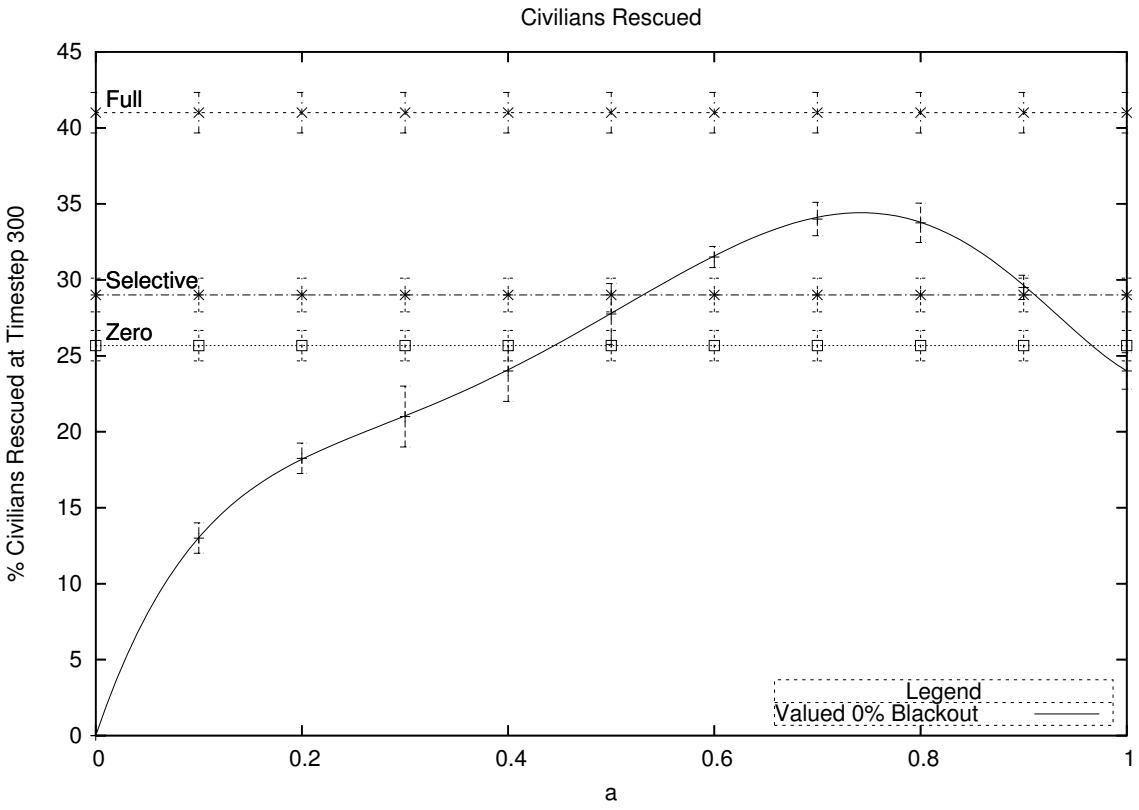


FIGURE 3.10: Percentage of civilians rescued at the end of the simulation averaged over 30 runs

With the utility of our method established in the simple case, we now consider the impact of communication restrictions. The communication restrictions we describe here more realistically model the sorts of communication conditions found in 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 Figures 3.11.

For blackouts ranging from 0-75%, the change in response to the α parameter is not significant because the nature of the communication restrictions means that there is not much change in the cost in this range (if the maps we considered were bigger then this would be different). 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** because the agents spend too much time travelling to communication points and too many communications are lost — so the value the communicating agent expects

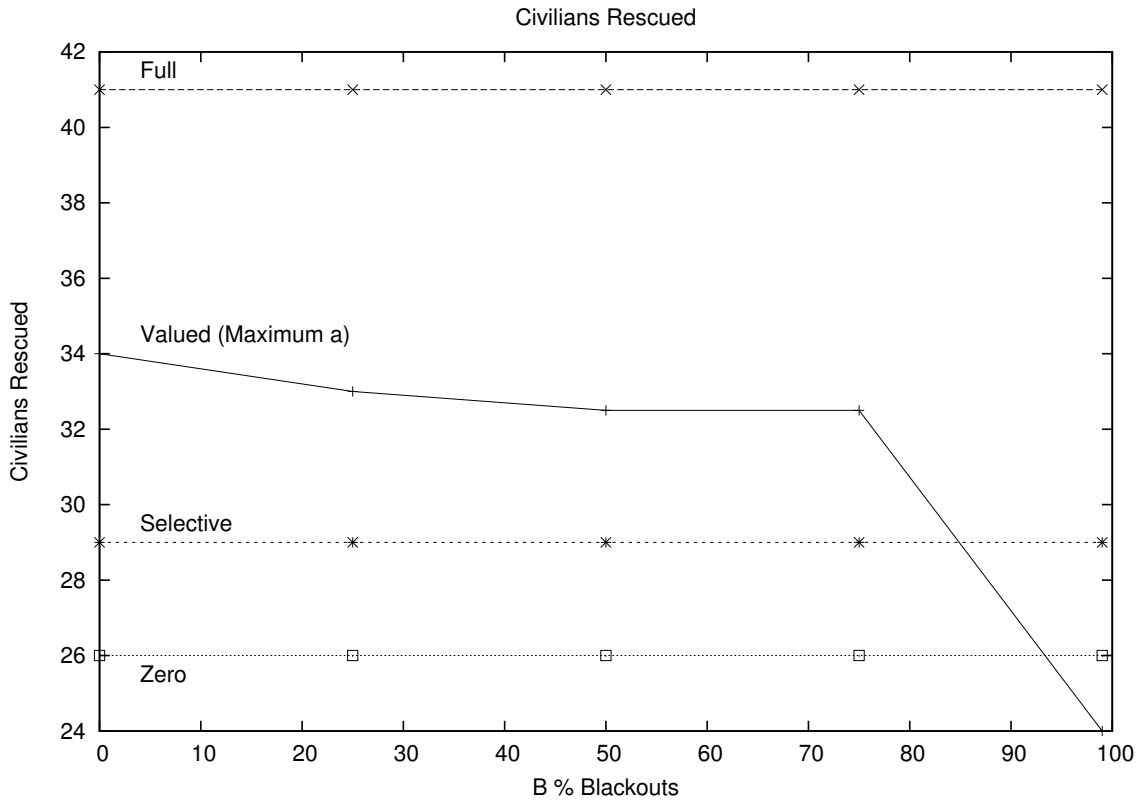


FIGURE 3.11: Percentage of civilians rescued at the end of the simulation averaged over 30 runs against blackout

to see is never attained. This suggests that when communication is very expensive, it is better to try to solve the problem in isolation. Figure 3.11 only compares **Valued** under communication restrictions, and so it useful to see how the simple communication strategies respond to the same restrictions.

Figure 3.12 shows the comparative performance of **Valued** versus **Selective** and **Full** as the communication restrictions grow. In more detail, we present the difference in civilians saved for **Valued** - **Selective** and **Valued** - **Full**. It can be seen that as the restrictions increase, **Valued** starts to do much much better than **Selective** because the latter does not accurately value communications and starts to do very poorly as communication becomes more expensive. **Valued** initially does worse than **Full**, but then increases to do better before arriving at a similar performance when the restrictions cover nearly all of the map. Remember that under zero restrictions, **Full** represents the optimal policy but this is highly dependent on communicating all the time — hence restrictions cause its performance to deteriorate much more quickly than **Valued**, and eventually it does worse because it never chooses to make communication possible.

To conclude, the evidence supports hypothesis 2. Specifically, for some $\alpha = 0.7$ value, performance is always better than the static valuation — and does not deteriorate as quickly. Furthermore, for high communication costs, our model is better than the **Full** policy (which is, in

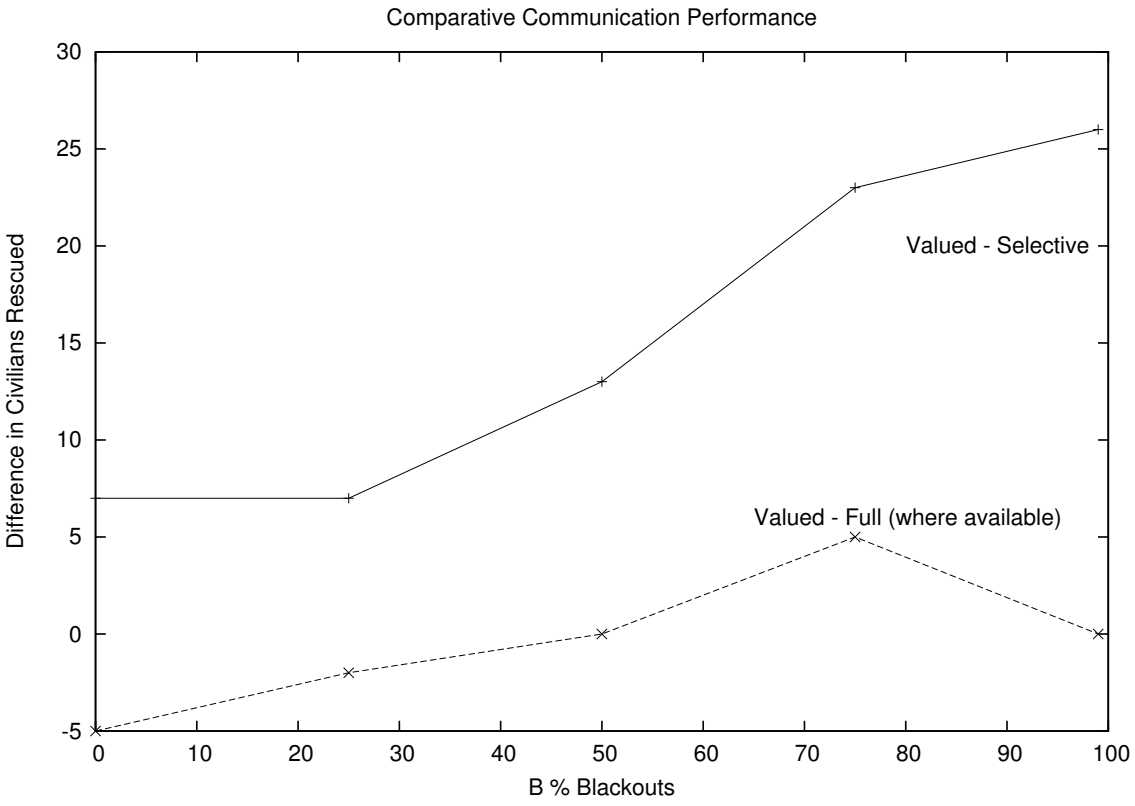


FIGURE 3.12: Difference in civilians rescued with communication restrictions

general, not possible to implement). Also, performance does not deteriorate as quickly.

3.4 Summary

This chapter introduced the *dec_POMDP_Valued_Com* model — a model of coordination using rational communication, and then detailed an action selection mechanism that can manipulate this model. Following this, we showed how the ambulance agent task could be represented as a *dec_POMDP_Valued_com* — which demonstrated how the communication valuation is used to communicate rationally. We also informally defined what it means to communicate rationally (in our example) and demonstrated that our model has these properties. Additionally, we demonstrated communication restrictions in our example and showed how our model can account for them to generate efficient solutions. As a result we have proposed a method which successfully solves research challenge 1 — capturing the cost of communication. Furthermore, in the solution to research challenge 2 that we have presented here we have partially solved research challenge 3. Future chapters will achieve this more successfully by developing techniques that give optimal solutions or bounded approximations.

In more detail, we analysed our model empirically and our show that valued communication leads to better policy computation in decentralised POMDP models than basic rule-based models. They also show that information theoretic valuations can be used to balance the cost of communicating, leading to performance which approaches a full communication model giving a partial solution to research challenge 3. This represents a promising empirical validation of the approach taken in this research. Specifically, these results indicate that communications can be valued by the sender and this valuation used to evaluate whether to send them. Furthermore, these experiments have shown that our model is capable of utilising a costly communication medium, with much less impact on performance than the simple polices considered here.

Now, in this model, we need to learn α offline. This has two drawbacks. First, only small scale problems can be tackled because the learning process involves running many simulations. Second, we do not know how close to optimal is the learned solution (research challenge 3). Because of this, we wish to develop an alternative model in which exact communication valuations can be defined without learning. This means larger problems can be solved and, furthermore, research challenge 3 can be tackled more readily. Specifically, the next chapter presents a general framework for valuing communications which can either solve much larger problems than the technique presented here or which can give a bounded approximation for larger problems and thus solves research challenge 3. Finally, we will also present a model that can give an optimal solution to this research challenge albeit for a constrained class of problems. That model uses a similar opportunity cost based approach to research challenge 1 as in this chapter.

Chapter 4

Reward Shaping for Valuing Coordination

The previous chapter showed that it is possible to use a principled approach to rational communication in order to coordinate a decentralised system. However, the technique used has two main problems. First, it requires tuning a parameter via an offline learning phase. Second, the solution generated lacks any information about how close it is to the optimal solution — the error may be unbounded. The first is a problem because it reduces the scalability of the approach and tuning may not be possible in practice. The latter is a problem because the solution learnt may be very poor relative to the optimal decentralised, and we would not know this is the case — which is the goal of research challenge 3.

To this end, this chapter introduces the *RS_dec_POMDP* model, an extension of the decentralised POMDP formalisms presented in Section 2.4.2, which utilises a novel reward shaping mechanism to compute decentralised policies using only local observations. In order to tackle research challenge 1, it follows the formalism from the last chapter to specify communication costs. In particular, communication has an opportunity cost like any other action. Thus, we first describe the intuition of reward shaping and detail how the model aligns with previous decentralised POMDPs. Then, we describe how belief divergence is measured and a reward shaping function might be derived for a given problem. In more detail, we will describe how approximate shaping functions can be used for a scalable approach that solves research challenge 2 or how we can trade-off some scalability for a technique with guaranteed error bounds in order to solve research challenge 3. This chapter presents the basic formalism and then we describe the approximate version further in Chapter 5, and a version of the algorithm for which we can formally bound the performance in Chapter 6. We end this chapter by describing how a reward shaping function is used with communication in the online policy generation algorithm from Section 3.1.2.

4.1 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, and hence, achieve research challenge 2. Now, the standard model of rational communication models the exact beliefs of other agents and analyses how communication would change their actions. However, as argued in Chapter 3, it is cheaper computationally to maintain an estimate of how coordinated the beliefs of the agents are and then use this to decide when to communicate. Unfortunately, the approach in Chapter 3 failed to identify the link between coordinated beliefs and the expected rewards for a joint action. Thus we address this issue here.

In more detail, two agents' beliefs are coordinated when they are identical. The greater the distance between those beliefs, the less coordinated they are. 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, 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. This was missing in Chapter 3 where the information value was scaled in line uniformly with other rewards without taking into account what it was the agent was trying to achieve with and without communication at the time of making a decision. By making this link we can avoid parameter tuning. This new model is much more natural with the value of communication related to what can be achieved after sending a message. Specifically, 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). It should be noted that we consider all parts of the belief space to be identical — although it would not be difficult to allow for the fact that differences about some parts of the belief space are more important than others.

Therefore, we first extend the *dec_POMDP_com* model by including communication actions into the standard action selection policy problem as in Chapter 3 to achieve research challenge 1. Furthermore, in a similar spirit to how information gain was calculated in Chapter 3, 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 4.2). We should make it clear that this is not the same as modelling the other agents' beliefs

— we are maintaining a difference parameter about the team as a whole which is much cheaper to calculate. Now, the main difference to the work in Chapter 3 is that we do not include a separate reward function for communications which is then weighted. Instead we assume communication itself has a zero positive value (but maybe a cost), but it changes the expected reward of other actions taken after it according to the reward shaping process we describe next. 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, as before, we restrict the communication alphabet to the observation alphabet.

Formally, *RS_dec_POMDP* is the tuple $RSDPM = \langle n, S, \mathcal{A}, P, \Omega, O, T, R, R_{rs}, \Sigma, C_\Sigma, \overrightarrow{L\omega}, B_d, \pi \rangle$ where the definitions are the same as those for the *dec_POMDP_com* (see Section 2.4.2) except:

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

$$R(s \in S, a \in \mathcal{A}, s' \in S) \in \mathbb{R} \tag{4.1}$$

  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 because, as we discuss in Chapters 1 and 3, it is more powerful to model communications as normal actions which have the same opportunity costs.

- $R_{rs} \in \mathbb{R}$ 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 the distance between the agent's individual beliefs. This allows us to transform the problem for all the benefits described earlier. This model is distinct to the weighted reward function used in Chapter 3.

- $B_d \in \mathbb{R}^+$ represents the agent's current estimation of the divergence in the beliefs of the agents. This is used in the reward shaping function to supply information about the current coordination of the agent team. Section 4.2 will explore this parameter further.

- π is a policy that relates joint actions (including communications) to belief states and belief divergences, $\pi : b(\overrightarrow{L\omega}) \times B_d \rightarrow \mathcal{A}$. This is the transformed problem.

Consequently, this model is distinct to the one proposed in Chapter 3 in that it adjusts the expected values of joint actions based on whether there is a given belief divergence in the team (which communication influences) rather than assigning a weighted value to information gain. This better because information gain is not uniformly important across the problem — it depends on what the agents are trying to achieve. 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. This is discussed next.

4.2 Expected Rewards using Belief Divergence

In this section we discuss how to measure belief divergence and how this is used in the general reward shaping function. In particular, we would like a principled metric that indicates the distance between two different belief states $b(\overrightarrow{L\omega})$ and $b(\overrightarrow{L\omega}')$ in a general fashion. Hence we need to consider:

- how to measure a distance in belief states, and
- how to estimate the difference in beliefs for a team of distributed agents (since our agents do not have full knowledge of the other agents' belief states).

Considering the first problem, since we are measuring the distance between probability distributions, it is appropriate to use information theory (as in Chapter 3). The actual measure is dependent on the problem, so simple domains might use an absolute difference ($B_d(\overrightarrow{L\omega}, \overrightarrow{L\omega}') = \sum_{s \in S} |Pr(s|\overrightarrow{L\omega}) - Pr(s|\overrightarrow{L\omega}')|$) in belief variables or relative entropy. In contrast, more complex belief spaces might use an aggregate measure like the KL Divergence measured used in Chapter 3:

$$B_d(\overrightarrow{L\omega}, \overrightarrow{L\omega}') = D_{KL}(\overrightarrow{L\omega}||\overrightarrow{L\omega}') = \sum_{s \in S} Pr(s|\overrightarrow{L\omega}) \cdot log \frac{Pr(s|\overrightarrow{L\omega})}{Pr(s|\overrightarrow{L\omega}')}$$

This is similar to the method used in Chapter 3 to measure the information gain of sending a message, the difference here is that we measure the difference between team and individual belief distributions in order to decide whether communication is needed.

In the second problem of estimating the belief state of a partially observed agent, 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. This is analogous to an agent assuming that it is the only entity that could have observed anything new since the last communication. To this end, we establish a reference point, $\overrightarrow{J\omega}^*$, which is the belief of the agent when it last synchronised its knowledge. We then compare the current belief state $\overrightarrow{L\omega}$ with this point. More formally, the approximate divergence AB_d is:

$$AB_d(\overrightarrow{L\omega}) = B_d(\overrightarrow{L\omega}, \overrightarrow{J\omega}^*) \tag{4.2}$$

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 initially 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. Nevertheless, in Chapter 6, under certain conditions, we will demonstrate how we can place bounds

on the error by considering the relationship between the observation function and the expected belief divergence. Furthermore, it may be possible to make the approximation more accurate using the observation function to obtain probabilities of features being commonly known, but, we leave this extension for future work (see Section 8.2) and concentrate on the basic assumption.

With the belief divergence measure established, we now consider the reward shaping function and where it fits into our model. As we introduced in the definition of *RS_dec_POMDP*, there is a reward shaping function which replaces the original reward function. We specify this function generally in terms of the expected rewards that are calculated for each action under reward shaping. In general B_d is normalised so that $0 < B_d < 1$, in which case the shaped expected reward for a joint action is given by the following function (which is problem-dependent):

$$R_{rs}(a, B_d) = f(B_d, E(a)_u, E(a)_r) \tag{4.3}$$

Here $E(a)_u$ and $E(a)_r$ represent the expected reward under the original reward function for a given action under certain conditions. In more detail, $E(a)_u$ is the reward when the agents are completely coordinated and $E(a)_r$ is the reward when they are mis-coordinated. The function f maps between these extremes using the parameter B_d. This is the function that is optimised by the policy generation algorithm in the next section.

Against this background, we can see the above model is a very general definition which allows considerable freedom in the specification of the reward shaping function for a given problem. In Chapter 5 we build on this and derive a specific heuristic shaping function which is approximate but scales very well. Furthermore, although the heuristic shaping function does not always perform optimally, in Chapter 6 we show how an error bounded shaping function can be derived which sacrifices some of the scalability of the heuristic version but has firm theoretical guarantees. However, in the rest of this section, we confine our discussion to the general specification and how it is used in policy generation (since it applies equally to both versions of the algorithm).

4.3 Communication within Policy Generation

We modify the online POMDP solution algorithm from the previous chapter to include the reward shaping algorithm. 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. As we stated earlier, we want to reduce the complexity of the problem by only considering local observations — the reward shaping transformation allows us to do this in a principled way. The expected reward for each action is calculated using Equation 4.3 and we assume we know f. Note that, 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 relatively low penalty (where

the average expected reward 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, an action is given by:

$$\pi(\overrightarrow{L\omega}, D) = argmax_{a \in \mathcal{A}} \sum_{\omega \in \Omega_i} P(\omega | \overrightarrow{L\omega}, a) \cdot \delta(\rho(\overrightarrow{L\omega}, a, \omega), D-1) \tag{4.4}$$

where $\delta(\overrightarrow{L\omega}, d)$ recursively calculates the payoff in the search tree and is defined by:

$$\delta(\overrightarrow{L\omega}, d) = \begin{cases} 0 & \text{, if } d = 0 \\ f(B_d, E(a)_u, E(a)_r) + \gamma \max_a \sum_{\omega \in \Omega_i} & \\ \quad [P(\omega | \overrightarrow{L\omega}, a) \cdot \delta(\rho(\overrightarrow{L\omega}, a, \omega), d-1)] & \text{, if } d > 0 \end{cases} \tag{4.5}$$

where γ is a discount factor and $\rho(\overrightarrow{L\omega}, a, \omega)$ gives a new belief state $b(\overrightarrow{L\omega'})$ when action a is performed in $b(\overrightarrow{L\omega})$ and ω is received. This is defined using Equation 3.2 for each state s. Incidentally, we must calculate a new belief divergence B'_d after each action:

$$B'_d = \begin{cases} 0 & \text{, if } a_i = \texttt{COM} \\ B_d \cup \omega & \text{, if } a_i \neq \texttt{COM} \end{cases} \tag{4.6}$$

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. This says that, if the agent communicates, its belief divergence is reset, (since we assume broadcast communication) else the observation that has been received must be integrated into the divergence estimate.

4.4 Summary

We have followed Chapter 3 in achieving research challenge 1 and, using this technique, we will show that our approach can handle problems as large as RoboCupRescue, yet it does not require extensive domain knowledge for solving decentralised POMDPs, nor do we need to learn anything offline as per the solution in Chapter 3. We can achieve 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). Now, in tackling research challenge 2, reward shaping in this framework can be approximate or exact. In the next chapter, we will discuss how an approximate reward shaping function can be used in our two exemplar domains — the Multi-Agent Tiger problem and RoboCupRescue. The approach presented here will be extended with the exact version of reward shaping in Chapter 6 in order to tackle research challenge 3 and give an error bounded version of this algorithm. In future work we will consider if this matches the theoretical value of communications from Chapter 2 — see Section 8.2).

Chapter 5

Reward Shaping with Heuristic Valuations

In this chapter we describe how to construct heuristic reward shaping functions for the formalism described in the previous chapter which scale very well. In particular, we first describe the general features of the heuristics — namely the limits of the shape function where behaviour is coordinated or mis-coordinated. Then, for the Multi–Agent Tiger problem, we construct a heuristic reward shaping function and present empirical results which show that heuristic reward shaping for coordination leads to significant improvements over the current state of the art. After this, for RoboCupRescue, again, we derive a heuristic shaping function and, finally, we present further empirical results showing the benefit of our approach. In this way we demonstrate a technique for addressing research challenge 2 (how to value communications) and show empirically how it addresses research challenge 3 (efficient global coordination).

5.1 Shaping Bounds

As explained in the last chapter, a heuristic reward shaping function parameterises (based on belief divergence) the expected reward between two extreme cases: the expected reward when the agents are fully coordinated, $E(a)_u$, and when the agents are completely mis-coordinated, $E(a)_r$. In this section we present the equations for these cases, which will subsequently be used to specify the full reward shaping function R_{rs}.

In more detail, each agent calculates the expected value for each joint action over its local belief state $b(\overrightarrow{L\omega})$ 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:

$$E(a)_u = \sum_{s \in S} \sum_{s' \in S} Pr(s|\overrightarrow{L\omega}) \cdot P(s,a,s') \cdot R(s,a,s') \tag{5.1}$$

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. In this case we assume that each of the joint actions by the other agents is equally likely, and consequently, an agent locally calculates the expected reward of a joint action assuming the other agents act randomly:

$$E(a)_r = \frac{1}{|\mathcal{A}_{-i}|} \sum_{a_{-i} \in \mathcal{A}_{-i}} \sum_{s \in S} \sum_{s' \in S} [Pr(s|\overrightarrow{L\omega}) \cdot P(s, \langle a_i, a_{-i} \rangle, s') \cdot R(s, \langle a_i, a_{-i} \rangle, s')] \tag{5.2}$$

where $\mathcal{A}_{-i}$ is the joint action space of all agents except i. The next sections give concrete examples of approximate reward shaping functions for our two exemplar domains.

5.2 The Multi–Agent Tiger Domain

This section demonstrates how our coordination mechanism from Chapter 4 can be used to facilitate agent teams in the Multi–Agent Tiger problem — a well known coordination problem described in Chapter 2 with theoretical and approximate solutions. After this, we describe our empirical evaluation and present the results.

5.2.1 Modelling as a *RS_dec_POMDP*

Basic aspects of the decentralised POMDP components of this model are already defined for this problem in Nair et al. (2003) and given in Chapter 2. Therefore we focus on the parts that are specific to *RS_dec_POMDP*.

Belief Space: 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 space is defined as $b_i = Pr(SL|\overrightarrow{L\omega})$, 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 4.2). More formally, $B_d(\overrightarrow{L\omega}, \overrightarrow{L\omega'}) = |Pr(SL|\overrightarrow{L\omega}) - Pr(SL|\overrightarrow{L\omega'})|$. We do not worry about direction since our reward shaping function will be insensitive to it.

Expected Rewards: Our heuristic shaping function uses belief divergence to select a shaped reward between two extremes. These extremes are when the agents are perfectly coordinated (belief divergence is zero) and when the agents are mis-coordinated (belief divergence is maximal). Consequently, we need to derive E_u and E_r for the joint actions available to the agents. These are defined using Equations 5.1 and 5.2.

TABLE 5.1: Expected Reward bounds

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

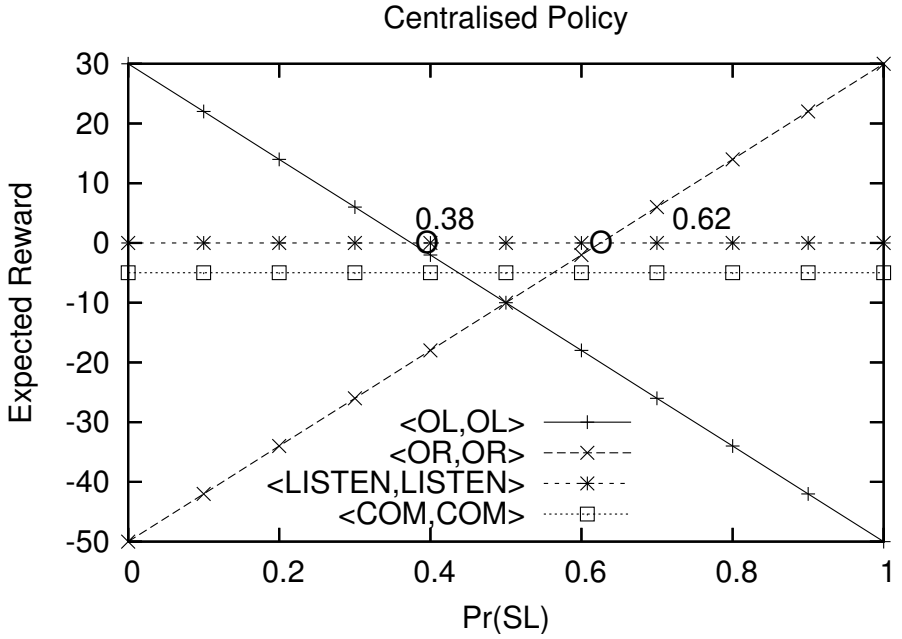


FIGURE 5.1: Centralised policy which assumes all agents have consistent beliefs

Reward Shaping Function: Now, we can construct f (from Equation 4.3) 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 uncoordinated expectation 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 multi–agent POMDP (which has a lower complexity class). For the Multi–Agent Tiger problem this policy can be represented by the alpha vectors for each action to a horizon of one (see Section 2.4.2), as shown in Figure 5.1. 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 that, if both agents have a belief $b_i \in [0, 0.38]$, then the best action for both of them is to open the left door. If both agents have a belief $b_i \in [0.62, 1.0]$ then the best action for both is to open the right door. Finally, for all other beliefs, the best action is to request an observation. Here,

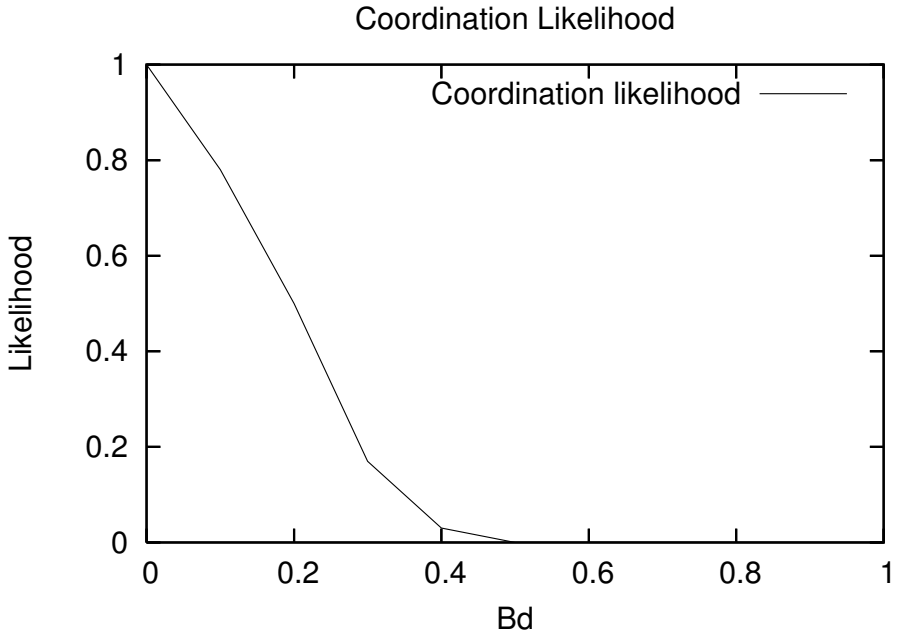


FIGURE 5.2: Probability of coordinating against belief divergence

it is interesting to note that, since the agents always assume that 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_i 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,

$$PC(B_d) = \sum_{i=1}^{n} max(d_i - d_{i-1} - B_d, 0) \tag{5.3}$$

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. Using an absolute belief divergence measure, this looks like Figure 5.2.

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:

$$R_{rs}(a, B_d) = f(B_d, E(a)_u, E(a)_r) = E(a)_r + PC(B_d) \cdot (E(a)_u - E(a)_r) \tag{5.4}$$

It is this equation that we maximise using the policy computation algorithm from Chapter 3. 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.

In the next sections we evaluate our model in the Multi–Agent Tiger domain. Initially we specify the hypotheses we consider in this analysis. Following this, we describe the experimental methodology we use in these tests. Finally, we give the results of the experiments against these hypotheses.

5.2.2 Hypotheses

We propose a set of experiments that allow us to measure the improvement over the state of the art as a result of using our mechanism. In more detail, the following hypotheses are to be tested in this set of experiments:

> *Hypothesis* 5.1. Our reward shaping mechanism is a more effective strategy for reducing the complexity of decentralised POMDPs than the current state of the art as well as several standard benchmarks, including models which communicate all the time or never.
>
> *Hypothesis* 5.2. Our reward shaping mechanism uses costly communication more efficiently than the current state of the art as well as several standard benchmarks, including models which communicate all the time or never.

The first hypothesis establishes the benefit of our approach to the problem of coordinating using a costly communication medium. It shows that research challenges 2 and 3 can be addressed by separating the full decentralised problem into several local agent problems in a principled manner that uses communication to manage the error in this transformation.

The second hypothesis is designed to show the utility of our valuation mechanism in balancing the use of an increasingly expensive communication medium. In achieving research challenge 2, we demonstrate that it better to plan communication actions based on their content, rather than relying on a rule-based approach commonly seen in the literature (as per Section 2.5.4).

5.2.3 Methodology

This section describes the methodology we employ in our experiments. Firstly, we describe the algorithms we experiment with — including some simple benchmarks and bounds. Next, we describe the control variables that we influence, and then we detail the dependent variables which we measure from the simulations. Finally, we describe the general methodology of the experiments, including how we achieve statistical significance.

5.2.3.1 Experimental Policies

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 (in a similar way to Chapter 3), and between these we analyse our mechanism (**RS_dec_POMDP**) for valuing communications and the state of the art benchmark solution (**ACE-PJB-Comm**) (see Chapter 2). In more detail:

- **Zero**: the agents do not communicate with each other, and essentially solve the problem in isolation. This is equivalent to having each agent solve an individual POMDP with all other agents treated as part of the environment.
- **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 ω at timestep t. At timestep $t+1$, a_i chooses an action and communicates ω 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 centralised multi–agent POMDP.
- **ACE-PJB-Comm**: the state of the art benchmark solution. Note that this model communicates in parallel like **Full**. As we describe in Chapter 2, this algorithm represents the current best solution for finding communication valuations in an online fashion. Results in Roth et al. (2005) show that it outperforms heuristic solutions (since no other approaches exist for our specific problem).
- **RS_dec_POMDP**: this is the our model as introduced in the previous chapter using the heuristic shaping function derived in the previous section.

In these models, the communication alphabets employed are identical to each other.

5.2.3.2 Control Variables

The major control variables that remain static during these experiments are:

1. The number of agents who must act together, n. we set $n = 2$ in order to compare with other work in the chosen test domain.
2. The time period of the simulation. A simulation time of 6 steps is standard in this test domain, and allows our results to be compared to other work in the Multi–Agent Tiger domain. This is because, after 6 timesteps, many instances of the problem will have reset.
3. The cost of communication C_Σ. In this case, the communication act itself requires a single timestep.

The following control variables are varied during the experiments:

1. The noise parameter of the observation function, w. We are interested in performance as the problem becomes increasingly difficult to observe directly. To this end, we vary w from exact observations ($w = 0$) to random observations ($w = 0.5$) with increments of 0.01.

5.2.3.3 Dependent Variables

The dependent variables are summarised at the end of each simulation run. The interesting variables at end of each simulation are:

1. The average reward obtained per timestep. We do not use total reward because we want to compare a model which treats communication as an action which takes time like an other action with a model which allows communication to happen in parallel.
2. The percentage fraction of the amount of communication that the **Full** model sends during a simulation run. This is interesting since it measures the actual volume of communication used by the algorithms.

5.2.3.4 Initial Configuration and Statistical Significance

Each simulation lasts for 6 timesteps, with the tiger placed randomly at the beginning (and after a door has been opened). The results show a mean of the dependent variables for each value of the noise parameter w. In general, 20000 runs are performed for statistical significance, which is computed using a standard t test for the 95% confidence interval (although we can reproduce the results with less runs we aim to match the experimental conditions used in Roth et al. (2005)).

5.2.4 Results

In this section we present the experimental results. We recap the hypotheses and then give the results for the experiment described previously.

5.2.4.1 Hypothesis 5.1: Multi–Agent Tiger performance

> *Our reward shaping mechanism is a more effective strategy for reducing the complexity of decentralised POMDPs than the current state of the art as well as several standard benchmarks, including models which communicate all the time or never.*

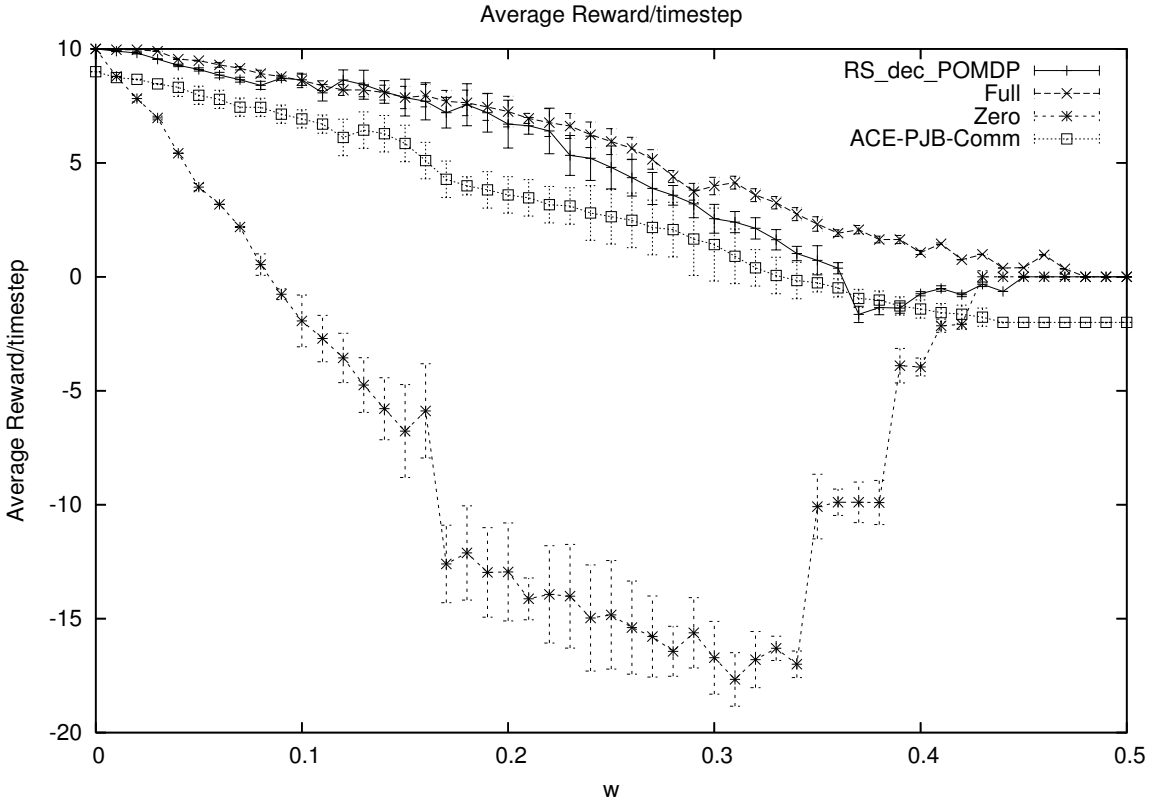


FIGURE 5.3: Performance of coordination models against noise, error bars are at 95% confidence intervals.

To this end, Figure 5.3 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 improvement is because there is an inherent weakness in trying to solve the problem without communication and then adding in communication later. Specifically, it does not allow for efficient exploitation of communication since it does not consider that communication might be useful yet costly. Furthermore, for $w > 0.35$ the performance of ACE-PJB-Comm drops significantly below zero, whilst our approach does not. This is because our model identifies that door opening may have potentially disastrous results when the agents might have the wrong coordinated impression (i.e. they both agree the tiger is in the wrong position), 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 values of 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.

To conclude, the evidence presented here supports hypothesis 1. Specifically, our model outperforms the state-of-the-art. Consequently, heuristic reward shaping is an efficient technique for

reducing the complexity of decentralised POMDPS.

5.2.4.2 Hypothesis 5.2: Multi–Agent Tiger communication

We now turn to the second hypothesis:

> *Our reward shaping mechanism uses costly communication more efficiently than the current state of the art as well as several standard benchmarks, including models which communicate all the time or never.*

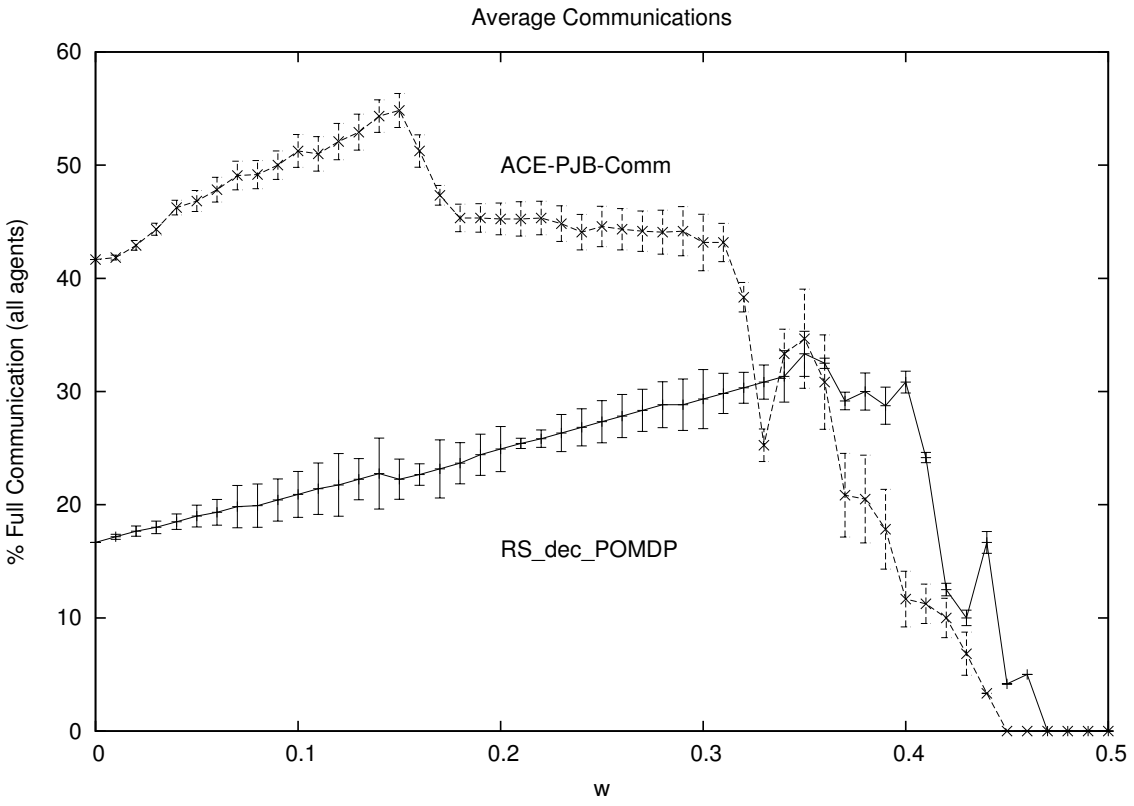


FIGURE 5.4: Communication in coordination models against noise, error bars are at 95% confidence intervals.

To address this hypothesis, Figure 5.4 compares the number of messages the team of agents send during the simulation. We present the percentage of the messages sent by *Full* since this is constant and naturally *Zero* always sends zero messages. As can be seen, for $w < 0.35$, our model sends fewer messages than ACE-PJB-Comm — 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 5.4 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 results 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 per ACE-PJB-Comm) means that the policy does not consider the *cost* of communicating, and consequently, exploits it inefficiently.

To conclude, the evidence presented here supports hypothesis 2. Specifically, our model sends less communications. Consequently, our model uses communication more rationally.

5.3 The RoboCupRescue Domain

Here we describe our method in 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. This problem has a different cost for communication, and so it represents our approach to research challenge 1. Furthermore, it shows how our method of solving research challenge 2 gives a partial solution to research challenge 3 in large problems. After this we present an empirical analysis of its performance.

5.3.1 Modelling as a *RS_dec_POMDP*

Most of the elements of the *RS_dec_POMDP* model the same as the *dec_POMDP_Valued_Com* from Chapter 3 and so we refer the reader to Section 3.2 for more details. Here we will describe the features unique to our new model.

Belief Space: 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 4.2). 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:

$$B_d(\overrightarrow{L\omega_i}, \overrightarrow{J\omega}^*) = D_{KL}(\overrightarrow{L\omega_i} || \overrightarrow{J\omega}^*) \tag{5.5}$$

Expected Rewards: $E_{u,r}$ can be trivially calculated using Equations 5.1 and 5.2.

Reward Shaping Function: In the Tiger problem we were able to define an exact reward shaping function (Equation 4.3), but this is not possible here since we need a probability of coordination and that involves solving the centralised POMDP (which as stated previously has PSPACE complexity). However we can estimate the function as a linear function of the belief divergence measure giving:

$$PC(B_d) = \frac{B_{dM} - B_d}{B_{dM}} \tag{5.6}$$

where B_{dM} is the maximum KL Divergence in this belief space. We choose this function to demonstrate how very little domain knowledge is needed to make the mechanism function at a reasonable level. More complex relationships could be defined, but as we will see, this one allows our mechanism to perform better than simple communication strategies such as always communicating new information or never communicating at all.

Given this, R_{rs} is:

$$R_{rs}(a, B_d) = f(B_d, E(a)_u, E(a)_r) = E(a)_r + PC(B_d) \cdot (E(a)_u - E(a)_r) \tag{5.7}$$

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 [1].

In the next sections we evaluate our model in the RoboCupRescue domain. Initially we specify the hypotheses we consider in this analysis. Following this, we describe the experimental methodology we use in these tests. Finally, we give the results of the experiments against these hypotheses.

5.3.2 Hypotheses

In Chapter 3 we showed that a rational communication method of coordinating works better than standard approaches in a problem of this size. However, that approach involved a costly offline learning phase and so, in Chapter 4, we specified a more principled method which did not require this. We will specify a hypothesis to test that the heuristic approach presented in this chapter can outperform the learning technique when α is set randomly. In more detail, the following hypothesis is to be tested in this set of experiments:

[1] It is clear this is not the case in all domains. For example, if there is only one civilian to find then that is the only feature of the belief space that we care about so differences elsewhere are unimportant.

> *Hypothesis* 5.3. Our reward shaping valuation mechanism for selective communication results in better performance than our previous model *dec_POMDP_Valued_Com* with a randomly set communication valuation α.

This hypothesis establishes that our method is, straight away, better than a model which requires a learning stage to do well. Furthermore, we aim to show that it does better than standard benchmarks — these are the only techniques available to us as a comparison model because of the size of the problem.

5.3.3 Methodology

This section describes the experimental methodology we employ in this set of experiments. Firstly, we describe the algorithms we experiment with — including some simple benchmarks and bounds. Next, we describe the control variables that we influence, and then we detail the dependent variables which we measure from the simulations. Finally, we describe the general methodology of the experiments, including how we achieve statistical significance.

5.3.3.1 Experimental Policies

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 as in Chapter 3, and between these we analyse our mechanism (**RS_dec_POMDP**) for valuing communications and our previous benchmark solution from Chapter 3 (**dec_POMDP_Valued_Com**). In more detail:

- **Zero**: as in Section 5.2.3.1.
- **Full**: as in Section 5.2.3.1.
- **dec_POMDP_Valued_Com**: this is the model from Chapter 3 with α set randomly.
- **RS_dec_POMDP**: this is the our model as introduced in the previous chapter using the heuristic shaping function derived in the previous section.

Note that, unlike in the Multi–Agent Tiger problem, there is no optimal solution for this problem, as the decentralised POMDP cannot be solved exactly by current techniques due to its extreme size (Bernstein et al., 2000).

To summarise, communication is completely free in **Full** — hence it is used all the time; the agents never communicate in **Zero**; **dec_POMDP_Valued_Com** and **RS_dec_POMDP** both use different communication valuation mechanisms. We benchmark against a randomly set α in

these results in order to negate the impact of tuning in that algorithm — our new approach requires no such tuning. **Full** represents an upper bound on performance because the agents are solving a simpler centralised POMDP (and intuitively agents should do better when they know everything the other agent does). **Zero** represents a lower bound since each treats the other agents as part of the environment in a single agent POMDP (a notoriously inefficient approach for sophisticated coordination) and intuitively, coordination cannot be achieved without some idea of what the other agent knows.

5.3.3.2 Control Variables

The major control variables that remain static during these experiments are defined in Section 3.3.2.2 although our new approach could employ larger teams [2].

5.3.3.3 Dependent Variables

Dependent variables are determined by simulation runs. The interesting variables at each time step of the simulation are:

1. The percentage of total civilians saved at that timestep. This represents a measure of how well the agents solve the problem.
2. The number of communications sent at the end of the simulation.

5.3.3.4 Initial Configuration and Statistical Significance

This setup is the same as in Section 3.3.2.4.

5.3.4 Results

In this section we present the experimental results. We repeat the hypothesis and then give the results and analysis.

5.3.4.1 Hypothesis 5.3: RoboCupRescue performance and communication

> *Our reward shaping valuation mechanism for selective communication results in better performance than our previous model* dec_POMDP_Valued_Com *with a randomly set communication valuation* α.

TABLE 5.2: 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)

The results in Table 5.2 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*.

To conclude, the evidence presented here supports the hypothesis in this section. Specifically, our new technique does well without a learning stage and better than simple benchmarks. Consequently, our new method is inherently better at rationalising the use of communication than the method presented in Chapter 3 and so should be viewed in preference to it in all cases.

5.4 Summary

These results show that reward shaping is a viable technique for reducing the complexity of policy computation and valuing communications in decentralised POMDPs. We have also demonstrated how, whilst maintaining the solution to research challenge 1 (specifying the cost of communication) from Chapter 3 we have improved the scalability of our solution to research challenge 2 (the value of communication). Specifically, this new work has employed heuristic reward shaping functions to separate a decentralised POMDP into individual agent POMDPs that allow the agents to coordinate better than the state of the art.

However, so far we have not discussed whether optimising with respect to a reward shaping function leads to an optimal policy in the original problem. In this context, it can be seen that the reward shaping functions used so far are heuristic (in that they are guided by the underlying problem) and do not allow for any guarantees about solution quality. Given this, the next chapter will discuss how the general formalisation in Chapter 4 can accept exact reward shaping

[2]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 standard desktop machine.

functions to provide such guarantees. However this comes at the expense of some scalability. Consequently, our powerful architecture can allow for both highly scalable heuristic solutions, as well as solutions with theoretical guarantees on smaller problems which solves research challenge 3 (optimal solutions).

Chapter 6

Reward Shaping for a Bounded Approximation

As seen in the previous chapter, reward shaping defined over belief divergence is an effective technique for reducing the complexity of decentralised POMDPs and providing more accurate online valuations of communications than is currently possible. However, as has been noted, it is an approximate approach due to both the definition of the shaping function and the estimation of the belief divergence. Given this, we would like to augment the model so that performance guarantees can be derived given some knowledge of the problem (the reward and observation function) and the computational power available to the agents (expressed as the planning horizon). This enables our approach to tackle research challenge 3 (global performance) successfully — which the models in Chapters 3 and 5 could not. To do so, however, we need to compute the exact reward shaping function in order to remove a source of error. However, we will see later that this computation is more demanding than using the heuristic techniques from the previous chapter, and consequently, we do sacrifice some scalability for these theoretical guarantees. Nevertheless, it should be clear that in smaller problems a bounded approach is more appropriate, but in larger domains a heuristic approach is necessary.

Against this background, in this chapter we present a general definition of the exact reward shaping function. This means that optimising with respect to this function is guaranteed to generate the same optimal policy as if the original decentralised POMDP were to be solved — the same could not be said of the previous heuristics in Chapter 5. In particular, we demonstrate how this function can always be specified if the reward function is known and there is an exact value for the belief divergence in the team. We then show how the observation function can be used to characterise the error in belief divergence using the reward shaping function. After this, we take the belief divergence error and show how it creates an error in the reward shaping transformation from the original decentralised POMDP to individual POMDPs. This is achieved by first specifying what would be the error if an optimal policy for the reward shaped POMDP was already provided and then, using this, what would be the error in our online policy generation

algorithm from Chapters 3 and 4. Finally, we provide empirical results showing that this exact method performs as well as the heuristic used in the previous section in the Multi–Agent Tiger problem, and explore the factors influencing the error bound in that problem.

6.1 Exact Reward Shaping

We desire an exact reward shaping function, defined over a measure of belief divergence and local beliefs, that would lead to the same policy as the original reward function defined over joint beliefs. That is to say that an agent, at the same point in the problem, should choose the same joint action using the reward shaped POMDP (parameterised by local beliefs and belief divergence) as if it was choosing a joint action in the original decentralised POMDP (parameterised by joint beliefs). One way to make sure that this is the case is to consider what would happen if the agents maximise their actions with respect to expected rewards — the same action in a given belief state should be selected in both the original problem and the new shaped problem so that the same policy is generated. Moreover, the expected rewards in each case should be the same (otherwise an error is introduced where the two models may select different actions in the same state). By making this equivalence, we can specify the general form of the reward shaping function.

If we consider the relationship between evaluating the expectation for an action based on local beliefs (of each agent) and based on the joint (fused) belief then we can make this equivalence. Specifically, the expected reward for an action $a \in A$ with a local belief $\overrightarrow{L\omega}$, which is a vector of observations $\omega \in \Omega$, is:

$$E(a, \overrightarrow{L\omega}) = \sum_{s \in S} \sum_{s' \in S} R(s,a,s') \cdot P(s,a,s') \cdot Pr(s|\overrightarrow{L\omega}) \tag{6.1}$$

where S is the set of states, $R(a,s,s')$ is the reward for performing action a in state s and arriving in state s', $P(s,a,s')$ is the probability of moving from s to s' given action a and $Pr(s|\overrightarrow{L\omega})$ is the probability of being in state s given the belief state $b(\overrightarrow{L\omega})$:

$$Pr(s|\overrightarrow{L\omega}) = \frac{Pr(s) \cdot \prod_i Pr(\overrightarrow{L\omega}_i|s)}{\sum_{s' \in S} Pr(s') \cdot \prod_i Pr(\overrightarrow{L\omega}_i|s')} \tag{6.2}$$

where $Pr(s)$ is the prior belief of being in state s and $Pr(\overrightarrow{L\omega}_i|s)$ is the likelihood of receiving observation $\overrightarrow{L\omega}_i$. Now, a joint belief is the fused observations of all members of the agent team, which, for two agents with local beliefs $\overrightarrow{L\omega}^1$ and $\overrightarrow{L\omega}^2$, we denote by the transpose concatenation as follows:

$$\overrightarrow{J\omega} = \left[\overrightarrow{L\omega}^1 : \overrightarrow{L\omega}^2\right]^T \tag{6.3}$$

where agent 1 has local belief state $b(\overrightarrow{L\omega}^1)$ and agent 2 has local belief state $b(\overrightarrow{L\omega}^2)$. Consequently, the expected reward for action a and joint belief state $b(\overrightarrow{J\omega})$ has the same form:

$$E(a,\overrightarrow{J\omega}) = \sum_{s\in S}\sum_{s'\in S} R(s,a,s')\cdot P(s,a,s')\cdot Pr(s|\overrightarrow{J\omega}) \tag{6.4}$$

When calculating the expected reward for an action a it is appropriate to use the joint belief state $b(\overrightarrow{J\omega})$. However, in distributed teams, an agent i may not know the other agent's local beliefs, so if it was to calculate the expectation based only on its local belief state $b(\overrightarrow{L\omega}^i)$ then an error would be introduced. This error is defined as the difference between the expectation for the joint belief and the local belief states:

$$\begin{aligned} Err(a,\overrightarrow{J\omega},\overrightarrow{L\omega}^i) &= E(a,\overrightarrow{J\omega}) - E(a,\overrightarrow{L\omega}^i) \\ &= \sum_{s\in S}\sum_{s'\in S} R(s,a,s')\cdot P(s,a,s')\cdot \left[Pr(s|\overrightarrow{J\omega}) - Pr(s|\overrightarrow{L\omega}^i)\right] \end{aligned} \tag{6.5}$$

Consequently, the error between the expectations is expressed as a function of the belief divergence between the local belief and the joint belief states, and the state s:

$$B_d(s,\overrightarrow{J\omega},\overrightarrow{L\omega}^i) = Pr(s|\overrightarrow{J\omega}) - Pr(s|\overrightarrow{L\omega}^i) \tag{6.6}$$

We now have all the components to derive a general reward shaping function which will allow us to later place theoretical guarantees on the quality of the solutions using it. Specifically, a reward shaping function assumes that if the belief divergence is 0 then the local beliefs are the same as the joint beliefs (because a communication has just happened), but as it increases the expected reward should diverge from the value based on local beliefs towards the actual value based on joint beliefs. We can use this definition of belief divergence to express this function exactly and generally in the case when an agent knows the belief divergence:

$$\begin{aligned} E_{sh}(a,\overrightarrow{L\omega}^i,\overrightarrow{J\omega}) &= f(B_d,E(a)_u,E(a)_r) \\ &= E(a,\overrightarrow{L\omega}^i) - |Err(a,\overrightarrow{J\omega},\overrightarrow{L\omega}^i)| \\ &= E(a,\overrightarrow{L\omega}^i) - |\sum_{s\in S}\sum_{s'\in S} R(s,a,s')\cdot P(s,a,s')\cdot B_d(s,\overrightarrow{J\omega},\overrightarrow{L\omega}^i)| \end{aligned} \tag{6.7}$$

This is the general form of the reward shaping function — which trivially reduces to the expected reward for the joint belief states in the case when this is fully known, but generally that is not the case. The next section will derive the properties of our approximation of belief divergence (described in Chapter 4) in a distributed team. After this we will see how this approximation leads to a bounded error on solution quality using our formalism.

6.2 Belief Divergence Error

As we have seen in the previous section, if we know $B_d(s, \overrightarrow{J\omega}, \overrightarrow{L\omega}^i)$ exactly then we can compute the exact reward shaping function. However, this requires that each agent in the team knows $\overrightarrow{J\omega}$ which is the same as knowing all other agents' local beliefs. This is clearly impractical in domains where communication is not free (such domains would make this reasoning unnecessary since agents would not need to communicate).

Thus we analyse a specific mechanism for inferring belief divergence. In more detail, as per Chapter 4, whenever communication occurs, we set the reference point $b(\overrightarrow{J\omega}^*)$ to the synchronised joint belief state. Then, we use the distance between an agent i's current local belief state $b(\overrightarrow{L\omega}^i)$ and $b(\overrightarrow{J\omega}^*)$ to approximate the belief divergence. We could use any other mechanism to approximate the belief divergence, such as referring to a predefined distribution over belief divergence which is parameterised by the time since communication and the observations that have been received, or even assuming a constant belief divergence (see Section 8.2). However, this particular mechanism is used here because it is intuitive to analyse and furthermore it avoids domain knowledge — making it easy to apply in other problems with little modification.

Note that, as discussed earlier, this approximation introduces two types of error:

- other agents may experience observations and take actions which that reasoning agent does not, and
- the reasoning agent may experience observations and take actions which the other agents do not.

We can analyse this error by examining the distance that the approximation moves from the *expected joint belief* of the team in both cases. In more detail, this is calculated by considering the dynamics of the problem to weight each possible history of actions and observations that each team member experiences. This concept is used in optimal (but intractable) solutions to the decentralised POMDP problem (see Section 2.4.2 for more details). Now, using the expected joint belief we can arrive at an estimation of what the true belief divergence is at any given point. It is accurate in the expected case, rather than at that particular point in time, which is fine for our error analysis because we want the expected worst case error in the performance of the algorithm since this will bound the maximum loss of utility from the optimal solution. Consequently, the error generated by assuming the belief divergence according to our approximation can be specified as the distance from our approximation to the expected belief divergence.

In this context, for the analysis, we assume that agents do not take actions and instead only receive observations. To include actions as well, we could use a pre-computed policy to give a distribution of actions that are taken in a given state (similar to the observation function). However, the specification of such a policy is non-trivial and so, in most cases, would be sub-optimal and, thus, introduces a new source of error. Because of this, we believe it is more

instructive to limit the analysis to observations. Given this, a history $\overrightarrow{J\omega}$ is defined as a sequence of observations $\omega \in \Omega$. Furthermore, the set of all possible histories $J\Omega^T$ contains all possible combinations of observations of length T.

We now calculate the expected error introduced by our approach. To do so, we use the underlying characteristics of the decentralised POMDP to derive the belief divergence in expectation (by calculating the expected joint belief) and then use the distance from this as the error in our approach. With this in mind, the expected joint belief, EJB, given the local belief state $b(\overrightarrow{L\omega})$ and the current state s is:

$$EJB(\overrightarrow{L\omega}, s) = \sum_{\overrightarrow{J\omega} \in J\Omega^T} Pr(\overrightarrow{J\omega}|\overrightarrow{L\omega}) \cdot Pr(s|\overrightarrow{J\omega}) \tag{6.8}$$

where $Pr(\overrightarrow{J\omega}|\overrightarrow{L\omega})$ is the likelihood of the joint history given the local history and is defined as:

$$Pr(\overrightarrow{J\omega}|\overrightarrow{L\omega}) = \sum_{s \in S} Pr(s|\overrightarrow{L\omega}) \cdot Pr(\overrightarrow{J\omega}|s) \tag{6.9}$$

We then use the expected joint belief to specify the true belief divergence in expectation, EB_d, which is given by:

$$EB_d(\overrightarrow{L\omega}, s) = \sum_{\overrightarrow{J\omega} \in J\Omega^T} \left[Pr(\overrightarrow{J\omega}|\overrightarrow{L\omega}) \cdot Pr(s|\overrightarrow{J\omega}) \right] - Pr(s|\overrightarrow{L\omega}) \tag{6.10}$$

Furthermore, the approximate belief divergence using our mechanism is given by:

$$AB_d(\overrightarrow{L\omega}, \overrightarrow{J\omega}^*, s) = \left[Pr(s|\overrightarrow{J\omega}^*) - Pr(s|\overrightarrow{L\omega}) \right] \tag{6.11}$$

We now have our definition of the approximate belief divergence and an expectation of the real belief divergence — the distance between these two values is the error (in expectation) on our approximate belief divergence. Thus, the error introduced in the approximation can be expressed for a specific horizon k as:

$$Err(AB_d, EB_d, k) = \sum_{s \in S} \left[\sum_{\overrightarrow{J\omega} \in J\Omega^k} \left| \left[Pr(\overrightarrow{J\omega}|\overrightarrow{L\omega}) \cdot Pr(s|\overrightarrow{J\omega}) \right] - Pr(s|\overrightarrow{J\omega}^*) \right| \right] \tag{6.12}$$

where we sum over all states to give an accurate measure for the entire belief space. Whilst this is the absolute error for a specific horizon, we will see later that it is also useful to have a definition of how the error grows with time and so the error as the horizon is extended from by one timestep is:

$$\delta^T_{T+1} = \sum_{s \in S} \left[\sum_{\overrightarrow{J\omega} \in J\Omega^T} \left[\sum_{\overrightarrow{J\omega}^1 \in J\Omega^1} \left| \left[Pr(\left[\overrightarrow{J\omega} : \overrightarrow{J\omega}^1\right] | \left[\overrightarrow{L\omega} : \overrightarrow{J\omega}^1_1\right]) \cdot Pr(s|\left[\overrightarrow{J\omega} : \overrightarrow{J\omega}^1\right]) \right] - \right. \right. \right. \tag{6.13}$$

$$\left. \left. \left. \sum_{\overrightarrow{J\omega} \in J\Omega^T} \left[Pr(\overrightarrow{J\omega}|\overrightarrow{L\omega}) \cdot Pr(s|\overrightarrow{J\omega}) \right] \right| \right] \right]$$

To summarise, this section has described the error in the estimation in belief divergence using our assumption that the other agent's beliefs do not progress after the communication point. This is a vital step in bounding the solution error in using reward shaping with this belief divergence assumption. Therefore the next section will describe how this causes a bounded error in solving the reward shaped POMDP (and consequently the underlying decentralised POMDP). We express this error first in terms of an optimal offline solution and then, using this, in terms of an online solution method. We will see later how this expansion relates the error in solution quality to the time since communication last occurred. This will allow us to bound the error not only to the solution horizon of the decentralised POMDP, but also to the frequency with which communication occurs. As a consequence, we can solve research challenge 3 (efficient global coordination) by presenting an algorithm which gives a bounded approximation of the global optimal solution.

6.3 Policy Generation Error using Reward Shaping

To understand the features of a problem which make a particular approximation appropriate or not, we need to characterise the error bound in our algorithm. Thus, in this section we place bounds on the loss of optimality if an agent team employs our reward shaping transformation and method of estimating the belief divergence in the team. Initially, we assume that we are provided with a solution to the belief divergence POMDP described by a set of α vectors over the belief space (as per Section 2.4.2). In this context, the α vector which is maximal at a particular belief point b is an action policy which should be followed — the action to take. Then, we can construct a bound on the error in following this policy using the estimated belief divergence. The derivation of this bound on reward proceeds in a similar fashion to Point-Based Value Iteration in single-agent POMDPs (see Section 3.1.2 for more details) because those algorithms must also contend with not knowing the exact belief point.

In more detail, the error in solution quality depends on the size of the error given by the belief divergence estimate. Recall from the previous section that this is defined as the distance from the expected belief divergence. As above, we define the error in the belief divergence as $\varepsilon_{BD}(k) = Err(AB_d, EB_d, k)$ for a specific horizon k. In the following section, it will be useful to think about this error as a distance in the belief divergence space BD — the distance between the estimate of the belief divergence, b, and the furthest point the actual belief divergence could be, b' (specified by $Err(AB_d, EB_d, k)$), so $\varepsilon_{BD}(k) = ||b - b'||_1$. Now, in order to bound the error on the reward, we need to think about how an incorrect estimation of the belief divergence would lead to the wrong action being selected. With this in mind, $\varepsilon_{BD}(k) = ||b - b'||_1$ represents the region where we think the belief divergence lies. It is the possible actions that may be taken in this region that we should concern ourselves with — especially if they lead to a worse outcome than the correct action at the true belief divergence. By way of illustration consider the example in Figure 6.1. In particular, imagine a different α vector is optimal at the extreme of the error region (marked as b' in the diagram). Here, α is the vector which is maximal at b, the true belief

divergence (so in Figure 6.1 this means that for beliefs in the darker shaded region the action corresponding to α should be selected), and α' is maximal at b' (in the lighter shaded region). By choosing α rather than α', the error is at most $\alpha'.b' - \alpha.b'$ (this is the distance between the heights of the lines at the selection points), and also note that $\alpha'.b \leq \alpha.b$ (we will use this fact in the proof). This is demonstrated in Figure 6.1 where it can be seen that if the agent's estimation of the belief divergence was equal to the true value b then it should select the action described by the policy vector α. Thus, due to the error in the belief divergence, in the worst case, it may select the α' vector maximal at b' mistakenly. As a result, we can bound the error that may result from this mistake.

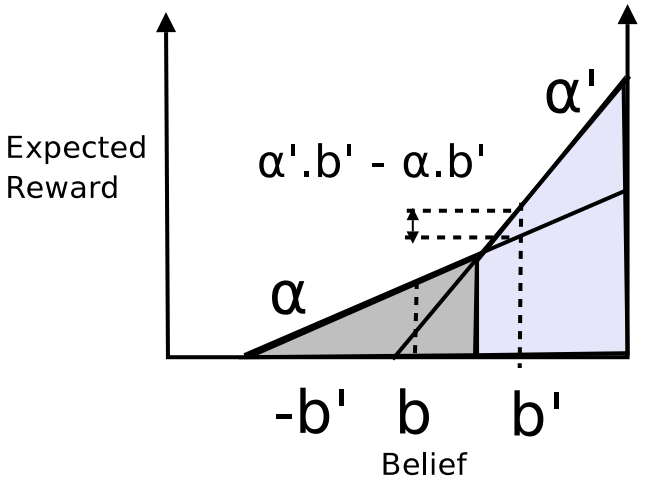


FIGURE 6.1: The maximum error possible using reward shaping

More formally:

Theorem 6.1. The error introduced by an estimate of the belief divergence when using an optimal policy is at most $\frac{R_{max}-R_{min}}{1-\gamma} \cdot \varepsilon_{BD}(k)$

Let R_{max} be the maximum possible reward available for a joint action, R_{min} be the minimum possible reward and γ the discount factor in the problem. Then:

Proof.

$$\begin{aligned}
err(k) &\leq \alpha'.b' - \alpha.b' \\
&= \alpha'.b' - \alpha.b' + (\alpha'.b - \alpha'.b) \\
&\leq \alpha'.b' - \alpha.b' + \alpha.b - \alpha'.b \\
&= (\alpha' - \alpha) \cdot (b' - b) \\
&\leq ||\alpha' - \alpha||_\infty \cdot ||b' - b||_1 \\
&\leq ||\alpha' - \alpha||_\infty \cdot \varepsilon_{BD}(k) \\
&\leq \frac{R_{max} - R_{min}}{1-\gamma} \cdot \varepsilon_{BD}(k)
\end{aligned}$$

□

From the theorem, it can be seen that the worst possible error between selecting the wrong α vector, and following it infinitely is $(R_{max} - R_{min})/(1-\gamma)$ (here we assume that the chosen α vector causes the worst possible action to be taken at each belief point).

So far we have only considered the error caused by the belief divergence estimate. This assumes, however, that the policy generation is computed optimally. Now, as we have discussed in previous chapters, we would like to generate solutions to these models in an online fashion as it is intractable to compute optimal solutions for large POMDPs (and decentralised ones) even offline. As a consequence, this error bound is not yet complete because there is an additional error introduced by the policy computation stage. Given this, we will describe how we can use this result in an online policy generation algorithm such as we have already detailed.

To this end, we assume that agents construct the optimal α vector for their current belief point b by using a k-lookahead search of possible joint actions and local observations (as per Chapter 4). At the fringes of this search tree, an approximate value function V^a is used to value those belief states. Consequently, in single-agent planning the error of this approach is bounded by $\gamma^k||V^a - V^*||$ (Puterman, 1994). However, as the previous section shows, there is a further error in V^a. More formally:

Theorem 6.2. The error using an online policy is at most $\gamma^k \cdot \frac{R_{max}-R_{min}}{1-\gamma} \cdot \varepsilon_{BD}(k)$

Proof.

$$\begin{aligned} err(k) &\leq \gamma^k||V^a - V^*|| \\ &\leq \gamma^k \cdot \frac{R_{max} - R_{min}}{1-\gamma} \cdot \varepsilon_{BD}(k) \end{aligned}$$

□

Consequently, the exact reward shaping function in Equation 6.7 has an error in estimating the belief divergence captured by Equation 6.12. Now, this error is used in Theorem 6.1 to bound the error whilst using Equation 6.7 for an existing policy. Finally, we show how this error is bound by Theorem 6.2 in the case when an online policy computation algorithm is used.

To conclude, we have shown that there exists a general reward shaping function defined over belief divergences which transforms a belief state decentralised POMDP into individual POMDPs with a bounded error. Furthermore, this transformation leads to a natural definition of belief divergence and using a simple and general technique for estimating distributed belief divergence we can bound the solution error when using both an existing optimal policy and an online policy generation algorithm to solve the reward shaped POMDP. This represents a unique solution, both in using reward shaping to aid coordination and communication valuations and in bounding approximate techniques in decentralised POMDP solutions. As a result, in this chapter, we have

proposed a method which builds upon the success of Chapter 3 in solving research challenge 1 (the cost of communication), along with the method of Chapter 4 and 5 in tackling research challenge 2 (the value of communication), in order to solve research challenge 3 (global coordination) for a general class of problems.

With this established, in the next section we will analyse how this error bound changes with the planning horizon k and other parameters in the Multi–Agent Tiger problem. This is a useful activity because there is a trade-off in the error function — namely, the error on the fringe is discounted by the planning horizon, but the maximum possible error in belief divergence increases. Consequently, by inspecting this function for a given problem we can find the planning horizon which minimises the error and, by doing so, gain an insight into the sorts of problems where our approximation is useful.

6.4 Error Bound Analysis for the Multi–Agent Tiger Problem

Earlier in the chapter we defined a new reward shaping function for use in our formalism. Building on this, in this section, we evaluate how well this modified model does against our previous heuristic algorithm from Chapter 5. To do so, we concentrate on the Multi–Agent Tiger problem since it is small enough to compute the actual error bounds. Now, we are interested in an empirical analysis of the performance in this setting since the theoretical bounds of the last section are for the worst case and, in practice, the performance may be significantly better. Also, since the other techniques do not have error bounds, we should compare under similar conditions (average reward over a number of simulations).

In addition to comparing the average performance and communications of the two approaches, we also consider what are the features of the domain which cause the error bound to change and in fact be minimised. This analysis is useful because the problem is small enough that we can fully analyse the relationship between a planning horizon of k and the observation function which are the two parameters that affect the error bound (see Theorem 6.2).

6.4.1 Empirical Methodology

Most of our experimental methodology remains the same as in Section 5.2.3, and so here we merely present the differences.

6.4.1.1 Hypotheses

We propose a set of experiments that allow us to measure the improvement over the state of the art as a result of using our bounded mechanism. In more detail, the following hypotheses are to be tested in this set of experiments:

Hypothesis 6.1. Our bounded reward shaping mechanism performs as well as the heuristic version.

Hypothesis 6.2. Our bounded reward shaping mechanism uses costly communication as efficiently as the heuristic version.

Our particular aim here is to show that the bounded version of the algorithm performs as well as the heuristic version (which is already known to outperform the state-of-the-art and standard benchmarks). It is important to remember that this bounded version of the algorithm is not as scalable as the heuristic version and they are therefore appropriate for different domains. In this way we improve upon our previous algorithm by offering the ability to bound the solution error.

6.4.1.2 Experimental Policies

Our experimental setup is the same as in Section 5.2.3, however, here we only to compare our previous heuristic reward shaping function with our new bounded reward shaping function (since the performance of the other benchmarks will remain unchanged).

- **RS_dec_POMDP**: this is the our heuristic model as introduced in the Chapter 4 using the heuristic shaping function derived in the previous chapter.
- **Bounded_RS_dec_POMDP**: this is the our bounded model as introduced in the Chapter 4 using the exact shaping function derived in Equation 6.7.

In our results we will present the same dependent variables, against the same control variables. However, we will present the difference between the algorithms. Specifically, we present the performance of **RS_Dec_POMDP** subtracted from that of **Bounded_RS_dec_POMDP**

6.4.2 Results

In this section we will present the results of the simulations against the hypotheses.

6.4.2.1 Hypothesis 6.1: Bounded Multi–Agent Tiger performance

Our bounded reward shaping mechanism performs as well as the heuristic version.

We can see in Figure 6.2 that the results suggest that the bounded version of the algorithm outperforms the previous heuristic version by an average of 9% across the observation noise $w < 0.1$ and $0.2 < w < 0.32$. For $0.32 < w < 0.35$ it performs slightly worse (14%) because the heuristic

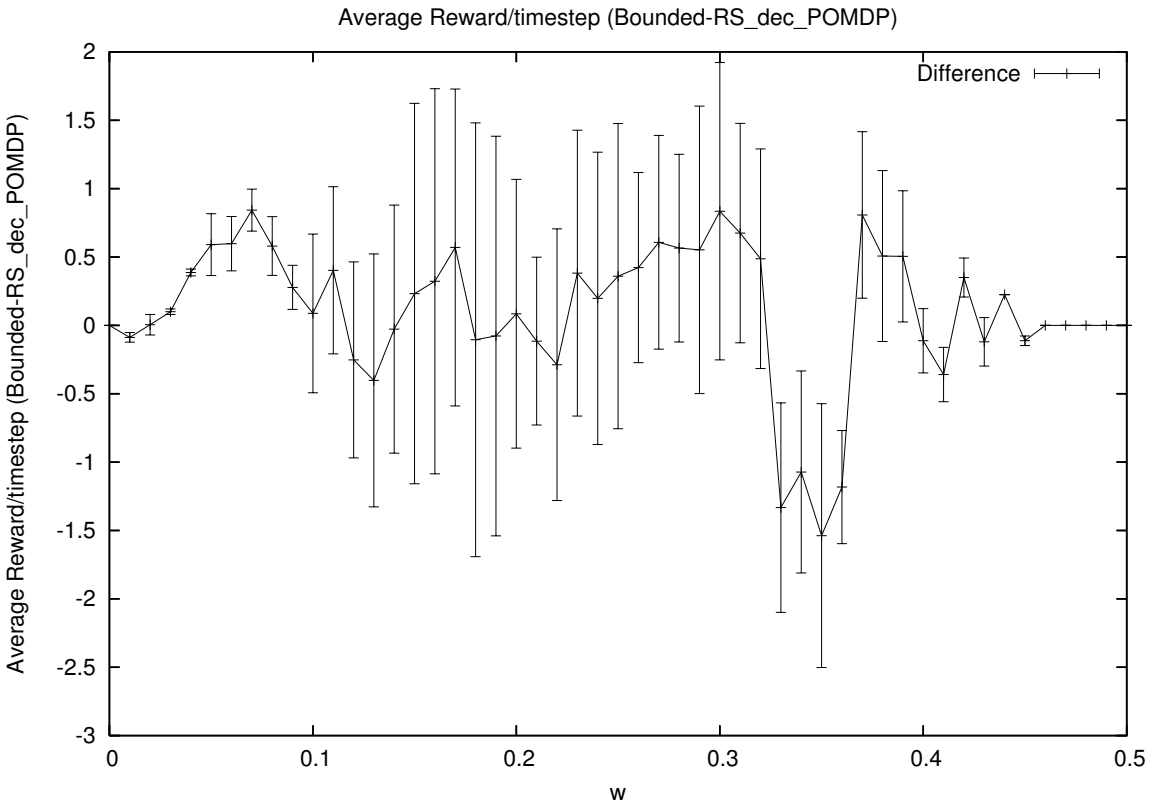


FIGURE 6.2: (Bounded—RS_Dec_POMDP) Performance of coordination models against noise, error bars are at 95% confidence intervals.

is conservative with observations which contain noisy but useful information. The rest of the time it performs the same (within the tolerance of the confidence intervals). The performance improvement should be expected since the heuristic was designed to approximate the exact shaping function we have defined in the last section — so they should have similar performance, but the heuristic makes an error in some places because it sometimes overestimates the penalty for mis-coordination, and so is more risk averse (however this does lead to a slight improvement in some cases). It should be noted that the middle of the parameter space $0.2 < w < 0.4$ is the hardest to coordinate in since communications do contain useful information but may be wrong. It is here that an exact approach does the best and yields the most benefit verses the heuristic since it can more accurately use the properties of the observation function to value receiving more observations against communicating to guarantee coordination. In contrast, the heuristic ignores the state of the observation function. Finally, both algorithms improve towards the end as they start to ignore observations and instead do nothing — although our new algorithm recognises that observations are too noisy to use earlier and improves first. However, the important point to note is that the new algorithm achieves this performance with a bounded error — something which cannot be said for the last algorithm (and different heuristics would not perform as well).

Consequently, the results suggest that we have confirmed the hypothesis that we do not lose much performance by introducing a bounded version of the algorithm compared to the heuristic.

6.4.2.2 Hypothesis 6.2: Bounded Multi–Agent Tiger communication

Our bounded reward shaping mechanism uses costly communication as efficiently as the heuristic version.

Here, an added effect is that communication is employed more efficiently, because it is valued more accurately. This is seen in Figure 6.3 where we plot the difference between the algorithms. Here, communication is generally lower in the bounded algorithm. This is the reason for the performance improvement since less time is spent communicating and more on opening correct doors. Our new method is particularly effective in the region $0.2 < w < 0.4$ where communication is used to verify noisy but useful observations. Just before this point, our new method sends more messages to try to extract value from very noisy observations. Finally, both algorithms send less messages at the end because the observations are too noisy to be useful and indeed communicate to each other.

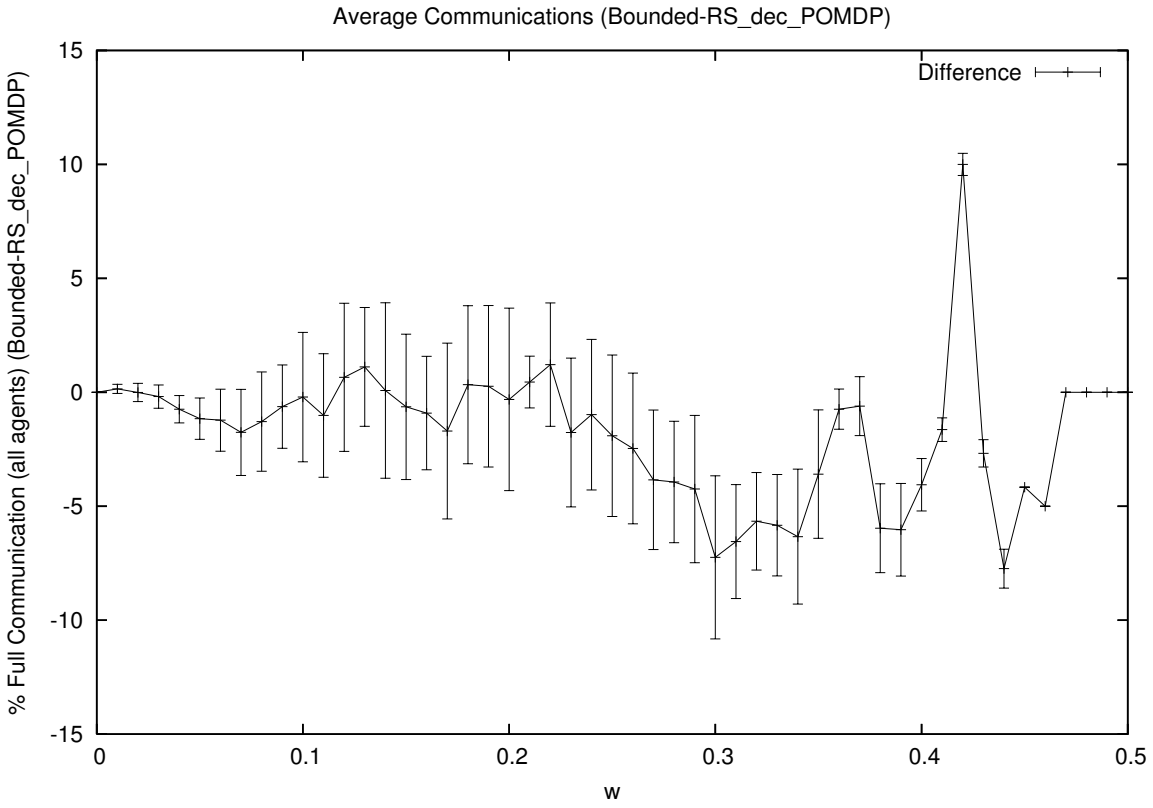


FIGURE 6.3: (Bounded—RS_Dec_POMDP) Communication in coordination models against noise, error bars are at 95% confidence intervals.

Consequently, we have demonstrated that an accurate shaping function can be defined which performs better than using domain intuition to make the function. And thus we verify the hypothesis.

Following these results, we would like to understand what is the dominating component of the error bound in this problem (the error bound in Theorem 6.2 tells us it is either the planning horizon k, the discount factor γ or the observation function). This information will allow us to better tune the approximation for the best performance and also understand which problems it would be inappropriate for — in this particular bound, the planning horizon is the only component we can alter and so we will be able to get the best value for a given problem. And so, in the next section, we will analyse how the error bound is influenced by the parameters of the formalisation.

6.4.3 Dominating Component of the Error Bound

As can be seen from the definition of the error bound (Theorem 6.2), there are several parameters that influence the size of the maximum error. Specifically, the planning horizon (k) discounts the error at the fringe (so larger horizons should reduce the maximum error). However, a larger planning horizon intuitively also causes the possible error in belief divergence to increase (as seen in Equation 6.13, where each timestep causes the set of possible joint beliefs in the team to increase). Thus, these two forces should cause a trade-off in the most desirable planning horizon for a given problem. Furthermore, the error in the observation function (w) should also influence the maximum possible error in the belief divergence. This is because an observation which distributes similar observations to all agents regardless of state will result in a smaller belief divergence error than one which gives very different observations to each agent.

In order to investigate this trade-off in the context of the Multi–Agent Tiger problem, we plot the error bound for $k = 1...11$ (the computation becomes too difficult for horizons > 11), against the same observation function used in previous experiments. It should be noted that, whereas previous experiments reported the average reward per timestep, this bound is calculated based on the infinite discounted reward. This is because of the $(R_{max} - R_{min})/(1-\gamma)$ term which is the penalty for taking the worst possible action at each timestep into the future. In bounds of this variety, we can only safely assume this worst case performance — since we cannot say at what point the correct actions might be taken.

As Figure 6.4 shows, the error does indeed decrease as the planning horizon is increased, for all observation noise w. Furthermore, for the horizons it is possible to compute, the influence of the planning horizon on the belief divergence error is not as important as the discounted fringe error. We can see this by the fact that, as k is increased, the maximum error decreases. However, we can safely assume that this does not carry on indefinitely — eventually we would have to see a trade-off between the planning horizon and the belief divergence error (it is just that we do not reach this point with the computational resources available to us — as the horizon is

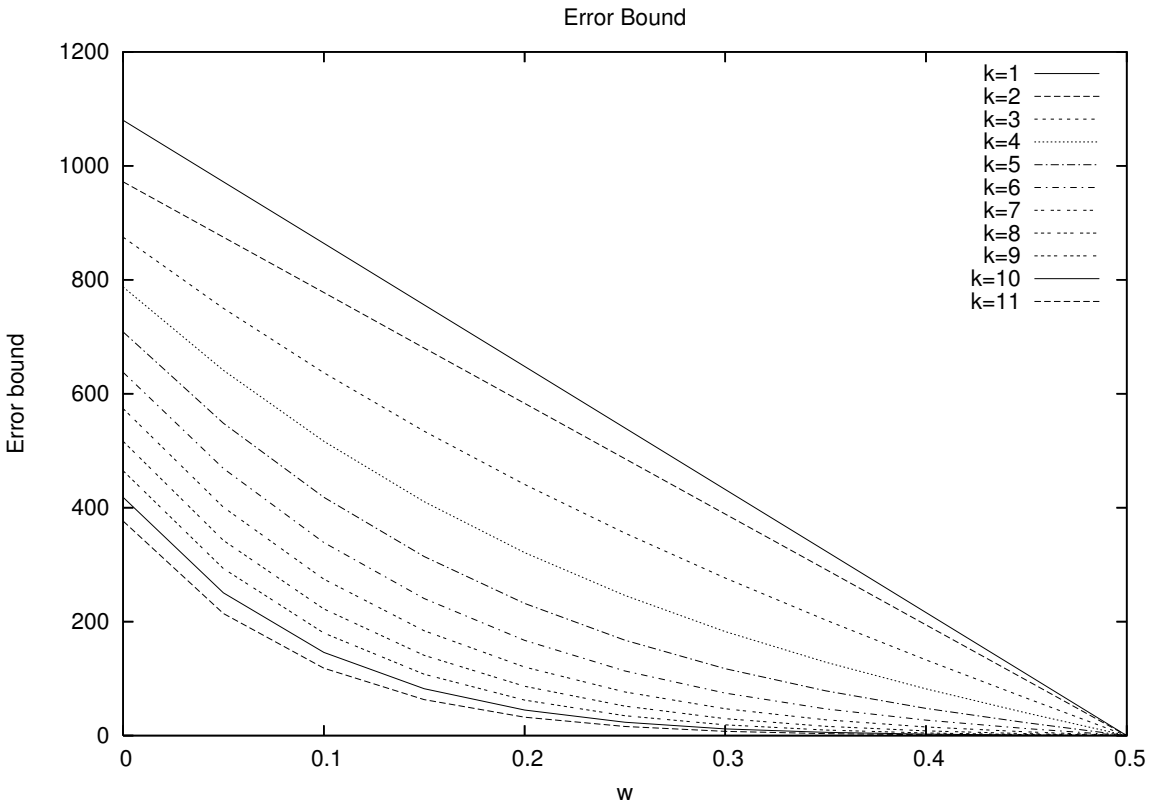


FIGURE 6.4: Error bound on the infinite discounted reward for k=0..11 against the observation function

increased, the number of belief states to consider increases exponentially with the number of joint observations). This can be seen clearly if the discount factor is very close to one, in which case extending the horizon will only cause the belief divergence error term to grow and not discount the error at the fringe by much. Consequently, where computational resources allow, it is generally better to plan for larger horizons (in reasonable static problems such as the Tiger problem). Also, as would be expected, as the observation noise increases the maximum error increases. This is a result of the belief divergence part of the term. When there is no noise, it is not possible to have a belief divergence error, so all horizons tend to 0. Similarly, the worst possible belief divergence error is possible with maximum noise. Finally, we can see that, as a result of the α vector error term $(R_{max} - R_{min})/(1-\gamma)$, the bound is not very tight because it assumes infinite worst case error.

This trade-off in maximum error can be seen more clearly in a slightly modified version of the Multi–Agent Tiger problem that we construct to show the trade-off. Specifically, we assume that observations become less reliable in the future — and that there is in fact a linear relationship between time and accuracy. Consequently, in Figure 6.5 the error in the belief divergence

increases linearly with the planning horizon:

$$\varepsilon_{BD}(k) = \frac{k}{10} \tag{6.14}$$

Here it can be seen that the maximum error increases with the planning horizon initially, but eventually the discount in the error starts to have an effect and the maximum error drops. This simple example shows that the trade-off can work in both directions.

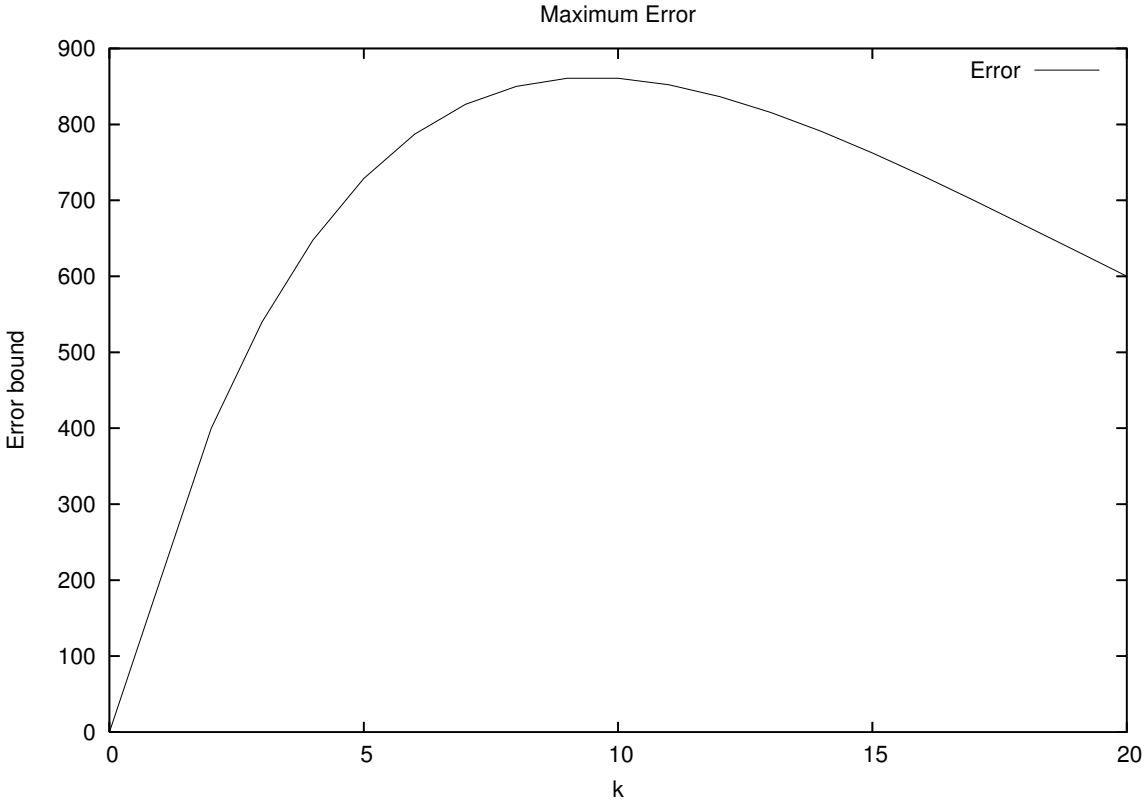


FIGURE 6.5: Error bound trade-off for linearly increasing belief divergence error and $\gamma = 0.9$

In short, this analysis shows that, in terms of error on the belief divergence, the observation function is the dominating factor. However, the discounted error on the planning horizon is also important in online solutions. Furthermore, we can see that the bound is dominated by the α vector error term — which is not very tight because it assumes worst case performance infinitely. In general an agent may err, incurring something close to a worst case penalty, and then correct itself and consequently the error will be less than the worst case bound. This is a similar weak bound used in other POMDP algorithms, as we have discussed in Chapter 2, and so we can only safely bound infinite horizon worst case performance. Given this, the $\frac{R_{max}-R_{min}}{1-\gamma}$ term is necessary and appropriate. With this established, the results we have derived can be regarded as generally applicable since a general feature of most domains is that if more time is spent planning then some accuracy will be gained by considering future consequences (discounted

rewards) yet some accuracy will also be lost because of events happening outside the control of the agent (belief divergence error).

6.5 Summary

This chapter has defined an exact version of the reward shaping algorithm presented in Chapter 4. The empirical results have demonstrated that it performs as well as the heuristic version in Chapter 5. Furthermore, we have presented an analysis of the error bound in the Multi–Agent Tiger problem which shows that the planning horizon is more important in this problem (this is to be expected in a problem which is essentially static).

Against this background, our bounded reward shaping algorithm successfully coordinates agent teams by estimating the cost of communication (research challenge 1) and then finding the value of communication (research challenge 2). Finally, it does this in a way which is error bounded and so we achieve (for the first time in this thesis) research challenge 3.

Now, the next chapter will present a different model which achieves all three research challenges, whilst arriving at the globally optimal solution. However, this is in a constrained problem class compared to the methods in this and the previous chapters. Specifically, in a Bayesian Game framework we specify how to model opportunity based costly communications (similar to how we use them in previous chapters). Following this we develop analytical forms of the exact value of communicating for this constrained class of games (see Section 8.2 for a discussion of the impact on more general settings). These games are constrained compared to the work in this chapter because we assume agents can only possibly send one communication and then take an action, rather than the more flexible mechanism using decentralised POMDPs in which planning over sequences of communications and other actions is possible.

Chapter 7

Optimal Communication Valuations in Bayesian Games

In previous chapters we have attempted solve research challenge 3 (global decentralised coordination) in problems which are very general and dynamic such as RoboCupRescue. Now the cost of this is that the solutions have been a bounded approximation of the decentralised optimal — because the problems are very large and inherently dynamic. Against this background, some still interesting problems are more static in nature. Essentially, this means that the agents control the environment entirely (unlike RoboCupRescue) and so actions can be pre-computed more easily. Specifically, a value of communication can be calculated in advance which allows for the calculation of the optimal decentralised solution, and consequently, to solve research challenge 3 optimally. This is possible because the static feature can be exploited to significantly reduce the amount of possible policies to consider. This technique is particularly useful in domains where the dynamism can be understood and modelled for instance in communication networks where we know the failure rate of nodes beforehand.

Consequently, in this chapter, we develop a method for valuing communication in a new class of games — iterated Bayesian Coordination games with explicit observing and communicating actions. In more detail, in a standard iterated Bayesian Coordination game, agents have to coordinate on different actions in different (partially observable) states of the game to receive a high payoff (see Section 2.3). In this case, coordination means that the payoff to both agents is identical. To enable coordination on the high payoff action, agents can request noisy observations of the state of the game or communicate with the rest of the team. Further to this, the game is iterated, meaning that it is reset and repeated after the agents make a decision of which action to take.

These games capture a broad class of problems that includes the canonical Multi–Agent Tiger problem from Chapter 2. Now, in our extension, observations are used to refine an agent's view of the world, and communication is used to coordinate each agent's beliefs about the state of the world before it commits to any action. However, as we have seen before, observing and

communicating take time, and may have to be carried out at the expense of some other payoff–earning action — these actions have opportunity costs. In our specific problem, then, agents must gather information about the state of the world before deciding on an action to take, because they know the state will only change if they take an action. This is the static feature that we can exploit — agents need to evaluate the value of communicating or not beforehand. They could coordinate by individually obtaining a very probable belief state or by using communication to share each others' beliefs about the world. However, depending on the relative costs of communication and observing, each policy will be more appropriate in different cases.

In more detail, we address the questions of: (i) measuring the *value* of broadcast communication in our new class of games — iterated Bayesian coordination games with explicit observing and communicating actions (research challenge 2), (ii) finding the optimal *communication policy* in such games, where a policy is composed of a combination of observing, communicating and acting, and (iii) showing that the optimal policy is an *equilibrium*. This last point is important because it means that an agent can find the optimal decentralised policy in a local fashion (research challenge 3). Furthermore, we can leverage the fact that there exist many efficient algorithms for finding such equilibriums in a distributed fashion (e.g. Sandholm et al. (2005) or Papadimitriou and Roughgarden (2004)).

In order to answer these questions, we provide the first characterisation of the opportunity cost of broadcast communication in such games. Second, based on these values, we develop a novel game–theoretic model of the interaction of agents' observing, communicating and acting policies. Third, we show that the optimal joint policy corresponds to the payoff–dominant Nash equilibrium. Fourth, we develop a new procedure that each agent can use to find this policy as a function of its payoff in the underlying Bayesian game and the noise in the observations which it receives about the state of the game. Finally, we demonstrate our method on the Multi–Agent Tiger problem and show that the value of communication increases as the relative cost of mis-coordination increases and, furthermore, decreases as the noise in the observation function decreases.

In what follows we first introduce Bayesian coordination games with explicit observing and communicating actions. Then the next section analyses broadcast communication in these games — including developing a procedure for finding the payoff–dominant Nash equilibria. Following this, we apply this analysis to the Multi–Agent Tiger problem. Finally, the last section concludes.

7.1 Communication in Bayesian Coordination Games

As mentioned before, our domain differs from the standard model of Bayesian games in two important ways, both of which allow agents to coordinate by achieving a similar view of the world. First, agents can explicitly choose to make observations of the world's state, which causes their beliefs to converge because they access the same observation function and be more

	L	R	O	C
U	a,a	c,c	δa,0	δa,0
D	c,c	b,b	c,0	c,0
O	0,δa	0,c	0,0	0,0
C	0,δa	0,c	0,0	0,0

FIGURE 7.1: Extension of the Bayesian coordination game for $s = l$, explicitly incorporating observation (O) and communication (C).

likely to take the high payoff action. Second, the agents can directly communicate (broadcast) their beliefs over the state of the world to each other. Furthermore, in our model, both of these actions take time. This is a key feature, and allows us to model more general problems in which communication consumes resources like any other action. Now, because there are only a finite number of time steps in the repeated game, the choice to observe or communicate must be made at the expense of forgoing a payoff–earning action. That is, the value of observing the state or communicating one's beliefs must be traded–off against the value of taking a less informed action.

Against this background, this section describes our extension of Bayesian coordination games to include observation and communication actions. We develop a model that is appropriate for an arbitrary number of agents, but we will often limit our discussion to two agents for ease of exposition. Formally, our model consists of a Bayesian coordination game, with the addition of explicit, time–consuming observing (O) and communicating (C) actions, repeated a finite number of times. An agent's utility function is the sum of its payoffs from each sub–game. In the sub–games, the payoffs to O and C are zero, regardless of the actions of other players in the game. In the two–player version of the game, if one agent plays O or C, we define the payoff for the second agent that takes the payoff–dominant equilibrium policy (e.g. U or L in $s = l$) as some fraction, $0 < \delta < 1$, of its equilibrium payoff. If the second agent takes a different policy, it receives the payoff for mis-coordinating. For the two–agent two–state case, these stage game payoffs are summarised in Figure 7.1 for $s = l$ (corresponding to the payoff–dominant equilibrium at $\{U, L\}$), where $a > 0 > c$, $a \geq b$ and $0 < \delta < 1$. Note that when the column player plays O or C, the payoff to the row player for playing the payoff–dominant equilibrium policy U is δa, and when it plays D its payoff is c.

We now define the components of the particular extension of Bayesian games addressed here — iterated Bayesian coordination games with explicit observing and communicating actions. To begin with, we are particularly interested in agent coordination, and so we focus on the class of *coordination games* (Mezzetti and Friedman, 2001). In these games, if the agents are able to coordinate then they receive a high payoff, and any mis-coordination leads to a low payoff, such as in Figure 7.2.

Specifically, in such games:

- **Each agent has the same size policy space** $m_i = m$ for all i;

	L	R
U	4, 3	1, 1
D	1, 0	2, 2

FIGURE 7.2: An example coordination game.

$p(s=l)$

l	L	R
U	a_1, a_2	c_1, d_2
D	d_1, c_2	b_1, b_2

$p(s=r)$

r	L	R
U	b_1, b_2	c_1, d_2
D	d_1, c_2	a_1, a_2

FIGURE 7.3: A two–player, two state Bayesian coordination game, where $a \geq b > c \lesseqgtr d$ for each player. When the state is l (left), the Nash equilibrium $\{U,L\}$ is preferred over $\{D,R\}$, while when $s = r$ (right), the opposite is the case.

- **There is the existence of a strict Nash equilibrium for each policy** The policies can be ordered so that $a^l = (l,\ldots,l)$ is a strict Nash equilibrium for all $l = 1,\ldots,m$;
- **There is a common ranking** For all $i, j \in N$ and all $h,l = 1,...,m$, $u_i(a^h) \geq u_i(a^l)$ if and only if $u_j(a^h) \geq u_j(a^l)$;
- **There exists diagonal dominance** $u(a^j) >> u(a)$ for all $a \in A/\{a^1,\ldots,a^m\}$.

These constraints on utility functions imply that in a Bayesian coordination game, different states only define different rankings of the Nash equilibria. For example, consider the two–player, two–state Bayesian game in Figure 7.3. When the state, s, is l (left), the Nash equilibrium $\{U,L\}$ is preferred over $\{D,R\}$, while when $s = r$ (right), the opposite is the case.

Next, we introduce the concept of time by considering the finite iterated version of the game. This is a structured way of formally defining the opportunity costs of actions — each action must take a timestep and it is not possible to conduct several actions in parallel. Now, in a finitely repeated game, the appropriate payoff function is the undiscounted sum of agents' payoffs. However, we consider the infinitely repeated version and so we need to maximise the payoff per timestep in one iteration of the game. As such, we can reduce the problem of finding an equilibrium in the repeated game to finding one in the sub–game. We will use this notion when performing an equilibrium analysis in this game.

Typically, the solution concept applied to Bayesian games is *Bayes–Nash equilibrium*. This concept implicitly assumes that each agent receives one signal indicating its type (payoff function), and the agents know how the state of the world and the observations are generated. From this information and the (commonly known) prior probability distribution over the states of the world, an agent can compute its expected utility for each action. However, as we will see in the next section, our model differs from the standard Bayesian game model because we treat observations as explicit actions which the agent chooses to take, and furthermore, we include communication between the agents (correlated (Bayesian) equilibria are often applied (Gerardi, 2004) in this situation). However, we are *not* trying to analyse the outcomes of a Bayesian game.

Rather, we seek to analyse the stability (i.e. whether it is an optimal policy) of communication policies, which are agreed upon before the Bayesian game is played. Given this, the payoff to a communication policy is defined at the level of expected utilities for (correlated) joint policies.

7.2 Observation and Communication Policies

In this section we analyse the interaction of various agent communication and observation policies. Specifically, we use the values and opportunity costs derived from the underlying Bayesian coordination game to construct an auxiliary game that represents the value of different communication protocols. To begin, we consider the two–state problem, and complete a full analysis of the equilibrium for n agents. Specifically, we show that the maximum payoff symmetric outcome of the game is an equilibrium (i.e. no agent has an incentive to deviate from this communication protocol), and furthermore, it is the optimal communication protocol in the game. We then consider the general d–state problem. This is more complicated than the two–state problem because the agents have to (i) choose which state to sample and (ii) map from a more complex belief space to an action. Nonetheless, we demonstrate that the same type of analysis can be completed as for the two–state problem, and to illustrate this we give an example using a uniform sampling distribution to search the space.

Before we begin the analysis, however, in order to tackle research challenge 3 optimally, we show how we can considerably collapse the set of policies admitted. First, we define the expected reward for policies in terms of whether an agent's most probable belief state is the true state of the world or not. Now, assume that when an agent takes a payoff generating action (i.e. not O or C) it takes the action with the highest expected reward given its beliefs. This allows us to reason over all the payoff generating actions as one, abstract 'act' action, which we write as A. For example, in the Multi–Agent Tiger problem, we express policies in terms of the act of opening a door (A) and not a specific door (L or R).

Second, in general, an agent's policy may be any combination of O and communication C actions followed by A, with the game resetting after this action. We only consider policies which conclude with an A and do not contain multiple As (because the game resets), as all other policies can be constructed by combining these policies, and are, therefore, redundant. Furthermore, we do not allow the agents to make any additional observations after communicating — they always act immediately after communicating — because (i) communicating more than once makes earlier Cs redundant, and (ii) for a fixed policy length, communicating later always dominates communicating earlier, because more information is transferred. Therefore, a single C immediately before A dominates all other combinations containing one or more Cs. Thus, we can restrict the agent's policies to the following combinations of actions:

- Observe m times and then act, (e.g. A or OOA) or
- Observe m times, communicate and then act (e.g. OCA).

In this case, observing m times means utilising a search policy of length m. An environment is made up of several observable features and a policy may observe a certain feature, all features in turn, or all features according to some distribution, or an information maximisation policy.

Against this background, the payoffs to agents for following combinations of these policies can be described as a normal form *auxiliary game* (a higher level game describing a game). Each outcome of this auxiliary game defines a combination of the agents' A, O and C policies, or a *communication policy*. The value of the payoff to an agent for an outcome in the auxiliary game, $\pi(a)$, is the average expected payoff per time step that the agent receives in the underlying Bayesian coordination stage game. This means that each expected value $\mathbb{E}[u_i(a_i, a_j)]$ derived in the coming sections needs to be divided by the length of its corresponding policy. We denote this value as $\pi_i(a_i, a_j)$, and it has the form:

$$\pi_i(a_i, a_j) = \frac{\mathbb{E}[u_i(a_i, a_j)]}{\min\{|a_i|, |a_j|\}} \tag{7.1}$$

where $|a_i|$ is the length of agent i's policy. Further to this, we will often drop the agent index on the expected payoff because the payoffs to all agents are symmetric and identical. In the case of two agents and two states, Figure 7.4 illustrates the generic payoff matrix to the row agent. We will refer to elements of this table in our analysis of the two–state problem. Now, in Section 7.2.1 we analyse the two–state problem, and show, using the auxiliary game, that the payoff–dominant symmetric outcome of the game is an equilibrium, and furthermore, it is the optimal communication policy in the game. Then in Section 7.2.2 we consider the more general d–state problem, in which we must reason about the choice over which state to sample alongside the problems of deciding when to observe and communicate.

7.2.1 Analysis of the Two–State Problem

We begin in Section 7.2.1.1 with expressions for the probability that two agents coordinate on the payoff dominant equilibrium or the other equilibrium, or mis-coordinate, given the level of noise in their observation function. Then, using these values, in 7.2.1.2 we reason over the expected payoff to an agent at each time step to construct the auxiliary game. Finally, in Section 7.2.1.3 using the auxiliary game, we show that the payoff–dominant symmetric outcome of the game is an equilibrium, and furthermore, it is the optimal communication policy in the game.

7.2.1.1 Observation Probabilities

To recap, an agent can make a noisy observation of the state of the world, which is true with probability $1-w$ (this function is known to the agents), and begins with a prior belief that places equal probability on every state of the world occurring. In what follows, $\hat{s}_i$ represents the most likely state according to agent i's belief state.

	A	OA	OCA	OOA	$\cdots$	O^mCA	O^mOA	$\cdots$
A	$\pi(A,A)$	$\pi(A,OA)$	$\pi(A,OCA)$	$\pi(A,OOA)$	$\cdots$	$\pi(A,O^mCA)$	$\pi(A,O^mOA)$	$\cdots$
OA	0	$\pi(OA,OA)$	$\pi(OA,OCA)$	$\pi(OA,OOA)$	$\cdots$	$\pi(OA,O^mCA)$	$\pi(OA,O^mOA)$	$\cdots$
OCA	0	0	$\pi(OCA,OCA)$	$\pi(OCA,OOA)$	$\cdots$	$\pi(OCA,O^mCA)$	$\pi(OCA,O^mOA)$	$\cdots$
OOA	0	0	$\pi(OOA,OCA)$	$\pi(OOA,OOA)$	$\cdots$	$\pi(OOA,O^mCA)$	$\pi(OOA,O^mOA)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	
O^mCA	0	0	0	0	$\cdots$	$\pi(O^mCA,O^mCA)$	$\pi(O^mCA,O^mOA)$	$\cdots$
O^mOA	0	0	0	0	$\cdots$	$\pi(O^mOA,O^mCA)$	$\pi(O^mOA,O^mOA)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\ddots$

FIGURE 7.4: Generic payoff table for row player in the 2–player auxiliary game

The probability an agent has a belief state which indicates the true state of the world more than any other state, given a history $\mathbf{m}$ of m observations, $Pr(\hat{s} = s|\mathbf{m})$ is:

$$Pr(\hat{s}_i = s|\mathbf{m}) = \begin{cases} \sum_{k=0}^{(m-1)/2} \binom{m}{k} (1-w)^{m-k}\, w^k & \text{if } m \text{ is odd,} \\ \sum_{k=0}^{m/2-1} \binom{m}{k} (1-w)^{m-k}\, w^k & \text{if } m \text{ is even.} \end{cases} \tag{7.2}$$

As can be seen, this follows a cumulative binomial distribution. This reflects all the ways that an agent can receive more observation indicating the true state of the world than any other given m. As such, the probability an agent has a belief that tends towards an incorrect state of the world given m observations, $Pr(\hat{a} \neq a|\mathbf{m})$, is:

$$Pr(\hat{a}_i \neq a|\mathbf{m}) = \begin{cases} \sum_{k=0}^{(m-1)/2} \binom{m}{k} (1-w)^{k}\, w^{m-k} & \text{if } m \text{ is odd} \\ \sum_{k=0}^{m/2-1} \binom{m}{k} (1-w)^{k}\, w^{m-k} & \text{if } m \text{ is even} \end{cases} \tag{7.3}$$

In a similar fashion to before, this reflects all the ways an agent can receive observations indicating the incorrect state of the world given m. Finally, the probability an agent has a uniform (or uninformative) belief, u, given m observations, $Pr(\hat{s} = u|\mathbf{m})$, is:

$$Pr(\hat{s}_i = u|\mathbf{m}) = \begin{cases} 0 & \text{if } m \text{ is odd} \\ \binom{m}{m/2} (1-w)^{m/2}\, w^{m/2} & \text{if } m \text{ is even} \end{cases} \tag{7.4}$$

This is a special case where an agent receives an equal number of observations for the true state and the false state given m.

7.2.1.2 Expected Payoffs

The way in which we restrict the policy space divides the expressions for the expected value of a policy into four cases.

1. The agents have identical policies that do not involve communication.
2. Agents have identical policies that do involve communication.
3. Agents' policies may be the same length and only their penultimate action differs — specifically, an agent either communicates or observes. In this case, the difference in the number of observations is never more than two, because an agent always takes an A action after communicating, at which point the episode is reset.

4. The agents may have policies of different lengths. Here, the shortest policy will terminate longer ones (since the game will repeat after an action is taken), so the shortest policy determines the payoff to all agents.

The expected payoffs to agents in each of these cases is discussed next. In what follows, we assume that agents have a common tie–breaking rule for choosing an A action when they have a uniform posterior, under which they play a predefined A action. This rule selects the correct action with probability equal to the prior; w.l.o.g. in all domains we assume a uniform prior, so the rule selects the correct action with probability 0.5 in the two door case.[1]

Case 1: Now, we first consider identical policies which do involve communication. Using payoffs from the underlying game in Figure 7.1 (a, b, and c), the expected payoff for each agent is:

$$\begin{aligned}\mathbb{E}[u_i(O^mA, O^mA)] \;=\; & a\,[Pr(\hat{s}=s|\mathbf{m}) + 1/2\; Pr(\hat{s}=u|\mathbf{m})]^2 \\ & + b\,[Pr(\hat{s}\neq s|\mathbf{m}) + 1/2\; Pr(\hat{s}=u|\mathbf{m})]^2 \\ & + 2c\,[Pr(\hat{s}=s|\mathbf{m}) + 1/2\; Pr(\hat{s}=u|\mathbf{m})] \\ & \times\; [Pr(\hat{s}\neq s|\mathbf{m}) + 1/2\; Pr(\hat{s}=u|\mathbf{m})] \end{aligned} \tag{7.5}$$

Case 2: We consider two agents, i and j, that do communicate. In this case, the expected payoff for each agent is:

$$\begin{aligned}\mathbb{E}[u_i(O^mCA, O^mCA)] \;=\; & a[Pr(\hat{s}=s|\mathbf{m}) \;+\; 1/2\; Pr(\hat{s}=u|\mathbf{m})] \\ & + b[Pr(\hat{s}\neq s|\mathbf{m}) \;+\; 1/2\; Pr(\hat{s}=u|\mathbf{m})] \end{aligned} \tag{7.6}$$

Note that, in contrast to Equation 7.5, the probability that they have different beliefs such that they mis–coordinate (and receive a payoff of c) is 0, because they have synchronised their beliefs by communicating.

Case 3: Consider different policies of the same length. As noted above, two policies of the same length can differ only in their penultimate action (e.g. *OOA* and *OCA*). Consequently, for any two differing policies of the same length, the only difference between them is that one makes one more observation than the other. Now consider the case of two agents. Because of the way policies are restricted, if agent i communicates, then agent j knows that it will act in the next time step, and knows what i will do with certainty. Therefore, its best action is to coordinate with i, ignoring it's own observations, because $c < a$ and $c < b$. The expected payoff for each agent in this case, given that i has made m_i observations, is:

$$\mathbb{E}[u_i(O^{m_i}A, O^{m_j}CA)] = a\,Pr(\hat{s}=s|\mathbf{m}_i) + b\,Pr(\hat{s}\neq s|\mathbf{m}_i) \tag{7.7}$$

[1] Any tie-breaking rule could be used, it is only important that all agents know what it is and follow it.

Furthermore, note that the symmetrically opposite outcome has the same expected payoff.

Case 4: For different length policies, we only need to define the expected payoffs to the agent with the shortest length policy, as all other agents receive a payoff of zero (as shown in Figures 7.1 and 7.4). Consequently, the expected payoff to the agent that acts is simply a function of its own beliefs:

$$\mathbb{E}[u_i(O^mCA, O^mCA)] = \delta a\, Pr(\hat{s}_i = s|\mathbf{m}_i) + c\, Pr(\hat{s}_i \neq s)|\mathbf{m}) \tag{7.8}$$

7.2.1.3 Equilibrium Analysis

We now proceed with the equilibrium analysis. As we have noted earlier, for ease of exposition, we omit the agent index since payoffs are identical to all agents. Note that expected payoffs to identical policies are located on the diagonal of the payoff matrix in Figure 7.4. From these expressions, we can derive several inequalities, which we use to show that the maximum–payoff policy on the diagonal of Figure 7.4 is a Nash equilibrium. Now, before we formally define the problem, note that for $m > 1$, the number of observations that the agents receive when both of them communicate is at least as great as the number when they do not. Consequently, when they communicate, the probability that their beliefs are accurate is at least as great as when they do not. Furthermore, when each of the agents communicate, they never mis-coordinate (i.e. they always act the same way). Therefore:

$$\pi(O^mCA, O^mCA) > \pi(O^mOA, O^mOA)\ \forall m \geq 1 \tag{7.9}$$

As a consequence, if a policy that has the maximum payoff on the diagonal involves more than one observation action, its penultimate action is always to communicate. This significantly reduces the number of points on the diagonal of Figure 7.4 that need to be compared to find the maximal element.

Therefore, the payoffs of interest on the diagonal are the maximum of $\pi(A,A)$, $\pi(OA,OA)$ and the set of $\pi(O^mCA, O^mCA)$ for $m \geq 1$. To find the maximum between these payoffs, we will solve the following integer program:

$$max\{\pi(A,A), \pi(OA,OA), \pi(O^mCA, O^mCA)\}\ \forall m \geq 1 \tag{7.10}$$

Note that, because of the form of the probabilities involved, the value of the set of $\pi(O^mCA, O^mCA)$ solutions is a bounded concave function in m. This is clear because an initial increase in observations will lead to more reward, but after a point the policy length will reduce the value of taking anymore observations. Therefore, a maximum of this set can be easily found using a simple one–dimensional search algorithm. As such, the integer program is reduced to selecting between $\pi(A,A)$, $\pi(OA,OA)$ and $\max_m\{\pi(O^mCA, O^mCA)\}$.

What interests us, however, is not only locating the maximum of the diagonal payoffs but to show that this is a Nash equilibrium. Now, the auxiliary game is symmetric, so we can do this by showing the payoff on the diagonal is greater than all other payoffs in that column. Furthermore, since we show that the maximum is on the diagonal, the equilibrium is symmetric, and so it holds for all numbers of agents. We now consider the three cases of the location of the maximum independently.

Maximum at $\pi(A,A)$: Note that every other value in the first column of Figure 7.4 is zero. Now, if $\pi(A,A)$ is the maximum diagonal element, it must be greater than zero. To see this, compare $\pi(A,A)$ to the value of the $\pi(O^mCA, O^mCA)$ elements as m grows. In the limit, the probability that the agents coordinate on the high payoff action is 1. However, as the number of time–steps tends to infinity, the average value of $\pi(O^mCA, O^mCA)$ is 0. This holds regardless of the payoffs in the underlying game or the level of noise in the observations. Consequently, if $\pi(A,A)$ is the maximum diagonal element, it must be greater than zero, so is a Nash equilibrium.

Maximum at $\pi(OA, OA)$: For $\pi(OA, OA)$, the argument is indirect. Note that:

$$\pi(A,A) = 1/2(a+b) > \pi(A, OA) = 1/2(\delta a + c) \tag{7.11}$$

because $a > \delta a$ and $b > c$. Then, if $\pi(OA, OA) > \pi(A,A)$, it is also greater than $\pi(A, OA)$. Finally, if $\pi(OA, OA)$ is the maximum diagonal element, then by the same reasoning as above, it must be greater than 0 and therefore a Nash equilibrium.

Maximum at $\pi(O^mCA, O^mCA)$: We now show that if an element of $\pi(O^mCA, O^mCA)$ is the maximum, denoted $\pi(O^{m^*}CA, O^{m^*}CA)$, then it is a Nash equilibrium. This requires several comparisons of payoffs. For example, in Figure 7.4, if $\pi(OCA, OCA)$ is the maximal payoff on the diagonal, to show that it is an equilibrium we need to show that it is greater than $\pi(A, OCA)$, $\pi(OA, OCA)$, $\pi(OOA, OCA)$ and zero. We do this indirectly, using the reasoning applied in the first two cases, the fact that $\pi(O^{m^*}CA, O^{m^*}CA)$ is the maximal element, and the following four sets of inequalities.

First, compare $\pi(A,A)$ to the payoffs in which one agent observes once and then acts, while the other observes m times, communicates and then acts, $\pi(OA, O^mCA)$. These policies are of different lengths, so we will directly compare them in terms of the observation probabilities. Now, the payoff to each policy is: $\pi(A,A) = 1/2(a+b)$ and $\pi(OA, OCA) = 1/2((1-w)\delta a + wc)$. Clearly, $a > (1-w)\delta a$ and $b > wc$, so:

$$\pi(A,A) > \pi(OA, OCA) = \pi(OA, OOCA) = \ldots \tag{7.12}$$

Second, for an arbitrary value of m, compare $\pi(O^mCA, O^mCA)$ to the payoffs to all the outcomes in which one agent observes m times, communicates and then acts (O^mCA), while the other either observes $m+l$ times ($l \geq 1$), communicates and then acts ($O^{m+l}CA$) or observes $m+l+1$ times and then acts ($O^{m+l}OA$). All of these outcomes have the same expected payoff because

they are determined by the shorter length policy, O^mCA. In Figure 7.4, these are the payoffs above the diagonal on the O^mCA row.

$$\pi(O^mCA, O^mCA) > \pi(O^mCA, O^{m+l}OA) \quad \forall m > 1,\ l \geq 1 \tag{7.13}$$

In the latter case, the agent that moves first is working from fewer observations than in the former case, so the probability that its belief is accurate is less than in the former case. Additionally, the payoffs for acting unilaterally (δa and c) are less than the payoffs when acting in a coordinated fashion (a and b). Therefore, the expected payoff is less.

Third, compare $\pi(O^mCA, O^mCA)$ to the payoffs to all the outcomes in which one agent observes $m+1$ times and then acts (O^mOA), while the other either observes $m+l$ times ($l \geq 1$), communicates and then acts ($O^{m+l}CA$) or observes $m+l+1$ times and then acts ($O^{m+l}OA$). As before, these outcomes all have the same expected payoff, which is determined by the shorter length policy, O^mOA. In Figure 7.4, these are the payoffs above the diagonal on the O^mOA row.

$$\pi(O^mCA, O^mCA) > \pi(O^mCA, O^{m+l}CA) \quad \forall m > 1,\ l \geq 1 \tag{7.14}$$

Note that in the latter case, the agent that moves first is working from, at most, the same number of observations as the former case, so the probability that its belief is accurate is at most equal to the former case. Similar to the previous comparison, the payoffs for acting unilaterally (δa and c) are less than the payoffs when acting in a coordinated fashion (a and b). Therefore, the expected payoff is less.

Fourth, compare $\pi(O^mCA, O^mCA)$ to the payoffs in which one agent communicates as the penultimate action and the other observes, $\pi(O^mOA, O^mCA)$. In the latter case, only one agent is communicating, so although coordination is guaranteed, the agents are only working from half the number of observations of the former case. Consequently, the probability that they have beliefs that are accurate is less than if they both communicate, and the following holds:

$$\pi(O^mCA, O^mCA) > \pi(O^mOA, O^mCA) = \pi(O^mCA, O^mOA)\ \forall m > 1, \tag{7.15}$$

where the equality holds by the same reasoning as above.

Given these four relations, we now show that the maximum diagonal element is a Nash equilibrium when it is an element of $\pi(O^mCA, O^mCA)$. To do this, we use the reasoning applied in the first two cases and the inequalities above to move down the column containing $\pi(O^{m^*}CA, O^{m^*}CA)$ showing that it is greater than the values in each position.

Top row: Because $\pi(O^{m^*}CA, O^{m^*}CA)$ is greater than $\pi(A, A)$, by Equation 7.11 it is also greater than $\pi(A, OA)$. Furthermore, the value of $\pi(A, OA)$ is equal to the payoff for all outcomes in which the agent immediately acts, while the other either observes m times, communicates and then acts or observes $m+1$ times and then acts (These are the payoffs above the diagonal on the

top row in Figure 7.4). Therefore, $\pi(O^{m^*}CA, O^{m^*}CA)$ is greater than any element in the top row of Figure 7.4:

Second row: By Equation 7.12, any diagonal element that is greater than $\pi(A,A)$ is also greater than the above–diagonal elements of the second row of Figure 7.4.

Remaining rows above row $O^{m^*}CA$: In this step we show that if $\pi(O^{m^*}CA, O^{m^*}CA)$ is the greatest diagonal element, then it is greater than all of the elements of its column between rows two and row $O^{m^*}CA$. For all $m < m^*$, $\pi(O^mCA, O^mCA)$ is less than $\pi(O^{m^*}CA, O^{m^*}CA)$. Then, The implication of Equations 7.13, 7.14 and 7.15 is that $\pi(O^{m^*}CA, O^{m^*}CA)$ is also greater than any of the above–diagonal elements of the O^mCA and O^mOA rows, because these values are always less than $\pi(O^mCA, O^mCA)$.

One row below row $O^{m^*}CA$: $\pi(O^{m^*}CA, O^{m^*}CA)$ is greater than the value immediately below it by Equation 7.15.

Remaining rows (all zeros): By the reasoning presented for $\pi(A,A)$, $\pi(O^mCA, O^mCA)$ is greater then zero.

Therefore, if $\pi(O^mCA, O^mCA)$ is the maximum diagonal element of Figure 7.4, it is a Nash equilibrium. This completes our analysis of the stability of the optimal communication policy for two agents. Furthermore, because the auxiliary game and the equilibrium are symmetric and these inequalities hold for all $n > 2$, the equilibrium analysis also holds for more than two agents.

7.2.2 Information Gathering in Larger Domains

We aim to develop a general model of communication valuations in coordination games and, as we mentioned earlier, coordination in these sorts of games can be achieved by communication or individually observing the state of the world. Now, as we have already highlighted, in some of these problems it may be inefficient for each agent to individually observe all parts of the problem (because many observation actions are required) and so a search policy is required which accounts for what the other agents are also searching. Given this, in this section we will show how the previous framework can be extended to include such problems and, thus, we can derive the value of communicating the results of a search policy. We will see that the same equilibrium analysis is possible because of the generality of our formalism.

Following this, the analysis so for has considered problems with two states. In that case, making an observation of one state gives us the same amount of information about the second state. However, in order to extend the analysis to problems with more states (and therefore more equilibria in the underlying Bayesian coordination game), we must consider the problem of choosing which state to sample (which partially observable feature of the problem), alongside the problems of deciding when to observe and when to communicate.

For example, consider a Multi–Agent Tiger problem comprising $d > 2$ doors and n agents, with a treasure behind only one door and tigers behind the rest. In making an observation of the state of the world, an agent has to choose which particular door to observe. To contrast with the previous problem, observing one door gives the agent much less information about the unobserved doors, and consequently the whole state. Furthermore, the agents now need to use a *search policy* to guide their choice of which door to observe. In this case a search policy is a policy that selects which features to observe — it could be deterministic, probabilistic or interactive. Moreover, the chosen search policy will have a very significant impact on the expected value to a communication policy and the subsequent equilibrium analysis. For example, agents may observe different doors, and then communicate, or they may choose to observe more doors and ignore the information gathered by other agents. Depending on the costs and benefits of communication (which themselves depend on the search policy in use), different communication policies will be better suited.

In this context, the model we defined and analysed in Section 7.2.1 is very general in the sense that the only thing that needs to be altered to complete the same analysis is the likelihood of ascertaining the correct state given a search policy of length m (as was done for two states in Equations 7.2, 7.3 and 7.4). Once this has been defined, the expected utility of a communication policy for a particular search policy is used as an input to the equilibrium analysis.

By way of example, consider a uniform search policy, in which an agent selects a door with equal likelihood m times. We can specify the probability that a uniform sample of size m^{uni} generates a belief with greatest likelihood on state s^j, as:

$$\begin{aligned} & Pr(\hat{s}_i = s|\mathbf{m}^{uni}) \\ = & \sum_{h \in H(\mathbf{m}^{uni})} \binom{m}{m^1, m^2, \ldots, m^d} \left(\frac{1}{d}\right)^m Pr(\hat{s}_i = s|\mathbf{m}^1, \mathbf{m}^2, \ldots, \mathbf{m}^d), \end{aligned} \tag{7.16}$$

where $\mathbf{m}^j$ is the number of observations of the payoff of equilibrium j and $H(\mathbf{m}^{uni})$ is the set of all samples of length $\mathbf{m}^{uni}$. In this expression, we weight each possible sample by the likelihood of it being selected using a uniform search policy, which gives a multinomial distribution over the likelihood of being able to determine the state given m observations. If the search policy was changed, this expression would need to be altered accordingly. This demonstrates the power of our approach, by allowing us to separate the search policy from the equilibrium analysis. Indeed, as long as we can specify the expected reward for a search policy in this format, then our equilibrium analysis holds and, consequently, we can always derive the optimal stable policy for each agent.

However, the specification of the probability of being able to determine the state given a sample, $Pr(\hat{s}_i = s|\mathbf{m}^1, \mathbf{m}^2, \ldots, \mathbf{m}^d)$, is a non–trivial task, involving computing the mode of a multinomial distribution. Thus, a derivation of these probabilities would distract from the main contribution

of the chapter, which is the general model for integrating costly communication and observations and the subsequent stability analysis. Nevertheless, given these probabilities, an equivalent equilibrium analysis to that presented in Section 7.2.1 follows directly, because the same relationships exist between expected payoffs to combinations of communication policies. Essentially, the stable policy is guaranteed to be on the diagonal of the game, and given any search policy specified this way, the relationship of the diagonal policies to the off-diagonal ones is always the same (as we have already shown).

7.3 The Multi–Agent Tiger Problem

Our analysis thus far has established the location of Nash equilibria for Bayesian coordination games that incorporate observation policies and communicating as explicit actions. To ground this, we now represent the Multi–Agent Tiger problem in this game (we consider both the two and multi–door settings). Specifically, we show how the location of the Nash equilibria in this game changes with the relative sizes of the payoffs and the level of noise in the observation function. This gives us an insight into the value of broadcast communication in this problem. Furthermore, it allows us to detail the conditions under which communication is a useful activity, compared to achieving a more certain impression of the state and acting with less information.

7.3.1 The Two Door Setting

We examine the location of the payoff–dominant equilibrium policy as the payoff for coordinating on the location of the tiger, b, and the level of noise, w, vary. Specifically, we present results for values of $b = \langle -20, -5, 2 \rangle$, fixing $a = 25$ and $c = -25$. The relative sizes of a and c are less interesting in this problem, since it is varying b that produces a change in behaviour. Figure 7.5 shows the location of the Nash equilibria in these three instantiations of the two–agent two–door problem.

As can be seen, for low levels of noise ($w < 0.08$) the optimal policy is $\{OA, OA\}$, regardless of payoffs (note that all figures initially have the symbol $\times$ for $m = 1$). This is because a single observation is true with high enough probability for it to act as a coordinating mechanism for the agents, without the need to communicate or make any further observations. When there is a high penalty for coordinating on the wrong door ($b = -20$), then, as noise increases, it is beneficial to obtain more observations before communicating (note the symbol $\circ$ for increasing m as noise increases). This trend is flatter for higher values of b because there is less incentive to get the correct door as b increases. Furthermore, when there is a small penalty, or even benefit, to coordinating on the wrong door ($b = -5, 2$), then the problem is dominated by the need to avoid mis-coordination, hence there is more communication compared to observations. Also, in this case, it is not worth spending time finding the tiger, so the agents just make a single (or no) observation, $m = 1$, and communicate straight away. In particular, when there is a positive

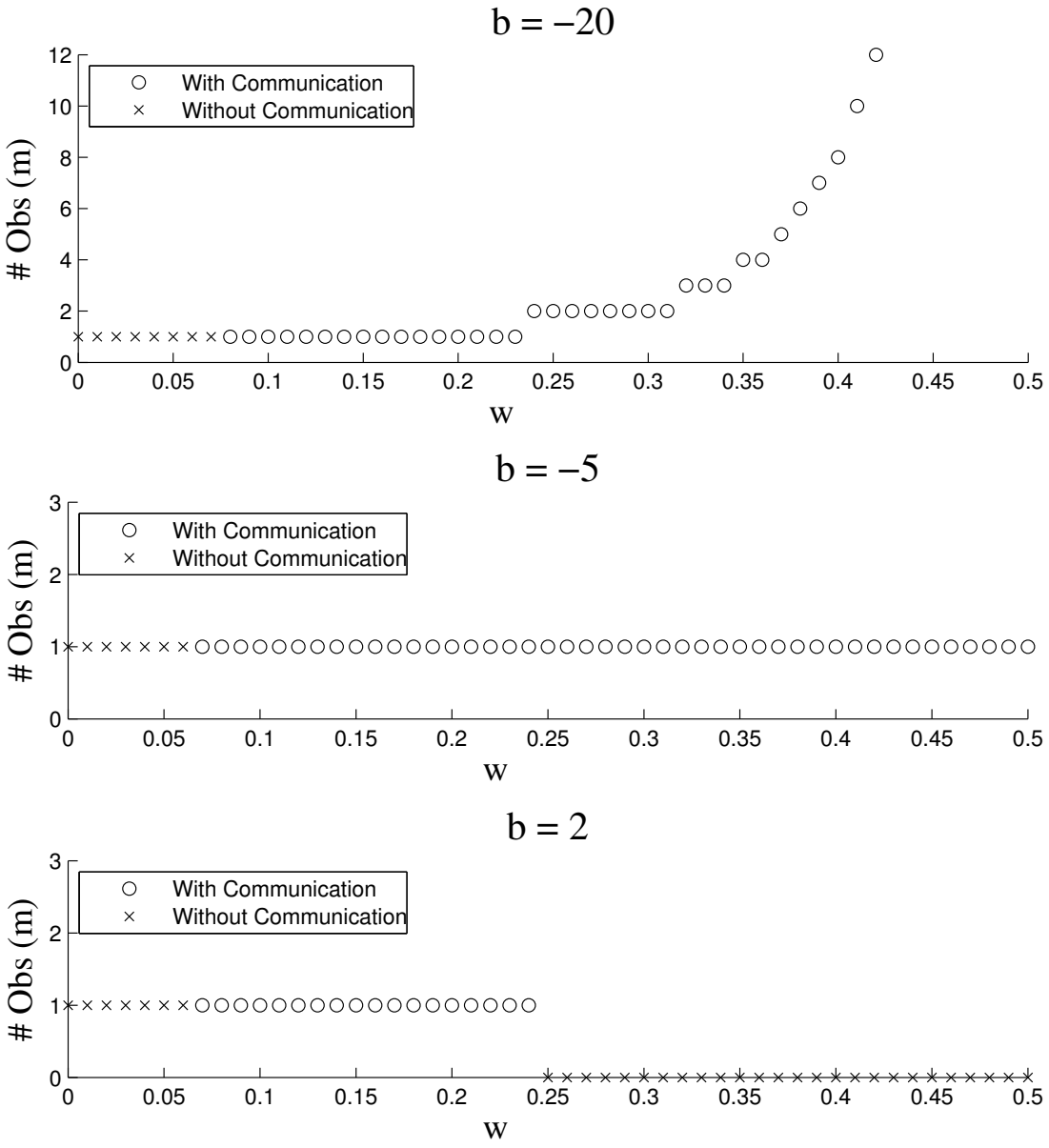


FIGURE 7.5: Three examples of the optimal communication policy in the two–player tiger problem. The symbol × indicates a policy with no communication, while ∘ indicates communication. For all, $a = 10$ and $c = -25$, with the value of b as specified.

reward for jointly opening the wrong door $b = 2$, a high level of noise ($w > 0.25$) erodes any benefit of observing, and the agents can simply rely on the tie–breaking rule to coordinate.

7.3.2 The Multi–Door Setting

Moving onto the multi–door problem, we present results as above with $a = 20$, $b = 5$ and $c = -100$. Specifically we consider the two door two agent case (labelled $2-2$ in Figure 7.6),

the three door four agent case (labelled $3-4$ in Figure 7.6) and the four door two agent case (labelled $4-2$ in Figure 7.6) as representative examples.

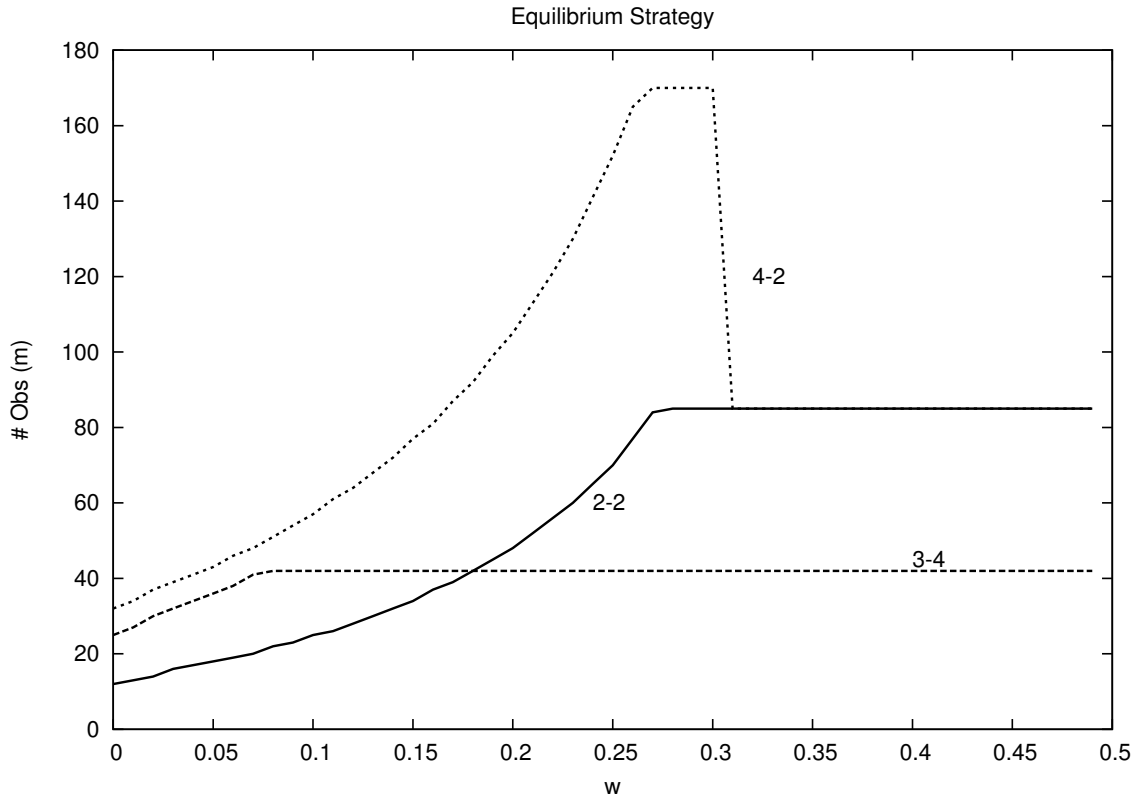


FIGURE 7.6: Optimal policy length m for The Multi–Agent Multi–Door problem for $2-2$, $3-4$ and $4-2$.

In Figure 7.6 we can see that the optimal policy length increases as noise increases (w increases). The policies for $2-2$ and $3-4$ are similar — to always communicate, however $3-4$ is eventually shorter than $2-2$ and that is because the problems are roughly similar but there are more agents to search in $3-4$. Therefore communication is more valuable to reduce the policy length. In $2-2$, once $w > 0.25$, there is no sense in gathering more observations (as they become noisier) so $m = 80$ holds. The same is true for $3-4$ at $w = 0.075$ where $m = 40$. The $4-2$ problem has a very large policy length because it is a much larger problem (four doors to search) and does not initially communicate until a phase change occurs at $w = 0.3$ where communication quickly becomes more important in order to confirm noisy observations with each other. Consequently, at this point, m drops from 170 to 80. Before this point it is important to gather more observations. Hence the phase change.

Finally, in the majority of cases there is a value to communicating — primarily because it guarantees coordination. In this setting, the observation function has to be virtually error–free for the agents to coordinate without communication — even in the cases where communication is costly. Consequently, the value of this communication policy increases as the relative cost of

mis-coordination increases, whilst decreasing as the noise in the observation function decreases. This highlights a general relationship between communication and coordination — that it is not needed if the agents can independently ascertain the global state of the problem, but it becomes more important as this gets harder. Furthermore, this trend is more pronounced as the cost of mis-coordination increases.

7.4 Summary

In this chapter we consider the cost and value of information gathering and exchange policies in coordination problems. Specifically, we argued for the use of opportunity costs as a principled and general formalisation of the impact of communication during multi–agent coordination. This model avoids arbitrary costs and valuations for using a communication medium, and instead, allows the benefits and penalties to be represented in terms of the increase or decrease in team utility as a result of using communication (research challenge 2).

In more detail, we developed a novel variant of Bayesian coordination games that incorporates search and communicating as explicit actions. Within this game, we derived expressions for the value of different communicating, observing and acting policies — or communication policies — using opportunity costs. We then showed that the optimal communication and search policy is the payoff-dominant Nash equilibrium. Finally, we provided a procedure for finding this equilibrium as a function of the noise in the underlying Bayesian game. We demonstrated that in the Multi–Agent Tiger problem, the value of communication increases with the relative cost of mis-coordination and decreases with the noise in the observation function. Our solution to research challenge 3 in this chapter is optimal (compared to a bounded approximation in the previous chapter), however that is because we restrict the generality of the problem by considering only static domains. Specifically, with this approach we can pre-compute plans because we know that the environment is completely controlled by only the agents. It is clear that in more dynamic domains, such as RoboCupRescue, our reward shaping based solution is more appropriate, since that can account for things happening outside the team's control.

Chapter 8

Conclusions and Future Work

In this chapter, we first present our concluding remarks regarding the work developed in this thesis. After this, we identify several avenues of future research.

8.1 Conclusions

In this thesis we argued that the coordination of multi–agent systems in the presence of an expensive or restricted communication medium is an important problem. Specifically, we posited the need for a rational approach to communication. That is, agents need to explicitly evaluate the utility of sending a specific communication, and if they do this, then they can balance the expected utility gain with the cost of communicating. In order to establish this, we identified three research challenges which should be addressed in order to achieve such rational communication: (i) how to specify, in a general framework, the cost of using the communication mechanism; (ii) how to specify the value, in terms of future utility, of sending a particular message to the rest of the team; and (iii) how to use this communication cost and valuation to control a decentralised system in a manner which represents an optimal or bounded approximation of the optimal solution. Now, in achieving these research challenges, our work has advanced the state of the art in the field of coordinating multi–agent systems in the presence of communication restrictions for solving decentralised control problems. In the following we summarise each chapter in relation to how it achieves these research challenges.

With this problem and proposed solution in mind, the decentralised POMDP formalisation was chosen as the point of departure for this work. This is because of its general nature and the ease with which a problem containing domain actions and communications can be modelled. This model also allows the integration of communication valuations. Following this, information theory, which is one such valuation technique, was chosen as a metric for making communication decisions because it does not place any demands on the coordination mechanism, making it a general method of valuing communication. To date, traditional approaches based on decentralised POMDPs do not explicitly value the communication acts that they allow. Rather,

they rely on the value being inferred during policy computation. In contrast, our work values communication directly so that it can be reasoned about explicitly. By so doing, large amounts of policy computation are not required to derive an implicit value. Also, by defining a general valuation, it can be used in domains where the dynamics are not known perfectly, and it forms a sound basis for a dynamic rational communication model.

Against this background, in Chapter 3, we introduced the *dec_POMDP_Valued_Com* model — a model of coordination using rational communication, and then detailed an action selection mechanism that can manipulate this model. We learnt a weighted information measure as a reward for communication in this method. Following this, we showed how the ambulance agent task from RoboCupRescue could be represented as a *dec_POMDP_Valued_Com* — which demonstrated how the communication valuation is used to communicate rationally. Finally, we analysed this model empirically and our results show that valued communication leads to better policy computation in decentralised POMDP models than basic rule-based models. As a result, we proposed a method which successfully solves research challenge 1 (the cost of communication) — by capturing this cost using opportunity costs in a sequential decision making framework. However, by valuing communications using a weighted information measure, we cannot solve research challenge 3 (decentralised coordination) because there is no formal link between how information influences coordination. Further to this, the learning process was slow making it hard to apply this method to larger problems.

To solve this flaw, in Chapter 4 we presented a general model of coordination inspired by *Reward Shaping*. Specifically, we defined how a traditional decentralised POMDP can be transformed into individual local knowledge POMDPs defined over the *belief divergence* in the team. This has the advantage of reasoning over only local knowledge, at the expense of ignoring the possible team experience. Within this transformation, belief divergence is used to shape individual estimations of the expected reward for joint actions based on how coordinated the team currently is. Furthermore, this framework allows communication to be cast as a function that explicitly modifies belief divergence (by synchronising beliefs across the team). This provides a principled way to both value communications and also reduce the computational complexity of the underlying team problem. Finally, the reward shaping function captures the value of communication and so no learning is required, unlike the model in Chapter 3. As a consequence, communication can be employed in an efficient rational manner on larger problems than was hitherto possible. Furthermore, we again used the opportunity cost based sequential decision making to solve research challenge 1 (how to accurately cost communications) as in Chapter 3. This framework was expanded to distinct problem domains in later chapters. Specifically, Chapter 5 used a heuristic approach to solve large problems, whilst Chapter 6 used an exact method to solve research challenge 3 (decentralised coordination).

In Chapter 5, using the model from Chapter 4, we have shown that the relationship between belief divergence and coordination for a given problem can be used in the form of problem specific shaping functions. We then implemented this in terms of the Multi–Agent Tiger and RoboCupRescue problems. Using heuristic shaping functions we extend the state of the art

in online communication valuations by providing a technique that outperforms existing work and also the technique from Chapter 3, whilst employing a more realistic costly communication medium (specifically that communication takes time like any other action). Also, because Chapter 3 employed an offline learning mechanism which the technique in this chapter does not, we can also solve larger scale problems. However, as we discussed earlier, using heuristic shaping functions contains an unknown error on the value of communication and so does not allow for a solution to research challenge 3 (optimal or bounded approximation coordination) since the heuristic functions have only informal information about the link between communication valuation and coordination. This link is made exact in the next chapter, at the cost of some scalability.

In Chapter 6 we proposed an alternative reward shaping function, which allows for theoretical bounds on the error in using the approach compared to an optimal decentralised POMDP. However, this bound comes at the expense of some degree of scalability because it is more difficult to compute than the heuristics in Chapter 5. Nevertheless, having a bound is particularly useful since it allows us to ascertain beforehand whether our technique is appropriate for a given problem, and what performance, in the worst case, we will lose by applying this approximation. Consequently, our approach has theoretical guarantees on the quality of its solutions. This is something that is lacking from all previous decentralised POMDP and communication valuation models. To ground this, an analysis of this error was presented for the Multi–Agent Tiger problem where it was found that the dominating term in the error is the planning horizon rather than how to estimate the belief divergence. Specifically, we found that longer horizons are more desirable, even with the larger error this introduces to the belief divergence. This will, in general, be true for other problems, although the point at which the belief divergence error becomes dominating will change with the observation function for that problem. As a result, this method allowed us to solve research challenge 3 in that we could find a bounded approximation of the optimal decentralised solution.

Finally, in Chapter 7, we explored a different (albeit complementary) direction in solving research challenge 2. Specifically, we develop a novel variant of Bayesian coordination games that incorporates search and communicating as explicit actions. Within this game, we derived expressions for the value of different communicating, observing and acting strategies — or communication policies — using opportunity costs (as in Chapter 3 for solving research challenge 1 — the cost of communication). We then showed that the optimal communication and search policy is the payoff-dominant Nash equilibrium. Finally, we provided a procedure for finding this equilibrium as a function of the noise in the underlying Bayesian game. We demonstrated that in the Multi–Agent Tiger problem, the value of communication increases with the relative cost of mis-coordination and decreases with the noise in the observation function. This approach solves research challenge 2 analytically and optimally for a constrained class of problems (specifically, they are static problems allowing the strategy to be decided in advance), and consequently, solves research challenge 3 — optimal decentralised coordination.

To summarise, in the context of restricted communication, we have contributed a heuristic solution to large scale dynamic problems (Chapters 4 and 5), a bounded approximation of a solution to slightly smaller dynamic problems (Chapters 4 and 6) and finally, an optimal solution to large scale static problems (Chapter 7). Against this background, we can justifiably claim to have achieved all of the research challenges in the interesting areas of the full problem space and consequently satisfied our central thesis that coordination in the presence of restricted communication calls for a rational approach to communication. Despite these achievements, a number of open issues remain. For instance, throughout this thesis we use the concept of broadcast synchronisation communication which in effect makes bandwidth free (but not availability). We have answered *when* to communicate but the most important follow–up would be to tackle the question of *what* to communicate in order to alleviate this problem. If these two engineering problems could be solved in a general fashion then we would have the first steps of understanding the impact of limited communication on bounded rationality in distributed artificial intelligence.

In more detail, this thesis has highlighted the important issue that coordination is inherently more difficult in the presence of restricted communication. This may seem like an obvious point, but often this is overlooked in coordination mechanisms where communication is treated as something that will be taken care of by a network protocol that will guarantee the receiving of a message (by multiple broadcasts and acknowledgements). In many domains this is simply not true and in fact highly counter-productive. This question is found most clearly in the more general question of bounded rationality. Specifically, if an agent's ability to compute a solution is bound by its processing power, time or information, then a limited communication infrastructure has an impact on these bounds in a distributed artificial intelligence setting. Our solution has shown how communication (with assumptions about the reasoning of the other agents) can be used to reduce the processing requirements of the team if that communication is unlimited — because there are no limits on information in the team. However, the more interesting question is when communication is limited and as a result the bounded processing requirements become larger in order to achieve the same solution. This is a fundamental problem in distributed decision making.

Moreover, as a result of the methods adopted in this work, the techniques that have been developed herein can be used in any problem where the agent must reason about the use of an expensive communication medium. Specifically, the methods of valuing communication can be extracted from their decentralised POMDP foundations and applied to other problem representations such as DCOPs or intentional teamwork models. It would be highly useful to develop distributed artificial intelligence algorithms whose solution quality is parameterised on the limits of the communication environment (in a manner similar to bounded rationality). Further to this, since we have presented a formal method for *separating* a team problem into individual problems, those researchers interested in studying fundamental questions about how much of a given team problem can be solved in isolation will find much of interest in this work.

8.2 Future Work

In the future, there are four main directions research could take from this thesis. Specifically, one direction could be to find different ways of estimating the belief divergence in the team and study the impact this has on the error bounded approach using reward shaping. A second direction would be to compare the theoretical value of communication defined in the literature with the value generated in our reward shaping scheme. A third direction would be to use our reward shaping method as the basis for an online error minimising algorithm (using communication) for controlling decentralised systems. A final direction would be to extend the game–theoretic analysis of communication valuations to more expressive domains and see if the analytical solutions remain tractable. We will now discuss each of these in more detail:

- **Belief Divergence Estimation:** In the future we intend to expand the analysis of this technique to other methods of estimating the belief divergence in distributed teams. In particular, our technique of using reward shaping to manage coordination and communication is very general, but its error is very dependent on the method of measuring and estimating belief divergence. At the moment we use a relatively straight-forward technique defined over an absolute measure of belief divergence, but this can be viewed as the first step toward a general model of using belief divergence to analyse communication in teams. Specifically, we could include prior knowledge of the environment, for a more advanced estimation of the belief divergence or use learning for a dynamic model. Furthermore, we could use other measures of belief divergence (our bounded solution use absolute difference and the heuristic uses KL Divergence) such as those from information theory (e.g. Entropy) since these will capture some of the information relationships implicit in the reward and state functions. At the moment, computing the divergence involves enumerating over all states — so some aggregate measure may be more efficient to compute and consequently allow both heuristic and bounded reward shaping to applied to still larger problems. However, we would need to estimate the error this introduces. Consequently, any change in the method of measuring or estimating the belief divergence would call for a new error bound analysis (if it is possible with the method used) and so this requires careful consideration.

- **Value of Communication:** There has been other work on the exact value of communication in decentralised POMDPs (see Section 2.4.2 for more details), and given this, it would be instructive to investigate if that valuation is preserved in our transformed model. In particular, this would shed light on the issue of whether, in general, decentralised POMDPs can be separated for each agent and then use communication in a principled fashion to maintain optimality. If we could establish this, then there would be scope for algorithms that solve independent parts of the complete decentralised POMDP in isolation (with a communication policy) and without loss of optimality in a truly decentralised fashion.

- **Communication Based Control:** In this work we have presented a definition of the error which can be computed offline to analyse the algorithm. It would also be useful if the agents could compute this error online and then use this as a parameterisation for the volume of communication required in a given problem. By doing so, we could develop a coordination algorithm which explicitly manages the error introduced by not communicating. Consequently, a desired level of coordination could be specified beforehand. This would be a useful technique since such an algorithm could allow the system designer to specify how much error is acceptable and then the model would decide how much of an expensive communication medium to use to ensure this. This would be a hard task for a system designer to do on her own as it would probably involve significant offline simulations and parameter tuning.

- **Game Theoretic Valuations:** Our general game-theoretic characterisation of opportunity cost based broadcast communication can easily be extended to consider other forms of communication. Specifically, it would be interesting to consider the issue of *who* to communicate with since some tasks may only require coordination between a subset of agents. This would involve extending the strategy space to include a decision over which agent to communicate with. Furthermore, it would be useful to consider *what* to communicate, since not all information is useful to coordination. Clearly, the content of a message influences the value of communicating that message, and a game-theoretic analysis of complex communication would be a useful line of research.

Appendix A

RoboCupRescue Dec POMDP

```
/*
* Implements Sebastien Paquet's RTBSS algorithm \citep{onlinePOMDP}for online lookahead making
* extended to Dec-POMDPs with communication valuations using the rescuecore interface for RCR
*/

package decpomdpRCR;
import java.lang.Math;
import java.util.*;

import rescuecore.CannotFindLocationException;
import rescuecore.Memory;
import rescuecore.RescueConstants;
import rescuecore.RescueObject;
import rescuecore.commands.AKLoad;
import rescuecore.commands.AKMove;
import rescuecore.commands.AKRescue;
import rescuecore.commands.AKUnload;
import rescuecore.commands.AgentCommand;
import rescuecore.commands.Update;
import rescuecore.objects.Civilian;
import rescuecore.objects.MovingObject;
import rescuecore.objects.Road;

public class RTBSSDecPOMDPRCR {

        public int      D;
        private double  gamma;
        private double  bestValue = Double.NEGATIVE_INFINITY;
        public JointAction best_action;
        private RescueObject agent;
        private POMDP_Ambulance_Agent   pom;
        public int bottom=0;
        public int nodes = 0;
        public int prunes=0;
        public int notprunes=0;
        double[] discount;
        boolean beenbottom=false;
        public boolean verbose=false;
        public boolean  verbose_heuristic = false;
        public static int USED=0;
        public static int NOTUSED=1;
```

```
RescueObject[] target;
public boolean[] target_civ;
public InformationMetrics inf;
public double alpha; // alpha is weighting of rescue

public RTBSS(int depth, double g, RescueObject agent, POMDP_Ambulance_Agent pom,
                double alpha){
        D = depth;
        gamma = g;
        this.agent = agent;
        this.pom = pom;
        discount = new double[D+1];
        target = new RescueObject[pom.belief.team.length];
        target_civ = new boolean[pom.belief.team.length];
        inf = new InformationMetrics(pom);
        this.alpha = alpha;
        int placeID[]=null;
        RescueObject[] position = pom.belief.mostLiklyTeamPosition(pom.belief,placeID);

}

/*
 * Main search algorithm
 *
 * Inputs: b - the current belief state
 *         d - the current depth
 *         rAcc - accumulated rewards
 *
 * Statics: D - the maximal depth search
 *          bestValue - the best value found in the search
 *          best_action: - the best action
 *
 */
public double rtbss_search(Belief_State b, int d, double rAcc,
                RescueObject[] was, Belief_State original){
        double finalValue;
        ArrayList<JointAction> actionList = new ArrayList<JointAction>();
        double max;
        double expReward;
        Belief_State b_tick;
        JointAction action;
        ArrayList<POMDPObservation> obs;
        boolean[] move;
        int[] ids;

        //bottom of search tree
        if(d==0){
                double utility = java.lang.Math.pow(gamma, D)*b.utility(was);
                finalValue = rAcc + utility; //leaf estimation function
                if (finalValue >= bestValue){
                        bestValue = finalValue;
                bottom++;
                return finalValue;
        }

        double reward = reward(b); //reward function from POMDP
        rAcc = rAcc + java.lang.Math.pow(gamma, D-d)*reward;

        actionList = sort(b,was); //possible actions at this belief
```

```
                max = 0.0;
                ListIterator aciter = actionList.listIterator();

                while (aciter.hasNext()) {
                   action = (JointAction)aciter.next();
                   move = action.getMoveBooleans();
                   expReward =0;

                    if(action.getMyAction(pom.myid).getType()==POMDPBehaviour.PICKUP){
                        expReward =expReward+ reward(b);
                    } else
                   if(action.getMyAction(pom.myid).getType()==POMDPBehaviour.DROPOFF){
                      expReward =expReward+ reward(b);
                    } else
                    if(action.getMyAction(pom.myid).getType()==POMDPBehaviour.COMMUNICATE){
                        //witha  cost function
                       double comv = pom.scale*inf.calculateDistance(pom.PriorBelief,b);
                       expReward = comv;
                     }
                   //recursive step
                   Belief_State b_tick_ac = transition_action(b, action);
                   ArrayList<ArrayList<POMDPObservation>> Obs = pom.god.state.getObservationsBelief
                         (agent,b_tick_ac,original);
                   ListIterator obsiter = Obs.listIterator();
                   double obmax = 0.0;
                   while (obsiter.hasNext()) {
                        obs = (ArrayList<POMDPObservation>)obsiter.next();
                        double pobs = probObs(Obs);
                         b_tick = transition_obs(b_tick_ac,obs,agent);
                        ids = new int[b.team.length];
                        RescueObject[] mostlikely = b_tick.mostLiklyTeamPosition(ids);
                        double s = rtbss_search(b_tick, (d-1), rAcc, mostlikely,original);
                        expReward = expReward + java.lang.Math.pow(gamma, (D-d))*pobs*s;
                        if(s>obmax){
                             obmax = s;
                        }
                    }
                     obmax=expReward;
                     if(expReward >= max){
                        if(D==d){
                            max = expReward;
                            best_action = action;
                        }
                return max;
        }
}
```

Bibliography

Artikis, A., Sergot, M. and Pitt, J. (2007), 'An executable specification of a formal argumentation protocol', *Artificial Intelligence* **171**(10-15), 776 – 804. Argumentation in Artificial Intelligence.

Austin, J. L. (1962), *How To Do Things With Words*, Oxford University Press.

Baxter, J. (2006), QinetiQ personal communication.

Becker, R., Carlin, A., Lesser, V. and Zilberstein, S. (2009), 'Analyzing Myopic Approaches for Multi-Agent Communication', *Computational Intelligence* **25**(1), 31–50.

Becker, R., Lesser, V. and Zilberstein, S. (2005), Analyzing myopic approaches for multi-agent communication, *in* 'Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology', Compiegne, France, pp. 550–557.

Becker, R., Zilberstein, S., Lesser, V. and Goldman, C. V. (2003), Transistion-independent decentralized markov decision processes, *in* 'Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Melbourne, Australia, pp. 41–48.

Bernstein, D. S., Zilberstein, S. and Immerman, N. (2000), The complexity of decentralized control of markov decision processes, *in* 'Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence', Stanford, USA, pp. 32–37.

Binmore, K. (1990), *Essays on Foundations of Game Theory*, Pittman.

Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999), *Swarm intelligence: from natural to artificial systems*, Oxford University Press, Inc., New York, NY, USA.

Boutilier, C. (1999), Sequential optimality and coordination in multi-agent systems, *in* 'Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)', San Franciso, USA, pp. 478–485.

Bratman, M. E., Israel, D. and Pollack, M. (1991), Plans and resource-bounded practical reasoning, *in* R. Cummins and J. L. Pollock, eds, 'Philosophy and AI: Essays at the Interface', The MIT Press, Cambridge, Massachusetts, pp. 1–22.

Brooks, R. A. (1999), *Cambrian Intelligence: The Early History of New AI*, The MIT Press.

Carlin, A. and Zilberstein, S. (2009), Value of communication in decentralized pomdps, *in* 'Proceedings of the AAMAS Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains', Budapest, Hungary, pp. 16–21.

Cohen, P. R. and Levesque, H. J. (1991), 'Teamwork', *Nous* **25**(4), 487–512.

Dash, R., Rogers, A., Jennings, N. R., Reece, S. and Roberts, S. (2005), Constrained bandwidth allocation in multi-sensor information fusion: A mechanism design approach, *in* 'Proceedings of the The Eighth International Conference on Information Fusion', Philadelphia, USA, pp. 8–16.

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

Durfee, E. and Lesser, V. (1991), 'Partial global planning: A coordination framework for distributed hypothesis formation', *IEEE Transactions on Systems, Man, and Cybernetics* **21**(5), 1167–1183.

Dutta, P., Jennings, N. R. and Moreau, L. (2005), 'Cooperative information sharing to improve distributed learning in multi-agent systems', *Journal of AI Research* **24**, 407–463.

Dutta, P. S., Goldman, C. and Jennings, N. R. (2007), Communicating effectively in resource-constrained multi-agent systems, *in* 'Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)', Hydrabad, India, pp. 1269–1274.

Dutta, P. S., Jennings, N. R. and Moreau, L. (2006), Adaptive distributed resource aladdress and diagnosis using cooperative information-sharing, *in* 'Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Hakodate, Japan, pp. 826–833.

Emery-Montemerlo, R., Gordon, G., Schneider, J. and Thrun, S. (2004), Approximate solutions for partially observable stochastic games with common payoffs, *in* 'Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', New York, USA, pp. 136–143.

Emery-Montemerlo, R., Gordon, G., Schneider, J. and Thrun, S. (2005), Game theoretic control for robot teams, *in* 'Proceedings of the 2005 IEEE International Conference on Robotics and Automation, (ICRA)', Barcelona, Spain, pp. 1163–1169.

Estlin, T., Gaines, D., Fisher, F. and Castano, R. (2005), Coordinating multiple rovers with interdependent science objectives, *in* 'Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Utrect, The Netherlands, pp. 879–886.

Finin, T., Fritzson, R., McKay, D. and McEntire, R. (1994), Kqml as an agent communication language, *in* 'Proceedings of the Third International Conference on Information and Knowledge Management (CIKM)', ACM, New York, pp. 456–463.

Fudenberg, D. and Tirole, J. (1991), *Game Theory*, The MIT Press.

Gerardi, D. (2004), 'Unmediated communication in games with complete and incomplete information', *Economic Theory* **114**(1), 104–131.

Gerkey, B. P. and Mataric, M. J. (2001), 'Sold!: Auction methods for multi-robot coordination', *IEEE Transactions on Robotics and Automation (Special Issue on Multi-Robot Systems)* **18**, 758 – 768.

Ghavamzadeh, M. and Mahadevan, S. (2004), Learning to communicate and act using hierarchical reinforcement learning, *in* 'Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Vol. 03, New York, USA, pp. 1114–1121.

Gmytrasiewicz, P. and Doshi, P. (2004*a*), A framework for sequential planning in multi-agent settings, *in* 'Eighth International Symposium on Artificial Intelligence and Mathematics'.

Gmytrasiewicz, P. and Doshi, P. (2004*b*), Interactive pomdps: Properties and preliminary results, *in* 'Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', New York, USA, pp. 1374–1375.

Gmytrasiewicz, P. and Doshi, P. (2005), 'A framework for sequential planning in multi-agent settings', *Journal of Artificial Intelligence Research* **23**, 49–79.

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

Goodie, A., Doshi, P. and Young, D. (2009), Recurseive reasoning by humans playing sequential fixed-sum games, *in* 'Proceedings of the AAMAS Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains', Budapest, Hungary, pp. 38–43.

Hansen, E. A. (1998), Solving pomdps by searching in policy space, *in* 'Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence', pp. 211–219.

Harsanyi, J. C. (1967), 'Games with incomplete information played by bayesian players', *Management Science* **14**, 159–182.

Hauskrecht, M. (2000), 'Value-function approximations for partially observable markov decision processes', *Journal of Artificial Intelligence Research* **13**, 33–94.

Hiroaki, K. (2000), Robocup rescue: A grand challenge for multi-agent systems, *in* 'Proceedings of the 4th International Conference on Multi-Agent Systems', Boston, MA, USA, pp. 5–12.

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

Jennings, N. R. (2001), 'An agent-based approach for building complex software systems', *Communications of the ACM* **44**(4), 35–41.

Kadane, J. and Larkey, P. (1982), 'Subjective probability and the theory of games', *Management Science* **28**, 113–120.

Kaelbling, L. P., Littman, M. L. and Cassandra, A. R. (1998), 'Planning and acting in partially observable stochastic domains', *Artificial Intelligence* **101**(1), 99–134.

Karim, S. and Heinze, C. (2005), Experiences with the design and implementation of an agent-based autonomous uav controller, *in* 'Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Utrect, The Netherlands, pp. 19–26.

Khoshnevis, B. and Bekey, G. (1998), 'Centralized sensing and control of multiple mobile robots', *Computers and Industrial Engineering* **35**(3-4), 503–506.

Kinney, M. and Tsatsoulis, C. (1998), 'Learning communication strategies in multi-agent systems', *Applied Intelligence* **9**(1), 71–91.

Krishna, R. (2007), 'Communication in games of incomplete information: Two players', *Economic Theory* **132**(1), 584–592.

Kullback, S. and Leibler, R. A. (1951), 'On information and sufficiency', *Annals of Mathematical Statistics* **22**, 79–86.

Laud, A. and DeJong, G. (2002), Reinforcement learning and shaping: Encouraging intended behaviors, *in* 'Proceedings of the Nineteenth International Conference on Machine Learning (ICML)', San Francisco, CA, USA, pp. 355–362.

Lesser, V., Decker, K., Wagner, T., Carver, N., Garvey, A., Horling, B., Neiman, D., Podorozhny, R., NagendraPrasad, M., Raja, A., Vincent, R., Xuan, P. and Zhang, X. Q. (2002), Evolution of the gpgp/tæms domain-independent coordination framework, *in* 'Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Bologna, Italy, pp. 1–2.

Lesser, V. R. and Erman, L. D. (1980), 'Distributed interpretation: A model and experiment.', *IEEE Transactions on Computers* **29**(12), 1144–1163.

Ljungberg, M. and Lucas, A. (1992), The OASIS air-traffic management system, *in* 'Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence (PRICAI '92)', Tokyo, Japan.

Mailler, R. and Lesser, V. (2004), Solving distributed constraint optimization problems using cooperative mediation, *in* 'Proceedings of the Third International Conference on Autonomous Agents and Multiagent Systems (AMAAS)', pp. 438–445.

Mataric, M., Nilsson, M. and Simsarin, K. (1995), 'Cooperative multi-robot box-pushing', *IEEE/RSJ International Conference on Intelligent Robots and Systems* **3**, 3556.

Mezzetti, C. and Friedman, J. W. (2001), 'Learning in games by random sampling', *Economic Theory* **98**(1), 55–84.

Modi, P., Shen, W., Tambe, M. and Yokoo, M. (2005), 'Adopt: Asynchronous distributed constraint optimization with quality guarantees', *Artificial Intelligence* **161**, 149–180.

Nair, R., Tambe, M., Yokoo, M., Pynadath, D. and Marsella, S. (2003), Taming decentralized POMDPs: Towards efficient policy computation for multi-agent settings, *in* 'Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)', Acapulco, Mexico, pp. 705–711.

Nash, J. (1950), Equilibrium points in n-person games, *in* 'National Academy of Sciences', Vol. 36, pp. 48–49.

Ng, A. Y., Harada, D. and Russell, S. (1999), Policy invariance under reward transformations: Theory and application to reward shaping, *in* 'Proceedings of the Sixteenth International Conference of Machine Learning (ICML)', Bled, Slovenia, pp. 278–287.

Noble, J. and Franks, D. W. (2004), 'Social learning in a multi-agent system', *Computing and Informatics* **22**(6), 561–574.

Ohko, T., Hiraki, K. and Anzai, Y. (1997), Addressee learning and message interception for communication load reduction in multiple robot environments, *in* 'ECAI '96: Selected papers from the Workshop on Distributed Artificial Intelligence Meets Machine Learning, Learning in Multi-Agent Environments', Springer-Verlag, Budapest, Hungary, pp. 242–258.

Oliehoek, F. and Vlassis, N. (2006), Dec-pomdps and extensive form games: equivalence of models and algorithms, Technical report, IAS-UVA-06-02 Informatics Institute, University of Amsterdam.

Owen, G. (1982), *Game Theory: Second Edition*, Academic Press.

Padhy, P., Dash, R., Martinez, K. and Jennings, N. R. (2006), A utility-based sensing and communication model for a glacial sensor network, *in* 'Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Utrect, The Netherlands, pp. 1353–1360.

Papadimitriou, C. H. and Roughgarden, T. (2004), Computing equilibria in multi-player games, *in* 'Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)', SIAM, pp. 82–91.

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

Paquet, S., Tobin, L. and Chaib-draa, B. (2005), An online pomdp algorithm for complex multiagent environments, *in* 'Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Utrect, The Netherlands, pp. 970–977.

Peshkin, L., Kim, K., Meuleau, N. and Kaelbling, L. (2000), Learning to cooperate via policy search, *in* 'Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence', San Francisco, USA, pp. 307–314.

Petcu, A. and Faltings, B. (2005), DPOP: A scalable method for multiagent constraint optimization, *in* 'Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)', Edinburgh, Scotland, pp. 266–271.

Pineau, J., Gordon, G. and Thrun, S. (2003), Point-based value iteration: An anytime algorithm for pomdps, *in* 'Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)', Hyderabad, India, pp. 1025 – 1032.

Pitt, J. and Mamdani, A. (1999), A protocol-based semantics for an agent communication language, *in* 'Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)', Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 486–491.

Poupart, P. and Boutilier, C. (2003), 'Bounded finite state controllers', *Advances in Neural Information Processing Systems* **16**, 823–829.

Puterman, M. L. (1994), *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley-Interscience.

Pynadath, D. V. and Tambe, M. (2002), Multi-agent teamwork: Analyzing the optimality and complexity of key theories and models, *in* 'Proceedings of the First Autonomous Agents and Multi-Agent Systems Conference (AAMAS)', Bologna, Italy, pp. 873–880.

Rogers, A., Dash, R., Jennings, N. R., Reece, S. and Roberts, S. (2006), Computational mechanism design for information fusion within sensor networks, *in* 'Proceedings of the Ninth International Conference on Information Fusion', Florence, Italy, pp. 1–7.

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

Rosenfeld, A., Kaminka, G. A. and Kraus, S. (2006), 'Adaptive robotic communication using coordination costs', *Proceedings of Distributed Autonomous Robotic Systems* **8**.

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

Roth, M., Simmons, R. and Veloso, M. (2005), Reasoning about joint beliefs for execution-time communication decisions, *in* 'Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Utrect, The Netherlands, pp. 786–793.

Sandholm, T., Gilpin, A. and Conitzer, V. (2005), Mixed-integer programming methods for finding nash equilibria, *in* 'The Twentieth National Conference on Artificial Intelligence (AAAI)', Pittsburgh, USA, pp. 495–501.

Schervish, M. J. (1995), *Theory of Statistics*, Springer.

Shannon, C. E. (1948), 'A mathematical theory of communication', *Bell System Technical Journal* **27**, 379–423.

Shen, J., Lesser, V. and Carver, N. (2001), Reasoning about remote data in cdps with distributed bayesian neworks, *in* 'Proceedings of the Multi-Agent Systems and Applications'.

Shen, J., Lesser, V. and Carver, N. (2003), Minimizing communication cost in a distributed bayesian network using a decentralized mdp, *in* 'Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Melbourne, Australia, pp. 678–685.

Simon, H. (1955), 'A behavioral model of rational choice', *The Quarterly Journal of Economics* **69**(1), 99–118.

Szer, D. and Charpillet, F. (2005), An optimal best-first search algorithm for solving infinite horizon dec-pomdps, *in* 'Proceedings of the Sixteenth European Conference on Machine Learning (ECML)', Porto, Portugal, pp. 389–399.

Tambe, M. (1997), 'Towards flexible teamwork', *Journal of Artificial Intelligence Research* **7**, 83–124.

Tambe, M., Bowring, E., Jung, H., Kaminka, G., Maheswaran, R., Marecki, J., Modi, J., Nair, R., Pearce, J., Paruchuri, P., Pynadath, D., Scerri, P., Schurr, N. and Varakantham, P. (2005), Conflicts in teamwork: Hybrids to the rescue, *in* 'Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Utrect, The Netherlands, pp. 3–10.

Washington, R. (1997), Bi-pomdp: Bounded, incremental, partially-observable markov-model planning, *in* 'Proceedings of the Fourth European Conference on Planning (ECP)', Toulouse, France, pp. 440–451.

Watson, N. R., John, N. W. and Crowther, W. J. (2003), Simulation of unmanned air vehicle flocking, *in* 'Proceedings of the Theory and Practice of Computer Graphics (TPCG)', Washington, DC, USA, p. 130.

Weiss, G., ed. (1999), *Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, Cambridge, MA, USA.

Wiewiora, E., Cottrell, G. W. and Elkan, C. (2003), Principled methods for advising reinforcement learning agents, *in* 'Proceedings of the Twentieth International Conference on Machine Learning (ICML)', Washington, DC, USA, pp. 792–799.

Wooldridge, M. (2002), *An Introduction to Multi-Agent Systems*, Wiley.

Xuan, P. and Lesser, V. (2002), Multi-agent policies: From centralized ones to decentralized ones, *in* 'Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Bologna, Italy, pp. 1098–1105.

Xuan, P., Lesser, V. and Zilberstein, S. (2001), Communication decisions in multi-agent cooperation: Model and experiments, *in* 'Proceedings of the Fifth International Conference on Autonomous Agents', Montreal, Canada, pp. 616–623.

Yen, J., Fan, X. and Volz, R. A. (2004), 'Information needs in agent teamwork', *Web Intelli. and Agent Sys.* **2**(4), 231–247.

Yokoo, M., Durfee, E. H., Ishida, T. and Kuwabara, K. (1998), 'The distributed constraint satisfaction problem: Formalization and algorithms', *Knowledge and Data Engineering* **10**(5), 673–685.

Yokoo, M. and Hirayama, K. (2000), 'Algorithms for distributed constraint satisfaction: A review', *Autonomous Agents and Multi-Agent Systems* **3**(2), 185–207.

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

Zhang, Y., Volz, R. A., Loerger, T. R. and Yen, J. (2004), A decision-theoretic approach for designing proactive communication in multi-agent teamwork, *in* 'SAC '04: Proceedings of the 2004 ACM symposium on Applied computing', Nicosia, Cyprus, pp. 64–71.

Zilberstein, S. and Goldman, C. V. (2003), Optimizing information exchange in cooperative muti-agent systems, *in* 'Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)', Melbourne, Australia, pp. 137–144.