OPTIMAL SEQUENTIAL PLANNING IN PARTIALLY OBSERVABLE MULTIAGENT SETTINGS

BY

PRASHANT J. DOSHI
M.S., Drexel University, 2001
B.E., University of Mumbai, 1999

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2005

Chicago, Illinois.

Dedicated to my wife, Shweta, and my parents for their constant and boundless love and support

ACKNOWLEDGEMENTS

I would like to acknowledge the contributions of my advisor, Piotr Gmytrasiewicz, in making this work possible. Of all the roles he plays for his students, his role of being a visionary has benefited me the most. Not only did he envision this research, and put the initial building blocks together, then onwards, he has guided me diligently, pointing out the right issues that must be addressed while always keeping the goal in mind. I would also like to thank Lloyd Greenwald for steering me during my initial months as a researcher, and being an excellent collaborator since then. His advice and encouragement provided me with the early inspiration. And of course, I would like to thank my thesis committee members, Peter Nelson, Gyorgy Turan, Bing Liu, and Avi Pfeffer for taking time out to read this thesis and provide me with their valuable suggestions for improving it. I would also like to mention the graduate student affairs staff of the CS department at UIC, especially Santhi Nannapaneni. She made the typical bureaucratic hassles almost invisible for the graduate students of the department.

I would also like to mention my colleagues in the MAS group, Bharanee, Kyle, and Nimesh. They made the weekly group meetings fun and intellectually challenging. Many thanks to Bharanee for reading almost the entire thesis in a day, and providing me some good tips on improving its clarity. A word of thanks is also due to Jack and Xin who made research in the AI Lab more enjoyable.

A note of appreciation is due to Richard Goodwin, manager of the Semantic eBusiness Middleware group at IBM's T. J. Watson Research Center. He was an excellent mentor during my two summer stints at IBM. He ensured that in addition to working, I also took time out to enjoy the summers. I would also like to mention the other members of the group: Rama Akkiraju, Juhnyoung Lee, Kunal Verma, and Sascha Roeder. Rama was and still remains an excellent collaborator. I still remember the endless conversations I had with Kunal and Sascha – some technical and some philosophical.

My parents' constant and unquestioning love and support played a key role in making this thesis possible. Finally, I would like to thank my wife, Shweta. She took care of everything else, so that I could concentrate solely on my thesis. She left no stone unturned in broadening my outlook toward life beyond the narrow confines of my research. Thank you for always being by my side.

PJD

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACC	Absolute Continuity Condition
DBN	Dynamic Bayesian Network
I-PF	Interactive Particle Filter
I-POMDP	Interactive Partially Observable Markov Decision Process
LP	Linear Program
MDP	Markov Decision Process
NI	Numerical Integration
PF	Particle Filter
POMDP	Partially Observable Markov Decision Process
POSG	Partially Observable Stochastic Game
PWLC	Piecewise Linear and Convex
SB	Sample based
RTS	Reachability Tree Sampling

SUMMARY

We develop a framework for sequential optimality of an autonomous agent interacting with other agents within a common, and possibly uncertain, environment. We use the normative paradigm of decision-theoretic planning under uncertainty as formalized by partially observable Markov decision processes (POMDPs) as a foundation. The new framework called *interactive POMDP* (I-POMDP) generalizes a POMDP to multiagent settings. I-POMDPs are applicable to autonomous self-interested agents who locally compute what actions they should execute to optimize their preferences given what they believe while interacting with others with possibly conflicting objectives. I-POMDPs ascribe models that are similar to types as used in Bayesian games, to other agents. Some of these models describe other agents in terms of their beliefs, capabilities, and preferences. Consequently, I-POMDPs replace the "flat" beliefs of POMDPs with nested hierarchical belief systems. Our approach of using a decision-theoretic framework and solution concept complements the equilibrium approach of analyzing interactions, as used in classical game theory. Specifically, we avoid the difficulties of non-uniqueness and incompleteness of the traditional Nash equilibrium approach, and offer solutions which are likely to be better than the solutions obtained from applying traditional POMDPs to multiagent settings. But these advantages come at the cost of processing and maintaining possibly infinitely nested interactive beliefs. We define a class of finitely nested I-POMDPs to form a basis for computable approximations to the infinitely nested ones. We show that a number of properties that facilitate solutions of POMDPs carry over to finitely nested I-POMDPs.

Analogous to POMDPs, optimal solutions to I-POMDPs are difficult to compute due to two sources of intractability: First is the complexity of the belief representation, sometimes called the *curse of dimensionality*, and the second is the complexity of the space of policies, also called the *curse of history*. The curse of dimensionality is especially acute for I-POMDPs because the beliefs may include beliefs about the physical environment, and possibly the agent's beliefs about other agents' beliefs, their beliefs about others, and so on. To address the curse of dimensionality, we resort to sampling methods which are typically immune to the high dimensionality of the underlying space. We adapt the particle filter, specifically, the bootstrap filter, to the multiagent setting, resulting in the *interactive particle filter* (I-PF). Mirroring the hierarchical character of interactive beliefs, the I-PF involves nested sampling and propagation at each of the hierarchical levels of beliefs. Our approximation method is applicable to agents that start with a prior belief and optimize

over finite horizons, and therefore finds applications for online plan computation. We derive error bounds of our approach, and empirically demonstrate its performance on simple test problems. In order to mitigate the curse of history, we present a complementary method based on sampling observations while building the look ahead reachability tree during value iteration.

Finally, we theoretically analyze the interactions taking place between agents participating in the infinite horizon partially observable stochastic game as formalized within the I-POMDP framework. Under the assumption of compatibility of agents' prior beliefs about future observations with the true distribution induced by the actual strategies of all agents, we show that the behavior of agents converges to the *subjective equilibrium*. Subjective equilibrium is stable with respect to learning and optimization. In trying to empirically validate the existence of subjective equilibrium, we run into obstacles. The difficulties arise because we are unable to guarantee the satisfiability of the truth compatibility condition, in practice. We believe that the practical difficulty for I-POMDPs to reach the subjective equilibrium also signifies a serious impediment to adopting equilibrium as a solution concept for multiagent planning.

Keywords: multiagent planning, interactive POMDPs, interactive belief systems, types, approximation methods, interactive particle filters, absolute continuity, subjective equilibrium

Chapter 1

INTRODUCTION

PLANNING in complex environments is receiving significant and sustained attention by the research community. One reason for the continuous focus is its myriad applications. The applications of planning are pervasive and affect all sections of the human society:

- **Security:** Planning may be used to coordinate troop movements in battlefields, anti-air missile defense units (Noh & Gmytrasiewicz, 2001), and search and rescue missions.
- **Health:** Planning algorithms may be used to formulate the course of a long-term treatment in an interacting multi-treatment therapy (Hauskrecht, 1997).
- **Sociology:** Planning frameworks may serve as computational models of mechanisms that generate anthropomorphic social behaviors such as trust and follow the leader, rebellion, and cultural traditions.
- **Economics:** Planning may be used to study the long term consequences of market reforms, generate strategies for entry-level companies in established markets, and establish long-term profit-making market prices for merchandise.

The aforementioned applications reveal several characteristics of environments, which planning algorithms must handle. While coordinating troop movements in battlefields, the exact ”ground situation” is seldom known. Additionally, actions of adversaries are usually only partially perceivable. Rescue bots will

be equipped with sensors and actuators that typically exhibit noisy readings or fail completely. In multi-treatment therapies, we hardly ever know all the side effects of any treatment. In order to handle such traits, we are motivated to develop planning algorithms that model and reason with uncertainty.

Mathematicians, economists, and other researchers have often taken recourse to probability theory to model and reason with uncertainty. Probability theory, founded on Kolmogorov's axioms of probability, has also become the *de-facto* tool for artificial intelligence researchers to tackle the uncertainty inherent in the real world. In many situations, we are confronted with several feasible candidate plans from which we must choose. Choice is usually governed by our preferences or rewards. We model the preferences using Von Neumann and Morgenstern utilities as governed by the axioms of utility theory. For the sake of simplicity, probability and utility theories have been unified into a single field called *decision theory*. The main focus of this thesis is on decision-theoretic planning.

1.1 Design of Planning Agents

We adopt a working model of the desicion-theoretic planning problem, illustrated in Fig. 1.1, that consists of two primary components:

- Environment
- Agent(s) – computational device(s) that act, perceive, and reason.

Agent(s) interact with the environment in a *sequential* manner – agent(s)' actions cause changes in the *state* of the problem. The state of the planning problem encompasses all information relevant to the agent(s)' decision making process. For e.g. in a *maze* problem, where a robot must navigate a maze filled with pits, the location of the robot and the pits constitute the state of the problem. As part of the agent(s)' interaction with the environment, the agent(s) also observes events that inform it of the state of the problem – possibly unreliably – and perhaps, even the actions of the other agent(s). In Fig. 1.1, we illustrate the model of the planning problem.

The task of the planning agent(s) is to choose actions that optimize its preferences over the long term, thereby producing a plan that achieves the agent(s)' goals in an optimal manner. In doing so, the agent(s) must appropriately address the **action outcome uncertainty** and the **state uncertainty** (also known as the partial observability) problems that were mentioned before.

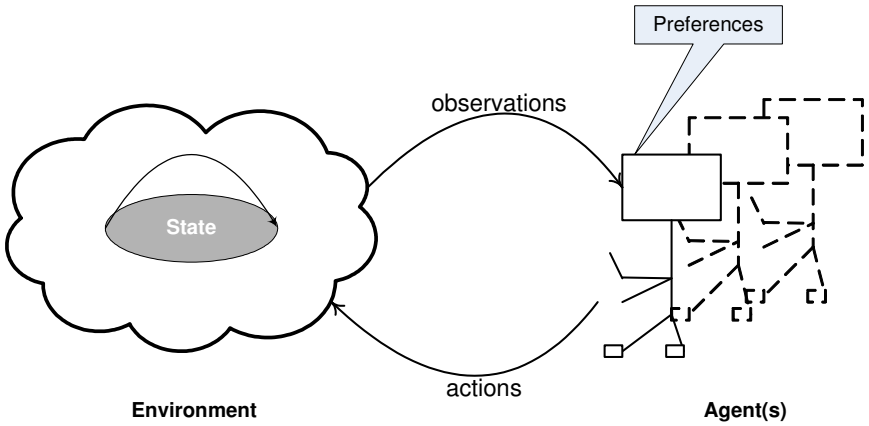

Figure 1.1: Model of the planning problem. Agent(s) interact with the environment through a series of actions and observations. The task for the agent is to select actions that are optimal over the long term with respect to its preferences.

For the sake of our arguments, we assume that the agent(s) has access to sufficient time and memory resources. Where appropriate, the computational time and memory requirements will be mentioned. Of course, realistically, agents may have limited resources at their disposal. In this regard, approximation techniques that trade-off computations with quality of the plans play a key role and also form the subject of this dissertation.

1.2 Planning in Uncertain Single Agent Settings

In order to plan in a single agent setting, we represent the planning problem computationally as a Markov decision process (MDP) (Puterman, 1994; Russell & Norvig, 2003). An MDP is a tuple consisting of four parameters:

Definition 1.1 (Markov decision process (MDP)). *A Markov decision process for an agent, say i, is*

$$MDP_i = \langle S, A_i, T_i, R_i \rangle$$

where:

- S is the set of states of the planning problem
- A_i is the set of agent i's actions

- $T_i : S \times A_i \times S \rightarrow [0, 1]$ is the transition function that captures the dynamics of the environment. It describes the possibly uncertain effects of i's actions on the states of the planning problem. Note that the transition function also encodes the *Markovian* assumption: The next state of the problem depends only on the current state and the agent's actions (and not on the history of previous states)

- $R_i : S \times A_i \rightarrow \mathbb{R}$ is the agent's reward function that expresses its preferences [1].

As we mentioned before, the agent's plan must optimize over its preferences. There are three types of optimality criteria that may be followed:

1. A *finite horizon* criterion according to which the agent maximizes the expected sum of the next H rewards – $E(\sum_{t=0}^{H} r_{i,t})$.

2. An *infinite horizon with discounting* criterion according to which the agent maximizes the expected discounted sum of infinite rewards – $E(\sum_{t=0}^{\infty} \gamma^t r_i)$, where $0 < \gamma < 1$.

3. Finally, an *infinite horizon with averaging* criterion according to which the agent maximizes the expected average of infinite rewards – $\lim_{h \rightarrow \infty} E(\frac{1}{h} \sum_{t=0}^{h} r_{i,t})$.

Of these criteria, we will focus on the finite horizon and infinite horizon with discounting, but our arguments can immediately be extended to the infinite horizon with averaging criterion, if required.

In order to solve MDPs and generate the optimal plan(s), we assume that the state is fully and reliably observable by the agent. There are three standard methods for solving MDPs: The method most commonly employed is *dynamic programming* using a technique called **value iteration**. MDPs can also be solved by solving a system of simultaneous linear equations using a technique called **policy iteration**, and through linear programming. Below we briefly describe value iteration, and refer the reader to (Russell & Norvig, 2003) for policy iteration, and (Littman, Dean, & Kaelbling, 1995b) for linear programming.

We associate with each state a payoff that reflects the maximum long term expected reward that the agent can accumulate beginning in that state. We call the function that associates the payoff a *value function*, and denote it as: $U : S \rightarrow \mathbb{R}$. The value function for the infinite horizon with discounting criterion satisfies the following:

[1] For some problems, the rewards may also depend on the resulting state. In this case, the reward function may be defined as, $R_i : S \times A_i \times S \rightarrow \mathbb{R}$

$$U(s) = \max_{a \in A_i} \left\{ R_i(s,a) + \gamma \sum_{s' \in S} T_i(s,a,s') U(s') \right\} \tag{1.1}$$

Equation 1.1 often called the Bellman update, embodies Bellman's Principle of Optimality – sub-solutions of a globally optimal solution are themselves globally optimal. Value iteration proceeds by initializing the value function with arbitrary values, and then repeatedly executing Eq. 1.1, until the change in the value assigned to each state is negligibly small. Note that the value function is guaranteed to converge to a fixed point, U^* after repeated iterations (Russell & Norvig, 2003). Once the value function converges, or the required number of iterations have been performed, the optimal action, a_i^*, for a state is an element of the set of optimal actions for that state, $OPT(s)$:

$$OPT(s) = \underset{a \in A_i}{argmax} \left\{ R_i(s,a) + \gamma \sum_{s' \in S} T_i(s,a,s') U^*(s') \right\}$$

We now turn our attention to planning problems where the current state of the situation cannot be reliably observed. A well known extension of the MDP framework to such partially observable settings is called the partially observable Markov decision process (POMDP) (Smallwood & Sondik, 1973; Cassandra, Kaelbling, & Littman, 1994; Hauskrecht, 1997). POMDPs address both, the action outcome uncertainty as well as the state uncertainty problems that were mentioned in Section 1.1. Similar to MDPs, the action outcome uncertainty is captured in the transition function; the state uncertainty is addressed through the use of information structures called *beliefs*. Beliefs are probability distributions over the states of the problem, and represent the agent's uncertainty about its true state. Beliefs also exhibit a very useful result: When updated in a Bayesian manner, they summarize an agent's entire observation history. [2] Because POMDPs constitute important background material for the topic of this thesis, they are described in detail in the next chapter.

1.3 Planning in Uncertain Multiagent Settings

In addition to its own action outcome uncertainty and state uncertainty, an agent in a multiagent setting must also contend with uncertainty over the other agents' actions. The dynamics as well as the payoffs in a problem are usually influenced by actions of all the agents, therefore optimal behavior of each agent depends

[2] This is the reason why POMDPs are sometimes referred to as belief-based MDPs.

on the behaviors of the other agents. Realistically, others' actions are not perfectly observable, therefore we base the agent's behavior on its expectation of others' actions.

The analysis of multiagent interactions has long been the center of attention of game theorists and economists. Starting with single play games, attention has shifted to the same game played repeatedly infinite number of times – infinitely repeated games – and repeated games with state transitions – stochastic games. In all these frameworks, **Nash equilibrium** has been and remains the solution concept of choice. However, Nash equilibrium as a solution paradigm suffers from two limitations:

- Non-uniqueness: In several decision-making problems, multiple Nash equilibria exist. This necessitates the utilization of an external synchronizing mechanism to ensure that the same Nash equilibrium is followed or reached by all agents.
- Incomplete: Nash equilibria do not specify an action when the agent believes that others may not act according to equilibria.

Additionally, as pointed out by Binmore (1990), researchers have an inadequate understanding of the intermediate stages of a game before Nash equilibrium is reached. Finally, in order to use Nash equilibrium as a solution, we must assume *common knowledge* of agent payoffs and beliefs. [3] Because of these limitations, the inadequacy of Nash equilibrium as a solution concept for planning is gradually seeping into mainstream thinking (Shoham, Powers, & Grenager, 2003; Russell & Norvig, 2003). To illustrate this viewpoint, we quote Russell and Norvig (2003) *ad verbatim*,

"... game theory has been used primarily to analyze *environments that are at equilibrium, rather than to* control *agents within an environment."*

The limitations of Nash equilibrium suggest the need for a different solution paradigm for planning in multiagent settings. We adopt the agent's best response to its state of knowledge as a solution concept. The agent's knowledge includes its knowledge about the current physical state of the problem as well as the expected actions of the other agents. This approach is also called the decision-theoretic approach to game theory (Kadane & Larkey, 1982). Our solution concept is complete – if the agent believes that others will act according to some Nash equilibrium, then it too will act out its part of the equilibrium, but if others choose to

[3] In some cases, mutual knowledge suffices to adopt a Nash equilibrium (see Aumann & Brandenburger, 1995).

diverge from their equilibrium behaviors, then the agent will perform its best response. The solution is also unique upto plans of equal expected utility, from which any one can be picked.

To predict others' actions, the agent utilizes models of other agents' behaviors. These models range from being naive - static probability distributions over actions - to sophisticated ones that ascribe to the other agent beliefs and rationality in action selection. When the sophisticated models are used to represent other agents, the agent's belief is a hierarchical or nested belief system. Such belief systems have been studied before in game theory and in theoretical computer science (Mertens & Zamir, 1985; Brandenburger & Dekel, 1993; Fagin, Halpern, Moses, & Vardi, 1995; Heifetz & Samet, 1998; Aumann, 1999) but never employed for sequential decision making. There are also models that lie between the naive and sophisticated ones mentioned above. These models are mappings from an agent's observation history to a distribution over its actions. An example of such a model is a finite state controllers. In this thesis, we focus on the sophisticated and sub-sophisticated models, demonstrate the usefulness of sophisticated models in comparison to the naive models, and utilize the hierarchical beliefs for sequential decision making. Furthermore, in contrast to Nash equilibrium, our approach does not require common knowledge of the true agent models or of beliefs.

1.4 Claims and Contributions

In the previous sections, we introduced the background and the foundational concepts relevant for this thesis. In this section, we present the primary claims and contributions of this work towards advancing the subject. This dissertation contributes significantly to existing pertinent literature in the fields of Artificial Intelligence and Economics. We group our contributions into three categories:

1.4.1 Framework

We present a new framework, called **interactive POMDPs** (Gmytrasiewicz & Doshi, 2005, 2004, 2004), for sequential rationality of autonomous agents interacting with other agents within a common, possibly uncertain, environment. We use the normative paradigm of decision-theoretic planning under uncertainty as a point of departure, and infuse it with notions from game theory to construct a framework for optimal multiagent planning. Our formalism is applicable to autonomous self-interested agents who locally compute what actions they should execute to optimize their preferences given what they believe while interacting with others with possibly conflicting objectives. We list the main contributions of this framework below:

- Interactive POMDPs generalize POMDPs to multiagent settings. When there is only a single planning agent, they reduce to the traditional POMDP planning framework.
- Interactive POMDPs adopt a solution paradigm centered on optimality and best response to anticipated actions of other agents. This solution approach addresses the shortcomings of non-uniqueness and incompleteness of equilibria-based solutions.
- Interactive POMDPs replace the "flat" beliefs in traditional decision-theoretic planning frameworks with interactive hierarchical ones that represent beliefs about others' beliefs and their beliefs about others'. This construction unifies long-term as well as strategic planning into a single framework.
- In contrast to other multiagent planning frameworks, interactive POMDPs are applicable to both cooperative and non-cooperative settings between agents.
- Interactive POMDPs are applicable to problems that are populated by both, sophisticated rational agents and "dumb" – possibly irrational – agents.

1.4.2 Algorithms and Analysis

The advantages of interactive POMDPs over traditional approaches come at a cost of processing and maintaining a possibly infinitely nested interactive belief system. We define a class of *finitely nested* interactive POMDPs to form a basis for computable approximations to the infinitely nested ones. Pertaining to this class of interactive POMDPs, we report the following contributions:

- We demonstrate that interactive POMDPs generate plans that are atleast as good, and typically better in value as compared to plans generated by applying the traditional POMDPs to multiagent settings (Gmytrasiewicz & Doshi, 2005).
- We illustrate solutions for several different non-cooperative and cooperative versions of the multiagent tiger problem modeled within the finitely nested interactive POMDP framework.
- We develop an online anytime approximation algorithm to address the prohibitive computational complexity of solving interactive POMDPs. Our algorithm addresses the curse of dimensionality by using the *interactive particle filter* (Doshi & Gmytrasiewicz, 2005b, 2005a) – a generalization of the traditional particle filter to the multiagent setting. The interactive particle filter reduces to the traditional

particle filter in single agent settings. We bound the error introduced by the approximation technique, and report on the computational savings.

- We develop a complementary approximation technique based on sampling the look ahead reachability tree that is constructed during value iteration. When combined with the previously mentioned approximation method, our approach reduces the impact of both the curses of dimensionality and history. We analyze the empirical performance of the approximation technique using the multiagent tiger and the multiagent machine maintenance problems.

- We theoretically analyze the infinite horizon play of agents in the interactive POMDP framework, and show that the play eventually converges to a *subjective* equilibrium that is stable with respect to learning and optimization. We point out some computational obstacles in empirically validating the equilibrium result.

1.4.3 Applications

As we mentioned before, the applications of multiagent planning are vast, and pervade all sections of the human society. In this thesis, we concentrate on the emerging area of human and social dynamics, for applications. We use interactive POMDPs to perform agent based simulation of anthropomorphic social behaviors. By successfully demonstrating through application of the framework, occurrence of human social behaviors or patterns, we achieve multiple objectives: We establish that the commonly observed behaviors are rational in regards to their respective settings. Our results serve to validate the framework as an important tool for explaining rational interactions in uncertain multiagent dynamic settings. Finally, and of key importance, the application will pave the way for deployment of the framework to new settings where rational social behavior has neither been established nor observed.

1.5 Structure of the Thesis

Our framework unites many concepts from game theory and decision theory to enable multiagent planning. Therefore, naturally, this dissertation surveys vast literatures addressing both game-theoretic decision-making and decision-theoretic planning frameworks, in addition to presenting the new framework for planning in multiagent settings. This document is structured so that we first present the background and related

work in the early chapters, and then introduce the new framework, exact and approximate algorithms in the subsequent chapters. Below, we briefly summarize rest of the chapters in this document.

In *Chapter 2*, we first present the well-known single agent decision-theoretic planning framework called partially observable Markov decision process (POMDP). Next, an exact method for solving POMDPs, that will be extended to a multiagent setting later, is presented. In the final section of the chapter, we briefly survey approximation techniques that reduce the computational complexity of solving POMDPs at the expense of solution quality. These techniques include approaches that exploit the structure of the problem, as well as approaches that address bottlenecks and bound the error introduced because of the approximation.

In *Chapter 3*, we succinctly survey the game- theoretic frameworks that address multiagent decision-making. We cover single play games involving complete and incomplete information, repeated games with incomplete information, and stochastic games. We review the learning algorithms that converge to Nash equilibrium in repeated and stochastic games, and discuss their shortcomings.

The new framework for planning in uncertain multiagent settings, interactive POMDP, is introduced in *Chapter 4*. We present the definition, properties, and value iteration for solving interactive POMDPs. Proofs of the properties are presented in *Appendix A*. Using the example of the multiagent tiger problem, we illustrate the concepts involved in the interactive POMDP framework, and present example solutions for cooperative and non-cooperative versions of the game.

In *Chapter 5*, we present solutions for several versions of the multiagent tiger problem. We group the different versions into two categories: the non-cooperative setting and the cooperative setting. We give value functions and policy trees for these settings, and also uncover simple behavioral insights.

Analogous to POMDPs, interactive POMDPs are also computationally prohibitive. Therefore, in *Chapter 6* we present an anytime method to compute approximately optimal plans while consuming less time and space. Our method approximates the interactive POMDP state estimation by extending the well-known technique of particle filtering to a multiagent setting. The reduction in quality of the solution as introduced by our approximation technique is bounded. The performance of the approximation method on two examples , the multiagent tiger and the multiagent machine maintenance problem (see *Appendix B*), is demonstrated. We also complement the interactive particle filter with a method that mitigates the policy space complexity.

In *Chapter 7*, we prove that agents' behaviors within the interactive POMDP framework converge to an

equilibrium. This equilibrium, called a subjective equilibrium, is the subjective counterpart of the objective Nash equilibrium, and is a natural consequence of the convergence of Bayesian learning in interactive POMDPs. We also point out computational limitations in achieving this equilibrium.

Finally, we conclude this thesis in *Chapter 8* with a brief summary of the important contributions of this work. We also lay out avenues of future work, which include supplementing the existing suite of algorithms for solving interactive POMDPs, and exploring multiagent planning when the agents are boundedly rational.

Chapter 2

BACKGROUND: SINGLE AGENT DECISION-THEORETIC PLANNING

MARKOV decision processes, briefly introduced in Section 1.2 of Chapter 1, are planning frameworks for single agent settings in which the physical state of the problem is always fully known to the agent. Because of this reason, they are sometimes also called fully observable Markov decision processes (FOMDPs). However, real-world sensors are typically noisy, thereby precluding complete observability of the state. An extension of the MDP, which addresses the state uncertainty problem also exists and is called the partially observable Markov decision process (POMDP) (Smallwood & Sondik, 1973; Cassandra et al., 1994; Hauskrecht, 1997).

POMDPs incorporate a new element into the definition of MDPs. They require the knowledge of a signaling function, called the *observation* function, that gives partial information about the state to the agent via the signals that an agent receives. POMDPs combine expected utility maximization with a psycho-cognitive process called the *belief update*. Because POMDPs form an important foundation on which our multiagent planning framework is based, we explore them in some detail. In Section 2.1, we formally define POMDPs, and present important properties of the framework. In Section 2.2, we briefly survey the principal exact methods of solving them, and present one such method (that we will use later) in detail. A simple toy problem called the single agent tiger problem is cast as a POMDP, and its solutions are shown. We also survey the approximation techniques for POMDPs in Section 2.3.

2.1 Partially Observable Markov Decision Processes (POMDPs)

POMDPs are one of the most general computational frameworks available for planning in uncertain single agent problem settings. They take into consideration the action outcome uncertainty as well as the state uncertainty problems that were mentioned in the previous chapter. However, their generality comes at a price – the computational cost of solving POMDPs is enormous, precluding their applications to all but the simplest settings.

2.1.1 Definition

A POMDP is typically defined as a six parameter tuple. The six parameters together capture all aspects of the decision making situation.

Definition 2.1 (POMDP). *A partially observable Markov decision process for an agent, say i, is:*

$$POMDP_i = \langle S, A_i, \Omega_i, T_i, O_i, R_i \rangle$$

where:

- S is the set of physical states of the environment
- A_i is the set of possible actions of i
- Ω_i is the set of possible observations of i
- $T_i : S \times A_i \times S \rightarrow [0, 1]$ is the transition function (similar to the one in MDPs), and represents the dynamics of the problem
- $O_i : S \times A_i \times \Omega_i \rightarrow [0, 1]$ is the observation function. O_i gives the probability distribution over the possible observations for each state-action pair
- $R_i : S \times A_i \rightarrow \mathbb{R}$ represents the agent's preferences or rewards.

The optimality criteria used for planning are the same as those used for the MDPs. In order to plan when the exact physical state of the environment is unknown, an agent maintains a *belief*. Beliefs – psychological constructs – are mathematically formalized as probability distributions over the physical states. Beliefs,

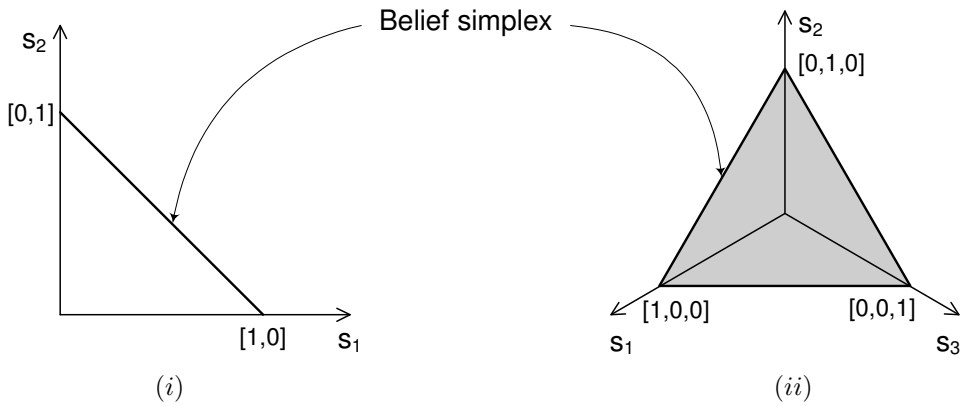

Figure 2.1: We show the belief simplices for (i) two states, and (ii) three states. Note that the belief simplices are geometric shapes of dimensions one less than the number of states. The coordinates of each point within the belief simplex sum to 1.

sometimes also called information states (Hauskrecht, 1997) or partitions (Aumann, 1999), give the likelihood with which the agent thinks that it's in a particular state of the environment. The space of beliefs is called a belief simplex; we illustrate the belief simplices for two and three states in Fig. 2.1. Computing a plan in the POMDP framework involves two steps:

1. **Belief Update:** Beliefs compactly represent all information available to the agent at the time of selection of the optimal action. Specifically, the belief succinctly represents the entire history of actions and observations as perceived by the agent. Formally:

$$b_i^t(s^t) = Pr(s^t | o_i^t, a_i^{t-1}, o_i^{t-1}, \ldots, o_i^1, a_i^0)$$

In order to plan optimally, the agent must continuously update its belief conditioned on the action it performs, a_i^{t-1}, and the observation, o_i^t, it perceives. The new belief, b_i^t, is computed using the Bayesian updating process abbreviated as $SE(b_i^{t-1}, a_i^{t-1}, o_i^t)$:

$$\begin{aligned} b_i^t(s^t) = \quad & Pr(s^t | o_i^t, a_i^{t-1}, b_i^{t-1}) \\ & = \frac{O_i(s^t, a_i^{t-1}, o_i^t) \sum_{s^{t-1} \in S} T_i(s^{t-1}, a_i^{t-1}, s^t) b_i^{t-1}(s)}{Pr(o_i^t | a_i^{t-1}, b_i^{t-1})} \\ & = \beta O_i(s^t, a_i^{t-1}, o_i^t) \sum_{s^{t-1} \in S} T_i(s^{t-1}, a_i^{t-1}, s^t) b_i^{t-1}(s^{t-1}) \end{aligned} \tag{2.1}$$

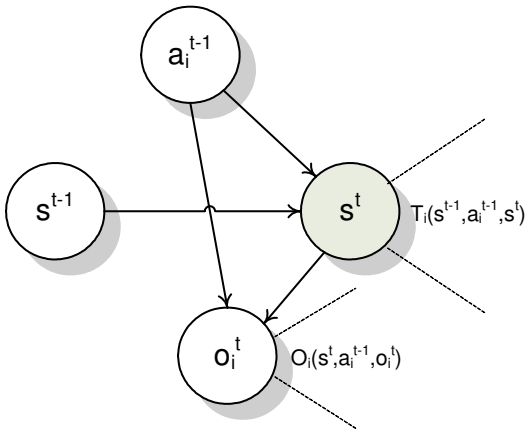

Figure 2.2: A 2-time slice DBN that graphically illustrates the belief update in POMDPs. The posterior belief is a distribution over the shaded random variable. The dotted lines enclose the functions that form CPTs for the respective random variables.

where β is a normalizing factor. A dynamic Bayesian network (DBN) (see Section 15.5 in Chapter 15 of Russell & Norvig, 2003, for information on DBNs) that graphically captures the updating process is given in Fig. 2.2.

2. **Policy Computation:** The solution of a POMDP produces a policy – conditional plan – that is a mapping from any belief state to the optimal distribution over the actions that must be performed in that belief state: $\pi^* : \mathcal{B}_i \rightarrow \Delta(A_i)$, where $\mathcal{B}_i$ is the agent's belief simplex (space of all beliefs), and $\Delta(\cdot)$ is the space of all probability distributions. Because a belief state compactly represents an agent's observation history, the policy may also be seen as a mapping from the agent's observation history to a distribution over its actions: $\pi^* : H_i \rightarrow \Delta(A_i)$, where H_i is the agent's observation history. In order to produce the policy, analogously to MDPs, we construct a value function that associates with each belief, a payoff that reflects the maximum long term reward that the agent can gain starting from that belief state. The value function is defined as: $U : \mathcal{B}_i \rightarrow \mathbb{R}$. In a manner analogous to MDPs, we will use **value iteration** to derive the value function, and thereafter the optimal policy. The Bellman equation for the discounted infinite horizon optimality criterion is shown:

$$U(b_i^t) = \max_{a \in A_i} \left\{ \rho_i(b_i^t, a) + \gamma \int_{b_i^{t+1} \in \mathcal{B}_i} Pr(b_i^{t+1} | b_i^t, a) U(b_i^{t+1}) db_i^{t+1} \right\} \tag{2.2}$$

where $\rho_i(b_i^t, a) = \sum_{s \in S} R_i(s, a_i^t) b_i^t(s)$.

Value iteration works because, as we show in the next subsection, the sequence of value functions, $\{U^n\}$, where n is the horizon, converges to a unique fixed point, i.e. $\lim_{n\to\infty} U^n = U^*$. Observing that there are only as many next belief states as there are observations ($|\Omega_i|$) for a particular action, allows us to write Eq. 2.2 more compactly. We can rewrite it by summing over all observations in the second term:

$$U(b_i^t) = \max_{a\in A_i}\left\{\rho_i(b_i^t, a) + \gamma \sum_{o_i^{t+1}\in\Omega_i} Pr(o_i^{t+1}|b_i^t, a)U(SE(b_i^t, a, o_i^t))\right\} \tag{2.3}$$

The set of all optimal actions, OPT, for a particular belief state using the infinite horizon with discounting criterion, is then calculated as:

$$OPT(b_i^t) = \operatorname*{argmax}_{a\in A_i}\left\{\rho_i(b_i^t, a) + \gamma \sum_{o_i^{t+1}\in\Omega_i} Pr(o_i^{t+1}|b_i^t, a)U^*(SE(b_i^t, a, o_i^t))\right\}$$

2.1.2 Properties

The value function exhibits several well-known properties that are central to solving POMDPs. We will first show that value iteration converges to a unique fixed point, and then prove that the value function is always piecewise linear and convex. These properties form the basis for solving POMDPs exactly, and generating and representing the optimal policies. Later on, we will extend these properties to the multiagent planning framework.

We start by defining the value function formally:

Definition 2.2 (Value Function). *A value function is defined as the mapping, $U : \mathcal{B}_i \rightarrow \mathbb{R}$ where $\mathcal{B}_i$ is the set of all i's beliefs in the belief simplex. The value function is real-valued and bounded. Let $B(\mathcal{B}_i)$ be the set of bounded real-valued value functions defined on $\mathcal{B}_i$.*

Let $U^n \in B(\mathcal{B}_i)$ be the n-horizon value function. The n-horizon value function satisfies the following equation:

$$U^n(b_i^t) = \max_{a\in A_i}\left\{\rho_i(b_i^t, a) + \gamma \sum_{o_i^{t+1}\in\Omega_i} Pr(o_i^{t+1}|a, b_i^t)U^{n-1}(SE(b_i^t, a, o_i^{t+1}))\right\} \quad \forall b_i^t \in \mathcal{B}_i \tag{2.4}$$

where $U^1(b_i^t) = \underset{a \in A_i}{max} \rho_i(b_i^t, a)$

Let,

$$h(b_i^t, a, U) = \rho_i(b_i^t, a) + \gamma \sum_{o_i^{t+1} \in \Omega_i} Pr(o_i^{t+1}|a, b_i^t) U(SE(b_i^t, a, o_i^{t+1}))$$

then define

$$H^a U^{n-1}(b_i^t) = h(b_i^t, a, U^{n-1})$$

and,

$$HU^{n-1}(b_i^t) = \underset{a \in A_i}{max} \; H^a U^{n-1}(b_i^t)$$

From Eq. 2.4 it follows,

$$U^n = HU^{n-1}$$

We will label H as the **backup** operator, $H : B(\mathcal{B}_i) \rightarrow B(\mathcal{B}_i)$. Let us get acquainted with the properties of the backup operator.

Lemma 2.1 (Isotonicity). *The backup operator, H, is an* **isotonic** *mapping. Formally, let* $V, U \in B(\mathcal{B}_i)$*, then if* $V(b_i) \leq U(b_i) \quad \forall b_i \in \mathcal{B}_i$*, denoted* $V \leq U$*, then* $HV \leq HU$*.*

The next property of H is of importance since it allows us to derive an important property of the value function. Let us first define the **sup** (supremum) norm, $|| \cdot ||_\infty$ on the value function.

$$||V||_\infty = sup\{|V(b_i)| : b_i \in \mathcal{B}_i\}$$

and

$$||V - U||_\infty = sup\{|V(b_i) - U(b_i)| : b_i \in \mathcal{B}_i\}$$

Lemma 2.2 (Contraction). *The backup operator,* H*, forms a* **contraction** *mapping. Formally, let* $V, U \in B(\mathcal{B}_i)$*, and* γ *be the discount factor, then*

$$||HV - HU||_\infty \leq \gamma ||V - U||_\infty$$

We refer the reader to (Section 3.2.3 of Hauskrecht, 1997) for proofs of Lemmas 2.1 and 2.2. Clearly, $B(\mathcal{B}_i)$ defines a vector space of all bounded real-valued value functions. We will now show that the vector space $B(\mathcal{B}_i)$, with the *sup* norm is a complete [1] normed space, i.e. $(B(\mathcal{B}_i), ||\cdot||_\infty)$ is a complete normed vector space. The space is complete w.r.t. the metric induced by the norm $||\cdot||_\infty$. The metric induced by the norm $||\cdot||_\infty$ is $d(V,U) = ||V-U||_\infty$. Such a complete normed space is called a *Banach* space (Aliprantis & Burkinshaw, 1998).

The next lemma establishes that the sequence of value functions in $B(\mathcal{B}_i)$ forms a Cauchy sequence. Since the proof of this lemma is rather straightforward, we will omit it.

Lemma 2.3 (Cauchy sequence). *Let $\{U^n\}$ be a sequence of value functions in $B(\mathcal{B}_i)$, where n is the horizon. Then,*

$$||U^{n+m} - U^n||_\infty = \frac{\gamma^n(1-\gamma^m)}{1-\gamma}||U^1 - U^0||_\infty$$

For each $\epsilon > 0$, there exists n_0 such that $||U^{n+m} - U^n||_\infty \leq \epsilon$ for all $n+m, n > n_0$. Thus $\{U^n\}$ forms a Cauchy sequence in $B(\mathcal{B}_i)$.

We will now show that the Cauchy sequence $\{U^n\}$ converges in $B(\mathcal{B}_i)$, i.e. $B(\mathcal{B}_i)$ is *complete*.

Theorem 2.1 (Banach space). *The normed vector space $(B(\mathcal{B}_i), ||\cdot||_\infty)$ is a Banach space.*

Proof. While this theorem is a well-known standard result, we include its proof for the sake of completeness. From Lemma 2.3, we note that $\{U^n\}$ is a Cauchy sequence. Because value functions are real-valued, and $\mathbb{R}$ is a complete space, therefore $\{U^n(b_i)\}$ converges in $\mathbb{R}$ for each $b_i \in B_i$. Now, let $U^*(b_i) = \lim_{n\to\infty} U^n(b_i) - U^*$ is real-valued. It follows from the inequality $|U^n(b_i) - U^m(b_i)| \leq \epsilon$ for all $n, m > n_0$ that $|U^n(b) - U^*(b)| \leq \epsilon$ for all $n > n_0$ and all $b_i \in \mathcal{B}_i$. The last inequality in turn implies U^* is bounded and therefore $U^* \in B$. Hence $\lim_{n\to\infty} ||U^* - U^n||_\infty = 0$, and so B is complete. □

Theorem 2.2 (Banach Fixed-point Theorem (Aliprantis & Burkinshaw, 1998)). *Let X be a Banach space. Let $F : X \to X$ be a contraction mapping and let $\{x_n\}$ be a sequence with arbitrary initial point $x_0 \in B$, such that $x_n = Fx_{n-1}$. Then:*

1. *F has a unique fixed point solution x^* such that $x^* = Fx^*$,*
2. *The sequence $\{x_n\}$ converges to x^*.*

[1] A space X is complete if every Cauchy sequence $\{x_n\}$ of X converges to a point in X.

Using the Banach fixed-point theorem, we can show that the sequence $\{U^n\}$ with an arbitrary initial point will always converge to a unique fixed point.

Theorem 2.3 (Convergence). *For any POMDP with a discount factor, $0 < \gamma < 1$, the value iteration starting from any arbitrary value function converges to a unique fixed-point.*

Proof. Let $X = B(\mathcal{B}_i)$, and $F = H$. Lemma 2.2 establishes the contraction property of H. Then using the Banach fixed-point theorem, the sequence $\{U^n\}$ converges in $B(\mathcal{B}_i)$ to $U^* \in B(\mathcal{B}_i)$ and U^* is unique and fixed. □

Computing the value of each belief, as required by value iteration, is computationally impossible because the space of all beliefs is a continuum. To address this problem, Smallwood and Sondik (1973) showed that the value function can be decomposed into a set of linear vectors, and dynamic programming can be carried out on these vectors. Formally, the value function for every horizon is piecewise linear and convex. In Theorem 2.4, we state this property, and restate the proof that first appeared in (Smallwood & Sondik, 1973), in a more intuitive way.

We start by defining an inner product:

Definition 2.3 (Inner Product). *Define the inner product, $\langle \cdot, \cdot \rangle : \Delta(S) \times B(S) \to \mathbb{R}$, by*

$$\langle b_i, \alpha \rangle = \sum_s b_i(s)\alpha(s)$$

where $\alpha \in B(S)$ is a bounded and real-valued value function defined on S, which we will refer to as the alpha vector.

The next lemma establishes the bilinearity of the inner product defined above.

Lemma 2.4 (Bilinearity). *For any $s, t \in \mathbb{R}$, $f, g \in B(S)$, and $b, \lambda \in \Delta(S)$ the following equalities hold:*

$$\langle sf + tg, b \rangle = s\langle f, b \rangle + t\langle g, b \rangle$$
$$\langle f, sb + t\lambda \rangle = s\langle f, b \rangle + t\langle f, \lambda \rangle$$

Theorem 2.4 (PWLC). *The value function in a POMDP, U^n, is always piecewise linear and convex (PWLC). Mathematically:*

$$U^n(b_i) = \max_{\alpha^n \in \Gamma^n} \sum_{s \in S} b_i(s)\alpha^n(s)$$

where Γ^n is a finite set of n-horizon alpha vectors.

Proof. The proof follows by induction.

Basis Case: The horizon 1 value function is:

$$\begin{aligned} U^1(b_i) &= \max_{a \in A_i} \rho(b_i, a) = \max_{a \in A_i} \textstyle\sum_{s \in S} b_i(s) R_i(s, a) \\ &= \max_{a \in A_i} \langle b_i, R_i^a \rangle \end{aligned}$$

The horizon 1 value function is an inner product. From Lemma 2.4, the value function is linear in b_i, and maximizing over a set of linear vectors makes the function piecewise linear and convex.[1]

Inductive Hypothesis: Assume $U^{n-1}(b_i)$ is PWLC. In other words,

$$U^{n-1}(b_i) = \max_{\alpha^{n-1} \in \Gamma^{n-1}} \sum_{s \varepsilon S} b_i(s)\alpha^{n-1}(s)$$

Inductive proof: We need to show that U^n is PWLC i.e. to show that it can be written in the form, $U^n(b_i) = \max_{\alpha^n \in \Gamma^n} \sum_{s \varepsilon S} b_i(s)\alpha^n(s)$.

We know that

$$U^n(b_i) = \max_{a \in A_i} \left\{ \sum_{s \in S} b_i(s) R_i(s, a) + \gamma \sum_{o_i \in \Omega_i} Pr(o_i | a, b_i) U^{n-1}(SE(b_i, a, o_i)) \right\}.$$

Replace U^{n-1} from the inductive hypothesis step:

$$U^n(b_i) = \max_{a \in A_i} \left\{ \sum_{s \in S} b_i(s) R_i(s, a) + \gamma \sum_{o_i \in \Omega_i} Pr(o_i | a, b_i) \max_{\alpha^{n-1} \in \Gamma^{n-1}} \sum_{s' \in S} SE(b_i, a, o_i)(s')\alpha^{n-1}(s') \right\}$$

[1] If $|S| = 2$, then the value function is composed of a set of lines, otherwise it is composed of a set of hyperplanes.

Let $l(b_i, a, o_i)$ be the index of the alpha vector that maximizes U^{n-1} at the updated belief $SE(b_i, a, o_i)$. Then

$$U^n(b_i) = \max_{a \in A_i} \left\{ \sum_{s \in S} b_i(s) R_i(s, a) + \gamma \sum_{o \in \Omega_i} Pr(o_i|a, b_i) \sum_{s'} SE(b_i, a, o_i)(s') \alpha^{n-1}_{l(b_i,a,o_i)}(s') \right\}$$

We substitute $SE(b_i, a, o_i)$ with the belief update shown in Eq. 2.1:

$$U^n(b_i) = \max_{a \in A_i} \left\{ \begin{array}{l} \sum_{s \in S} b_i(s) R_i(s, a) + \gamma \sum_{o_i \in \Omega_i} Pr(o_i|a, b_i) \sum_{s' \in S} \frac{O_i(s', a, o_i) \sum_s T_i(s, a, s') b_i(s)}{Pr(o_i|a, b_i)} \\ \quad \times \alpha^{n-1}_{l(b_i,a,o_i)}(s') \end{array} \right\}$$

Rearranging the terms of the equation we get:

$$\begin{aligned} U^n(b_i) &= \max_{a \in A_i} \left\{ \begin{array}{l} \sum_{s \in S} b_i(s) R_i(s, a) + \gamma \sum_{o_i \in \Omega_i} \sum_{s' \in S} O_i(s', a, o_i) \sum_{s \in S} T_i(s, a, s') \\ \quad \times b_i(s) \alpha^{n-1}_{l(b_i,a,o_i)}(s') \end{array} \right\} \\ &= \max_{a \in A_i} \left\{ \sum_{s \in S} b_i(s) \left(R_i(s, a) + \gamma \sum_{o_i \in \Omega_i} \sum_{s' \in S} O_i(s', a, o_i) T_i(s, a, s') \alpha^{n-1}_{l(b_i,a,o_i)}(s') \right) \right\} \end{aligned}$$

Let the innermost summand

$$R_i(s, a) + \gamma \sum_{o_i \in \Omega_i} \sum_{s' \in S} O_i(s', a, o_i) T_i(s, a, s') \alpha^{n-1}_{l(b_i,a,o_i)}(s') = \alpha^n(s), \tag{2.5}$$

and the set of all α^n be Γ^n. Note that Γ^n is a finite set for a finite number of actions and observations.

We can rewrite the equation as:

$$U^n(b_i) = \max_{\alpha^n \in \Gamma^n} \sum_{s \in S} b_i(s) \alpha^n(s) = \max_{\alpha^n \in \Gamma^n} \langle b_i, \alpha^n \rangle \tag{2.6}$$

Eq. 2.6 is an inner product, and using Lemma 2.4, U^n is linear in b_i. Futhermore, maximizing over a set of linear alpha vectors produces the piecewise linear and convex value function. □

2.2 Algorithms for Solving POMDPs

A wide spectrum of exact and approximate solution techniques that compute the optimal and approximately optimal policy, respectively, for a POMDP exist. Approximate solution techniques trade off quality of the solution with computation time. They are critically required, since POMDPs suffer from a high computational complexity. Specifically, the task of computing a finite horizon policy for a POMDP is PSPACE-hard in general, and PSPACE-Complete for the case where the number of horizons is less than or equal to the number of states of the POMDP (Papadimitriou & Tsitsiklis, 1987a). For the infinite horizon case, the decision of whether a policy of some given value can be computed is undecidable (Madani, Hanks, & Condon, 2003). In this section, we will briefly survey the exact algorithms, and present one such algorithm in detail. In the next section, we will survey the approximation techniques.

All exact algorithms derive the n^{th} horizon policy from the $(n-1)^{th}$ horizon one, by computing the n^{th} horizon alpha vectors from the $(n-1)^{th}$ horizon ones. They differ in the method by which this derivation takes place. The earliest and the least sophisticated of all exact algorithms, the **enumeration** algorithm (Monahan, 1982), performs a "brute force" combinatorial approach. Specifically, it derives every possible n^{th} horizon policy tree from the previous horizon ones. The policy trees that are not optimal at any belief are rejected (pruned) using a linear program. The **witness** algorithm (Cassandra et al., 1994), on the other hand, incrementally generates the correct set of policy trees, thereby taking less time, in practice. The **incremental pruning** algorithm (Cassandra, Littman, & Zhang, 1997) exploits an important insight – several linear programs with less constraints are computationally less expensive than a single linear program with a large number of constraints. Consequently, the incremental pruning algorithm, interleaves pruning with generation of new policy trees, as opposed to first generating all possible policy trees, and then pruning them. Because we extend the incremental pruning algorithm to multiagent settings later on, we explain it in detail in the next section.

The previously mentioned algorithms iterate over the value function space and converge to the optimal value function. The policy graph, is then derived from the value function. A contrasting approach (Hansen, 1998) is to directly iterate over the policy space. Specifically, the policy iteration algorithm represents a policy as a finite state machine, and attempts to incrementally improve this policy.

One method to speed up the exact computation of solutions, is to utilize the structure of the problem domain. Such structure, is usually, captured using *factored* (feature-based) state representations such as

DBNs and decision trees. In (Boutilier & Poole, 1996), Boutilier et al. present an algorithm that represents each alpha vector as a decision tree, where the decision nodes are the factored state variables, and performs a dynamic programming update of these decision trees. A slightly different approach is to perform incremental pruning on the factored state representations, as shown in (Hansen & Feng, 2000).

2.2.1 An Exact Method for Solving POMDPs

The incremental pruning algorithm (Cassandra et al., 1997) improves on other exact algorithms by interleaving the generation of new alpha vectors with pruning. The algorithm utilizes a dynamic programming approach by initializing the set of alpha vectors with horizon 1 vectors, and using the Bellman update, as given in Eq. 2.5, to generate new vectors of succeeding horizons. Each iteration of the algorithm can be decomposed into three steps:

Step 1: We first generate intermediate sets $\Gamma^n_{a,*}$, and $\Gamma^n_{a,o}$, $\forall a \in A_i$, $\forall o \in \Omega_i$,

$$\Gamma^1_{a,*} \overset{\cup}{\leftarrow} \alpha^1_{a,*} = R_i(s,a)$$

$$\Gamma^n_{a,o_i} \overset{\cup}{\leftarrow} \alpha^n_{a,o_i} = \gamma \sum_{s' \in S} T_i(s,a,s') O_i(s,a,o_i) \alpha^{n-1}(s') \;\; \forall \alpha^{n-1} \in \Gamma^{n-1}_p$$

where Γ^{n-1}_p is the set of horizon $n-1$ optimal alpha vectors.

Step 2: Next, we generate Γ^n_a $\forall a \in A_i$, by taking the cross-sum over all possible observations,

$$\Gamma^n_a = \Gamma^1_{a,*} \oplus \Gamma^n_{a,o^1_i} \oplus \Gamma^n_{a,o^2_i} \oplus \ldots \oplus \Gamma^n_{a,o^{|\Omega_i|}_i}$$

where $\Gamma_1 \oplus \Gamma_2 = \{\alpha_1 + \alpha_2 | \alpha_1 \in \Gamma_1, \alpha_2 \in \Gamma_2\}$

Step 3: Normally, we would take the union of all vector sets generated so far:

$$\Gamma^n = \bigcup_{a \in A_i} \Gamma^n_a$$

Many of the vectors in the final set may be completely *dominated* by others. A vector is dominated by another if at all belief states, the value of the latter is greater than the value of the former. The domination

test can be carried out using *linear programming* (LP). Hence, as a final step, we would apply the domination test on the vector set, and retain a parsimonious set of vectors.

$$\Gamma_p^n = prune(\Gamma^n)$$

Rather than pruning the vector set after all cross-sums have been computed (*Step 2*), the incremental pruning algorithm interleaves pruning with cross-sum computations:

$$prune(A \oplus B \oplus C) = prune(prune(A \oplus B) \oplus C) \tag{2.7}$$

Using Equation 2.7, *Step 2* of the algorithm can be rewritten:

$$\begin{aligned}\Gamma_a^n &= \Gamma^1_{a,*} \oplus \Gamma^n_{a,o_i^1} \oplus \Gamma^n_{a,o_i^2} \oplus \ldots \oplus \Gamma^n_{a,o_i^{|\Omega_i|}} \\ &= prune(\ldots prune(prune(\Gamma^1_{a,*} \oplus \Gamma^n_{a,o_i^1}) \oplus \Gamma^n_{a,o_i^2}) \oplus \ldots \oplus \Gamma^n_{a,o_i^{|\Omega|}})\end{aligned}$$

To understand the time complexity of this algorithm, let us ascertain the maximum number of vectors generated in each step prior to the pruning step. In *Step 1* we generate $|A_i||\Omega_i||\Gamma_p^{n-1}|$ vectors. The original *Step 2* generates $|A_i||\Gamma_p^{n-1}|^{|\Omega_i|}$ vectors. Hence, set Γ^n will contain a maximum of $|A_i||\Gamma_p^{n-1}|^{|\Omega_i|}$ vectors in time $|S|^2|A_i||\Gamma_p^{n-1}|^{|\Omega_i|}$, and occupy maximum space, $|S||A_i||\Gamma_p^{n-1}|^{|\Omega_i|}$. The worst case time complexity is exponential in the number of observations and doubly exponential in the number of horizons making the problem of computing exact solutions intractable. However, the complexity of incremental pruning is $\Theta(|\Gamma_a^n||\Gamma_p^{n-1}||\Omega_i|)$ LPs, and $O(|\Gamma_a^n|^2|\Gamma_p^{n-1}||\Omega|)$ total number of constraints. Though, in the worst case, these bounds are identical to those of the original algorithm, there are POMDPs for which the best-case total number of constraints is asymptotically better than for the original algorithm.

2.2.2 Example: The Single Agent Tiger Problem

In order to see what solutions to POMDPs – policies – look like, we use the single agent tiger problem for illustration. Our purpose is to build on the insights that POMDP solutions provide in this simple case to illustrate solutions to interactive versions of this problem later.

Definition 2.4 (Single Agent Tiger Problem). *The traditional tiger problem resembles a game-show situation in which the decision maker has to choose to open one of two doors behind which lies either a valuable*

prize or a dangerous tiger. Apart from actions that open doors, the subject has the option of listening for the tiger's growl coming from the left, or the right, door. However, the subject's hearing is imperfect, with given percentages (say, 15%) of false positive and false negative occurrences. Following (Kaelbling, Littman, & Cassandra, 1998), we assume that the value of the prize is 10, that the pain associated with encountering the tiger can be quantified as -100, and that the cost of listening is -1. See Fig. 2.3 for an illustration.

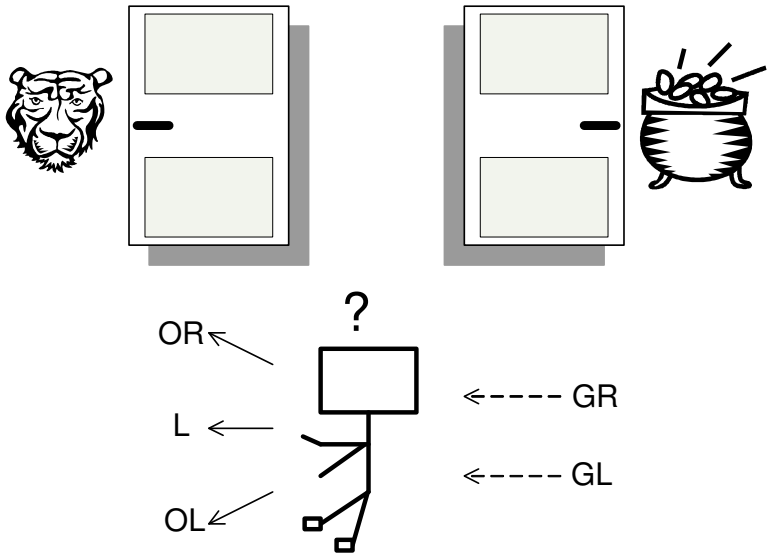

Figure 2.3: An illustration of the tiger problem. At each step, agent i must make a decision: open the left door (OL), listen (L), or open the right door (OR). To aid its decision, the agent receives observations – the tiger's growls from either the left (GL) or right (GR) depending on where the tiger is. However, the agent's sensors are noisy.

We represent the single agent tiger problem as a POMDP. The transition, reward, and observation functions that quantify the problem are shown in Table 2.1.

We solve the POMDP representing the single agent tiger problem using the incremental pruning algorithm described in Section 2.2.1, for finite horizons. Specifically, we solve the game for 1,2 and 3 horizons and show the corresponding value functions in Figs. 2.4, 2.5, and 2.6. Values of beliefs are based on the best conditional plan (policy tree) available in that belief state, as specified in Eq. 2.4. Because the tiger problem has only two physical states, the agent's belief is completely specified using a single number: $0 \leq p(TL) \leq 1$. Each vector is labeled with the conditional plan that it represents. Note that the symmetric nature of the agent's POMDP produces symmetric value functions, and that the value function is piecewise linear in belief and convex.

Action	State	TL	TR
OL	*	0.5	0.5
OR	*	0.5	0.5
L	TL	1.0	0
L	TR	0	1.0

Transition function (T_i) for agent i.

Action	TL	TR
OR	10	-100
OL	-100	10
L	-1	-1

Reward function (R_i) for agent i.

Action	State	GL	GR
L	TL	0.85	0.15
L	TR	0.15	0.85
OL	*	0.5	0.5
OR	*	0.5	0.5

Observation function (O_i) for agent i.

Table 2.1: Transition, reward, and observation functions for agent i playing the tiger problem.

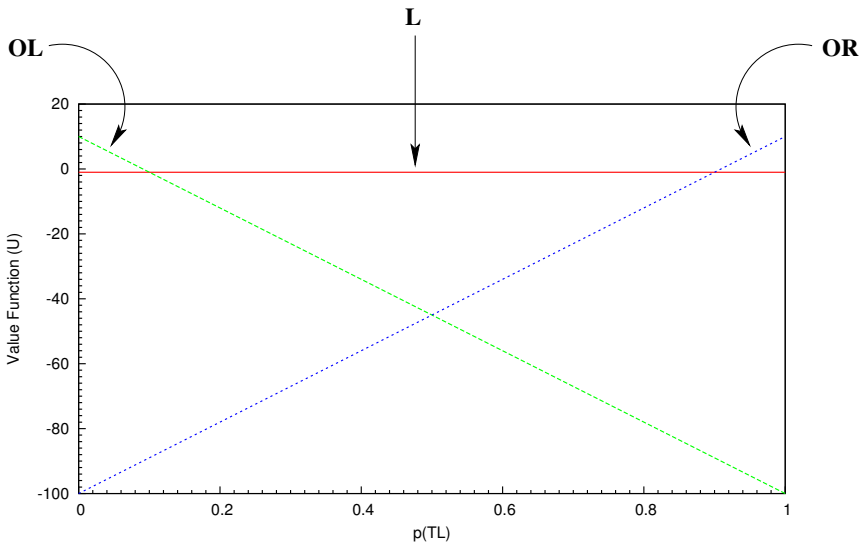

Figure 2.4: Value function for horizon 1. The agent performs OL when the probability that it assigns to the tiger being on the left, $p(TL) < 0.1$. Listens when $0.1 < p(TL) < 0.9$, and does OR when $p(TL) > 0.9$. At values of $p(TL)$ where two vectors intersect, the conditional plans represented by the intersecting vectors are followed with equal probabilities.

The value function, in Figure 2.4, shows values of various belief states when the agent's time horizon is equal to 1. The state of certainty is most valuable – when the agent knows the location of the tiger it can open the opposite door and claim the prize which certainly awaits. Thus, when the probability of tiger location is 0 or 1, the value is 10. When the agent is sufficiently uncertain, its best option is to play it safe and listen; the value is then -1. The agent is indifferent between opening doors and listening when it assigns probabilities of

0.9 or 0.1 to the location of the tiger. Note that when the time horizon is equal to 1, listening does not provide any useful information since the game does not continue to allow for the use of this information.

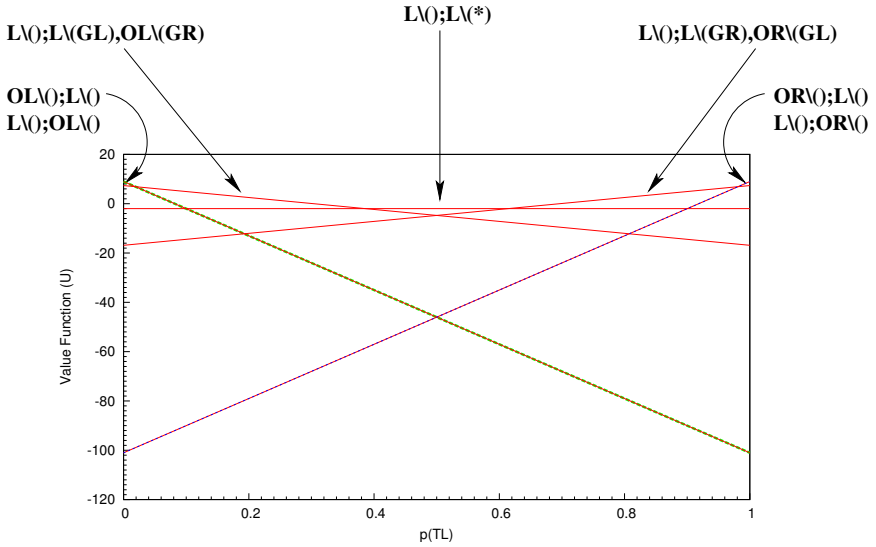

Figure 2.5: Value function for horizon 2. When $p(TL) < 0.02$ the agent starts with a OL and then listens no matter what the observations are, or listens first and then does OL no matter what, with equal probabilities. For $0.02 < p(TL) < 0.39$ a conditional plan that starts by listening and on hearing a GL listen again or OL on hearing a GR, is followed by the agent. When $0.39 < p(TL) < 0.61$, the agent will always listen, and for $0.61 < p(TL) < 0.98$, the agent will start by listening, and OR when it hears a GL or L when it hears a GR. If $0.98 < p(TL) < 1.0$, the agent will execute the two conditional plans of first performing OR and then listening no matter what, or first listening and then always performing OL, with equal probabilities. Note that at values of $p(TL)$ where the vectors intersect, the conditional plans represented by the intersecting vectors will be carried out with equal probabilities.

When the time horizon is 2 (Fig. 2.5), listening and hearing the tiger's growls does provide useful information to the agent. The result is that the agent makes full use of its listening action and the agent's policy when it has two steps to go becomes more cautious: It opens doors only when its certainty level is greater than 0.98.

2.3 Approximation Techniques for POMDPs

There are two distinct but correlated sources of intractability that we encounter while applying POMDPs to large planning domains. The first one is aptly called the *curse of dimensionality* – the number of physical states of the problem which make up the dimensions of the belief simplex, and the second is called the *curse of history* – possible beliefs that the agent could have in the future depending on its anticipated actions and

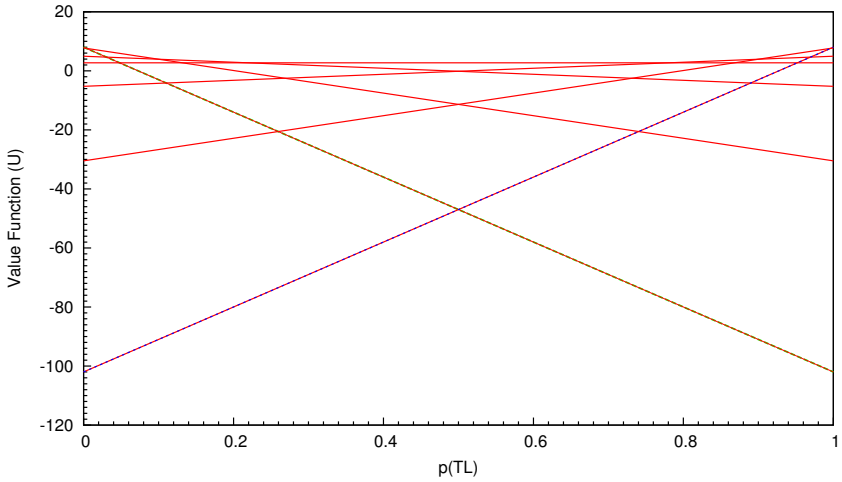

Figure 2.6: Value function for horizon 3. We avoid describing the policy here because of its complexity. However, the conditional plan associated with each vector may be obtained from the policy graph in Fig. 2.7.

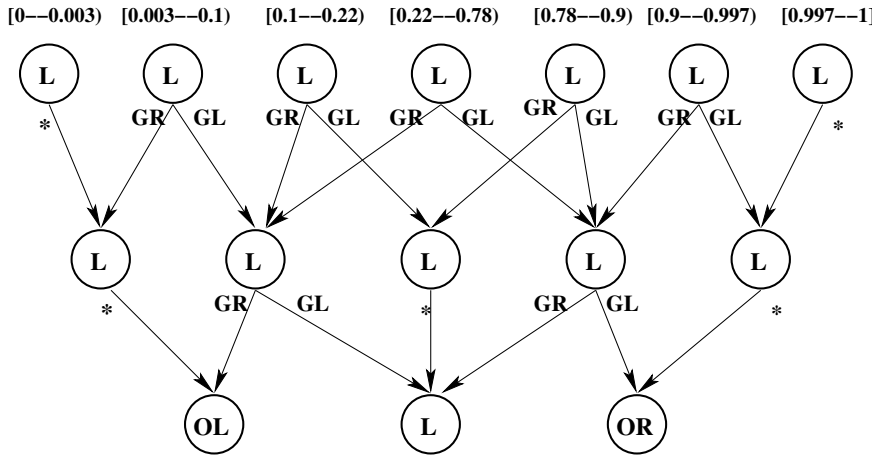

Figure 2.7: One of 16 optimal horizon 3 policies. The belief intervals are over the probability of the tiger being behind the left door.

observations. The latter bottleneck is same as the complexity of the policy space. As the number of physical states increase, the dimensionality of the belief simplex grows, which not only affects the time complexity of the LPs but also adversely affects the naive approaches that solve POMDPs by discretizing the belief space. The number of possible future beliefs that the agent i may have starting from a particular belief is proportional to $(|A_i||\Omega_i|)^{T-1}$, which grows exponentially with the horizon T. The two curses are also correlated: A larger dimensioned belief simplex naturally implies a larger number of possible beliefs that an

agent may have, thereby contributing to the curse of history. If we extend POMDPs to multiagent settings, these curses will carry over, and possibly become even more acute. In this regard, it is useful to briefly survey the vast literature on approximation techniques for POMDPs, with the aim of extending some of the underlying ideas to multiagent settings. For excellent surveys of approximation techniques see (Hauskrecht, 2000) and (Aberdeen, 2003).

Let us first look at approximations that address the curse of dimensionality. The least sophisticated of all such approximation techniques treats POMDPs as completely observable MDPs (Littman, Cassandra, & Kaelbling, 1995a). The algorithm therefore assumes that any uncertainty in the agent's current belief reduces to zero after the next action. Since the solution assumes perfect observability, the value function is overly optimistic, and provides an upper bound on the optimal POMDP value function. Another class of algorithms prescribe substituting the more complex belief space with a simpler feature space (Bertsekas, 1995; Tsitsiklis & Roy, 1996). The feature space is usually smaller, and summarizes the important characteristics of the belief space with regard to control. One such method (Roy & Gordon, 2002) utilizes the technique of principal component analysis to identify the relevant important features. Another method (Poupart & Boutilier, 2003) investigates lossless and lossy compressions of the belief space through its impact on decision quality.

Several algorithms also exist that beat back the curse of history. Rather than operating over the entire belief simplex, these algorithms pick sample beliefs and approximate the value function on the basis of the selected beliefs. One such set of algorithms (Lovejoy, 1991; Brafman, 1997; Zhou & Hansen, 2001) approximate the POMDP belief space by superimposing a regular grid on the belief simplex. The grid divides the belief simplex into a set of equal-sized sub-simplices. The key idea in this technique is to perform value iteration over the belief states that lie on the intersection of grid lines, and *interpolate* the value of the non-grid belief states. Zhou and Hansen (2001) consider a variable-resolution grid. The resolution of the grid is increased in those sub-simplices where lower error bound is required. Recently, Pineau et. al. (2003b) suggested performing point-based value iteration, by selecting belief points from the belief simplex and retaining only the alpha vectors that are optimal over those belief points. This not only reduces the set of alpha vectors at each time step, but also simplifies the pruning step by eliminating the need for linear programs. The selected belief points are those which lie on the reachability tree generated from the starting belief points. They also explored the use of metric trees (Pineau, Gordon, & Thrun, 2003a) (using the supremum norm as the distance metric between belief points) for making a smarter selection of relevant belief points.

Recently, a method that tackles both the curses of dimensionality and history was proposed (Poupart & Boutilier, 2004). The method combines compression of the belief simplex, reducing its dimensionality, with bounded policy iteration to reduce the policy space complexity. The resulting approach is highly scalable; its application was shown on a network management problem of 33 million states. Hierarchical approaches that decompose a global task into a set of subtasks, solve the subtasks using smaller POMDPs, and piece together the resulting sub-policies, have also appeared in the literature (Pineau, Roy, & Thrun, 2001).

2.4 Summary

POMDPs are general decision-theoretic frameworks for planning in single agent settings that address both, the action outcome uncertainty and the state uncertainty problems. However, they are computationally very complex; exact solutions have been reported only for very simple problem domains that contain less than ten states. Therefore, approximate techniques that trade off complexity with the quality of the solution are critically required if POMDPs are to move beyond toy problems. A vast suite of approximation techniques exist that address both the sources of intractibility in POMDPs: the curse of dimensionality, and the policy space complexity. Many of these techniques also provide useful error bounds for the approximations. Though, POMDPs yet do not find mainstream recognition, their applications to ever larger problems is encouraging.

Chapter 3

BACKGROUND: GAME THEORY

STRATEGIC interaction in multiagent settings has been the central problem of interest for game theorists. Though we adopt a decision theoretic approach to multiagent planning rather than game theoretic, we borrow several concepts from game theory which we briefly review in this chapter. From single play interactions (typically called *games*) in which agents (players) act only once, attention has shifted to infinitely repeated games in which the same game is played repeatedly. An assumption that pervades all of game theory is that all agents are rational – they always maximize their (expected) utility – all agents know that all agents are rational, all agents know that all know that all are rational, and so on. This assumption is called the *common knowledge* of rationality. Under the umbrella of this assumption, the solution concept of choice when analyzing games has been Nash equilibrium, and continues to remain so. A pair of strategies (actions for single play games) of two agents is in Nash equilibrium if each is a best response to the other. This circular definition ensures that if each agent performs its part of the Nash equilibrium, then there is no incentive for the other to deviate from its part. Therefore, Nash equilibrium is stable. On the other hand, many games have multiple Nash equilibria necessitating all agents to act out the same Nash equilibrium; this, in the absence of centralized control, requires some kind of synchronizing mechanism among agents which researchers have usually shied away from addressing.

Interest in game theory was spurred by Von Neumann and Morgenstern's foundational book (1953) first published in 1944. A vast literature has spawned from then onwards until now which cannot be reviewed in its entirety here due to lack of space. Therefore, we concentrate only on those parts of game theory that find a direct application elsewhere in this thesis. The remainder of this chapter is structured as follows. We

very briefly review single play games in Section 3.1, using an example game for illustration. We then move onto repeated games in Section 3.2, and focus on the learning algorithms for incomplete information repeated games. In Section 3.3 we survey the various algorithms that exist for learning Nash equilibria in stochastic games. We summarize this chapter in Section 3.4.

3.1 Single Play Games

Games in which each agent can act only once are called single play or single shot games. All agents that play the game act *simultaneously* and *independently*, and their play is usually guided by their payoffs which is a function of actions of all agents. Single play games are commonly represented in two ways: the matrix or normal form, and the extensive form. Recently, graphical models such as influence diagrams (Tatman & Shachter, 1990) have also been extended to represent and solve games (Koller & Milch, 2001; Gal & Pfeffer, 2003). Graphical models provide a descriptive approach to analyzing games by explicitly capturing the structure of the game and decision making models of the players. In this chapter, we will utilize the normal form of games, and refer the reader to (Fudenberg & Tirole, 1991) for an explanation of the extensive form.

3.1.1 Games of Complete Information

We will first restrict our attention to games in which each player knows its own and others' payoff functions. Let us summarize the important solution concepts for such games using an example:

Definition 3.1 (Market Niche). *Two firms, I and J, are competing for a single market niche. If one firm completely occupies the niche, then its profit is quantified by 10, while the other firm does not make any profit or loss. If both the firms occupy the niche, then both suffer losses worth -5 each. However, if both firms choose to stay out then each breaks even. The payoff (reward) function of each firm is shown in Table 3.1. In each cell, the first number denotes the payoff to firm I, and the second number denotes the payoff to firm J.*

Let us start our analysis of the symmetric game by defining the *strategy* for each firm. A strategy π_I of firm I is a probability distribution over I's actions: $\pi_I \in \Delta(A_I)$, where $A_I = \{\text{Enter}, \text{Stay out}\}$. Firm J's strategy is defined analogously. If a strategy is a point mass distribution, then it is called a *pure* strategy[1],

[1] If the space of actions is continuous, then strategies are probability density functions (p.d.f.s), and a pure strategy is a Dirac-delta p.d.f.

		FIRM J	
		Enter	**Stay out**
FIRM I	**Enter**	-5, -5	10, 0
	Stay out	0, 10	0, 0

Table 3.1: The market niche game between two firms in normal form. We show the payoff functions, $R_I(a_i, a_j)$ and $R_J(a_i, a_j)$, of each firm. All aspects of the interaction are captured using the payoff functions.

otherwise it is called a *mixed* (randomized) strategy. Next, we define a firm's, say I's, best response strategy to J's strategy:

Definition 3.2 (Best Response). *A strategy, π_I, of firm I with a payoff function R_I is a best response to the strategy π_J if $\pi_I \in OPT$ where:*

$$OPT(R_I, \pi_J) = \underset{\pi_I \in \Delta(A_I)}{argmax} \sum_{a_I, a_J} R_I(a_I, a_J)\pi_I(a_I)\pi_J(a_J) \tag{3.1}$$

Firm I's best response function for a given payoff function and J's strategy can be computed using a linear program.[2] A solution for the game would be a prescription: the strategy that firm I should perform and the strategy that firm J should carry out, that will maximize their individual utilities. However, selection of firm I's strategy will depend on the strategy selected by firm J, which in turn will depend on the strategy selected by firm I and so on, resulting in an infinite regress. A solution concept that "cuts" through this regress is that of Nash equilibrium. We define Nash equilibrium below:

Definition 3.3 (Nash Equilibrium). *A pair of strategies, $[\pi_I, \pi_J]$, are in Nash equilibrium if each strategy of the pair is a best response to the other. Formally,*

$$\pi_I \in OPT(R_I, \pi_J) \text{ and } \pi_J \in OPT(R_J, \pi_I)$$

where OPT is defined according to the Definition 3.2.

Nash equilibrium is stable: Because of common knowledge of rationality, there is no incentive for each firm to deviate from its part of the equilibrium. For the market niche game of Table. 3.1, there are two

[2] If the game were to be a zero-sum game – $R_I(a_I, a_J) + R_J(a_I, a_J) = 0 \quad \forall a_I, a_J$ – Eq. 3.1 would be $OPT(R_I) = \underset{\pi_I \in \Delta(A_I)}{argmax} \underset{\pi_J \in \Delta(A_J)}{min} \sum_{a_I, a_J} R_I(a_I, a_J)\pi_I(a_I)\pi_J(a_J)$

pure strategy, and one mixed strategy Nash equilibria. The two pure strategy equilibria are $\Big[\langle 1,0\rangle,\langle 0,1\rangle\Big]$ and $\Big[\langle 0,1\rangle,\langle 1,0\rangle\Big]$, where $\langle x, y\rangle$ indicates a strategy where the firm will choose to enter the market with a probability of x, and stay out of the market with the probability of $y(= 1 - x)$. The mixed strategy Nash equilibrium is $\Big[\langle \frac{2}{3}, \frac{1}{3}\rangle, \langle \frac{2}{3}, \frac{1}{3}\rangle\Big]$. At this point, the reader may ask what epistemic conditions are sufficient for the firms to adopt the Nash equilibrium as a solution. Aumann and Brandenburger (1995) observed that for games such as the one that we are currently considering, each firm must be rational, know its own payoff function, and know the strategy selected by the other firm (mutual knowledge of strategy choices). Note that these conditions are sufficient but not necessary: in the absence of these conditions, the firms may still select the Nash equilibrium profile accidentally. We note that fulfilling the key epistemic condition of knowing the other firm's strategy may be difficult in practice.

The existence of multiple Nash equilibria for the market niche game raises a question. How do the firms come to expect to play the same Nash equilibrium. This question is significant, because in the absence of coordination their play need not correspond to any equilibrium at all. One may adopt synchronizing schemes such as the common selection procedure of Harsanyi and Selten (1988), but such approaches are also plagued with unresolved issues (see Chapter 1 in Fudenberg & Levine, 1997). Consequently, we view the non-uniqueness of Nash equilibrium as a significant impediment to its adoption as *the* solution concept for planning.

3.1.2 Games of Incomplete Information: Bayesian Games

In the market niche game of Section 3.1.1, each firm was aware of the other's payoff function, and subsequently of all the parameters of the game. Realistically, this assumption may not hold, with one or both firms being unaware of the other's payoff function (or other parameters). To illustrate this situation, let us modify the market niche game so that firm I has one of two payoff functions depending on whether its aggressive or submissive. We illustrate the modified market niche game in Table 3.2.

Such games of (one or two-sided) incomplete information were first addressed by Harsanyi (1967), who proposed encompassing all of the agent's private information relevant to its decision making in a attribute

FIRM J

FIRM I		Enter	Stay out
	Enter	1.5, -1	3.5, 0
	Stay out	2, 1	3, 0

FIRM I is aggressive

FIRM J

FIRM I		Enter	Stay out
	Enter	0, -1	2, 0
	Stay out	2, 1	3, 0

FIRM I is submissive

Table 3.2: The modified market niche game. Firm I's payoff function will depend on whether it's aggressive or submissive. Firm J's payoff function is fixed. Firm J is unaware of which of the two possible payoff functions is that of I.

vector called the *type* [3]. For our modified market niche game, firm I has two types:

$$\Theta_I = \{R_{I_{\text{aggressive}}}, R_{I_{\text{submissive}}}\}$$

Additionally, Harsanyi, in the same paper, also suggested – as a special case – using a prior distribution (provided by nature) over the types that is common knowledge to all agents playing the game. In this way, the game of incomplete information is turned into a game of imperfect information, and we can now compute a Bayesian Nash equilibrium for the game. If firm J also has more than one possible payoff function, then I's type space would additionally include its beliefs over J's payoffs. These beliefs would be derived from the common prior.

Let us solve the modified market niche game as shown in Table 3.2. First note that when firm I is submissive, choosing to stay out of the market is a dominant strategy no matter what firm J does. Let p_I be the commonly known prior probability that the firm I is aggressive. If x is the probability that firm I will choose to enter the market when it is aggressive, then firm J will enter if $x < \frac{1}{2(1-p_I)}$, stay out if $x > \frac{1}{2(1-p_I)}$, and be undecided if $x = \frac{1}{2(1-p_I)}$. Analogously, the aggressive firm I will enter if firm J chooses to enter with a probability of less than 0.5. The Bayesian Nash equilibrium for our game would be a pair (π_I, π_J), in which π_I is the optimal response of the aggressive firm I to π_J, and π_J is the optimal response of firm J to firm I's response and J's belief of firm I being aggressive with a probability p_I. There are again three Bayesian Nash equilibria in the modified market niche game. The two pure strategy equilibria are $\left[\langle 0,1\rangle,\langle 1,0\rangle\right]$ for any p_I, and $\left[\langle 1,0\rangle,\langle 0,1\rangle\right]$ iff $p_I \leq 0.5$. The mixed strategy equilibria is $\left[\langle \frac{1}{2(1-p_I)}, \frac{1-2p_I}{2(1-p_I)}\rangle, \langle \frac{1}{2}, \frac{1}{2}\rangle\right]$ iff $p_I \leq 0.5$.

[3] In Harsanyi's own words: " ...we can regard the attribute vector c_i as representing certain physical, social, and psychological attributes of player i himself in that it summarizes some crucial parameters of player i's own payoff function U_i as well as main parameters of his beliefs about his social and physical environment ..."

3.2 Learning in Repeated Games

In this section, we turn our attention to a more realistic framework of interaction. In this framework, the single play games considered in Section 3.1 are played repeatedly for an infinite number of times. Each agent is able to perfectly observe the previous plays (perfect monitoring) before making its next decision. Therefore, strategies such as tit-for-tat now become possible. Because each agent observes the history of the game, it may learn an assessment (belief) of the other agents' strategies, and use its assessment to generate its own best response strategy. There are two models of learning that are predominant in game theory: the *fictitious play* model (Fudenberg & Levine, 1997) and the *rational (Bayesian) learning* model (Kalai & Lehrer, 1993a; Nyarko, 1997). The two models differ in the way each agent learns about the other's behavior from its past observation history.

3.2.1 Fictitious play

Fictitious play (Fudenberg & Levine, 1997) is one of the simplest model of learning, but the first to show the relevance of Nash equilibrium as a predictive solution concept. In this process, each agent thinks that the other agent(s) behaves according to a stationary, but unknown (possibly mixed) strategy. For illustration, we will restrict our fictitious play model to two agents, i and j. An agent, say i, in the fictitious play model maintains a belief at each time t, $b_i^t \in \Delta(A_j)$. The agent updates its belief according to the following rule: for some action played by j, a_j^{t-1}, we add 1 to the action's frequency count and normalize. The belief update is shown below:

$$\widehat{b}_i^t(a_j) = \widehat{b}_i^{t-1}(a_j) + \begin{cases} 1 & a_j^{t-1} = a_j \\ 0 & a_j^{t-1} \neq a_j \end{cases} \quad \forall a_j \in A_j$$

$$b_i^t(a_j) = \frac{\widehat{b}_i^t(a_j)}{\sum_{a_j \in A_j} \widehat{b}_i^t(a_j)}$$

Once i's belief over j's action space is updated, its own play is simply a best response to its belief: $\pi_i \in OPT_i(R_i, b_i^t)$, where OPT_i is as defined in Eq. 3.1, and R_i is i's stage game payoff function. For traces of agent behaviors resulting from applying the fictitious play model to common games such as matching pennies, see (Shoham & Lleyton-Brown, 2002).

The fictitious play model though naive, exhibits some important asymptotic properties. Beliefs of agents using the fictitious play learning rule will necessarily converge in zero-sum games (though not necessarily

in other types of games). Additionally, if the belief of each agent converges, when using the fictitious play model, they will converge to a Nash equilibrium of the stage game. Variations on the fictitious play model also exist that strengthen the results on convergence (see Fudenberg & Levine, 1997).

3.2.2 Rational Learning

The learning model described in Section 3.2.1 ascribes a stationary mixed strategy to the other agent, and assumes that the other agent's actions are sampled i.i.d. from its distribution. Realistically, the agents' strategy may not be stationary, but instead may change depending on the history of play. Define an agent's behavioral strategy as $\pi_i : H \rightarrow \Delta(A_i)$, where H is the set of histories of play consisting of all sequences of pairs of actions of both agents. The rational learning model (Kalai & Lehrer, 1993a) postulates that each agent maintains a belief over the possible behavioral strategies of the other agent, $b_i^t \in \Delta(\Pi_j)$, where Π_j is the set of all behavioral strategies of j.

At each stage of the game, after play, the beliefs are updated in a Bayesian fashion using the observed actions of other players, and each agent optimizes according to its belief. Note that if play reaches a history which is not accounted by the player's belief, then the Bayesian update rule becomes undefined. Hence an important constraint of the learning rule is that an agent's beliefs should be *absolutely continuous* [4] w.r.t. the possible true histories of the game. Once this constraint is satisfied, the beliefs about the strategies of the other players will converge, though not necessarily to the true strategy of the other agent. The asymptotic behaviors of the agents participating in this process will converge to a *subjective* equilibrium that is stable with respect to learning and optimization. Note that if a profile of strategies is in Nash equilibrium, then it is also in subjective equilibrium. In (Kalai & Lehrer, 1993a), it is shown that the converse is also true (though in a weaker sense). Fudenberg and Levine (1993) demonstrate a related equilibrium called a *self-confirming* equilibrium by essentially using the same model but relaxing an important assumption that is made in the rational learning framework of Kalai and Lehrer: other agents' strategies are assumed independent of each other. We also discuss this framework in greater detail in Chapter 7.

For the sake of completeness, we note that while the previous algorithms addressed the question of learning of individual agents, literature in game theory also addresses the question of learning of a population of agents. A learning model that is well-studied in such a setting is the *replicator dynamic*, and a particular

[4] Two probability measures, μ_1 and μ_2 are absolutely continuous, $\mu_1 \ll \mu_2$ if for any measurable event A, $\mu_2(A) = 0 \Rightarrow \mu_1(A) = 0$.

stability concept inspired by such a setting is that of *evolutionary stable strategy*. However, such a setting is not directly relevant to our work, hence we direct the interested reader to (Fudenberg & Levine, 1997).

3.3 Learning in Stochastic Games

Stochastic games (Owen, 1982) may be viewed as a generalization of Markov decision processes and normal form games. In Fig. 3.1, we illustrate the relation between stochastic games, MDPs, and the normal form games. Because our work concentrates on partially observable stochastic games – a generalization of stochastic games to partially observable settings – we briefly review the stochastic game literature. Much of the work to date has addressed stochastic games in which the state space is perfectly observable, and (similar to the frameworks in previous sections) assume that actions of other agents are perfectly observable. Our work relaxes both of these (unrealistic) assumptions.

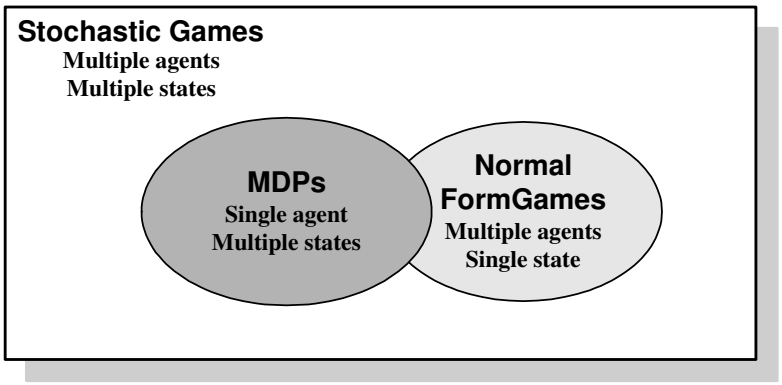

Figure 3.1: Stochastic games as a generalization of MDPs and normal form games to situations of multiple states and multiple agents.

Many algorithms (Littman, 1994; Hu & Wellman, 1998) for learning to play optimally in stochastic games perform model-free reinforcement learning (Sutton & Barto, 1998), and extend directly, the single-agent Q-learning algorithm (Watkins, 1989) to the multiagent setting. Recently, Bowling and Veloso (2002), suggested two properties that every learning algorithm in stochastic games must aspire for. The properties are: (1) convergence – the algorithm must converge to a stationary policy under suitable assumptions of play by other agents, (2) rationality – the convergent policy must be the best response to the other agents's stationary policies. Learning algorithms that exhibit these two properties will necessarily converge to the

Nash equilibrium. Note that these two properties are sufficient but not necessary for convergence to Nash equilibrium.

We will briefly review several algorithms that satisfy one or both the criteria. The minimax Q-learning algorithm (Littman, 1994), was the first algorithm to directly extend the single-agent Q-learning algorithm to zero-sum stochastic games. It does so by simply maintaining a Q-value for each combination of the state and actions of all agents. The learning rule for updating the Q-table computes the expected payoff to the agent resulting from following the equilibrium mixed strategy, for the single stage zero-sum game defined for the particular state. The computation of the agent's payoff is done using the standard minimax approach (implemented using a linear program). The algorithm converges in self-play in the limit of infinite exploration, but is not rational.

The above mentioned approach loses its significance for general-sum stochastic games. An important contribution for such games is the Nash Q-learning (Hu & Wellman, 1998). Nash Q-learning adopts the same idea of updating Q-values using Nash equilibrium payoffs. However, in order to do so, it must maintain Q-tables not only for itself, but for all other agents as well. Nash Q-learning suffers from the non-determinism of multiple Nash equilibria, and is applicable for only a special class of games. However, for this class, Nash Q-learning is shown to converge to Nash equilibrium in self-play in the limit of infinite exploration. Another algorithm also applicable to general-sum games, but restricted to cooperative ones, is (Claus & Boutilier, 1997). The algorithm differs from Nash Q-learning by maintaining beliefs about the other agent's policies, and updating these beliefs in a fashion similar to fictitious play.

Recently, the properties of convergence and rationality of learning algorithms have come under criticism (Shoham et al., 2003). These properties are seen as restrictive: they stress on the importance of achieving Nash equilibria at the expense of more realistic approaches, and most algorithms that conform to these properties do so under the assumption of self-play. Tesauro (2003) suggested utilizing an algorithm called Hyper Q-learning which relaxed several assumptions in the previously mentioned algorithms in order to develop a more practical approach. Powers and Shoham (2005) have suggested a new set of criteria on the rewards accumulated by the agent (as opposed to the criteria on play of the agent) that learning algorithms should satisfy, and present an algorithm under these criteria.

3.4 Summary

Traditionally, multiagent interactions have been extensively studied in the subject of game theory. Nash equilibria has been the central solution concept for most of the work (a notable exception is Kadane & Larkey, 1982). However, recently researchers have begun to question the relevance of Nash equilibrium as a solution paradigm for controlling agents: In games with multiple Nash equilibria, there is no clear explanation of why all the agents should come to expect the same equilibrium. Secondly, Nash equilibrium does not prescribe what the agent should do if others fail to follow their part of the equilibrium. Finally, all of the assumptions (epistemic and otherwise) under which the games have been analyzed such as common knowledge of rationality and payoffs, and perfect monitoring are being questioned on whether they are realistic (see Chapter 1 of Fudenberg & Levine, 1997). Having said the above, we utilize several other game theoretic concepts to develop our multiagent planning framework.

Chapter 4

INTERACTIVE POMDPS: MULTIAGENT DECISION-THEORETIC PLANNING

We develop a framework for sequential rationality of autonomous agents interacting with other agents within a common, and possibly uncertain, environment. We use the normative paradigm of decision-theoretic planning under uncertainty formalized as partially observable Markov decision processes (POMDPs) (see Chapter 2) as a point of departure. As we mentioned before, solutions of POMDPs are mappings from an agent's belief to actions. While POMDPs can be used in environments populated by other agents, the drawback is that other agents' actions have to be represented implicitly as environmental noise within the, usually static, transition model. Such restricted modeling disregards the fact that the other agents may also be learning and consequently their behavior may be dynamic. Thus, an agent's beliefs about the other agent are not part of solutions to POMDPs.

The main idea behind our formalism, called **interactive POMDPs** (I-POMDPs) (Gmytrasiewicz & Doshi, 2005, 2004, 2004), is to allow agents to use more sophisticated constructs to model and predict behavior of other agents. Thus, we replace the "flat" beliefs about the state space used in POMDPs with beliefs about the physical environment *and* about the other agent(s), possibly in terms of their preferences, capabilities, and beliefs. Such beliefs could include others' beliefs about others, and thus can be nested to arbitrary levels. They will be called interactive beliefs. While the space of interactive beliefs is very rich and updating these beliefs is more complex than updating their "flat" counterparts, we use the value function plots to show that solutions to I-POMDPs are at least as good as, and in usual cases superior to, comparable

solutions to POMDPs. The reason is intuitive – maintaining sophisticated models of other agents allows more refined analysis of their behavior and better predictions of their actions.

I-POMDPs are applicable to autonomous self-interested agents who locally compute what actions they should execute to optimize their preferences given what they believe while interacting with others with possibly conflicting objectives. The multiagent setting within which we study I-POMDPs extends stochastic games (Section 3.3) to partially observable environments, and we call it the partially observable stochastic game (POSG). Our approach of using a decision-theoretic framework and solution concept complements the equilibrium approach to analyzing interactions as used in classical game theory (see Chapter 3). As we mentioned before, the drawback of equilibria is that there could be many of them (non-uniqueness), and that they describe agent's optimal actions only if, and when, an equilibrium has been reached (incompleteness). Our approach, instead, is centered on optimality and best response to anticipated action of other agent(s), rather then on stability (Binmore, 1990; Kadane & Larkey, 1982). The question of whether, under what circumstances, and what kind of equilibria could arise from solutions to I-POMDPs is addressed in Chapter 7.

Our approach avoids the difficulties of non-uniqueness and incompleteness of traditional equilibrium approach, and offers solutions which are likely to be better than the solutions of traditional POMDPs applied to multiagent settings. But these advantages come at the cost of processing and maintaining possibly infinitely nested interactive beliefs. Consequently, only approximate belief updates and approximately optimal solutions to planning problems are computable in general. We define a class of finitely nested I-POMDPs to form a basis for computable approximations to infinitely nested ones. We show that a number of properties that facilitate solutions of POMDPs carry over to finitely nested I-POMDPs. In particular, the interactive beliefs are sufficient statistics for the histories of agent's observations, the belief update is a generalization of the update in POMDPs, the value function is piece-wise linear and convex, and the value iteration algorithm converges at the same rate.

The remainder of this chapter is structured as follows. We start with a brief review of related work in Section 4.1, and formalize the concept of agent types in Section 4.2. Section 4.3 introduces the I-POMDP framework and its solution. The finitely nested I-POMDPs, and their key properties are introduced in Section 4.4. We continue with an example application of finitely nested I-POMDPs to a multiagent version of the tiger problem in Section 4.5. There, we show examples of belief updates and value functions. Applications of I-POMDPs demonstrating the emergence of social behaviors are given in Section 4.6. We conclude with

a brief summary in Section 4.7, point out the contributions of our work in Section 4.8, and lay out directions of future work in Section 4.9.

4.1 Related Work

Our work draws from prior research on partially observable Markov decision processes, which recently gained a lot of attention within the AI community. We explained POMDPs in appropriate detail in Chapter 2.

As we mentioned before, the formalism of Markov decision processes has been extended to multiple agents giving rise to stochastic games or Markov games (Section 3.3 of Chapter 3). The algorithms for planning in stochastic games perform model-free reinforcement learning and assume complete observability of state. In contrast, our approach is model-based, and assumes partial state observability. Traditionally, the solution concept used for stochastic games is that of Nash equilibria. However, as we mentioned before, and as has been pointed out by some game theorists (Binmore, 1990; Kadane & Larkey, 1982), while Nash equilibria are useful for describing a multiagent system when, and if, it has reached a stable state, this solution concept is not sufficient as a general control paradigm. The main reasons are that there may be multiple equilibria with no clear way to choose among them (non-uniqueness), and the fact that equilibria do not specify actions in cases in which agents believe that other agents may not act according to their equilibrium strategies (incompleteness).

Other extensions of POMDPs to multiple agents appeared in (Bernstein, Givan, Immerman, & Zilberstein, 2002; Nair, Tambe, Yokoo, Pynadath, & Marsella, 2003). They have been called decentralized POMDPs (DEC-POMDPs), and are related to decentralized control problems (Ooi & G.W.Wornell, 1996). DEC-POMDP framework assumes that the agents are fully cooperative, i.e., they have common reward function and form a team. Furthermore, it is assumed that the optimal joint solution is computed centrally and then distributed among the agents.

From the game-theoretic side, we are motivated by the subjective approach to probability in games (Kadane & Larkey, 1982), Bayesian games of incomplete information (see Section 3.1.2 of Chapter 3 and and references therein), work on interactive belief systems (Harsanyi, 1967; Mertens & Zamir, 1985; Brandenburger & Dekel, 1993; Fagin et al., 1995; Aumann, 1999; Fagin, Geanakoplos, Halpern, & Vardi, 1999), and insights from research on learning in game theory (Section 3.2 of Chapter 3). We relax assumptions of these previous works such as perfect monitoring, and perfect observability of state. Our approach, closely related

to decision-theoretic (Myerson, 1991), and epistemic (Ambruster & Boge, 1979; Battigalli, 1996; Brandenburger, 2002) approaches to game theory, consists of predicting actions of other agents given all available information, and then of choosing the agent's own action (Kadane & Larkey, 1982). Thus, the descriptive aspect of decision theory is used to predict others' actions, and its prescriptive aspect is used to select agent's own optimal action.

The work presented here also extends previous work on the Recursive Modeling Method (RMM) (Gmytrasiewicz & Durfee, 2000), but adds elements of sequential update and planning.

4.2 Agent Types and Frames

The POMDP definition includes parameters that permit us to compute an agent's optimal behavior,[1] conditioned on its beliefs. Let us collect these implementation independent factors into a construct we call an agent i's *type*.

Definition 4.1 (Type). *A type of an agent i is, $\theta_i = \langle b_i, A_i, \Omega_i, T_i, O_i, R_i, OC_i \rangle$, where b_i is agent i's state of belief (an element of $\Delta(S)$), OC_i is its optimality criterion, and the rest of the elements are as defined before in Section 2.1.1 of Chapter 2. Let Θ_i be the set of agent i's types.*

Given type, θ_i, and the assumption that the agent is Bayesian-rational, the set of agent's optimal actions will be denoted as $OPT(\theta_i)$. In the next section, we generalize the notion of type to situations which include interactions with other agents; it then coincides with the notion of type used in Bayesian games (see Section 3.1.2 of Chapter 3).

It is convenient to define the notion of a *frame*, $\widehat{\theta}_i$, of agent i:

Definition 4.2 (Frame). *A frame of an agent i is, $\widehat{\theta}_i = \langle A_i, \Omega_i, T_i, O_i, R_i, OC_i \rangle$. Let $\widehat{\Theta}_i$ be the set of agent i's frames.*

For brevity one can write a type as consisting of an agent's belief together with its frame: $\theta_i = \langle b_i, \widehat{\theta}_i \rangle$.

In the context of the tiger game described previously in Section 2.2.2 of Chapter 2, the agent's type describes the agent's actions and their results, the quality of the agent's hearing, its payoffs, and its belief about the tiger location.

[1] The issue of computability of solutions to POMDPs has been a subject of much research (Papadimitriou & Tsitsiklis, 1987b; Madani et al., 2003). It is of obvious importance when one uses POMDPs to model agents; we return to this issue later.

Realistically, apart from the implementation-independent factors grouped in a type, an agent's behavior may also depend on implementation-specific parameters, like the processor speed, memory available, etc. These can be included in the (implementation dependent, or *complete*) type, increasing the accuracy of predicted behavior, but at the cost of additional complexity. Definition and use of complete types is a topic of ongoing work.

4.3 Interactive POMDPs

As we mentioned, our intention is to generalize POMDPs to handle the presence of other agents. We do this by including descriptions of other agents (their types for example) in the state space. We formally define the I-POMDP framework below.

4.3.1 Definition

For simplicity of presentation, we again consider an agent i, that is interacting with one other agent, j. The formalism easily generalizes to larger number of agents.

Definition 4.3 (I-POMDP). *An* interactive POMDP *of agent i, $I\text{-}POMDP_i$, is:*

$$I\text{-}POMDP_i = \langle IS_i, A, T_i, \Omega_i, O_i, R_i \rangle \tag{4.1}$$

where:

- IS_i is a set of **interactive** states defined as $IS_i = S \times M_j$,[2] interacting with agent i, where S is the set of states of the physical environment, and M_j is the set of possible models of agent j. Each model, $m_j \in M_j$, is defined as a triple $m_j = \langle h_j, f_j, O_j \rangle$, where $f_j : H_j \rightarrow \Delta(A_j)$ is agent j's function, assumed computable, which maps possible histories of j's observations to distributions over its actions. h_j is an element of H_j, and O_j is a function specifying the way the environment is supplying the agent with its input. Sometimes we write model m_j as $m_j = \langle h_j, \widehat{m}_j \rangle$, where $\widehat{m}_j$ consists of f_j and O_j. It is convenient to subdivide the set of models into two classes. The subintentional models, SM_j, are relatively simple, while the intentional models, IM_j, use the notion of rationality to model the other agent. Thus, $M_j = IM_j \cup SM_j$.

[2] If there are more agents, say $N > 2$, then $IS_i = S \times_{j=1}^{N-1} M_j$.

Simple examples of subintentional models include a no-information model and the fictitious play model, both of which are history independent. A no-information model (Gmytrasiewicz & Durfee, 2000) assumes that each of the agent j's actions is executed with equal probability. Fictitious play (see Section 3.2.1 of Chapter 3) assumes that j chooses actions according to a fixed but unknown distribution, and that i's prior belief over that distribution takes the form of a Dirichlet distribution.[3] An example of a more powerful subintentional model is a finite state machine.

The intentional models are more sophisticated in that they ascribe to the other agent beliefs, preferences, and rationality in action selection.[4] Intentional models are thus j's types, $\theta_j = \langle b_j, \widehat{\theta}_j \rangle$, under the assumption that agent j is Bayesian-rational.[5] We introduced types in Section 3.1.2 of Chapter 3 where they encompassed the agent's reward function only. Here – in the context of partially observable stochastic games – the agent's I-POMDP captures all relevant aspects of its decision-making process. Agent j's belief is a probability distribution over states of the environment and the models of the agent i; $b_j \in \Delta(S \times M_i)$. In particular, if agents' beliefs are private information, then their types involve possibly infinitely nested beliefs over others' types and their beliefs about others (Mertens & Zamir, 1985; Brandenburger & Dekel, 1993; Aumann, 1999; Aumann & Heifetz, 2002).[6] They are related to the recursive model structures in (Gmytrasiewicz & Durfee, 2000). The definition of the interactive state space is consistent with the notion of a completely specified state space put forward by Aumann (1999). Similar state spaces have been proposed in (Mertens & Zamir, 1985; Brandenburger & Dekel, 1993).

- $A = A_i \times A_j$ is the set of joint moves of all agents

- T_i is the transition model. The usual way to define the transition probabilities in POMDPs is to assume that the agent's actions can change any aspect of the state description. In case of I-POMDPs, this would mean actions modifying any aspect of the interactive states, including other agents' observation histories and their functions, or, if they are modeled intentionally, their beliefs and reward functions. Allowing agents to directly manipulate other agents in such ways, however, violates the notion of agents' autonomy. Thus, we make the following simplifying assumption:

[3] Technically, according to our notation, fictitious play is actually an ensemble of models.

[4] (Dennett, 1986) advocates ascribing rationality to other agent(s), and calls it "assuming an intentional stance towards them".

[5] Note that the space of types is by far richer than that of computable models. In particular, since the set of computable models is countable and the set of types is uncountable, many types are not computable models.

[6] Implicit in the definition of interactive beliefs is the assumption of coherency (Brandenburger & Dekel, 1993).

Assumption 4.1 (Model Non-manipulability Assumption (MNM)). *Agents' actions do not change the other agents' models directly.*

Given this simplification, the transition model can be defined as $T_i : S \times A \times S \rightarrow [0, 1]$

Autonomy, formalized by the MNM assumption, precludes, for example, direct "mind control", and implies that other agents' belief states can be changed only indirectly, typically by changing the environment in a way observable to them. In other words, agents' beliefs change, like in POMDPs, but as a result of belief update after an observation, not as a direct result of any of the agents' actions[7]

- Ω_i is defined as before in the POMDP model
- O_i is an observation function. In defining this function we make the following assumption:

  Assumption 4.2 (Model Non-observability (MNO)). *Agents cannot observe other's models directly.*

  Given this assumption the observation function is defined as $O_i : S \times A \times \Omega_i \rightarrow [0, 1]$.

  The MNO assumption formalizes another aspect of autonomy – agents are autonomous in that their observations and functions, or beliefs and other properties, say preferences, in intentional models, are private and the other agents cannot observe them directly[8]

- R_i is defined as $R_i : IS_i \times A \rightarrow \mathbf{R}$. We allow the agent to have preferences over physical states and models of other agents, but usually only the physical state will matter

As we mentioned, we see interactive POMDPs as a subjective counterpart to an objective external view in stochastic games (Fudenberg & Tirole, 1991), to approaches in (Boutilier, 1999) and (Koller & Milch, 2001), and to decentralized POMDPs (Bernstein et al., 2002; Nair et al., 2003). Interactive POMDPs represent an individual agent's point of view on the environment and the other agents, and facilitate planning and problem solving at the agent's own individual level.

[7] The possibility that agents can influence the observational capabilities of other agents can be accommodated by including the factors that can change sensing capabilities in the set S.

[8] Again, the possibility that agents can observe factors that may influence the observational capabilities of other agents is allowed by including these factors in S.

4.3.2 Belief Update in I-POMDPs

We will show that, as in POMDPs, an agent's beliefs over their interactive states are sufficient statistics, i.e., they fully summarize the agent's observation histories. Further, we need to show how beliefs are updated after the agent's action and observation, and how solutions are defined.

The new belief state, b_i^t, is a function of the previous belief state, b_i^{t-1}, the last action, a_i^{t-1}, and the new observation, o_i^t, just as in POMDPs. There are two differences that complicate belief update when compared to POMDPs. First, since the state of the physical environment depends on the actions performed by both agents the prediction of how the physical state changes has to be made based on the probabilities of various actions of the other agent. The probabilities of other's actions are obtained based on their models. Thus, unlike in Bayesian and stochastic games, we do not assume that actions are fully observable by other agents. Rather, agents can attempt to infer what actions other agents have performed by sensing their results on the environment. Second, changes in the models of other agents have to be included in the update. These reflect the other's observations and, if they are modeled intentionally, the update of the other agent's beliefs. In this case, the agent has to update its beliefs about the other agent based on what it anticipates the other agent observes and how it updates. As could be expected, the update of the possibly infinitely nested belief over other's types is, in general, only asymptotically computable.

Proposition 4.1. (Sufficiency) *In an interactive POMDP of agent i, i's current belief, i.e., the probability distribution over the set $S \times M_j$, is a sufficient statistic for the past history of i's observations.*

The next proposition defines the agent i's belief update function, $b_i^t(is^t) = Pr(is^t|o_i^t, a_i^{t-1}, b_i^{t-1})$, where $is^t \in IS_i$ is an interactive state. We use the belief state estimation function, SE_{θ_i}, as an abbreviation for belief updates for individual states so that $b_i^t = SE_{\theta_i}(b_i^{t-1}, a_i^{t-1}, o_i^t)$. $\tau_{\theta_i}(b_i^{t-1}, a_i^{t-1}, o_i^t, b_i^t)$ will stand for $Pr(b_i^t|b_i^{t-1}, a_i^{t-1}, o_i^t)$. Further below we also define the set of type-dependent optimal actions of an agent, $OPT(\theta_i)$.

Proposition 4.2. (Belief Update) *Under the MNM and MNO assumptions, the belief update function for an interactive POMDP $\langle IS_i, A, T_i, \Omega_i, O_i, R_i \rangle$, when m_j in is^t is intentional, is:*

$$\begin{aligned} b_i^t(is^t) &= \beta \sum_{is^{t-1}:\widehat{m}_j^{t-1}=\widehat{\theta}_j^t} b_i^{t-1}(is^{t-1}) \sum_{a_j^{t-1}} Pr(a_j^{t-1}|\theta_j^{t-1}) O_i(s^t, a^{t-1}, o_i^t) \\ &\quad \times T_i(s^{t-1}, a^{t-1}, s^t) \sum_{o_j^t} \tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t) O_j(s^t, a^{t-1}, o_j^t) \end{aligned} \tag{4.2}$$

When m_j in is^t is subintentional the first summation extends over $is^{t-1} : \widehat{m}_j^{t-1} = \widehat{m}_j^t$, $Pr(a_j^{t-1}|\theta_j^{t-1})$ is replaced with $Pr(a_j^{t-1}|m_j^{t-1})$, and $\tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t)$ is replaced with the Kronecker delta function $\delta_K(APPEND(h_j^{t-1}, o_j^t) - h_j^t)$.

Above, b_j^{t-1} and b_j^t are the belief elements of θ_j^{t-1} and θ_j^t, respectively, β is a normalizing constant, and $Pr(a_j^{t-1}|\theta_j^{t-1})$ is the probability that a_j^{t-1} is Bayesian rational for the agent described by type θ_j^{t-1}. This probability is equal to $\frac{1}{|OPT(\theta_j)|}$ if $a_j^{t-1} \in OPT(\theta_j)$, and it is equal to zero otherwise. We define OPT in Section 4.3.3.[9] For the case of j's subintentional model, $is = (s, m_j)$, h_j^{t-1} and h_j^t are the observation histories which are part of m_j^{t-1}, and m_j^t respectively, O_j is the observation function in m_j^t, and $Pr(a_j^{t-1}|m_j^{t-1})$ is the probability assigned by m_j^{t-1} to a_j^{t-1}. APPEND returns a string with the second argument appended to the first. To maintain clarity of the exposition, the proofs of the propositions are presented in Appendix A. A two time-slice DBN that represents the belief update in I-POMDPs is given in Fig. 4.1.

Proposition 4.2 and Eq. 4.2 have a lot in common with the belief update in POMDPs (Eq. 2.1), as should be expected. Both depend on agent i's observation and transition functions. However, since agent i's observations also depend on agent j's actions, the probabilities of various actions of j have to be included (in the first line of Eq. 4.2.) Further, since the update of agent j's model depends on what j observes, the probabilities of various observations of j have to be included (in the second line of Eq. 4.2.) The update of j's beliefs is represented by the τ_{θ_j} term. Since the agent i's beliefs could be infinitely nested, the belief update in I-POMDPs can, in general, be calculated only asymptotically. The belief update can easily be generalized to the setting where more than one other agents co-exist with agent i.

[9] If the agent's prior belief over IS_i is given by a probability density function then the $\sum_{is^{t-1}}$ is replaced by an integral. In that case $\tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t)$ takes the form of Dirac delta function over argument b_j^{t-1}: $\delta_D(SE_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t) - b_j^t)$.

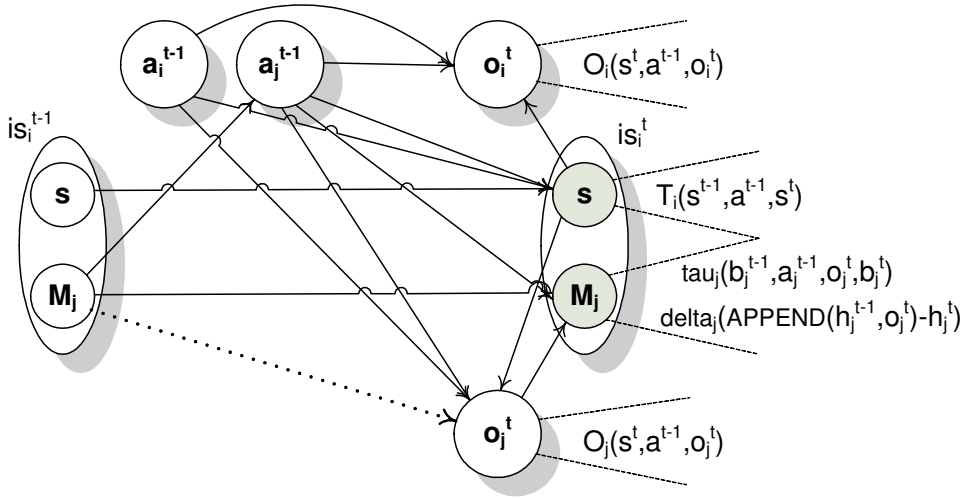

Figure 4.1: A 2-time slice DBN that graphically illustrates the belief update in I-POMDPs. The posterior belief is a distribution over the shaded random variables. The dashed lines enclose the functions that form the CPTs for the respective random variables. If $m_j \in M_j$ under consideration is intentional, then τ forms the CPT, otherwise δ. The dotted link indicates that the function O_j is the one that is contained in m_j. Causal links within the interactive states have been omitted for clarity.

4.3.3 Value Functions and Solutions to I-POMDPs

Analogously to POMDPs, each belief state in I-POMDP has an associated value reflecting the maximum payoff the agent can expect in this belief state:

$$U(\theta_i) = \max_{a_i \in A_i} \left\{ \sum_{is} ER_i(is, a_i) b_i(is) + \gamma \sum_{o_i \in \Omega_i} Pr(o_i|a_i, b_i) U(\langle SE_{\theta_i}(b_i, a_i, o_i), \widehat{\theta}_i \rangle) \right\} \tag{4.3}$$

where, $ER_i(is, a_i) = \sum_{a_j} R_i(is, a_i, a_j) Pr(a_j|m_j)$. Eq. 4.3 is a basis for value iteration in I-POMDPs.

Agent i's optimal action, a_i^*, for the case of infinite horizon criterion with discounting, is an element of the set of optimal actions for the belief state, $OPT(\theta_i)$, defined as:

$$OPT(\theta_i) = \underset{a_i \in A_i}{argmax} \left\{ \sum_{is} ER_i(is, a_i) b_i(is) + \gamma \sum_{o_i \in \Omega_i} Pr(o_i|a_i, b_i) U(\langle SE_{\theta_i}(b_i, a_i, o_i), \widehat{\theta}_i \rangle) \right\} \tag{4.4}$$

As in the case of the belief update, due to possibly infinitely nested beliefs, a step of value iteration and optimal actions are only asymptotically computable.

4.4 Finitely Nested I-POMDPs

Possible infinite nesting of agents' beliefs in intentional models presents an obvious obstacle to computing the belief updates and optimal solutions. Since the models of agents with infinitely nested beliefs correspond to agent functions which are not computable, it is natural to consider finite nestings.

4.4.1 Definition

We follow (Aumann, 1999; Brandenburger & Dekel, 1993; Fagin et al., 1999), extend (Gmytrasiewicz & Durfee, 2000), and construct finitely nested I-POMDPs bottom-up. Assume a set of physical states of the world S, and two agents i and j. Agent i's 0^{th} level beliefs, $b_{i,0}$, are probability distributions over S. Its 0^{th} level types, $\Theta_{i,0}$, contain its 0^{th} level beliefs, and its frames, and analogously for agent j. 0-level types are, therefore, POMDPs.[10] 0-level models include 0-level types (i.e., intentional models) and the subintentional models, SM. An agent's first level beliefs are probability distributions over physical states and 0-level models of the other agent. An agent's first level types consist of its first level beliefs and frames. Its first level models consist of the types upto level 1 and the subintentional models. Second level beliefs are defined in terms of first level models and so on. Formally, define spaces:

$$\begin{array}{lllllllllll}
IS_{i,0} & = & S, & \Theta_{j,0} & = & \{\langle b_{j,0}, \widehat{\theta}_j\rangle : b_{j,0} \in \Delta(IS_{j,0})\}, & M_{j,0} & = & \Theta_{j,0} \cup SM_j \\
IS_{i,1} & = & S \times M_{j,0}, & \Theta_{j,1} & = & \{\langle b_{j,1}, \widehat{\theta}_j\rangle : b_{j,1} \in \Delta(IS_{j,1})\}, & M_{j,1} & = & \Theta_{j,1} \cup M_{j,0} \\
 & . & & & . & & & & \\
 & . & & & . & & & & \\
 & . & & & . & & & & \\
IS_{i,l} & = & S \times M_{j,l-1}, & \Theta_{j,l} & = & \{\langle b_{j,l}, \widehat{\theta}_j\rangle : b_{j,l} \in \Delta(IS_{j,l})\}, & M_{j,l} & = & \Theta_{j,l} \cup M_{j,l-1}
\end{array}$$

Definition 4.4. (Finitely Nested I-POMDP) *A finitely nested I-POMDP of agent i, I-POMDP$_{i,l}$, is:*

$$\text{I-POMDP}_{i,l} = \langle IS_{i,l}, A, T_i, \Omega_i, O_i, R_i\rangle \tag{4.5}$$

The parameter l will be called the *strategy level* of the finitely nested I-POMDP. The belief update, value function, and the optimal actions for finitely nested I-POMDPs are computed using Equation 4.2 and Equation 4.4, but recursion is guaranteed to terminate at the 0^{th} level and subintentional models.

[10] In 0-level types the other agent's actions are folded into the T, O and R functions as noise.

Agents which are more strategic are capable of modeling others at deeper levels (i.e., all levels up to their own strategy level l), but are always only boundedly optimal. As such, these agents could fail to predict the strategy of a more sophisticated opponent. The fact that the computability of an agent function implies that the agent may be suboptimal during interactions has been pointed out by Binmore (1990), and proved more recently in (Nachbar & Zame, 1996). Intuitively, the difficulty is that an agent's unbounded optimality would have to include the capability to model the other agent's modeling the original agent. This leads to an impossibility result due to self-reference, which is very similar to Gödel's incompleteness theorem and the halting problem (Brandenburger, 2002).

As we mentioned, the 0^{th} level intentional models are POMDPs. They provide probability distributions over actions of the agent modeled at that level to models with strategy level of 1. Given probability distributions over other agent's actions, the level 1 models can themselves be solved as POMDPs, and provide probability distributions to yet higher level models. Assume that the number of intentional models considered at each level is bound by a number, $|\Theta|$. Solving an $\texttt{I-POMDP}_{i,l}$ in then equivalent to solving $O(|\Theta|^l)$ POMDPs. If there are K other agents, then we must solve $O((K|\Theta|^K)^l)$ POMDPs. Hence, the complexity of solving an $\texttt{I-POMDP}_{i,l}$ is PSPACE-hard for finite time horizons [11], and undecidable for infinite horizons, just like for POMDPs.

4.4.2 Properties

In this section we establish two important properties, namely convergence of value iteration and piecewise linearity and convexity of the value function, for finitely nested I-POMDPs. These properties are analogous to those of POMDPs (see Section 2.1.2 of Chapter 2), and subsequently their proofs follow the same approach. All proofs are given in Appendix A.

Convergence of Value Iteration:

For an agent i and its $\texttt{I-POMDP}_{i,l}$, we can show that the sequence of value functions, $\{U^n\}$, where n is the horizon, obtained by value iteration defined in Eq. 4.3, converges to a unique fixed-point, U^*.

[11] Usually PSPACE-complete since the number of states in I-POMDPs is likely to be larger than the time horizon (Papadimitriou & Tsitsiklis, 1987b).

Let us define a *backup* operator $H : B(\Theta_i) \rightarrow B(\Theta_i)$ such that $U^n = HU^{n-1}$, and $B(\Theta_i)$ is the set of all bounded value functions. In order to prove the convergence result, we first establish some of the properties of H.

Lemma 4.1 (Isotonicity). *For any finitely nested* $\texttt{I-POMDP}_{i,l}$ *value functions* V *and* U*, if* $V \leq U$*, then* $HV \leq HU$.

Another important property exhibited by the backup operator is the property of contraction.

Lemma 4.2 (Contraction). *For any finitely nested* $\texttt{I-POMDP}_{i,l}$ *value functions* V*,* U *and a discount factor* $\gamma \in (0, 1)$*,* $||HV - HU|| \leq \gamma||V - U||$.

The proof of this lemma makes use of Lemma 4.1. $||\cdot||$ is the supremum norm.

Under the contraction property of H, and noting that the space of value functions along with the supremum norm forms a complete normed space (Banach space), we can apply the Banach fixed-point theorem to show that value iteration for I-POMDPs converges to a unique fixed-point (optimal solution). The following theorem captures this result.

Theorem 4.1 (Convergence). *For any finitely nested* $\texttt{I-POMDP}_{i,l}$*, the value iteration algorithm starting from any arbitrary well-defined value function converges to a unique fixed-point.*

The detailed proof of this theorem is included in Appendix A.

As in the case of POMDPs (Russell & Norvig, 2003), the error in the iterative estimates, U^n, for finitely nested I-POMDPs, i.e., $||U^n - U^*||$, is reduced by the factor of at least γ on each iteration. Hence, the number of iterations, N, needed to reach an error of at most ϵ is:

$$N = \lceil \log(R_{max}/\epsilon(1-\gamma))/\log(1/\gamma) \rceil \tag{4.6}$$

where R_{max} is the upper bound of the reward function.

Piecewise Linearity and Convexity:

Another property that carries over from POMDPs to finitely nested I-POMDPs is the piecewise linearity and convexity (PWLC) of the value function. Establishing this property allows us to decompose the I-POMDP value function into a set of *alpha* vectors, each of which represents a policy tree. The PWLC property enables

us to work with sets of alpha vectors rather than perform value iteration over the continuum of agent types. Theorem 4.2 below states the PWLC property of the I-POMDP value function.

Theorem 4.2 (PWLC). *For any finitely nested I-POMDP$_{i,l}$, U^n is piecewise linear and convex.*

The complete proof of Theorem 4.2 is included in Appendix A. The proof is similar to the one shown in Section 2.1.2 for POMDPs and proceeds by induction. The basis case is established by considering the horizon 1 value function. Showing the PWLC for the inductive step requires substituting the belief update (Eq. 4.2) into Eq. 4.3, followed by factoring out the belief from both terms of the equation.

4.5 Example: The Multiagent Tiger Problem

To illustrate optimal sequential behavior of agents in multiagent settings we apply our I-POMDP framework to the multiagent tiger problem, a traditional version of which we described before in Section 2.2.2.

4.5.1 Definition

Let us denote the actions of opening doors and listening as OR, OL and L, as before. TL and TR denote states corresponding to tiger located behind the left and right door, respectively. The transition, reward and observation functions depend now on the actions of both agents. Again, we assume that the tiger location is chosen randomly in the next time step if any of the agents opened any doors in the current step. We also assume that the agent hears the tiger's growls, GR and GL, with the accuracy of 85%. To make the interaction more interesting we added an observation of door creaks, which depend on the action executed by the other agent. Creak right, CR, is likely due to the other agent having opened the right door, and similarly for creak left, CL. Silence, S, is a good indication that the other agent did not open doors and listened instead. See Fig. 4.2 for an illustration. We assume that the agent's payoffs are identical to the single agent versions described in Section 2.2.2 to make these cases comparable. Note that the result of this assumption is that the other agent's actions do not impact the original agent's payoffs directly, but rather indirectly by resulting in states that matter to the original agent. Table 4.1 quantifies these factors.

When an agent makes its choice in the multiagent tiger problem, it may find it useful to consider what it believes about the location of the tiger, as well as whether the other agent will listen or open a door, which in turn depends on the other agent's beliefs, reward function, optimality criterion, etc. [12] In particular, if

[12] We assume an intentional model of the other agent here.

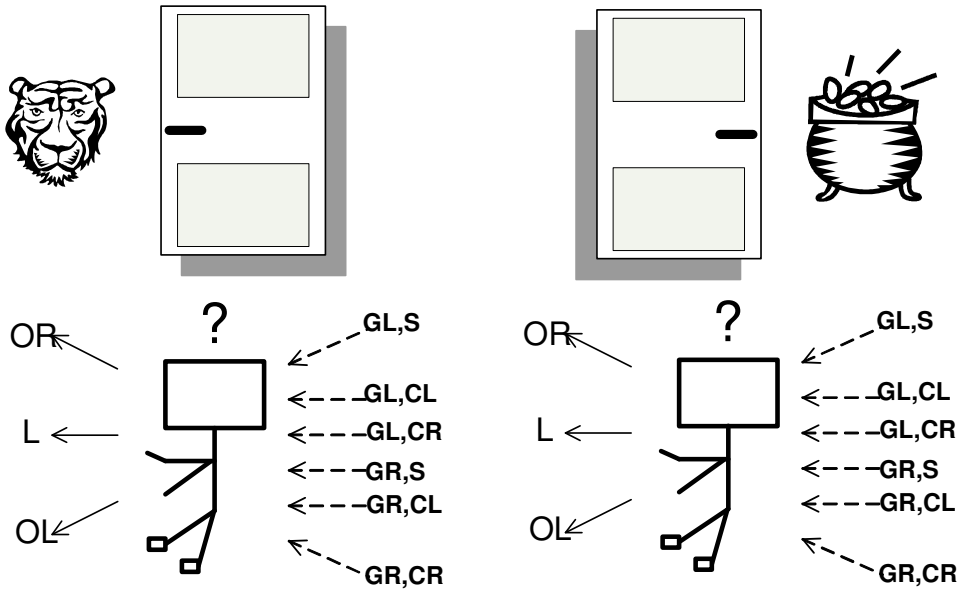

Figure 4.2: An illustration of the multiagent tiger problem. At each step, each agent must make a decision: open the left door (OL), listen (L), or open the right door (OR). To aid its decision, the agent receives one of six observations – a combination of the tiger's growls and creaks resulting from the other agent opening doors or listening. Note that the agents' sensors are noisy.

the other agent were to open any of the doors the tiger's location in the next time step would be chosen randomly. Therefore, the information obtained from any previous observations has to be discarded. We will simplify the situation by considering i's `I-POMDP` with a single level of nesting, assuming that all of the agent j's properties, except for beliefs, are known to i, and that j's time horizon is equal to i's. In other words, i's uncertainty pertains only to j's beliefs and not to its frame. Agent i's interactive state space is, $IS_{i,1} = S \times \Theta_{j,0}$, where S is the physical state, S=$\{$TL, TR$\}$, and $\Theta_{j,0}$ is a set of intentional models of agent j's, each of which differs only in j's beliefs over the location of the tiger.

4.5.2 Modeling the Other Agent as Static Noise

We may apply POMDPs directly to multiagent settings by ascribing a stationary behavior to the other agent j, and folding (marginalizing) the behavior into the agent i's POMDP definition. This is akin to treating j as noise in the environment. For illustration, let us assume that i thinks that j listens with a probability of 0.8, opens the left door with a probability of 0.1, and opens the right door with a probability of 0.1. The result of this noise is that i's transition function changes: even when i is listening, one of the two doors may open,

$\langle a_i, a_j \rangle$	State	TL	TR
$\langle$ OL,* $\rangle$	*	0.5	0.5
$\langle$ OR,* $\rangle$	*	0.5	0.5
$\langle$ *,OL $\rangle$	*	0.5	0.5
$\langle$ *,OR $\rangle$	*	0.5	0.5
$\langle$ L,L $\rangle$	TL	1.0	0
$\langle$ L,L $\rangle$	TR	0	1.0

Transition function: $T_i = T_j$

$\langle a_i, a_j \rangle$/State	TL	TR
$\langle$ OR,OR $\rangle$	10	-100
$\langle$ OL,OL $\rangle$	-100	10
$\langle$ OR,OL $\rangle$	10	-100
$\langle$ OL,OR $\rangle$	-100	10
$\langle$ L,L $\rangle$	-1	-1
$\langle$ L,OR $\rangle$	-1	-1
$\langle$ OR,L $\rangle$	10	-100
$\langle$ L,OL $\rangle$	-1	-1
$\langle$ OL,L $\rangle$	-100	10

$\langle a_i, a_j \rangle$/State	TL	TR
$\langle$ OR,OR $\rangle$	10	-100
$\langle$ OL,OL $\rangle$	-100	10
$\langle$ OR,OL $\rangle$	-100	10
$\langle$ OL,OR $\rangle$	10	-100
$\langle$ L,L $\rangle$	-1	-1
$\langle$ L,OR $\rangle$	10	-100
$\langle$ OR,L $\rangle$	-1	-1
$\langle$ L,OL $\rangle$	-100	10
$\langle$ OL,L $\rangle$	-1	-1

Reward functions of agents i and j

$\langle a_i, a_j \rangle$	State	$\langle$ GL, CL $\rangle$	$\langle$ GL, CR $\rangle$	$\langle$ GL, S $\rangle$	$\langle$ GR, CL $\rangle$	$\langle$ GR, CR $\rangle$	$\langle$ GR, S $\rangle$
$\langle$L,L$\rangle$	TL	0.85*0.05	0.85*0.05	0.85*0.9	0.15*0.05	0.15*0.05	0.15*0.9
$\langle$L,L$\rangle$	TR	0.15*0.05	0.15*0.05	0.15*0.9	0.85*0.05	0.85*0.05	0.85*0.9
$\langle$L,OL$\rangle$	TL	0.85*0.9	0.85*0.05	0.85*0.05	0.15*0.9	0.15*0.05	0.15*0.05
$\langle$L,OL$\rangle$	TR	0.15*0.9	0.15*0.05	0.15*0.05	0.85*0.9	0.85*0.05	0.85*0.05
$\langle$L,OR$\rangle$	TL	0.85*0.05	0.85*0.9	0.85*0.05	0.15*0.05	0.15*0.9	0.15*0.05
$\langle$L,OR$\rangle$	TR	0.15*0.05	0.15*0.9	0.15*0.05	0.85*0.05	0.85*0.9	0.85*0.05
$\langle$OL,*$\rangle$	$*$	1/6	1/6	1/6	1/6	1/6	1/6
$\langle$OR,*$\rangle$	$*$	1/6	1/6	1/6	1/6	1/6	1/6

$\langle a_i, a_j \rangle$	State	$\langle$ GL, CL $\rangle$	$\langle$ GL, CR $\rangle$	$\langle$ GL, S $\rangle$	$\langle$ GR, CL $\rangle$	$\langle$ GR, CR $\rangle$	$\langle$ GR, S $\rangle$
$\langle$L,L$\rangle$	TL	0.85*0.05	0.85*0.05	0.85*0.9	0.15*0.05	0.15*0.05	0.15*0.9
$\langle$L,L$\rangle$	TR	0.15*0.05	0.15*0.05	0.15*0.9	0.85*0.05	0.85*0.05	0.85*0.9
$\langle$OL,L$\rangle$	TL	0.85*0.9	0.85*0.05	0.85*0.05	0.15*0.9	0.15*0.05	0.15*0.05
$\langle$OL,L$\rangle$	TR	0.15*0.9	0.15*0.05	0.15*0.05	0.85*0.9	0.85*0.05	0.85*0.05
$\langle$OR,L$\rangle$	TL	0.85*0.05	0.85*0.9	0.85*0.05	0.15*0.05	0.15*0.9	0.15*0.05
$\langle$OR,L$\rangle$	TR	0.15*0.05	0.15*0.9	0.15*0.05	0.85*0.05	0.85*0.9	0.85*0.05
$\langle$*,OL$\rangle$	$*$	1/6	1/6	1/6	1/6	1/6	1/6
$\langle$*,OR$\rangle$	$*$	1/6	1/6	1/6	1/6	1/6	1/6

Observation functions of agents i and j.

Table 4.1: Transition, reward, and observation functions for the multiagent tiger problem.

and the tiger may change its location with a probability of 0.1. We compare the value functions obtained from the original POMDP (see Section 2.2.2), with those obtained by solving the POMDP with the noise factor.

The value function of the POMDP with noise coincides with that of the original POMDP for horizon 1 as shown in Fig. 4.3. Because this is horizon 1, listening does not provide any useful information since the problem does not continue to allow for the use of this information. Therefore, the effect of noise in the listening action is not visible.

In Fig. 4.4 we present a comparison of value functions for horizon of length 2 for the original agent, and for the agent facing the noisy environment. Consequences of folding noise are two-fold. First, the effectiveness of the agent's optimal policies declines since the value of hearing growls diminishes over many

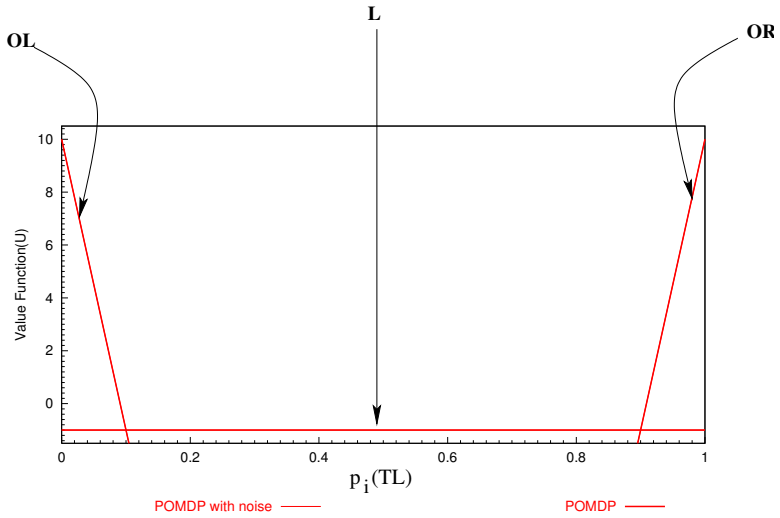

Figure 4.3: The value functions for the POMDP with the noise factor and the POMDP for the single agent tiger problem, with time horizon of length 1. Actions are: open right door - OR, open left door - OL, and listen - L. For this value of the time horizon the value function for a POMDP with noise factor is identical to the single agent POMDP.

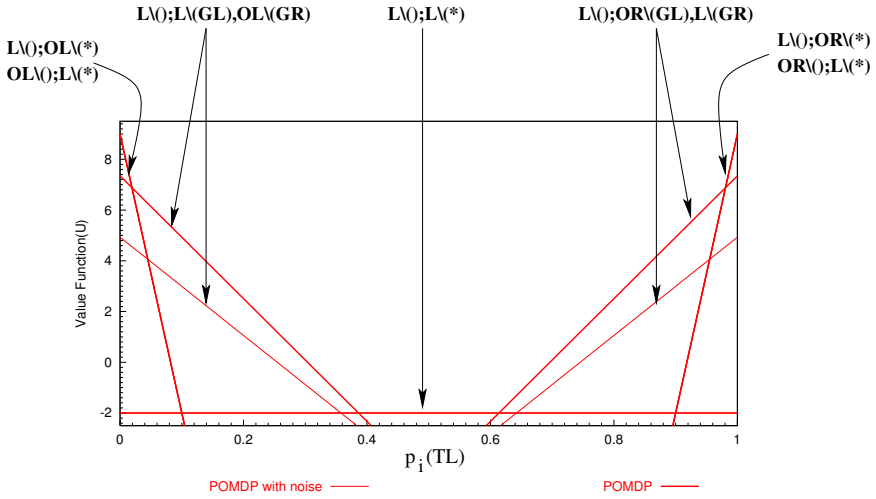

Figure 4.4: The value function for the single agent tiger problem compared to an agent facing a noise factor, for horizon of length 2. Policies corresponding to value lines are conditional plans. Actions, L, OR or OL, are conditioned on observational sequences in parenthesis. For example L\();L\(GL),OL\(GR) denotes a plan to perform the listening action, L, at the beginning (list of observations is empty), and then another L if the observation is growl from the left (GL), and open the left door, OL, if the observation is GR. $*$ is a wildcard with the usual interpretation.

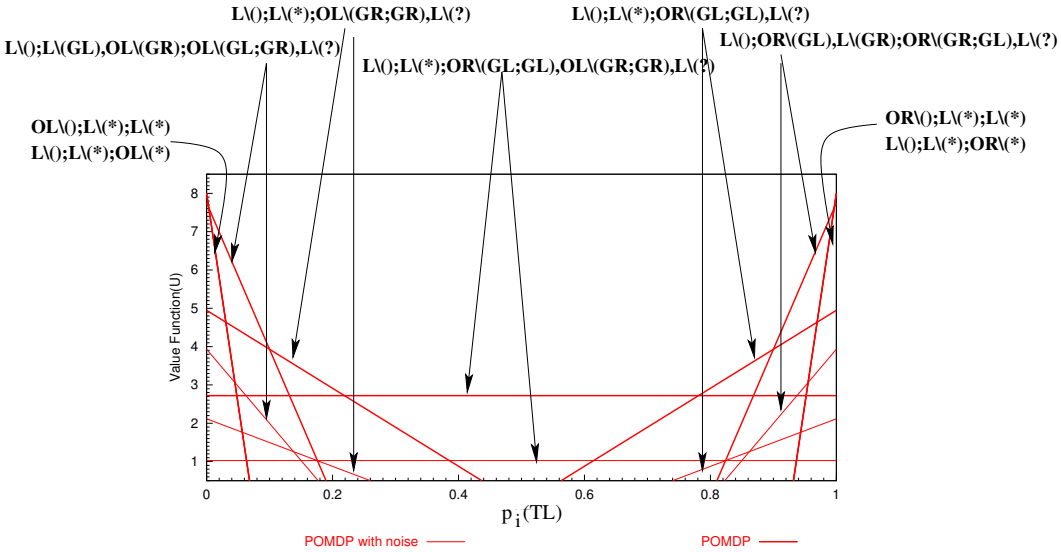

Figure 4.5: The value function for single agent tiger problem compared to an agent facing a noise factor, for horizon of length 3. The "?" in the description of a policy stands for any of the perceptual sequences not yet listed in the description of the policy.

time steps. Fig. 4.5 depicts a comparison of value functions for horizon of length 3. Here, for example, two consecutive growls in a noisy environment are not as valuable as when the agent knows it is acting alone since the noise may have perturbed the state of the system between the growls.

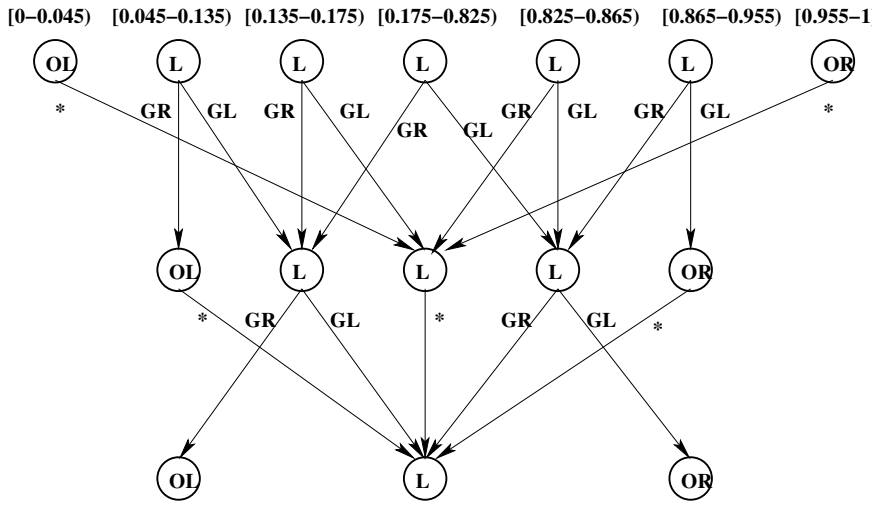

Figure 4.6: The policy graph corresponding to the horizon 3 value function of POMDP with noise depicted in Fig. 4.5.

Second, since the presence of another agent is implicit in the static transition model, agent i cannot update its model of j's actions during repeated interactions, i.e. i cannot model the learning process of j. This effect becomes more important as time horizon increases. Our approach addresses this issue by allowing explicit modeling of the other agent(s). This results in policies of superior quality, as we show later.

Figure 4.6 shows a policy for an agent facing a noisy environment for time horizon of 3. We compare it to the corresponding I-POMDP policy in Section 4.5.4. Note that it is slightly different than the policy without noise in (Kaelbling et al., 1998) due to differences in value functions.

4.5.3 Examples of the I-POMDP Belief Update

In Section 4.3.2, we presented the belief update equation for I-POMDPs (Eq. 4.2). Here we consider examples of level 1 beliefs, $b_{i,1}$, of agent i, which are probability distributions over $S \times \Theta_{j,0}$. Each 0^{th} level type of agent j, $\theta_{j,0} \in \Theta_{j,0}$, contains a "flat" belief as to the location of the tiger, which can be represented by a single probability assignment – $b_{j,0} = Pr_j(TL)$.

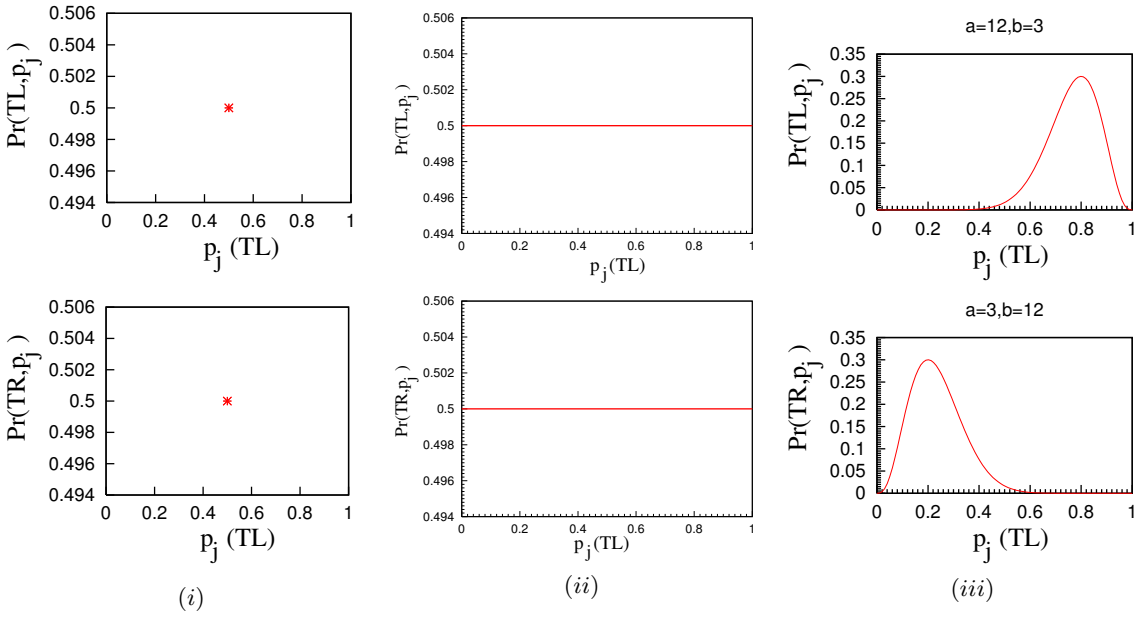

Figure 4.7: Three examples of singly nested belief states of agent i. In each case i has no information about the tiger's location. In (i) agent i knows that j does not know the location of the tiger; the single point (star) denotes a Dirac delta function which integrates to the height of the point, here 0.5. In (ii) agent i is uninformed about j's beliefs about tiger's location. In (iii) agent i believes that j is likely informed about the location of the tiger; for this case we used beta density functions, $\beta(a, b)$, for the beliefs.

In Fig. 4.7 we show some examples of level 1 beliefs of agent i. In each case i does not know the location of the tiger so that the marginals in the top and bottom sections of the figure sum up to 0.5 for probabilities of TL and TR each. In Fig. 4.7(i), i knows that j assigns 0.5 probability to tiger being behind the left door. This is represented using a Dirac delta function. In Fig. 4.7(ii), agent i is uninformed about j's beliefs. This is represented as a uniform probability density over all values of the probability j could assign to state TL. Fig. 4.7(iii) represents i's belief that j is informed about the location of the tiger. We use the beta density function to represent this belief. Modifying the values of the beta parameters, a and b, allows us to represent a wide variety of i's beliefs about j's beliefs. [13]

To make the presentation of the belief update more transparent we decompose the Eq. 4.2 into two steps:

- *Prediction:* When agent i performs an action a_i^{t-1}, and given that agent j performs a_j^{t-1}, the predicted belief state is:

$$\begin{aligned}\widehat{b}_i^t(is^t) = Pr(is^t|a_i^{t-1}, a_j^{t-1}, b_i^{t-1}) \quad &= \textstyle\sum_{is^{t-1}:\widehat{m}_j^{t-1}=\widehat{\theta}_j^t} b_i^{t-1}(is^{t-1}) Pr(a_j^{t-1}|\theta_j^{t-1}) T_i(s^{t-1}, a^{t-1}, s^t) \\ &\times \sum_{o_j^t} O_j(s^t, a^{t-1}, o_j^t) \tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t)\end{aligned} \tag{4.7}$$

- *Correction:* When agent i perceives an observation, o_i^t, the predicted belief states, $Pr(\cdot|a_i^{t-1}, a_j^{t-1}, b_i^{t-1})$, are combined according to:

$$b_i^t(is^t) = Pr(is^t|o_i^t, a_i^{t-1}, b_i^{t-1}) = \beta \sum_{a_j^{t-1}} O_i(s^t, a^{t-1}, o_i^t) Pr(is^t|a_i^{t-1}, a_j^{t-1}, b_i^{t-1}) \tag{4.8}$$

  where β is the normalizing constant.

In Fig. 4.8, we display an example trace through the update of a singly nested belief. In the first column of Fig. 4.8, labeled (a), is an example of agent i's prior belief we introduced before, according to which i knows that j is uninformed of the location of the tiger.[14] Let us assume that i listens and hears a growl from the left and no creaks, $\langle$GL,S$\rangle$. The second column of Fig. 4.8, (b), displays the *predicted* belief after i performs the listen action (Eq. 4.7). As part of the prediction step, agent i must solve j's model to obtain j's optimal action when its belief is 0.5 (term $Pr(a_j^{t-1}|\theta_j^{t-1})$ in Eq. 4.7). Given the value function in Fig. 4.4, this evaluates

[13] However, beta density functions cannot be used to represent all possible beliefs that i could have about j's beliefs. For e.g., beta cannot be used to represent multi-modal beliefs (i.e. belief shapes with more than one peak). In Chapter 6, we adopt a polynomial based representation for the nested beliefs which is more general.

[14] The points in Fig. 4.8 and in Fig. 4.9 again denote Dirac delta functions which integrate to the value equal to the points' height.

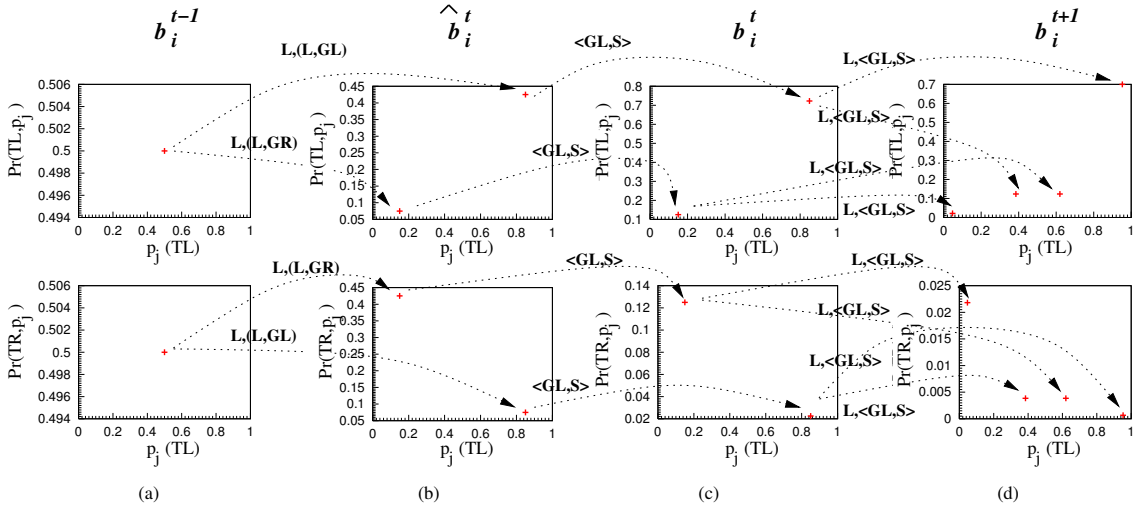

Figure 4.8: A trace of the belief update of agent i. (a) depicts the level 1 prior. (b) is the result of prediction given i's listening action, L, and a pair denoting j's action and observation. i knows that j will listen and could hear tiger's growl on the right or the left, and that the probabilities j would assign to TL are 0.15 or 0.85, respectively. (c) is the result of correction after i observes tiger's growl on the left and no creaks, $\langle$GL,S$\rangle$. The probability i assigns to TL is now greater than TR. (d) depicts the results of another update (both prediction and correction) after another listen action of i and the same observation, $\langle$GL,S$\rangle$.

to probability of 1 for listen action, and zero for opening of any of the doors. i also updates j's belief given that j listens and hears the tiger growling from either the left, GL, or right, GR, (term $\tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t)$ in Eq. 4.7). Agent j's updated probabilities for tiger being on the left are 0.85 and 0.15, for j's hearing GL and GR, respectively. If the tiger is on the left (top of Fig. 4.8 (b)) j's observation GL is more likely, and consequently j's assigning the probability of 0.85 to state TL is more likely (i assigns a probability of 0.425 to this state.) When the tiger is on the right j is more likely to hear GR and i assigns the lower probability, 0.075, to j's assigning a probability 0.85 to tiger being on the left. The third column, (c), of Fig. 4.8 shows the posterior belief after the *correction* step. The belief in column (b) is updated to account for i's hearing a growl from the left and no creaks, $\langle$GL,S$\rangle$. The resulting marginalised probability of the tiger being on the left is higher (0.85) than that of the tiger being on the right. If we assume that in the next time step i again listens and hears the tiger growling from the left and no creaks, the belief state depicted in the fourth column of Fig. 4.8 results.

In Fig. 4.9 we show the belief update starting from the prior in Fig. 4.7 (ii), according to which agent i initially has no information about what j believes about the tiger's location.

The traces of belief updates in Fig. 4.8 and Fig. 4.9 illustrate the changing state of information agent i has

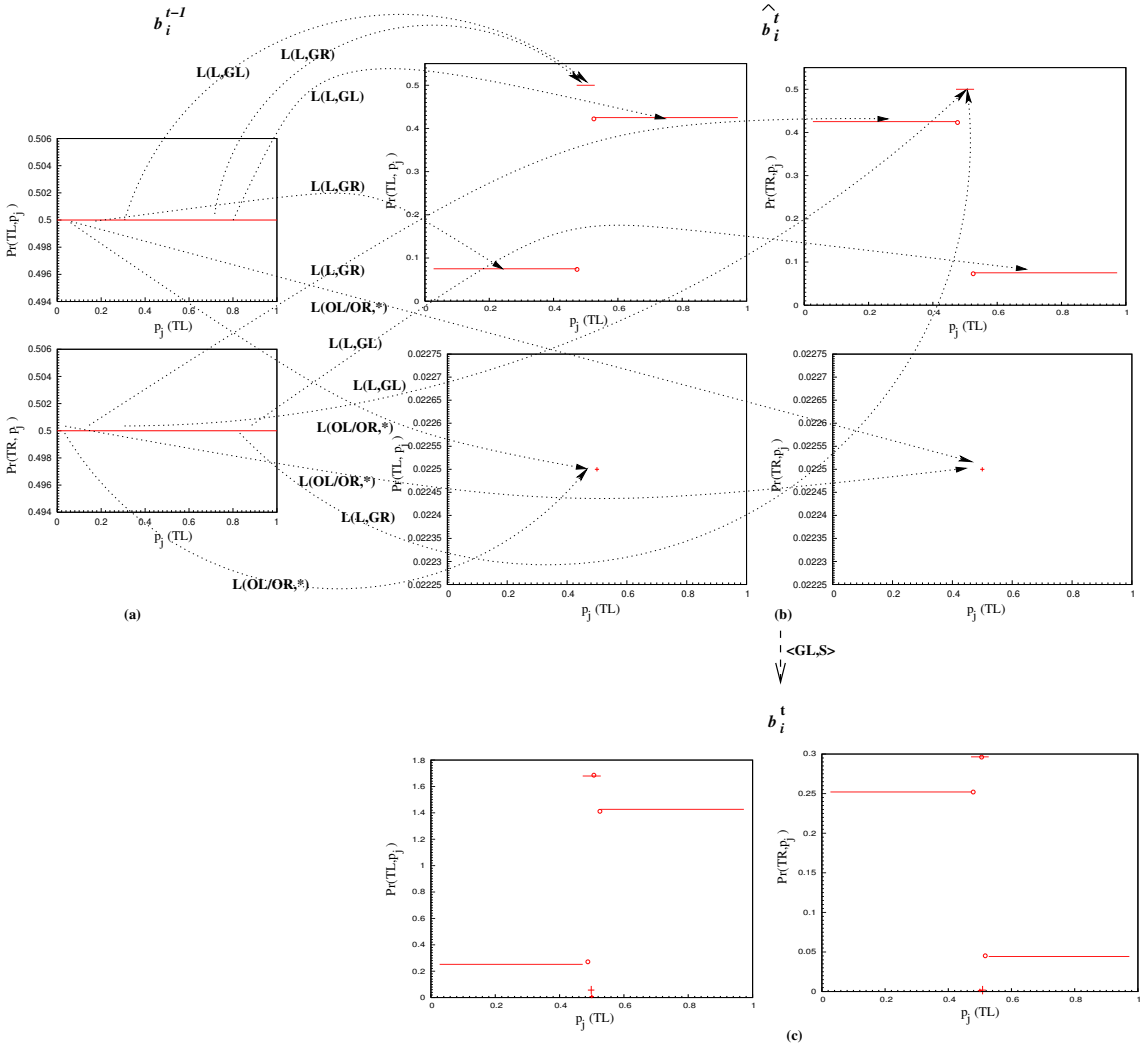

Figure 4.9: Another trace of the belief update of agent i. (a) depicts the prior according to which i is uninformed about j's beliefs. (b) is the result of the prediction step after i's listening action (L). The top half of (b) shows i's belief after it has listened and given that j also listened. The two observations j can make, GL and GR, each with probability dependent on the tiger's location, give rise to flat portions representing what i knows about j's belief in each case. The increased probability i assigns to j's belief between 0.472 and 0.528 is due to j's updates after it hears GL and after it hears GR resulting in the same values in this interval. The bottom half of (b) shows i's belief after i has listened and j has opened the left or right door (plots are identical for each action and only one of them is shown). i knows that j has no information about the tiger's location in this case. (c) is the result of correction after i observes tiger's growl on the left and no creaks $\langle$GL,S$\rangle$. The plots in (c) are obtained by performing a weighted summation of the plots in (b). The probability i assigns to TL is now greater than TR, and information about j's beliefs allows i to refine its prediction of j's action in the next time step.

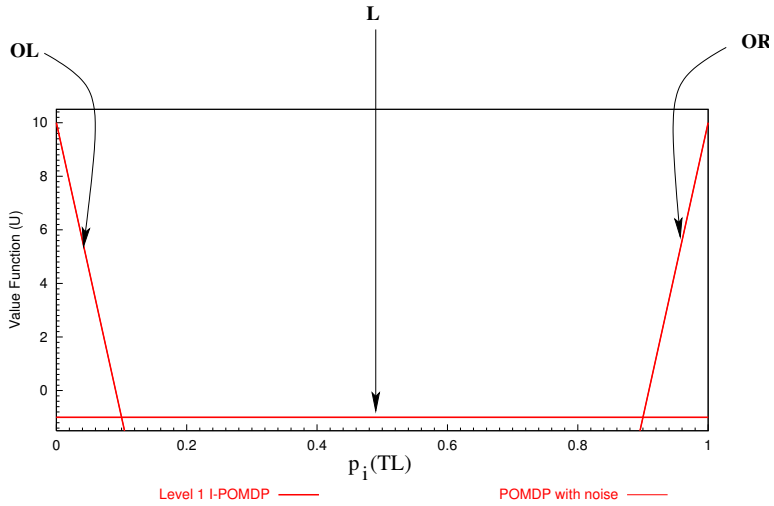

Figure 4.10: For time horizon of 1 the value functions obtained from solving a singly nested I−POMDP and a POMDP with noise factor overlap.

about the other agent's beliefs. The benefit of representing these updates explicitly is that, at each stage, i's optimal behavior depends on its estimate of probabilities of j's actions. The more informative these estimates are the more value agent i can expect out of the interaction. In the next section, we show the increase in the value function for I−POMDPs compared to POMDPs with the noise factor.

4.5.4 Examples of Value Functions

This section compares value functions obtained from solving a POMDP with a static noise factor, accounting for the presence of another agent,[15] to value functions of level-1 I−POMDP. We use incremental pruning (see Section 2.2.1 of Chapter 2) appropriately extended for the I−POMDP framework, to compute the solutions. The advantage of more refined modeling and update in I−POMDPs is due to two factors. First is the ability to keep track of the other agent's state of beliefs to better predict its future actions. The second is the ability to adjust the other agent's time horizon as the number of steps to go during the interaction decreases. Neither of these is possible within the classical POMDP formalism.

We continue with the simple example of $\text{I−POMDP}_{i,1}$ of agent i. In Fig. 4.10 we display i's value function for the time horizon of 1, assuming that i's initial belief as to the value j assigns to TL, $p_j(TL)$, is as depicted in Fig. 4.7 (ii), i.e. i has no information about what j believes about the tiger's location. This value function is

[15] The POMDP with noise is the same as level 0 I−POMDP.

identical to the value function obtained for an agent using a traditional POMDP framework with noise as well as the single agent POMDP, which we described in Section 4.5.2. The value functions overlap since agents do not have to update their beliefs and the advantage of more refined modeling of agent j in i's I-POMDP does not become apparent. Put another way, when agent i models j using an intentional model, it concludes that agent j will open each door with probability 0.1 and listen with probability 0.8. This coincides with the noise factor we described in Section 4.5.2.

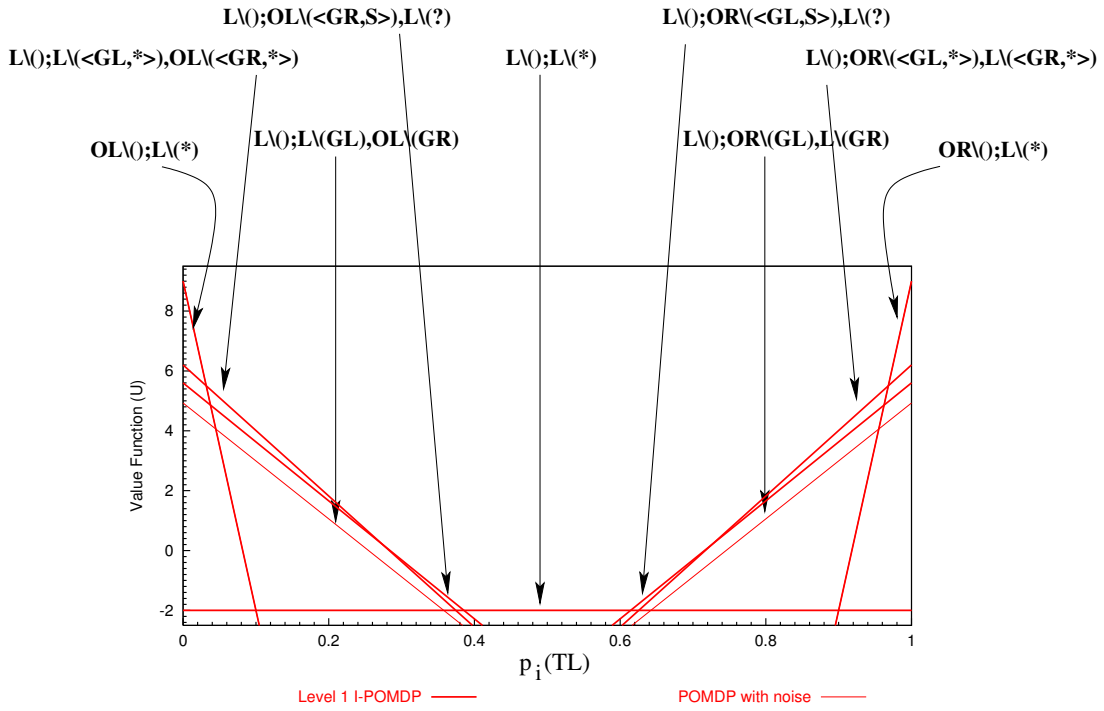

Figure 4.11: Comparison of value functions obtained from solving a singly nested I-POMDP and a POMDP with noise for time horizon of 2. I-POMDP value function dominates due to agent i adjusting the behavior of agent j to the remaining steps to go in the interaction.

In Fig. 4.11 we display i's value functions for the time horizon of 2. The value function of I-POMDP$_{i,1}$ is higher than the value function of a POMDP with the noise factor. The reason is not related to the advantages of modeling agent j's beliefs – this effect becomes apparent at the time horizon of 3 and longer. Rather, the I-POMDP solution dominates due to agent i modeling j's time horizon during interaction: i knows that at the last time step j will behave according to its optimal policy for time horizon of 1, while with two steps to go j will optimize according to its 2 steps to go policy. As we mentioned, this effect cannot be modeled using a POMDP with a static noise factor included in the transition function.

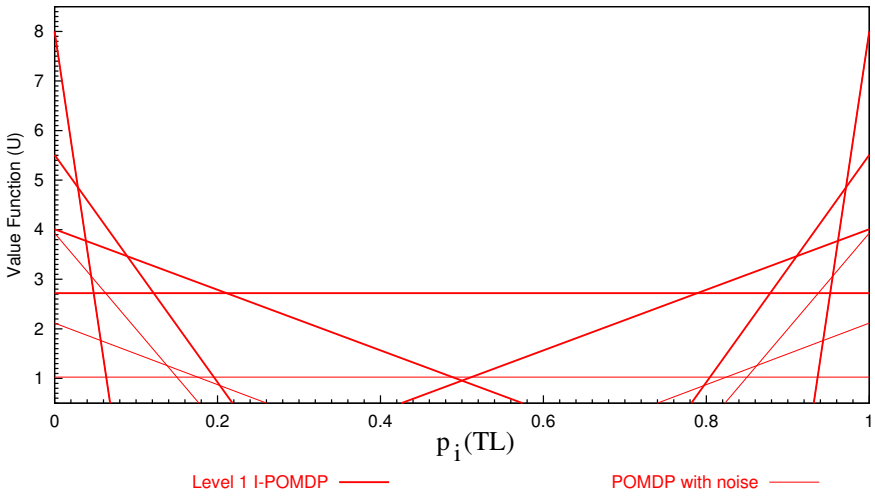

Figure 4.12: Comparison of value functions obtained from solving a singly nested I−POMDP and a POMDP with noise for time horizon of 3. The I−POMDP value function dominates due to agent i's adjusting j's remaining steps to go, and due to i's modeling j's belief update. Both factors allow for better predictions of j's actions during interaction. The descriptions of individual policies were omitted for clarity; they can be read off of Fig. 4.13.

Fig. 4.12 shows a comparison between the I−POMDP and the noisy POMDP value functions for horizon 3. The advantage of the more refined agent modeling within the I−POMDP framework has increased.[16] Both factors, i's adjusting j's steps to go and i's modeling j's belief update during interaction are responsible for the superiority of values achieved using the I−POMDP. In particular, recall that at the second time step i's information as to j's beliefs about the tiger's location is as depicted in Fig. 4.9 (c). This enables i to make a high quality prediction that, with two steps left to go, j will perform its actions OL, L, and OR with probabilities 0.009076, 0.96591 and 0.02501, respectively (recall that for POMDP with noise these probabilities remained unchanged at 0.1, 0,8, and 0.1, respectively.)

Fig. 4.13 shows agent i's policy graph for the time horizon of 3. As usual, it prescribes the optimal first action depending on the initial belief about the tiger's location. The subsequent actions depend on the observations received. The observations include creaks that are indicative of the other agent's having opened a door. The creaks contain valuable information and allow the agent to make more refined choices, compared to ones in the noisy POMDP in Fig. 4.6. Consider the case when agent i starts out with a fairly strong belief as to the tiger's location, decides to listen (according to the four off-center top row "L" nodes in Fig. 4.13)

[16] Note that the I−POMDP solution is not as good as the solution of a POMDP for an agent operating alone in the environment as shown in Fig. 4.5.

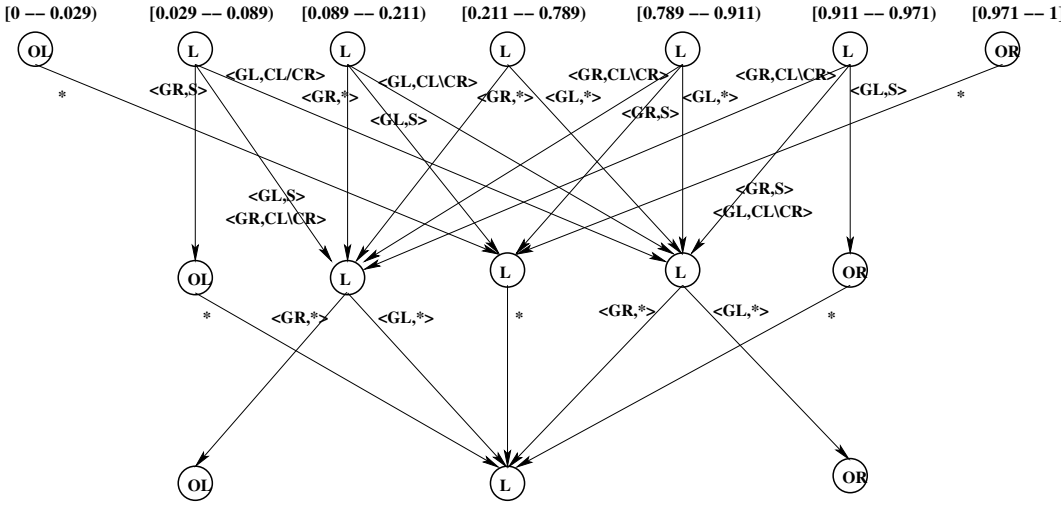

Figure 4.13: The policy graph corresponding to the I-POMDP value function in Fig. 4.12.

and hears a door creak. The agent is then in the position to open either the left or the right door, even if that is counter to its initial belief. The reason is that the creak is an indication that the tiger's position has likely been reset by agent j and that j will then not open any of the doors during the following two time steps. Now, two growls coming from the same door lead to enough confidence to open the other door. This is because agent i's hearing of tiger's growls are indicative of the tiger's position in the state following the agents' actions.

Note that the value functions and the policy above depict a special case of agent i having no information as to what probability j assigns to tiger's location (Fig. 4.7 (ii)). Value functions for some other shapes of i's belief over j's beliefs are shown in Chapter 5. Accounting for and visualizing all possible beliefs i can have about j's beliefs is difficult due to the complexity of the space of interactive beliefs. As our ongoing work indicates, a drastic reduction in complexity is possible without loss of information, and consequently representation of solutions in a manageable number of dimensions is indeed possible. We discuss more on this topic in Chapter 8.

4.6 Application: Agent Based Simulation of Social Behaviors

We apply the I-POMDP framework to empirically demonstrate the emergence of commonly observed anthropomorphic social behaviors among rational interacting agents.[17] By successfully demonstrating the

[17]We believe that the framework can be used to simulate both agent based as well as mixed agent-human environments.

occurrence of social behaviors or patterns among rational agents, we achieve multiple objectives: We establish that the commonly observed behaviors are rational in regards to their respective settings. The results will serve to validate the framework as an important tool for studying and explaining rational interactions in uncertain multiagent dynamic settings. Finally, and of key importance, it will pave the way for deployment of the framework to new settings where rational social behavior has neither been established nor observed.

One of the most intuitive human social behavior is to *follow the leader*. It can arise when a follower believes that a leader possesses a superior ability of some kind and following him may improve the follower's payoff (power relationship). To model this behavior we used the multiagent tiger game with two agents. We gave the agent j (the leader) a better hearing capability, while curtailing the hearing accuracy of i. Additionally, we also assumed that the tiger persists in its location with a high probability after the opening of any door. We solved this game over three horizons under the assumption that i is initially unaware of where the tiger is, and knows that j is also unaware of the tiger's location. The resulting policy tree is in Fig. 4.14(a), where GL(R) stands for i's hearing the tiger's growl on the left (right), and CL(R) denotes i's hearing the creak of the left(right) door opened by j. Note that, in this case, the optimal policy of i is to wait for a creak, and then open the door which i believes j opened previously, but only if the information provided by the creak and i's own hearing of the growls come from the same side. This conditional following of the leader changes if i believes that j's hearing of growls is even more accurate or its own hearing even worse. In that case the optimal policy for i is to follow j's creaks (if they are considered reliable) and ignore its own perception of where the growls come from. We demonstrate this behavior in Fig. 4.14(b).

Other social phenomena that we intend to focus on include the role of technological innovations in competitive settings. Our existing results show that in single agent settings (monopolies), an agent will adopt a technological innovation immediately when its long term usage is expected to justify its initial adoption cost and maintenance costs. We intend to study these results in the context of multiagent settings (oligopolies), and investigate the effect of the presence of other agents on decisions of when to adopt technological innovations. The decision of whether and when to embrace innovations is further complicated when a power relationship exists between agents. In such scenarios, we conjecture that the agent in the position of power will have no incentive to adopt new technology (unless others are likely to adopt it), and will obstruct the adoption of innovations by other agents in order to preserve its position of power. Establishing the accuracy of these and other commonly practiced social behaviors signifies an important step in our understanding of

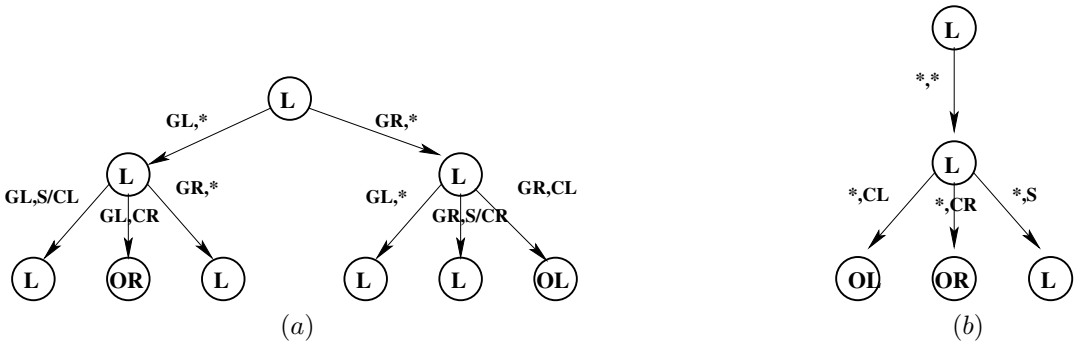

Figure 4.14: (a) *Conditional follow the leader* behavior in the multiagent tiger problem when the agent knows that the other agent is better at hearing the tiger's growls, and the tiger persists behind its original door when any door is opened. The agent opens the same door as the one previously opened by the leader and only when its own observations of the tiger's location are consistent with the door opened by the leader. (b) *Unconditional follow the leader* behavior when the agent receives no information about the tiger location from its own observations. The agent has no choice but to follow the leader and it chooses to open the same door as the one previously opened by the leader.

the causative events that rationally lead to these phenomena.

4.7 Summary

We presented a framework for optimal sequential decision-making suitable for controlling autonomous agents interacting with other agents within an uncertain environment. We used the normative paradigm of decision-theoretic planning under uncertainty formalized as partially observable Markov decision processes (POMDPs) as a point of departure. We extended this framework to make it applicable to agents interacting with other agents by allowing them to have beliefs not only about the physical environment, but also about the other agents. This could include beliefs about the others' beliefs, abilities, sensing capabilities, preferences, and intended actions. Our framework shares numerous properties with POMDPs, has analogously defined solutions, and reduces to POMDPs when agents are alone in the environment. In contrast to much of the recent work (see Section 4.1), our approach is subjective and amenable to agents independently computing their optimal solutions. We demonstrated, using the multiagent tiger problem, that modeling other's beliefs is beneficial: the refined modeling permits more informative decisions than a traditional POMDP based approach, and this generates plans of larger value. We also explored the use of `I-POMDPs` to demonstrate social behaviors that are typically observed in human settings.

4.8 Contributions

Novel framework: We proposed a novel framework, called interactive POMDP, for optimal sequential planning in multiagent settings. Interactive POMDPs reduce to POMDPs in the single agent setting. Our framework is applicable to partially observable stochastic games, in which actions of others agents are not perfectly observable. Additionally, in contrast to Bayesian games, we do not require the unrealistic assumptions of common knowledge of rationality, beliefs, and payoffs of all agents.

Addresses limitations of Nash equilibria: We have adopted a decision-theoretic approach to solving games, and a solution concept centered on optimality and best response to anticipated action(s) of other agents, rather than the prevailing concept of Nash equilibrium. Consequently, our approach does not suffer from the limitations of Nash equilibria such as non-uniqueness and incompleteness.

Better quality plans: Our formalism replaces the "flat" beliefs in POMDPs with an interactive belief system that contains beliefs about others' beliefs, and their beliefs about others'. While such belief systems have been investigated as formalizations of Harsanyi's notion of a type space, they have not been employed for sequential decision-making before. We empirically demonstrated the advantage of this nested modeling, by showing that I-POMDPs generate plans that are of significantly better value than those generated by using the traditional POMDP framework.

Broad applicability: A natural result of adopting a decision-theoretic approach is that our framework is applicable to both cooperative and non-cooperative games – such distinctions as well as categorization of games into zero-sum or constant sum, and general sum is no longer needed. Additionally, we do not restrict the scope of the framework to settings populated by just rational agents, but include intentional as well as subintentional agents. Clearly, subintentional agents that may be simple memoryless finite state automatons need not be modeled using sophisticated constructs.

Application: We demonstrated the emergence of some intuitive social behaviors among rational agents modeled using the I-POMDP framework. This suggests that I-POMDPs can be employed to understand and formalize the mechanisms that give rise to such behaviors, and also to investigate what behaviors may arise in new unexplored settings.

4.9 Future Work

The line of work presented here opens a wide area of future research on integrating frameworks for sequential planning with elements of game theory and Bayesian learning in interactive settings. In particular, one of the avenues of our future research centers on proving further formal properties of I-POMDPs, and establishing clearer relations between solutions to I-POMDPs and various flavors of equilibria. Another concentrates on developing efficient approximation techniques for solving I-POMDPs. As for POMDPs, development of approximate approaches for solving I-POMDPs is crucial for moving beyond toy problems. Both these lines of research are addressed in the remainder of this thesis. Another research issue is the suitable choice of priors over models. We are looking at Kolmogorov complexity (Li & Vitanyi, 1997) as a possible way to assign priors. We are also interested in developing models that capture the boundedness of the resources realistically available to an agent. In this respect, developing algorithms for generating bounded optimal plans becomes important.

Chapter 5

SOLUTIONS TO OTHER MULTIAGENT TIGER PROBLEMS

THE tiger problem is a classical example for illustrating planning frameworks that deal with uncertainty. It's attractive because of its simplicity, flexibility, and richness in plan structure. In Section 4.5 of Chapter 4, we extended the traditional single agent tiger problem to the multiagent setting to illustrate our multiagent planning framework. We created a particular version in which the agents' immediate reward was not directly influenced by other agents' actions. However, other agent's actions affected the location of the tiger which in turn could influence the behavior of the original agent. Let's call this version of the multiagent tiger problem as a *neutral* setting.

In this chapter, we provide solutions for a suite of non-cooperative and cooperative versions of the multiagent tiger problem, including the neutral setting. We utilize the transition, and observation functions of the previously mentioned multiagent tiger problem, and create a variety of reward functions that encourage coordination or friendly behaviors, and mis-coordination or conflicting behaviors. Traditionally – in game theory – the type of interaction between agents is established through the reward function only. In the context of I-POMDPs, as we mentioned in Section 4.5.4 of Chapter 4, the effect of an agent's actions on the physical state of the problem also influences the behavior of the agents. Our objective in showing the solutions to various multiagent tiger problems is to illustrate the effect of the type of interaction on the solution. In doing so, we demonstrate the broad applicability of the I-POMDP framework to both non-cooperative and cooperative settings. We also demonstrate the influence that different i's beliefs over j's beliefs have on the

plan structure. Furthermore, we uncover behavioral insights that are intuitive of real-world interactions.

The remainder of this chapter is structured in the following manner. In Section 5.1 we present value functions for non-cooperative versions of the multiagent tiger problem. Specifically, we concentrate on two settings: *enemy* and *neutral*. We present value functions for the cooperative versions of the multiagent tiger problem in Section 5.2. We again look at two settings: *friend* and *team*. We then summarize the chapter in Section 5.3, give the contributions of this work in Section 5.4, and the future lines of research in Section 5.5.

5.1 Non-cooperative Versions

We develop two versions of the multiagent tiger problem that promote non-cooperation between the agents. First is the enemy setting, in which agent i gets larger rewards if j opens the wrong door than if j opens the correct door, thereby encouraging non-cooperation between the two. Additionally, as we mentioned before, j's action of opening doors may reset the location of the tiger, thereby making i's accumulated observations uninformative. Second is the neutral setting in which, though i's payoffs are indifferent to j's actions, j's actions may change the state in a way that is detrimental to i.

For each of the settings below, the transition and observation functions for the agents i and j are as given previously in Fig. 4.1 of Chapter 4. We will also adopt the assumptions mentioned previously: agent i is singly nested, and is uncertain only about j's beliefs (not j's frames). Agent i assumes that the level 0 agent j assigns a static distribution of 0.1, 0.1, and 0.9 to the opening of left, right, and no doors due to the noise. Each of the settings below differs in the rewards that each agent receives given the joint action and the location of the tiger.

5.1.1 Enemy

The reward functions for the agents in the enemy setting are given in Table 5.1. In this setting, it is most beneficial for i if it opens the correct door and j opens the wrong door.

In Fig. 5.1, we show the value function plots for the horizons 1 and 2, when agent i thinks that j is likely to be uninformed about the location of the tiger. To model i's belief, we used a beta p.d.f. that peaks at $p_j(TL) = 0.5$. Policies corresponding to the value lines are conditional plans. The actions L, OR, or OL, are conditioned on observational sequences in parenthesis. For example, L\();L\(GL,*),OL\(GR,S), L\(?) denotes a plan to perform the listening action, L, at the beginning (list of observations is empty), and then

$\langle a_i, a_j \rangle$ /State	TL	TR
$\langle$ OR,OR $\rangle$	5	-50
$\langle$ OL,OL $\rangle$	-50	5
$\langle$ OR,OL $\rangle$	60	-105
$\langle$ OL,OR $\rangle$	-105	60
$\langle$ L,L $\rangle$	-0.5	-0.5
$\langle$ L,OR $\rangle$	-6	49
$\langle$ OR,L $\rangle$	10.5	-99.5
$\langle$ L,OL $\rangle$	49	-6
$\langle$ OL,L $\rangle$	-99.5	10.5

$\langle a_i, a_j \rangle$ /State	TL	TR
$\langle$ OR,OR $\rangle$	5	-50
$\langle$ OL,OL $\rangle$	-50	5
$\langle$ OR,OL $\rangle$	-105	60
$\langle$ OL,OR $\rangle$	60	-105
$\langle$ L,L $\rangle$	-0.5	-0.5
$\langle$ L,OR $\rangle$	10.5	-99.5
$\langle$ OR,L $\rangle$	-6	49
$\langle$ L,OL $\rangle$	-99.5	10.5
$\langle$ OL,L $\rangle$	49	-6

Table 5.1: Reward functions for the agents i and j in the *enemy* setting.

another L if the observation is growl from the left (GL) and regardless of whether a creak is heard or not, open the left door, OL, if the observation is growl from the right (GR) and no creak (S), and L for all the remaining observations. $*$ is a wildcard with the usual interpretation and ? denotes all remaining observations.

We note that the horizon 2 value function refines the horizon 1 value function: Of the 2178 possible conditional plans that i could follow, 7 were found optimal for the different regions of i's beliefs. Notice that the optimal plan L\();L\(GL,*),OL\(GR,*) and its symmetric counterpart L\();L\(GR,*),OR\(GL,*), ignore the creaks and uses only growls heard in the next time step to guide i's actions. This is because i is certain that j will listen and consequently not change the location of the tiger. Any creak that is heard will be dismissed as a noisy observation.

In Fig. 5.2, we present the horizon 1 and 2 value function plots for the case in which i thinks that j is likely to be informed about the location of the tiger. In this state of belief, i believes that j likely assigns a high probability to the correct location of the tiger. Again, i's belief is modeled using a beta p.d.f. that peaks at $p_j(TL) = 0.98$ when the tiger is on the left, and at $p_j(TL) = 0.02$ when the tiger is on the right. We point out the presence of the plan L\();OL\(GR,S),L\(?) and its symmetric counterpart as part of the policy. i's belief that j is likely informed causes i to be almost certain that j will open doors. j's opening of doors will cause the tiger to reset forcing i to listen in the next time step. Because there is a small chance that j will listen, i will open doors on hearing no creaks rather than dismiss it as a noisy observation.

When i's certainty that j is informed of the location of the tiger increases, the belief region for which the plan L\();OL\(GR,S),L\(?) is optimal gradually reduces. In the limit (i knows that j is informed – Dirac-delta p.d.f.), the conditional plan mentioned above is no longer part of the policy. This is because i knows that j will open doors and therefore dismisses the observation of no creaks as noisy.

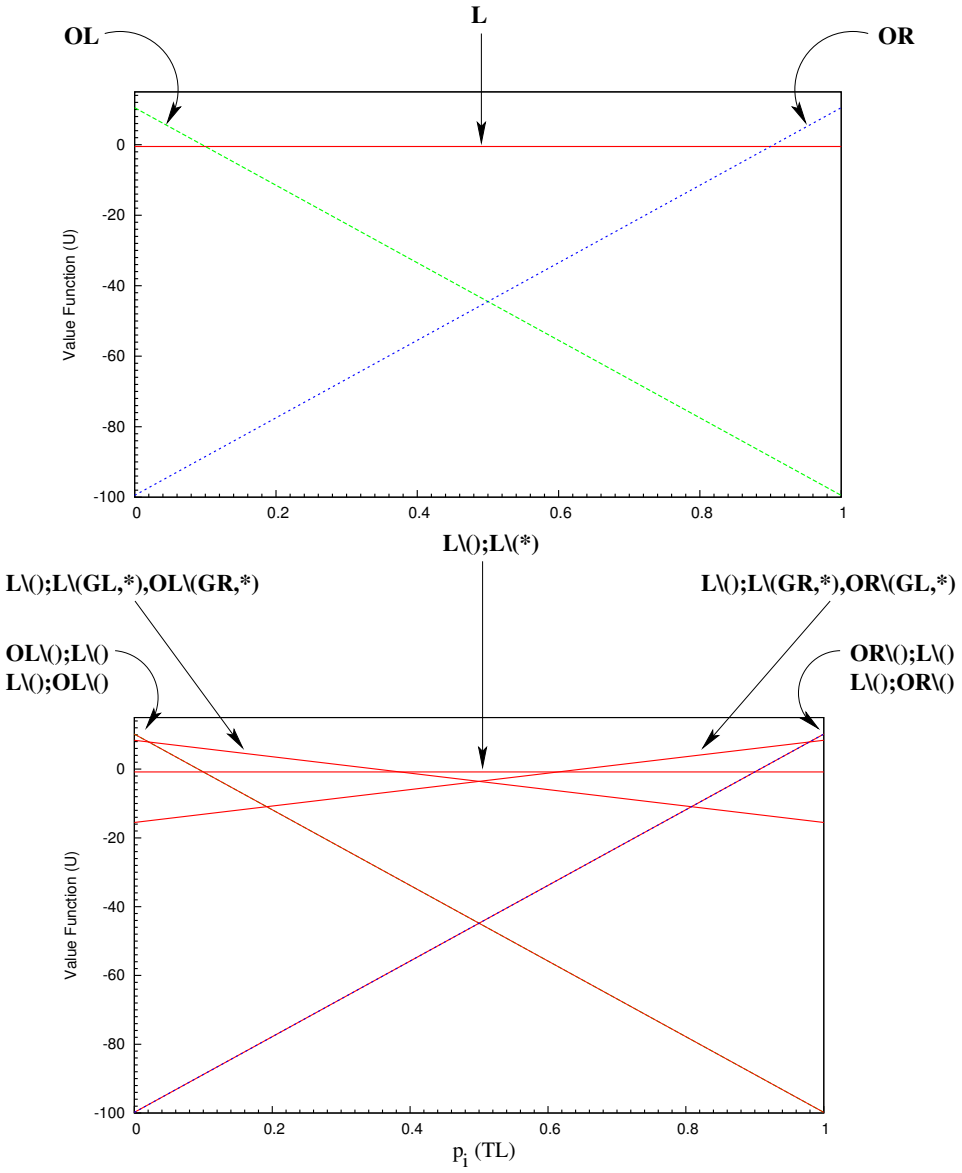

Figure 5.1: Horizon 1 and 2 value functions for the enemy setting when i believes that j is likely uninformed about the tiger's location.

On comparing the value functions in Fig. 5.1 with the corresponding ones in Fig. 5.2, we make an insightful observation: *The value of the interaction for an agent is more when its enemy is uninformed about the state of the problem as compared to when the enemy is informed.* In Fig. 5.3, we highlight the difference between the two value functions. It clearly shows that the expected reward when the enemy is uninformed

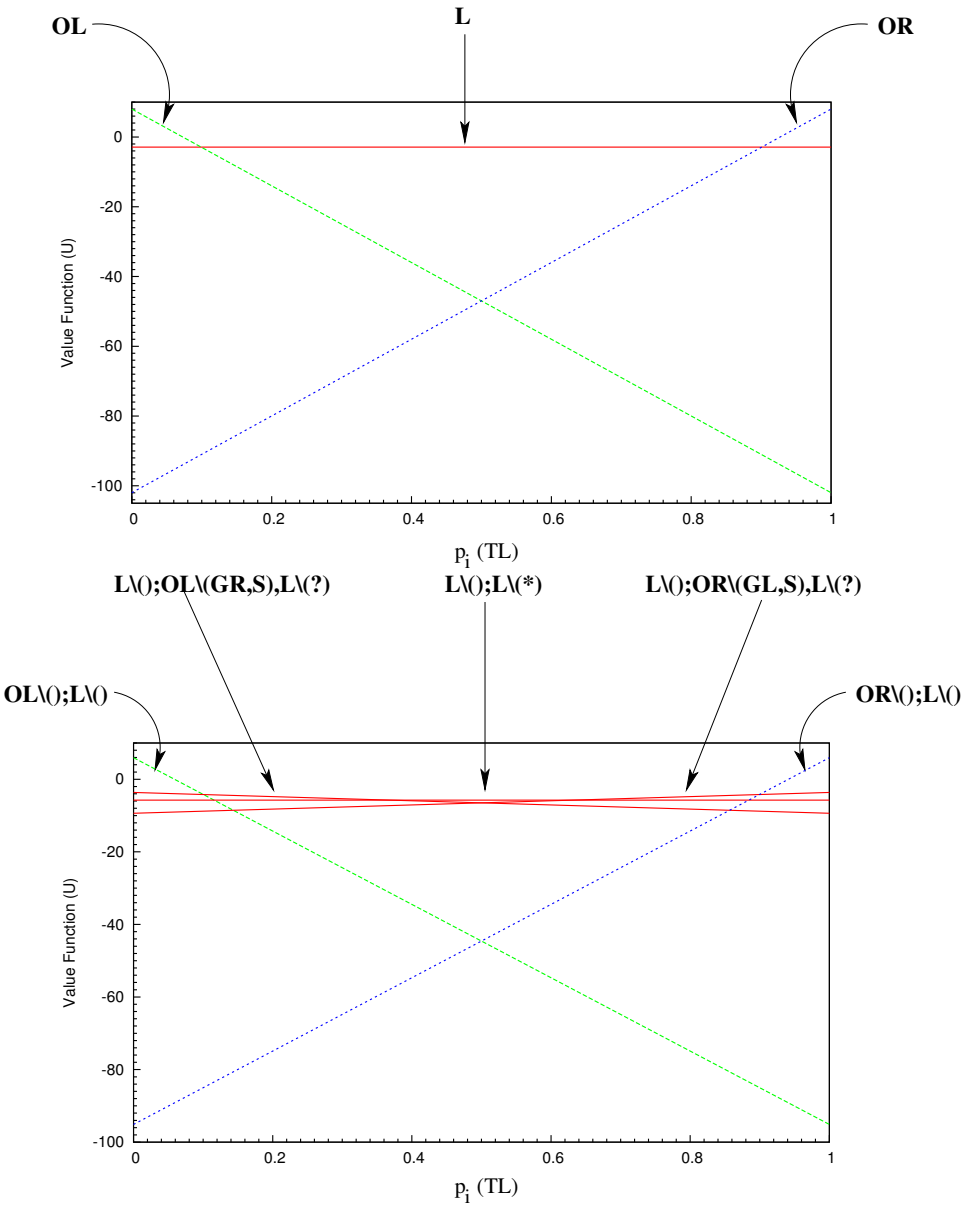

Figure 5.2: Horizon 1 and 2 value functions for the enemy setting when i believes that j is likely informed about the tiger's location.

is greater than when it is informed. The policy trees for the vectors can be read off the original plots. The difference is, of course, also present between the horizon 1 value functions. We note that this observation is intuitive of real world interactions.

Finally, in Fig. 5.4, we show the value functions for horizons 1 and 2 when i thinks that j is partly

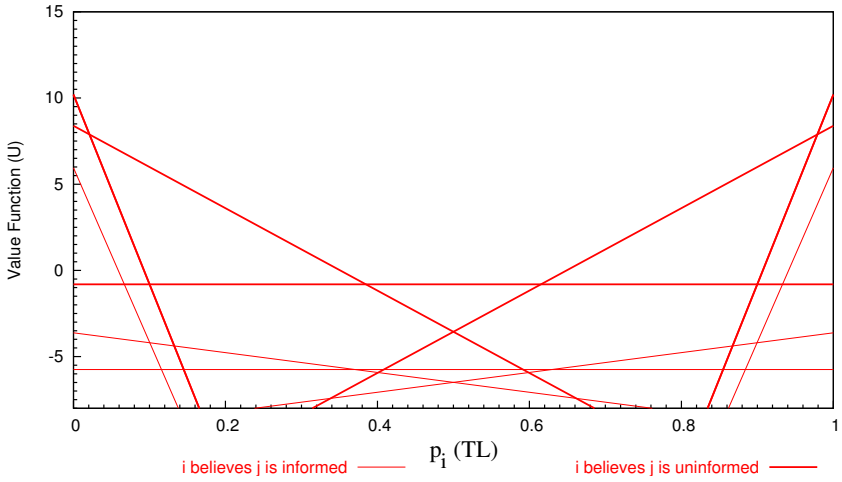

Figure 5.3: A comparison of the horizon 2 value functions obtained when i believes that j is likely informed about the tiger's location versus when i believes that j is uninformed. We observe that the value of the plan is more when the enemy is uninformed compared to when it is informed.

informed: i believes that j likely assigns a high probability to the tiger being on the left regardless of whether the tiger is indeed on the left or the right. We model this belief of i using a beta p.d.f. that peaks at $p_j(TL) = 0.98$ no matter what the tiger's location is. As a result of its beliefs about j, agent i thinks that j will likely open the right door.

In contrast to the plots in Fig. 5.1 and 5.2, the value functions are not symmetric about $p_i(TL)$ = 0.5. Agent i assigns a larger value to its beliefs about the location of the tiger that cause it to open the left door (implying that j is opening the wrong door). This is because of i's asymmetric beliefs about j's and payoffs which reward i opening the correct door and j's opening of the wrong door. The "skewness" of the value functions increases as i's belief of j being partly informed becomes stronger.

5.1.2 Neutral

For the neutral setting, the reward function is as given in Fig. 4.1 of Chapter 4. As we mentioned previously, though i's rewards are independent of j's actions, actions of j may alter the physical state of the problem in a way that is detrimental to i. For example, let i believe that the tiger is likely to be on the left. On hearing a creak, i believes that j has very likely opened a door causing the tiger to reset. Agent i must now discard the information that it previously had about the tiger's location. This simple example demonstrates the effects that actions of agents have on each other through the state.

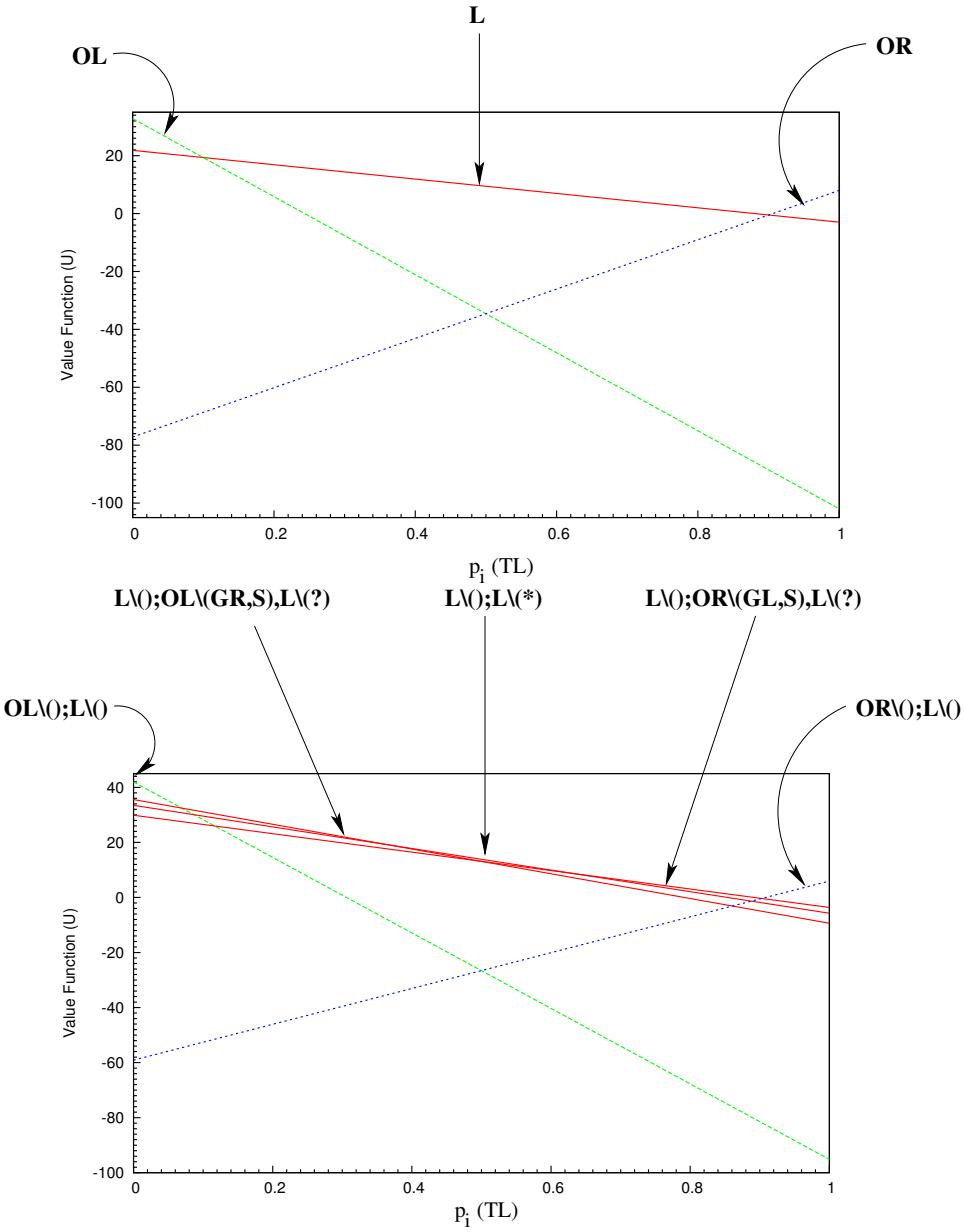

Figure 5.4: Horizon 1 and 2 value functions for the enemy setting when i believes that j is partly informed about the tiger's location.

In Section 4.5.4, we showed the value functions for horizons 1, 2 and 3 when i is uninformed about j's beliefs over the location of the tiger. In Fig. 5.5, we show the value function plots for horizons 1 and 2 when i thinks that j is likely uninformed about the location of the tiger. The horizon 1 value function is similar to

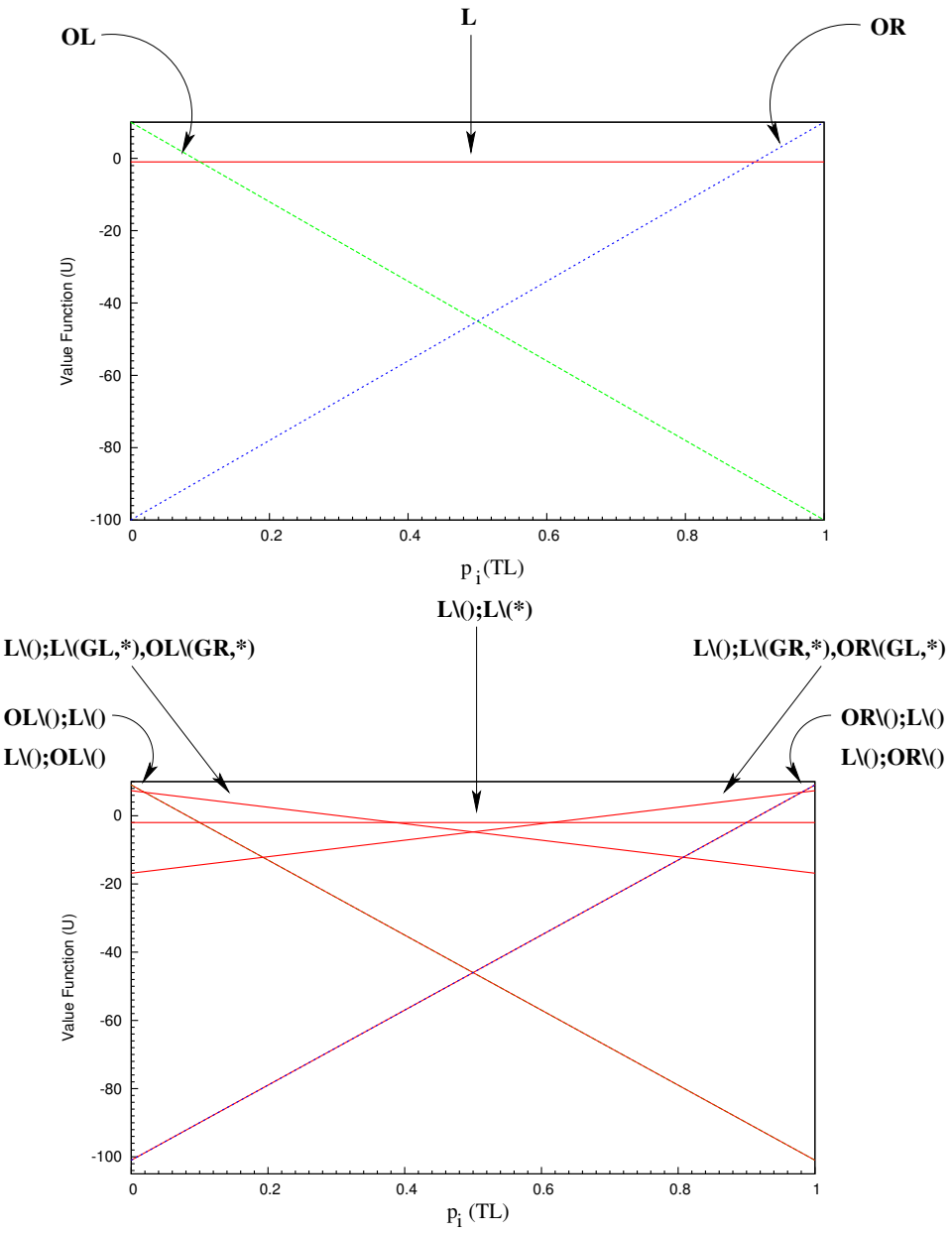

Figure 5.5: Horizon 1 and 2 value function plots for the neutral payoffs when i believes that j is likely uninformed about the tiger's location.

the one in Fig. 2.4 for the single agent tiger problem. This is because i's rewards are not dependent on j's actions, and the problem does not extend beyond the single time step for i to feel the effect of j's actions. For the same reason, policy trees that are common in the horizon 2 value function plot shown here and in Fig. 2.5

have the same value.

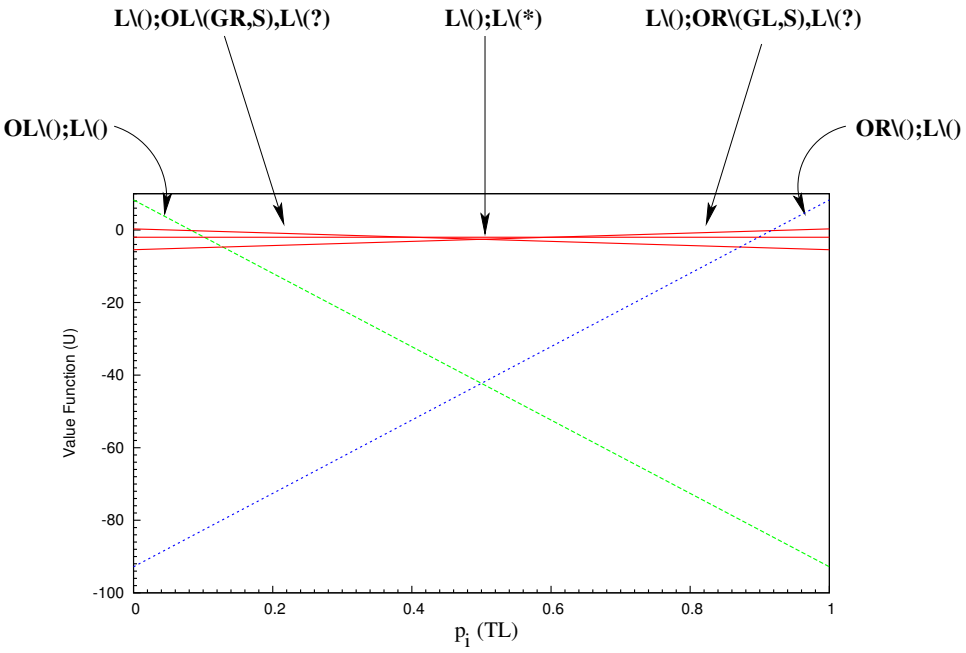

Figure 5.6: Horizon 2 value function plot for the neutral payoffs when i believes that j is likely informed. An identical value function plot results when i believes that j is partly informed.

We show the horizon 2 value function when i believes that j is likely informed about the tiger's location, in Fig. 5.6. The horizon 1 value function is, of course, identical to the one shown in Fig. 5.5.

Let us compare the horizon 2 value function in Fig. 4.11 with those in Figs. 5.5 and 5.6. While both the conditional plans, L\();L\(GL,*),OL\(GR,*) and L\();OL\(GR,S),L\(?) (and their symmetric counterparts), are present in Fig. 4.11, one or the other is absent from the value functions in Figs. 5.5 and 5.6. The reason is intuitive: Since i is uninformed about j's beliefs in the former setting, i thinks that j may listen or open doors with a probability distribution given by j's policy. Hence neither the observation of creaks nor of no creaks can be dismissed as noise. In Fig. 5.5, because i believes that j certainly listens, its observations of creaks are dismissed as noise, and the conditional plan L\();OL\(GR,S),L\(?) is absent from the corresponding value function. In Fig. 5.6, since i believes that j almost certainly opens doors causing the tiger's location to reset, the creaks cannot be ignored. Therefore the conditional plan L\();L\(GL,*),OL\(GR,*) which ignores creaks is absent from the corresponding value function.

For the case when i believes that j is partly informed – i believes that j likely opens the right door

regardless of the tiger's location – the horizon 1 value function is, as usual, identical to the one in Fig. 5.5. The horizon 2 value function turns out to be identical to the horizon 2 value function in Fig. 5.6. This is because, in both these cases i believes that j opens doors. Since it does not matter to i which door j opens, i's value function remains the same. Additionally, unlike the corresponding value function in the enemy setting, this function is not skewed because i's rewards do not depend on j's actions.

5.2 Cooperative Versions

In this section, we show solutions for two cooperative versions of the multiagent tiger problem. The first version is a *friend* setting in which the payoffs encourage cooperation between the agents. The second version is the *team* setting that has appeared previously in the literature (Nair et al., 2003). While our transition and reward functions in the team setting are identical to those in (Nair et al., 2003), the observation function differs due to the presence of creaks. The team setting encourages coordination among the agents.

For the settings below, the transition and observation functions for the agents i and j are as given previously in Fig. 4.1 of Chapter 4. We will adopt the assumptions that agent i is singly nested, and is uncertain only about j's beliefs (not j's frames). Furthermore, agent i assumes that the level 0 agent j assigns a static distribution of 0.1, 0.1, and 0.9 to the opening of left, right, and no doors due to the noise.

5.2.1 Friend

The reward functions for the agents i and j in the *friend* setting are given in Table. 5.2. We note that unlike the team setting that we shall see later, the friend setting does not reward strict coordination, though it does promote cooperation. Also, in contrast to the team setting, the reward functions of the two agents differ.

$\langle a_i, a_j \rangle$ /State	TL	TR
$\langle$ OR,OR $\rangle$	15	-150
$\langle$ OL,OL $\rangle$	-150	15
$\langle$ OR,OL $\rangle$	-40	-95
$\langle$ OL,OR $\rangle$	-95	-40
$\langle$ L,L $\rangle$	-1.5	-1.5
$\langle$ L,OR $\rangle$	4	-51
$\langle$ OR,L $\rangle$	9.5	-100.5
$\langle$ L,OL $\rangle$	-51	4
$\langle$ OL,L $\rangle$	-100.5	9.5

$\langle a_i, a_j \rangle$ /State	TL	TR
$\langle$ OR,OR $\rangle$	15	-150
$\langle$ OL,OL $\rangle$	-150	15
$\langle$ OR,OL $\rangle$	-95	-40
$\langle$ OL,OR $\rangle$	-40	-95
$\langle$ L,L $\rangle$	-1.5	-1.5
$\langle$ L,OR $\rangle$	9.5	-100.5
$\langle$ OR,L $\rangle$	4	-51
$\langle$ L,OL $\rangle$	-100.5	9.5
$\langle$ OL,L $\rangle$	-51	4

Table 5.2: Reward functions for the agents i and j for the *friend* setting.

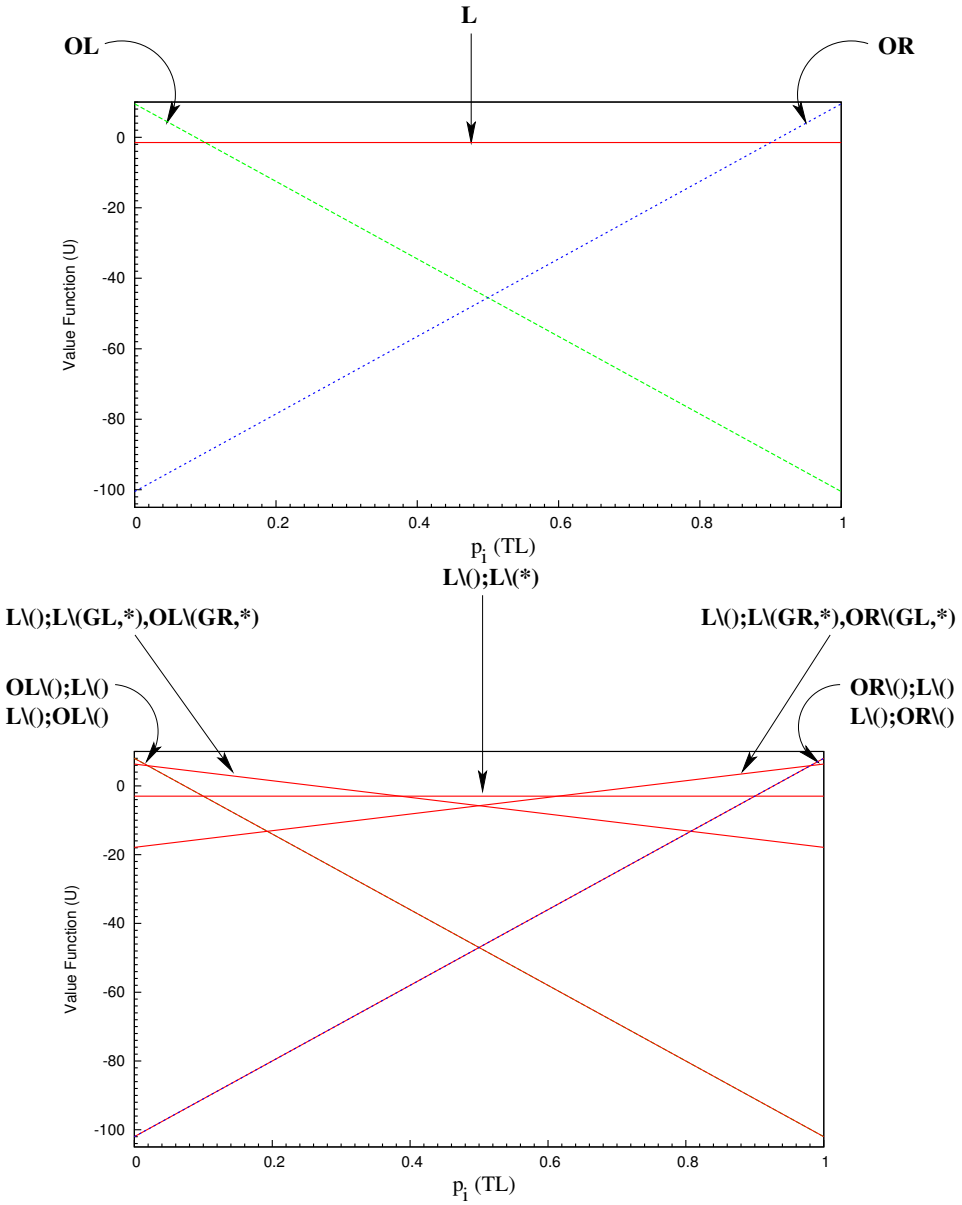

Figure 5.7: Horizon 1 and 2 value functions for the friend setting when i believes that j is likely uninformed about the tiger's location.

As before, we start with the value function plots (Fig. 5.7) when i believes that j is likely to be uninformed about the location of the tiger. As we mentioned before, beliefs of i will be modeled using beta p.d.f.s. We note that the conditional plans that are part of the optimal policy of i remain unchanged when compared with the analogous cases in the non-cooperative settings. In Fig. 5.8, we show the value functions when i believes

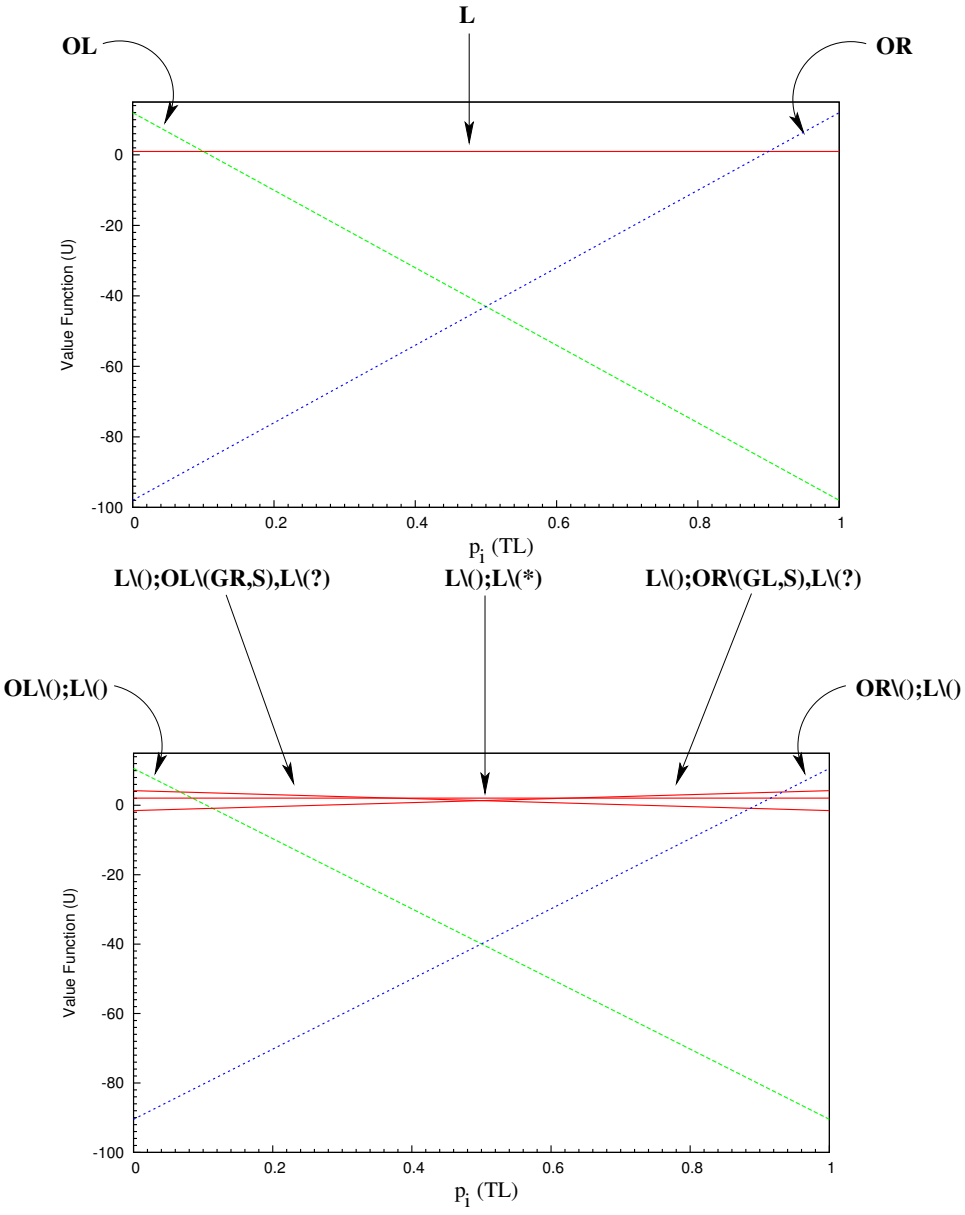

Figure 5.8: Horizon 1 and 2 value functions for the friend setting when i believes that j is likely informed about the tiger's location.

that j is likely informed about the location of the tiger. When we compare the plots in Fig. 5.7 with those in Fig. 5.8, we again uncover a key insight: *The value of the interaction for an agent is more when its friend is informed about the state of the problem as compared to when the friend is uninformed.* This observation is true of real-world interactions, and is the counterpart of the observation we made in the enemy setting. A

comparison of the two value functions is shown in Fig. 5.9. The plans denoted by each vector can be read off from the original value functions. The difference in values is also evident between the horizon 1 value functions.

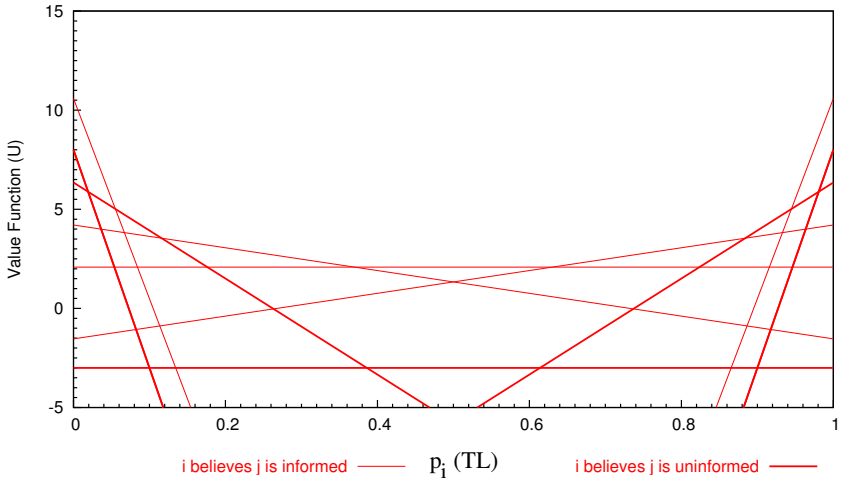

Figure 5.9: A comparison of the horizon 2 value functions when i believes that j is likely informed about the tiger's location versus when i believes that j is uninformed. The values of the plans are more when the friend is informed compared to when it is uninformed.

In Fig. 5.10, we present the value functions for the case when i believes that j likely assigns a high probability to the tiger being on the left. The value function plot is asymmetric about $p_i(TL) = 0.5$ due to the asymmetric nature of i's beliefs over j's beliefs and the reward function which encourages cooperation. The "skewness" of the value function plots increases as the belief of i about j's becomes stronger. i assigns larger value to beliefs that require it to open the same door as j. This is because the reward for i when both agents open the same correct door is higher than when only i opens the correct door. As expected, the direction of the skewness is opposite of that in Fig. 5.4.

5.2.2 Team

The reward functions for the team setting (Nair et al., 2003) are given in Table 5.3. Both agents have the same reward function, which encourages cooperation between the two. Also, notice that coordinated actions of the two agents are rewarded: the reward is doubled when both agents open the correct door, and the penalty is halved when both agents open the wrong door.

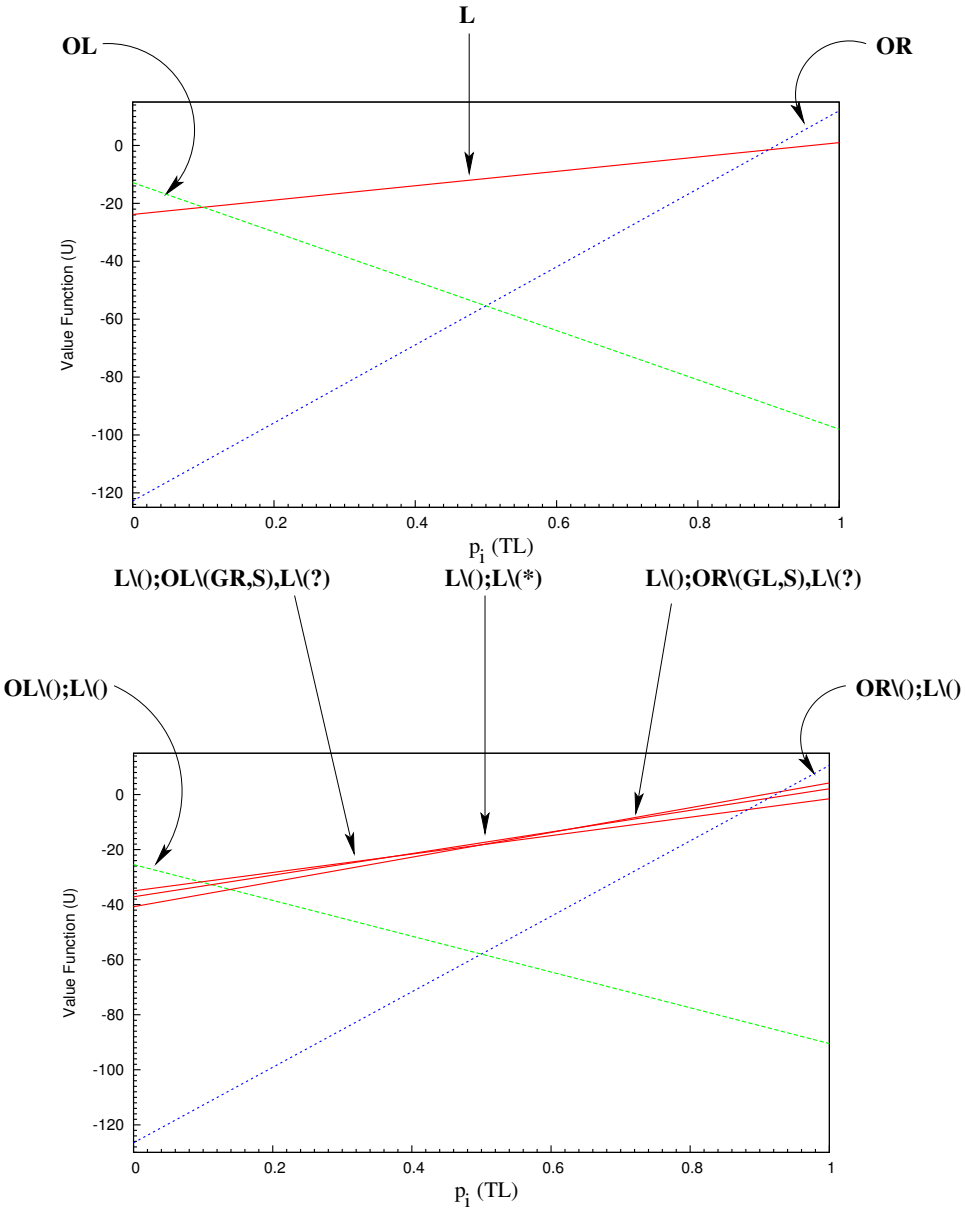

Figure 5.10: Horizon 1 and 2 value functions for the friend setting when i believes that j is likely partly informed about the tiger's location.

Fig. 5.11 shows the value function plots for horizons 1 and 2, when i believes that j is likely uninformed about the location of the tiger. When we compare these plots with the corresponding ones in Fig. 5.12, we again uncover a similar observation as before: It is beneficial if team members are informed rather than be uninformed about the state of the problem.

$\langle a_i, a_j \rangle$ /State	TL	TR
$\langle$ OR,OR $\rangle$	20	-50
$\langle$ OL,OL $\rangle$	-50	20
$\langle$ OR,OL $\rangle$	-100	-100
$\langle$ OL,OR $\rangle$	-100	-100
$\langle$ L,L $\rangle$	-2	-2
$\langle$ L,OR $\rangle$	9	-101
$\langle$ OR,L $\rangle$	9	-101
$\langle$ L,OL $\rangle$	-101	9
$\langle$ OL,L $\rangle$	-101	9

$\langle a_i, a_j \rangle$ /State	TL	TR
$\langle$ OR,OR $\rangle$	20	-50
$\langle$ OL,OL $\rangle$	-50	20
$\langle$ OR,OL $\rangle$	-100	-100
$\langle$ OL,OR $\rangle$	-100	-100
$\langle$ L,L $\rangle$	-2	-2
$\langle$ L,OR $\rangle$	9	-101
$\langle$ OR,L $\rangle$	9	-101
$\langle$ L,OL $\rangle$	-101	9
$\langle$ OL,L $\rangle$	-101	9

Table 5.3: Reward functions for the agents i and j for the *team* setting. Both agents have the same reward function indicating that the team setting is purely cooperative.

In Fig. 5.13, we give the value functions for the case when i believes that j is partly informed about the location of the tiger. In other words, i believes that j likely thinks that the tiger is on the left, and will therefore likely open the right hand side door. As we saw in the friend setting, the value functions plots are asymmetric. Agent i assigns a larger value to beliefs that require it to open the right hand side door – the same door as agent j. This is a natural result because coordination between the two agents is encouraged. Because the payoffs of the team setting encourage coordination more than those of the friend setting, the "skewness" is more pronounced here.

5.3 Summary

In this chapter, we presented and analyzed solutions – value functions and policies – of cooperative and non-cooperative versions of the multiagent tiger problem. Specifically, we developed two settings in each category: the *enemy* and *neutral* settings as non-cooperative versions, and the *friend* and *team* settings as cooperative versions. Note that the team setting first appeared elsewhere in the literature. For each of these settings, we showed value functions for specific shapes of i's beliefs over j's level 0 beliefs over the location of the tiger. On comparing the solutions, we uncovered some key insights that are intuitive of real-world interactions. We also illustrated the influence that different i's beliefs over j's beliefs and the type of setting have on the plan value and structure.

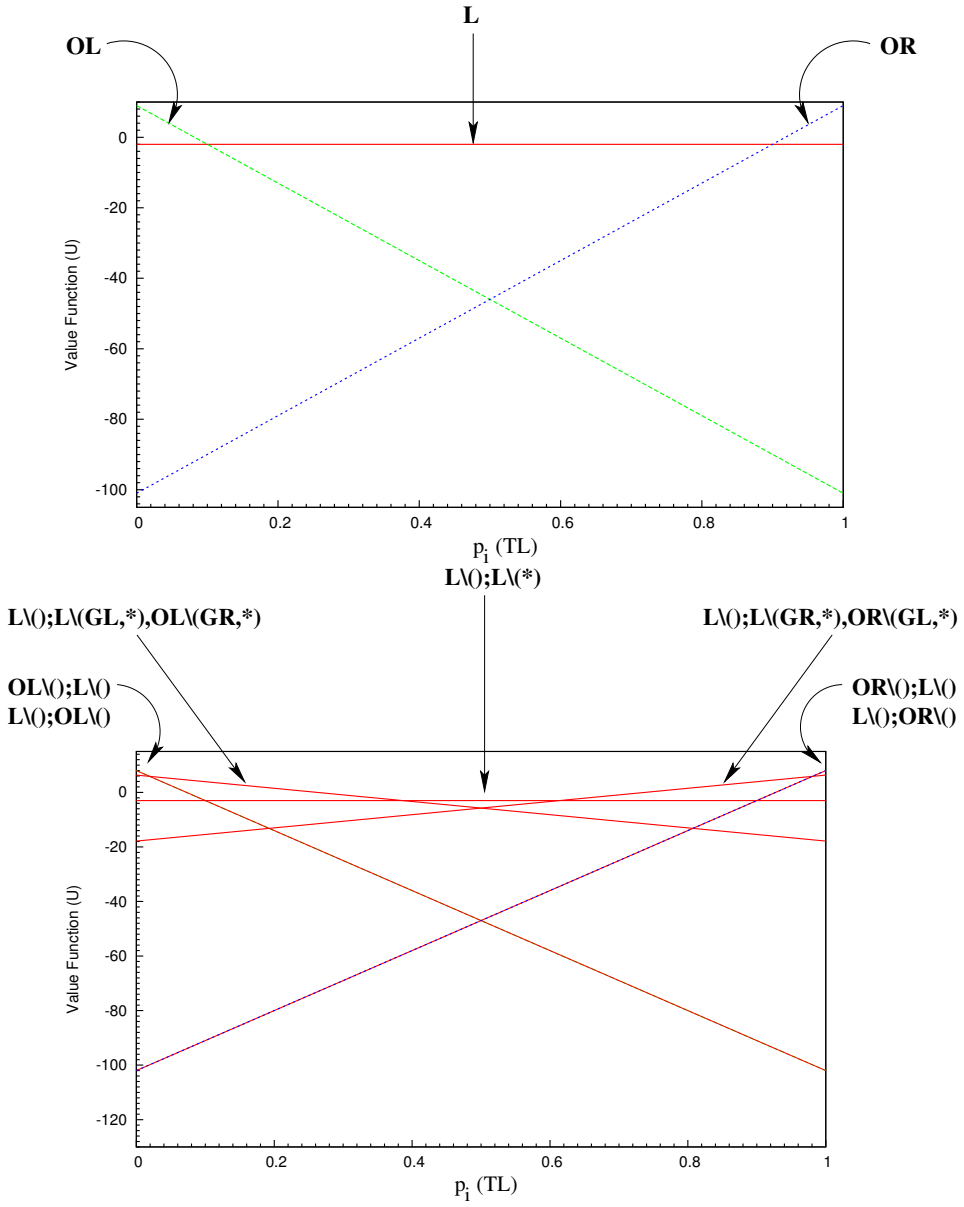

Figure 5.11: Horizon 1 and 2 value functions for the team setting when i believes that j is likely uninformed about the tiger's location.

5.4 Contributions

Behavioral insights: We made some key observations when comparing solutions of the various multiagent tiger problems. Specifically, we showed that the value of a cooperative interaction to an agent is more

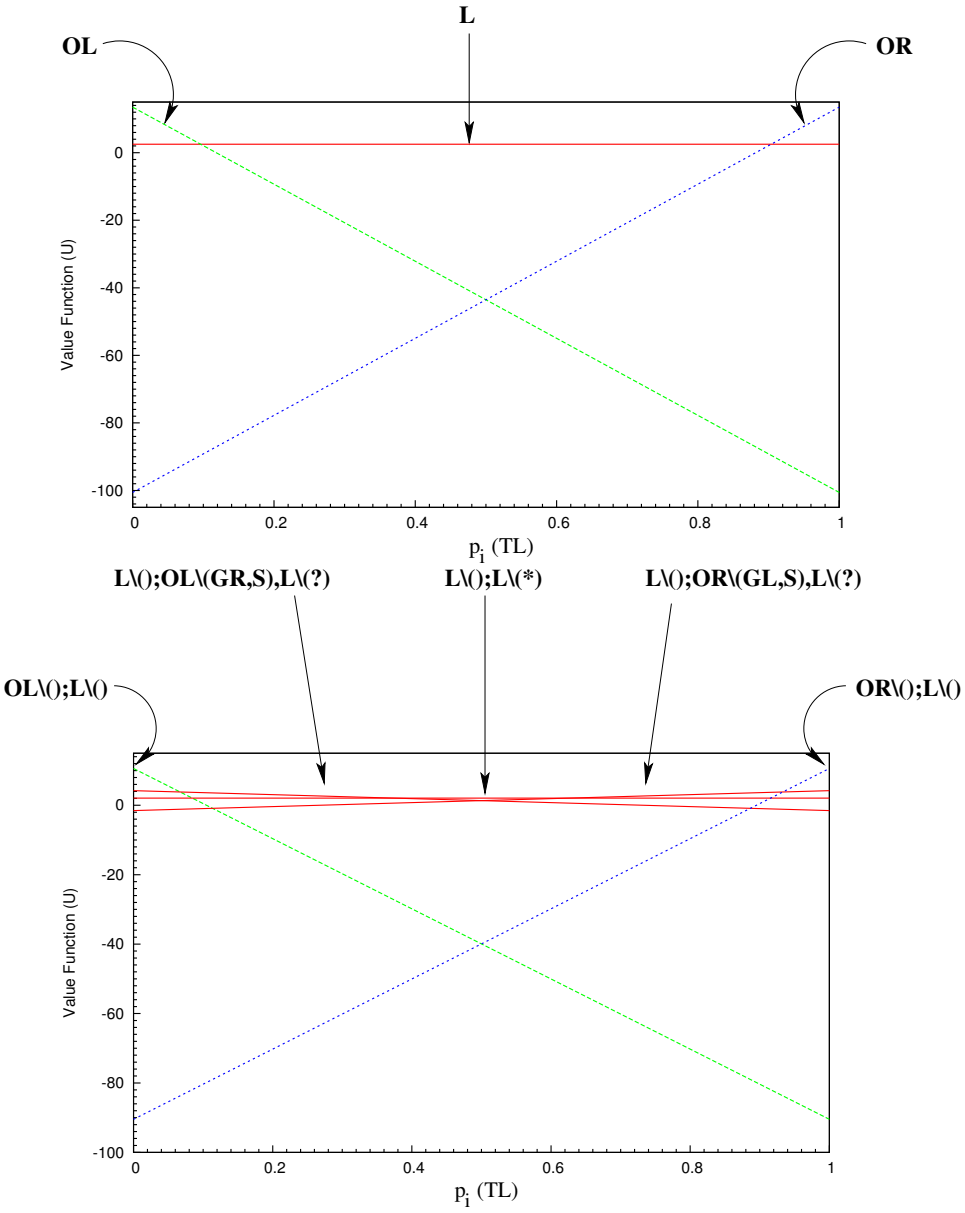

Figure 5.12: Horizon 1 and 2 value functions for the team setting when i believes that j is likely informed about the tiger's location.

when the other friendly agent is informed as compared to when it is uninformed. For a non-cooperative setting, the opposite is true: the value of a non-cooperative interaction is more when the enemy is uninformed as compared to when it is informed.. These observations, of course, reflect real-world interactions, and serve to validate `I-POMDPs` as useful tools for analyzing and explaining social behaviors.

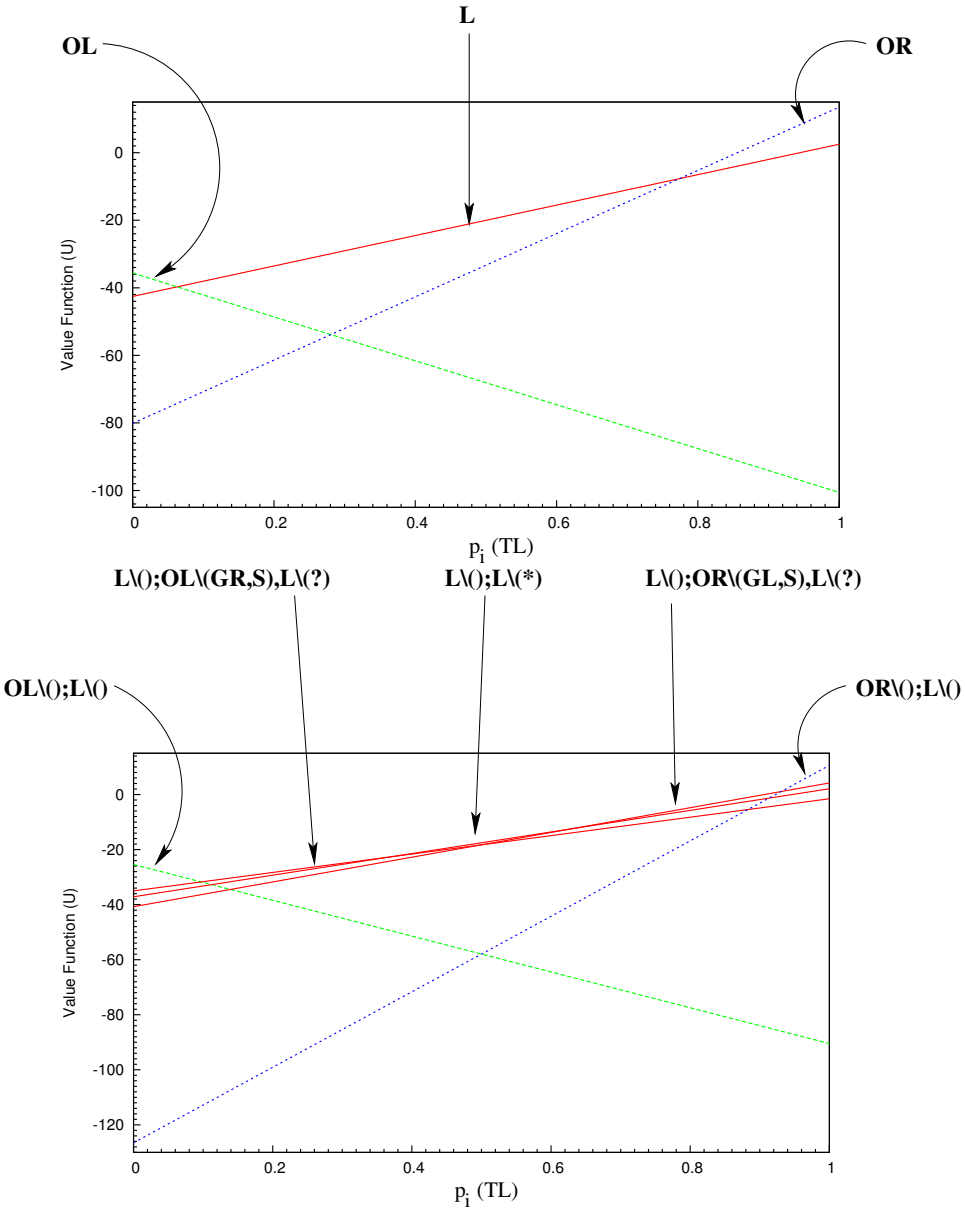

Figure 5.13: Horizon 1 and 2 value functions for the team setting when i believes that j is partly informed about the tiger's location.

Broad applicability of `I-POMDPs`: By computing solutions for non-cooperative and cooperative problems, we have demonstrated the broad applicability of the `I-POMDP` framework. This is in contrast to many other methods in the literature (see Section 4.1 of Chapter 4) which are applicable to only either one of the two settings.

5.5 Future Work

An important line of future work is to devise methods for representing I-POMDP solutions without assumptions about what's believed about other agents' beliefs. Inspite of the complexity of the interactive state space, there seem to be intuitive representations based on belief partitions corresponding to optimal policies of the other agents. We are also looking into techniques for representing the solutions when beliefs are nested to levels greater than one. Partitions of others' multiply-nested beliefs based on their policies, again show promise for scaling up the solutions to higher levels of belief nestings.

Chapter 6

APPROXIMATING I-POMDPS USING PARTICLE FILTERS

INTERACTIVE POMDPs, a generalization of POMDPs to multiagent settings, offer a principled framework for sequential decision-making in uncertain multiagent settings. Their solutions map an agent's states of belief about the environment and other agents' models to policies, but optimal solutions are difficult to compute due to two sources of intractability: First is the complexity of the belief representation, sometimes called the *curse of dimensionality*. The second is the complexity of the space of policies, also called the *curse of history*. Both these sources of intractability exist in POMDPs also (see Section 2.3 in Chapter 2), but the curse of dimensionality is especially more acute in `I-POMDPs`. This is because in `I-POMDPs` the complexity of the belief space is even greater; the beliefs may include beliefs about the physical environment, and possibly the agent's beliefs about other agents' beliefs, their beliefs about others, and so on.

In this chapter, we present a method for computing *approximately* optimal policies for the finitely nested `I-POMDP` framework. Since an agent's belief is defined over other agents' models, which may be a complex continuous space, sampling methods, which are immune to the high dimensionality of the underlying space are a promising approach. We adapt particle filtering (Doucet, Freitas, & Gordon, 2001; Gordon, Salmond, & Smith, 1993), and more specifically the bootstrap filter (Gordon et al., 1993), to the multiagent setting, resulting in the *interactive particle filter* (Doshi & Gmytrasiewicz, 2005a, 2005b). Mirroring the hierarchical character of interactive beliefs, our approach involves nested sampling at each of the hierarchical levels of beliefs. Our method is applicable to agents that start with a prior belief and optimize over finite horizons.

Consequently, our method finds applications for online plan computation. We derive error bounds of our approach, and empirically demonstrate its performance on simple test problems. In order to beat back the curse of history, we present a complementary method based on sampling observations while building the look ahead reachability tree during value iteration. This translates into considering only those future beliefs during value iteration, that are likely . We report on the additional computational savings obtained when we combine this method with the interactive particle filter, and also provide empirical results.

Rest of this chapter is structured in the following manner. We briefly review the various sampling methods and their relevance, and the use of particle filters in previous works in Section 6.1. In Section 6.2, we introduce the traditional particle filtering technique concentrating on bootstrap filters in particular. We then present the interactive particle filter that approximates the I−POMDP belief update in Section 6.4. This is followed by a method that utilizes the interactive particle filter to compute solutions to I−POMDPs, in Section 6.5. We also comment on the asymptotic convergence and compute error bounds of our approach. In Section 6.6, we report on the performance of our approximation method on simple test problems. In Section 6.7, we provide a technique for mitigating the curse of history. We summarize the chapter in Section 6.8, lay out the contributions of this work in Section 6.9, and outline future research directions in Section 6.10.

6.1 Related Work

Several approaches to non-linear recursive Bayesian estimation exist. Amongst these, the extended Kalman filter (EKF) (Sorenson, 1985), is most popular. The EKF linearises the estimation problem so that the Kalman filter can be applied. The required p.d.f. is still approximated by a Gaussian, which may lead to filter divergence, and therefore an increase in the error. Other approaches include the Gaussian sum filter (Sorenson & Alspach, 1971), and superimposing a grid over the state space with the belief being evaluated only over the grid points (Kramer & Sorenson, 1988). In the latter approach, the choice of an efficient grid is non-trivial, and the number of grid points that must be considered is exponential in the dimensions of the state space. Recently, techniques that utilize Monte Carlo (MC) sampling for approximating the Bayesian state estimation problem have received much attention. These techniques are general enough, in that, they are applicable to both linear, as well as, non-linear problem domains. Amongst the spectrum of MC techniques, two that have been particularly well-studied in sequential settings are Markov chain Monte Carlo (MCMC) (Hastings, 1970), and particle filters (Doucet et al., 2001; Gordon et al., 1993). Approximating the I−POMDP belief

update using the former technique, may turn out to be computationally exhaustive. Specifically, MCMC algorithms (Hastings, 1970) utilize rejection sampling, that may cause a large number of intentional models to be sampled, solved, and rejected, before one is utilized for transition. However, particle filters do not employ rejection sampling, and produce reasonable approximations of the posterior while being computationally feasible.

Particle filters have previously been successfully applied to approximate the belief update in continuous state single agent POMDPs (Thrun, 2000; Poupart, Ortiz, & Boutilier, 2001). While Thrun integrates particle filtering with Q-learning to learn the policy, Poupart et al. assume the existence of an exact value function and present an error bound analysis of using particle filters. Loosely related to our work are the sampling algorithms that appear in (Ortiz & Kaelbling, 2000) for selecting actions in influence diagrams, but this work does not focus on sequential decision making. In the multiagent setting, particle filters have been employed for collaborative multi-robot localization (Fox, Burgard, Kruppa, & Thrun, 2000). In this application, the emphasis was on predicting the position of the robot, and not the decisions and actions of the other robots (which is the focus of our work). Additionally, to facilitate fast localization, beliefs of other robots encountered during motion were considered to be fully observable.

6.2 Particle Filter for the Single Agent Setting

Particle filters represent a specific implementation of Bayes filters (Eq. 2.1), tailored towards making Bayes filters applicable to non-linear dynamic systems. Rather than sampling directly from the target distribution, which is often difficult to compute, particle filters adopt the method of importance sampling (Geweke, 1989), which allows samples to be drawn from a more tractable distribution called the *proposal distribution*, π. Specifically, if $Pr(S^t|o_i^t, a_i^{t-1}, b_i^{t-1})$ is the target posterior distribution, and $\pi(S^t|o_i^t, a_i^{t-1}, b_i^{t-1})$ the proposal distribution, and the support of $\pi(S^t|o_i^t, a_i^{t-1}, b_i^{t-1})$ includes the support of $Pr(S^t|o_i^t, a_i^{t-1}, b_i^{t-1})$, we can approximate the posterior by sampling N i.i.d. particles $\{s^{(n)}, n = 1...N\}$ according to $\pi(S^t|o_i^t, a_i^{t-1}, b_i^{t-1})$, and assigning to each particle a normalized importance weight:

$$w^{(n)} = \frac{\tilde{w}(s^{(n)})}{\sum_{n=1}^{N} \tilde{w}(s^{(n)})} \quad \text{where} \quad \tilde{w}(s^{(n)}) = \frac{Pr(s^{(n)}|o_i^t, a_i^{t-1}, b_i^{t-1})}{\pi(s^{(n)}|o_i^t, a_i^{t-1}, b_i^{t-1})}$$

Each probability, $Pr(s'|o_i^t, a_i^{t-1}, b_i^{t-1})$, is then approximated by:

$$Pr_N(s'|o_i^t, a_i^{t-1}, b_i^{t-1}) = \sum_{n=1}^{N} w^{(n)} \delta(s' - s^{(n)})$$

where $\delta(\cdot)$ is the Dirac-delta function. As $N \rightarrow +\infty$, $Pr_N(s'|o_i^t, a_i^{t-1}, b_i^{t-1}) \stackrel{a.s.}{\rightarrow} Pr(s'|o_i^t, a_i^{t-1}, b_i^{t-1})$. When applied recursively over several steps, importance sampling leads to a large variance in weights. To avoid this degeneracy, Gordon et al. (1993) suggested inserting a resampling step, which would increase the population of particles that had high importance weights, thereby increasing the tracking ability of the particle filter. Since particle filtering extends importance sampling sequentially and appends a resampling step, it has also been called sequential importance sampling and resampling (SISR).

The general algorithm for the particle filtering technique is given in (Doucet et al., 2001). We shall concentrate on a specific implementation of this algorithm, that has previously been studied under various names such as Monte Carlo localization, survival of the fittest, and bootstrap filter. The implementation maintains a set of N particles denoted by $\widetilde{b}_i^{t-1}$ independently sampled from the prior, b_i^{t-1}. Each particle is then propagated forwards in time, using the transition kernel T_i of the environment. Each particle is then weighted by the likelihood of perceiving the observation from the state that the particle represents, as given by the observation function O_i. This is followed by the (unbiased) resampling step, in which particles are picked proportionately to their weights, and a uniform weight is attached to each particle. We outline the algorithm of the bootstrap filter in Fig. 6.1. A rigorous proof of the convergence of this algorithm towards the true posterior as $N \rightarrow \infty$ is outlined in (Crisan & Doucet, 2002).

Let us understand the working of the bootstrap filter in the context of a simple example – the single agent tiger problem described in Section 2.2.2 of Chapter 2. Let the agent have a prior belief according to which it is uninformed about the location of the tiger. In other words, it believes with a probability of 0.5 that the tiger is behind the left door, and with a similar probability that the tiger is behind the right door. We will see how the agent approximately updates its belief using the particle filter when, say, it listens and hears a growl from the left. Fig. 6.2 illustrates the particle filtering process. Since the agent is uninformed about the tiger's location, we start with an equal number of particles (samples) denoting TL (red) and TR (blue). The initial sample set is approximately representative of the agent's prior belief of 0.5. Since listening does not change the location of the tiger, the composition of the sample set remains unchanged after propagation. On

Function PARTICLEFILTER($\widetilde{b}_i^{t-1}, a_i^{t-1}, o_i^t$) **returns** $\widetilde{b}_i^t$

1. $\widetilde{b}_i^{tmp} \leftarrow \phi, \widetilde{b}_i^t \leftarrow \phi$

   Importance Sampling

2. **for all** $s^{(n),t-1} \in \widetilde{b}_i^{t-1}$ **do**
3. Sample $s^{(n),t} \sim T_i(S^t | a_i^{t-1}, s^{(n),t-1})$
4. Weight $s^{(n),t}$ with the importance weight: $\widetilde{w}_t^{(n)} = O_i(o_i^t | s^{(n),t}, a_i^{t-1})$
5. $\widetilde{b}_i^{tmp} \overset{\cup}{\leftarrow} (s^{(n),t}, \widetilde{w}_t^{(n)})$
6. Normalize all $\widetilde{w}_t^{(n)}$ so that $\sum_{n=1}^{N} w_t^{(n)} = 1$

   Selection

7. Resample with replacement N particles $\{s^{(n),t}, n = 1...N\}$ from the set $\widetilde{b}_i^{tmp}$ according to the importance weights.
8. $\widetilde{b}_i^t \leftarrow \{s^{(n),t}, n = 1...N\}$
9. **return** $\widetilde{b}_i^t$

end function

Figure 6.1: Particle filtering for approximating the Bayes filter. Note that the Bayes filter is precisely the POMDP belief update that we have seen previously.

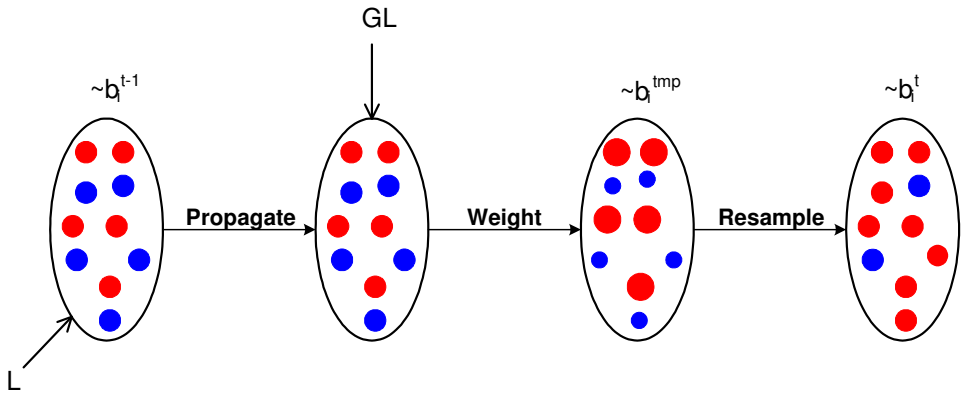

Figure 6.2: Particle filtering for state estimation in the single agent tiger problem. The red and blue particles denote the states TL and TR respectively. The particle filtering process consists of three steps: *Propagation* (line 3 of Fig. 6.1), *Weighting* (line 4), and *Resampling* (line 7).

hearing a growl from the left (GL), the red particles denoting TL will be tagged with a larger weight (0.85) because they are more likely to be responsible for GL, than the blue particles denoting TR (0.15). Here, the size of the particle is proportional to the weight attached to the particle. Finally, the resampling step yields the sample set at time step t, which contains more particles denoting TL than TR. This sample set approximately represents the updated belief of 0.85 of the agent.

6.3 Sampling from Nested Beliefs

As we mentioned, there is a continuum of intentional and subintentional models of an agent. Since an agent is unaware of the true models of interacting agents *ex ante*, it must maintain a belief over all possible candidate models. The complexity of this space precludes practical implementations of I-POMDPs for all but the simplest settings. Approximations based on sampling use a finite set of sample points to represent a complete belief state.

In order to sample from nested beliefs we need a language to represent them. We introduce polynomials as a basic building block and assume that all probability density functions representing agent i's beliefs nested to some level l are polynomials.[1][2] Using the definitions of nested interactive state spaces in Section 4.4 as a basis, we construct the polynomial based belief representation in a bottom-up manner, as follows:
Level 0 belief: i's level 0 belief, $b_{i,0} \in \Delta(S)$, is a vector of probabilities over each physical state.

$$b_{i,0} = \langle p_{i,0}(s_1), p_{i,0}(s_2), \ldots, p_{i,0}(s_{|S|}) \rangle$$

Level 1 belief: i's first level belief, $b_{i,1} \in \Delta(S \times \{\Theta_{j,0} \cup SM_j\})$, is a vector of densities over j's level 0 beliefs for each state and j's intentional frame, as well as densities over j's histories for each state and j's subintentional frame. We will represent the densities using polynomials. Formally,

$$b_{i,1} = \langle p_{i,1}^{\langle s,\widehat{m}_j \rangle_1}, p_{i,1}^{\langle s,\widehat{m}_j \rangle_2}, \ldots, p_{i,1}^{\langle s,\widehat{m}_j \rangle_{|S||\widehat{M}_j|}} \rangle$$

such that,

$$\begin{array}{lll} \forall \langle s, \widehat{m}_j \rangle : & & \\ \text{if } \widehat{m}_j \in \widehat{\Theta}_j & \text{then } p_{i,1}^{\langle s,\widehat{\theta}_j \rangle}(b_{j,0}) & = c_1 + c_2 p_{j,0}(s_1) + c_3 p_{j,0}(s_2) + \ldots + c_{(d+1)^{|S|-1}} \prod_{q=1}^{|S|-1} p_{j,0}(s_q)^d \\ & \text{else } p_{i,1}^{\langle s,\widehat{m}_j \rangle}(h_j) & = c_1 + c_2 h_j + \ldots + c_{d+1} h_j^d \end{array}$$

Because belief is a probability distribution the areas of the polynomials must sum to 1. Formally,
$\sum_{s,\widehat{m}_j} \int p_{i,1}^{\langle s,\widehat{m}_j \rangle} = 1$.

[1] We use polynomials because it is a well known representation capable of approximating any function over Euclidean space to arbitrary accuracy.

[2] An n-variable polynomial f of degree d has $(d+1)^n$ coefficients.

Level 2 belief: i's second level belief, $b_{i,2} \in \Delta(S \times \{\Theta_{j,1} \cup \Theta_{j,0} \cup SM_j\})$, is a vector of densities over j's level 1 and level 0 beliefs for each state and j's intentional frame, as well as densities over j's histories for each state and j's subintentional frame. Formally,

$$b_{i,2} = \langle p_{i,2}^{\langle s,\widehat{m}_j\rangle_1}, p_{i,2}^{\langle s,\widehat{m}_j\rangle_2}, \ldots, p_{i,2}^{\langle s,\widehat{m}_j\rangle_{|S||\widehat{M}_j|}} \rangle$$

such that

$\forall \langle s, \widehat{m}_j \rangle$:

if $\widehat{m}_j \in \widehat{\Theta}_j$ then

$$\begin{aligned} p_{i,2}^{\langle s,\widehat{\theta}_j\rangle}(b_{j,1}) = \; & c_1 + c_2 p_{j,1}^{\langle s,\widehat{\theta}_i\rangle_1}(b_{j,0}) + c_3 p_{j,1}^{\langle s,\widehat{\theta}_i\rangle_2}(b_{j,0}) + \ldots \\ & + c_{(d+1)^{|S||\widehat{M}_i|}} \textstyle\prod_{q=1}^{|S||\widehat{\Theta}_i|} p_{j,1}^{\langle s,\widehat{\theta}_i\rangle_q}(b_{j,0})^d \prod_{q=|S||\widehat{\Theta}_i|+1}^{|S||\widehat{M}_i|} p_{j,1}^{\langle s,\widehat{m}_i\rangle_q}(h_j)^d \\ p_{i,2}^{\langle s,\widehat{\theta}_j\rangle}(b_{j,0}) = \; & c_1 + c_2 p_{j,0}(s_1) + c_3 p_{j,0}(s_2) + \ldots + c_{(d+1)^{|S|-1}} \textstyle\prod_{q=1}^{|S|-1} p_{j,0}(s_q)^d \end{aligned}$$

else

$$p_{i,1}^{\langle s,\widehat{m}_j\rangle}(h_j) = \; c_1 + c_2 h_j + \ldots + c_{d+1} h_j^d$$

Belief being a probability distribution the areas of the polynomials must sum to 1. Since polynomials are described by their parameters, each of i's level 2 polynomial defined over j's level 1 belief is a density over the degrees, coefficients, and variables of j's level 1 polynomials that constitute j's level 1 belief.

Level l belief: We generalize our polynomial representation to any level $l > 0$. i's level l belief, $b_{i,l} \in \Delta(S \times \{\Theta_{j,l-1} \cup \ldots \cup \Theta_{j,0} \cup SM_j\})$ is a vector of densities over j's level $l-1$, level $l-2$,..., downto level 0 beliefs, depending on the state and j's frame, as well as densities over j's histories for each state and j's frame. Formally,

$$b_{i,l} = \langle p_{i,l}^{\langle s,\widehat{m}_j\rangle_1}, p_{i,l}^{\langle s,\widehat{m}_j\rangle_2}, \ldots, p_{i,l}^{\langle s,\widehat{m}_j\rangle_{|S||\widehat{M}_j|}} \rangle$$

such that

$$
\begin{array}{lll}
\forall \langle s, \widehat{m}_j \rangle : & & \\
\text{if } \widehat{m}_j \in \widehat{\Theta}_j \quad \text{then} \quad p_{i,l}^{\langle s,\widehat{\theta}_j \rangle}(b_{j,l-1}) = & c_1 + c_2 p_{j,l-1}^{\langle s,\widehat{\theta}_i \rangle_1}(b_{j,l-2}) + c_3 p_{j,l-1}^{\langle s,\widehat{\theta}_i \rangle_2}(b_{j,l-2}) + \ldots \\
& + c_{(d+1)^{|S||\widehat{M}_i|}} \prod_{q=1}^{|S||\widehat{\Theta}_i|(l-1)} p_{j,l-1}^{\langle s,\widehat{\theta}_i \rangle_q}(b_{j,l-2})^d \\
& \times \prod_{q=|S||\widehat{\Theta}_i|(l-1)+1}^{|S||\widehat{M}_i|} p_{j,l-1}^{\langle s,\widehat{m}_i \rangle_q}(h_j)^d \\
p_{i,l}^{\langle s,\widehat{\theta}_j \rangle}(b_{j,l-2}) = & c_1 + c_2 p_{j,l-2}^{\langle s,\widehat{\theta}_i \rangle_1}(b_{j,l-3}) + c_3 p_{j,l-2}^{\langle s,\widehat{\theta}_i \rangle_2}(b_{j,l-3}) + \ldots \\
& + c_{(d+1)^{|S||\widehat{M}_i|}} \prod_{q=1}^{|S||\widehat{\Theta}_i|(l-2)} p_{j,l-2}^{\langle s,\widehat{\theta}_i \rangle_q}(b_{j,l-3})^d \\
& \times \prod_{q=|S||\widehat{\Theta}_i|(l-2)+1}^{|S||\widehat{M}_i|} p_{j,l-2}^{\langle s,\widehat{m}_i \rangle_q}(h_j)^d \\
& . \\
& . \\
& . \\
p_{i,l}^{\langle s,\widehat{\theta}_i \rangle}(b_{j,0}) = & c_1 + c_2 p_{j,0}(s_1) + c_3 p_{j,0}(s_2) + \ldots + c_{(d+1)^{|S|-1}} \prod_{q=1}^{|S|-1} p_{j,0}(s_q)^d \\
\text{else} \quad p_{i,l}^{\langle s,\widehat{m}_j \rangle}(h_j) = & c_1 + c_2 h_j + \ldots + c_{d+1} h_j^d
\end{array}
$$

As we mentioned before, the areas of the polynomials must sum to 1.

Example 6.1. *To illustrate our polynomial based representation, we utilize the multiagent tiger problem introduced previously in Section 4.5 of Chapter 4. We will again assume that each agent knows that other's possible models are intentional nested to one level lower than its own, and it is uncertain of only the other's lower level beliefs.*

An example level 0 belief of i, $b_{i,0} = \langle p_{i,0}(TL), p_{i,0}(TR) \rangle$, is $\langle 0.7, 0.3 \rangle$ that assigns a probability of 0.7 to TL and 0.3 to TR.

An example level 1 belief of i, $b_{i,1} = \langle p_{i,1}^{\langle TL,\widehat{\theta}'_j \rangle}, p_{i,1}^{\langle TR,\widehat{\theta}'_j \rangle} \rangle$, in the tiger problem is one according to which i is uninformed about j's level 0 beliefs and about the location of the tiger (see Fig. 4.7(ii)). The corresponding polynomials are, $p_{i,1}^{\langle TL,\widehat{\theta}'_j \rangle}(b_{j,0}) = p_{i,1}^{\langle TR,\widehat{\theta}'_j \rangle}(b_{j,0}) = 0.5$. Here, the polynomials for each location of the tiger and j's frame are identical, and are of degree 0 with $c_1 = 0.5$.

A level 2 belief of i, $b_{i,2} = \langle p_{i,2}^{\langle TL,\widehat{\theta}'_j \rangle}, p_{i,2}^{\langle TR,\widehat{\theta}'_j \rangle} \rangle$, is one in which i considers increasingly complex level 1 beliefs of j (for example, $p_{j,1}$ of higher degrees) as less likely (Occam's Razor), and is uninformed of the location of the tiger. A level 2 polynomial of i is defined over $b_{j,1}$ which as we saw above, is a vector of two polynomials. Since a polynomial is described by its degree, coefficients, and variables,

$p_{i,2}^{\langle TL,\widehat{\theta}'_j\rangle}(b_{j,1}) = p_{i,2}^{\langle TL,\widehat{\theta}'_j\rangle}(d_1, d_2, \langle c_1, c_2, \ldots, c_{d_{max}+1}\rangle_1, \langle c_1, c_2, \ldots, c_{d_{max}+1}\rangle_2, p_{i,0}(TL))$ *and analogously for* $p_{i,2}^{\langle TR,\widehat{\theta}'_j\rangle}(b_{j,1})$. d_{max} *is an upper bound on the degree. We express the example belief using a normalized Taylor series expansion of* $2^{-(d_1+d_2)}$*, i.e.* $p_{i,2}^{\langle TL,\widehat{\theta}'_j\rangle}(d_1, d_2, \langle c_1, c_2, \ldots, c_{d_{max}+1}\rangle_1, \langle c_1, c_2, \ldots, c_{d_{max}+1}\rangle_2,$ $p_{i,0}(TL)) = p_{i,2}^{\langle TR,\widehat{\theta}'_j\rangle}(d_1, d_2, \langle c_1, c_2, \ldots, c_{d_{max}+1}\rangle_1, \langle c_1, c_2, \ldots, c_{d_{max}+1}\rangle_2, p_{i,0}(TL))$
$= \beta \sum_{n=0}^{5}(-1)^n ln(2)^n \left(\frac{1}{n!}(-1)^n (d_1-3)^n + \sum_{m=0}^{n} \frac{1}{n!}(d_2-3)^m (d_1-3)^{n-m} \right)$ *where* β *is the normalizing constant and* $d_{max} = 3$. [3]

Now that we have a representation for agent i's nested beliefs, we turn our attention to sampling from these beliefs. In Fig. 6.3 we present the algorithm, PRIORSAMPLE [4] for recursively sampling from nested beliefs using the polynomial based representation of the beliefs. In a nutshell, PRIORSAMPLE generates nested sample sets by recursively sampling the parameters of the lower level polynomials from the higher level ones. The recursion bottoms out when a level 0 belief is sampled from a level 1 polynomial density. In the algorithm, k will stand for either agent i or j, and $-k$ for the other agent, j or i, as appropriate. For a visualization of the nested sample sets, see Fig. 6.6.

For singly nested beliefs, the sampling is straightforward: For each of the N sampled physical states, $s^{(n)}$, and j's frame, $\widehat{m}_j^{(n)}$, either the level 0 belief vector, $\langle p_{j,0}(s_1), \ldots, p_{j,0}(s_{|S|})\rangle$, or the observation history, h_j is sampled from the polynomial $p_{j,1}^{\langle s^{(n)},\widehat{\theta}_j^{(n)}\rangle}$ or $p_{j,1}^{\langle s^{(n)},\widehat{m}_j^{(n)}\rangle}$, respectively (lines 2–15). Analogously, from i's level 2 beliefs, we will sample j's level 1 belief or its observation history, depending on whether j's sampled frame is intentional or subintentional. Let's consider the case where j's sampled frame is intentional. Because j's level 1 belief is represented using $|S||\widehat{M}_i|$ polynomials, we must sample the parameters – degree and coefficients – of these polynomials. Since the number of coefficients depends in part on the degree of the polynomial, we must sample the degrees first by marginalizing the level 2 polynomial over the coefficients. Therefore, we sample the joint degree $\langle d_1, d_2, \ldots, d_{|S||\widehat{M}_i|}\rangle \sim p_{i,2}^{\langle s^{(n)},\widehat{\theta}_j^{(n)}\rangle}(b_{j,1})$ and use the individual degrees to sample the coefficients $\langle c_1, \ldots, c_{(d_1+1)^{|S|-1}}\rangle_1, \ldots, \langle c_1, \ldots, c_{(d_{|S||\widehat{\Theta}_i|}+1)^{|S|-1}}\rangle_{|S||\widehat{\Theta}_i|}$,

[3] We use $2^{-K(x)}$ where $K(\cdot)$ is the Kolmogorov complexity as a mathematical formalization of Occam's razor (Li & Vitanyi, 1997). For simplicity and computability, we use the degrees of the lower level polynomials as approximate measures of the complexity.

[4] In favor of simplicity and clarity, in PRIORSAMPLE and in all other algorithms in this chapter, for level l beliefs we restrict the intentional models of the other agent to level $l-1$ only.

Function PRIORSAMPLE($\langle p_{k,l}^{\langle s,\widehat{\theta}_{-k}\rangle_1}, \ldots, p_{k,l}^{\langle s,\widehat{\theta}_{-k}\rangle_{|S||\widehat{M}_{-k}|}}\rangle, l > 0$) **returns** $\widetilde{b}_{k,l}$

1. **for** n **from** 1 **to** N **do**
2. $\quad b_{-k,l-1}^{(n)} \leftarrow \phi$
3. $\quad$ **if** $(l = 1)$ **then**
4. $\quad\quad$ **for each** $s \in S, \widehat{m}_{-k} \in \widehat{M}_{-k}$ **do**
5. $\quad\quad\quad Pr(s) \stackrel{+}{\leftarrow} \int p_{k,1}^{\langle s,\widehat{m}_{-k}\rangle}$
6. $\quad\quad$ Sample $s^{(n)} \sim Pr(S)$
7. $\quad\quad$ **for each** $\widehat{m}_{-k} \in \widehat{M}_{-k}$ **do**
8. $\quad\quad\quad Pr(\widehat{m}_{-k}) \stackrel{+}{\leftarrow} \int p_{k,1}^{\langle s^{(n)},\widehat{m}_{-k}\rangle}$
9. $\quad\quad$ Sample $\widehat{m}_{-k}^{(n)} \sim Pr(\widehat{M}_{-k})$
10. $\quad\quad$ **if** $\widehat{m}_{-k}^{(n)} \in \widehat{\Theta}_{-k}$ **then** $\quad$ /* If other's frame is intentional */
11. $\quad\quad\quad$ Sample $b_{-k,0}^{(n)} = \langle p_{j,0}(s_1), p_{j,0}(s_2), \ldots, p_{j,0}(s_{|S|-1})\rangle \sim p_{k,1}^{\langle s^{(n)},\widehat{\theta}_{-k}^{(n)}\rangle}$
12. $\quad\quad\quad is_k^{(n)} \leftarrow \langle s^{(n)}, \langle b_{-k,0}^{(n)}, \widehat{\theta}_{-k}^{(n)}\rangle\rangle$
13. $\quad\quad$ **else** $\quad$ /* If other's frame is subintentional */
14. $\quad\quad\quad$ Sample $h_{-k}^{(n)} \sim p_{k,1}^{\langle s^{(n)},\widehat{m}_{-k}^{(n)}\rangle}$
15. $\quad\quad\quad is_k^{(n)} \leftarrow \langle s^{(n)}, \langle h_{-k}^{(n)}, \widehat{m}_{-k}^{(n)}\rangle\rangle$
16. $\quad$ **else if** $(l = 2)$ **then**
17. $\quad\quad$ **for each** $s \in S, \widehat{m}_{-k} \in \widehat{M}_{-k}$ **do**
18. $\quad\quad\quad Pr(s) \stackrel{+}{\leftarrow} \int p_{k,2}^{\langle s,\widehat{m}_{-k}\rangle}$
19. $\quad\quad$ Sample $s^{(n)} \sim Pr(S)$
20. $\quad\quad$ **for each** $\widehat{m}_{-k} \in \widehat{M}_{-k}$ **do**
21. $\quad\quad\quad Pr(\widehat{m}_{-k}) \stackrel{+}{\leftarrow} \int p_{k,2}^{\langle s^{(n)},\widehat{m}_{-k}\rangle}$
22. $\quad\quad$ Sample $\widehat{m}_{-k}^{(n)} \sim Pr(\widehat{M}_{-k})$
23. $\quad\quad$ **if** $\widehat{m}_{-k}^{(n)} \in \widehat{\Theta}_{-k}$ **then** $\quad$ /* If other's frame is intentional */
24. $\quad\quad\quad$ Sample $\langle d_1, \ldots, d_{|S||\widehat{\Theta}_k|}, \ldots, d_{|S||\widehat{M}_k|}\rangle \sim p_{k,2}^{\langle s^{(n)},\widehat{\theta}_{-k}^{(n)}\rangle}$
25. $\quad\quad\quad$ Sample $\langle\langle c_1, c_2, \ldots, c_{(d_1+1)^{|S|-1}}\rangle_1, \ldots, \langle c_1, c_2, \ldots, c_{(d_{|S||\widehat{\Theta}_k|}+1)^{|S|-1}}\rangle_{|S||\widehat{\Theta}_k|}, \langle c_1, \ldots, c_{d_{|S||\widehat{\Theta}_k|+1}+1}\rangle_{|S||\widehat{\Theta}_k|+1}, \ldots, \langle c_1, c_2, \ldots, c_{d_{|S||\widehat{M}_k|}+1}\rangle_{|S||\widehat{M}_k|}\rangle \sim p_{k,2}^{\langle s^{(n)},\widehat{\theta}_{-k}^{(n)}\rangle}$
26. $\quad\quad\quad$ **for** i **from** 1 **to** $|S||\widehat{\Theta}_k|$ **do**
27. $\quad\quad\quad\quad p_{-k,1}^{\langle s,\widehat{\theta}_k\rangle} \leftarrow \langle d_i, c_1, c_2, \ldots, c_{(d_i+1)^{|S|-1}}\rangle, b_{-k,1}^{(n)} \stackrel{\cup}{\leftarrow} p_{-k,1}^{\langle s,\widehat{\theta}_k\rangle}$
28. $\quad\quad\quad$ **for** i **from** $|S||\widehat{\Theta}_k|$+1 **to** $|S||\widehat{M}_k|$ **do**
29. $\quad\quad\quad\quad p_{-k,1}^{\langle s,\widehat{m}_k\rangle} \leftarrow \langle d_i, c_1, c_2, \ldots, c_{d_i+1}\rangle, b_{-k,1}^{(n)} \stackrel{\cup}{\leftarrow} p_{-k,1}^{\langle s,\widehat{m}_k\rangle}$
30. $\quad\quad\quad$ Normalize $\widetilde{b}_{-k,1}^{(n)}$
31. $\quad\quad\quad \widetilde{b}_{-k,1}^{(n)} \leftarrow$ PRIORSAMPLE($b_{-k,1}^{(n)}$,1)
32. $\quad\quad\quad is_k^{(n)} \leftarrow \langle s^{(n)}, \langle \widetilde{b}_{-k,1}^{(n)}, \widehat{\theta}_{-k}^{(n)}\rangle\rangle$
33. $\quad\quad$ **else** $\quad$ /* If other's frame is subintentional */
34. $\quad\quad\quad$ Sample $h_{-k}^{(n)} \sim p_{k,2}^{\langle s^{(n)},\widehat{m}_{-k}^{(n)}\rangle}$

```
            /* If other's frame is subintentional */
35.         is_k^(n) ← ⟨s^(n), ⟨h_{-k}^(n), m̂_{-k}^(n)⟩⟩
36.     else
37.       for each s ∈ S, m̂_{-k} ∈ M̂_{-k} do
38.           Pr(s) +← ∫ p_{k,l}^{⟨s,m̂_{-k}⟩}
39.       Sample s^(n) ~ Pr(S)
40.       for each m̂_{-k} ∈ M̂_{-k} do
41.           Pr(m̂_{-k}) +← ∫ p_{k,l}^{⟨s^(n),m̂_{-k}⟩}
42.       Sample m̂_{-k}^(n) ~ Pr(M̂_{-k})
43.       if m̂_{-k}^(n) ∈ Θ̂_{-k} then          /* If other's frame is intentional */
44.         Sample ⟨d_1, ..., d_{|S||Θ̂_k|}, ..., d_{|S||M̂_k|}⟩ ~ p_{k,l}^{⟨s^(n),θ̂_{-k}^(n)⟩}
            Sample ⟨⟨c_1, ..., c_{(d_1+1)^f(l)}⟩1, ..., ⟨c_1, ..., c_{(d_{|S||Θ̂_k|}+1)^f(l)}⟩_{|S||Θ̂_k|}
45.                 , ⟨c_1, ..., c_{d_{|S||Θ̂_k|+1}+1}⟩_{|S||Θ̂_k|+1}, ..., ⟨c_1, ..., c_{d_{|S||M̂_k|}+1}⟩_{|S||M̂_k|}⟩
                    ~ p_{k,l}^{⟨s^(n),θ̂_{-k}^(n)⟩}
46.         for i from 1 to |S||Θ̂_k| do
47.           p_{-k,l-1}^{⟨s,θ̂_k⟩} ← ⟨d_i, c_1, ..., c_{d_i+1^f(l)}⟩
48.           b_{-k,l-1}^(n) +← p_{-k,l-1}^{⟨s,θ̂_k⟩}
49.         for i from |S||Θ̂_k|+1 to |S||M̂_k| do
50.           p_{-k,l-1}^{⟨s,m̂_k⟩} ← ⟨d_i, c_1, ..., c_{d_i+1}⟩
51.           b_{-k,l-1}^(n) +← p_{-k,l-1}^{⟨s,m̂_k⟩}
52.         Normalize b_{-k,l-1}^(n)
53.         b̃_{-k,l-1}^(n) ← PRIORSAMPLE(b_{-k,l-1}^(n), l − 1)
54.         is_k^(n) ← ⟨s^(n), ⟨ b̃_{-k}^(n), θ̂_{-k}^(n)⟩⟩
55.       else                                  /* If other's frame is subintentional */
56.         Sample h_{-k}^(n) ~ p_{k,2}^{⟨s^(n),m̂_{-k}^(n)⟩}
57.         is_k^(n) ← ⟨s^(n), ⟨h_{-k}^(n), m̂_{-k}^(n)⟩⟩
58.     b̃_{k,l} +← {is_k^(n)}
59.return b̃_{k,l}
end function
```

Figure 6.3: Algorithm for sampling from a nested belief that is represented using polynomial densities at each level. Here, k denotes either agent i or j, and $-k$ denotes j or i respectively.

$\langle c_1, \ldots, c_{d_{|S||\widehat{\Theta}_i|+1}+1}\rangle_{|S||\widehat{\Theta}_i|+1}$, ..., $\langle c_1, \ldots, c_{d_{|S||\widehat{M}_i|}+1}\rangle_{|S||\widehat{M}_i|} \sim p_{i,2}^{\langle s^{(n)}, \widehat{\theta}_j^{(n)}\rangle}(b_{j,1})$ (lines 16–35). The degrees and coefficients are then assembled to form the $|S||\widehat{M}_i|$ polynomials that represent j's level 1 belief. We generalize the sampling procedure to all levels greater than two in the remainder of the algorithm. We observe that we are faced with sampling an exploding number of coefficients for increasing

nesting levels. If the other agent's sampled frame is intentional, then the number of coefficients for describing its polynomial is $(d+1)^{f(l)}$ where d is the sampled degree and $f(l) = |S||\widehat{\Theta}|[1+(d_{max}+1)^{l\times^{.^{.^{.^{|S||\widehat{\Theta}|[1+(d_{max}+1)^{|S|-1}]+|S||\widehat{SM}|[1+(d_{max}+1)]}}}}}] + |S||\widehat{SM}|[1+(d_{max}+1)]$.

6.4 Interactive Particle Filter for the Multiagent Setting

We presented the algorithm for the traditional bootstrap filter in Section 6.2. As we mentioned before, the bootstrap filter is a MC sampling based randomized implementation of the POMDP belief update (Bayes filter). We extend this implementation to the I-POMDP belief update presented previously in Section 4.3.2 of Chapter 4.

6.4.1 Description

Our extension of the boostrap filter to the multiagent case, which we call an *interactive particle filter* (I-PF), similar to basic particle filtering, requires the key steps of *importance sampling* and *selection*. The resulting algorithm, inherits the convergence properties of the original algorithm (Doucet et al., 2001). Specifically, the approximate posterior belief generated by the filter converges to the truth as the number of particles (N) tends to infinity. The extension to the multiagent setting turns out to be non-trivial because we are faced with an interactive belief hierarchy. Analogously to the I-POMDP belief update, the I-PF reduces to the traditional PF when there is only one agent in the environment.

The I-PF, described in Fig. 6.4, requires an initial set of N particles, $\widetilde{b}_{k,l}^{t-1}$, that is approximately representative of the agent's belief, along with the action, a_k^{t-1}, the observation, o_k^t, and the level of belief nesting, $l > 0$. As per our convention, k will stand for either agent i or j, and $-k$ for the other agent, j or i, as appropriate. Each particle, $is_k^{(n)}$, in the sample set represents the agent's possible interactive state, in which the belief, if present, may itself be a set of particles. Formally, $is_k^{(n)} = \langle s^{(n)}, m_{-k}^{(n)} \rangle$ where if $m_{-k}^{(n)} \in \Theta_{-k}$ (other's model is intentional), then $m_{-k}^{(n)} = \langle \widetilde{b}_{-k,l-1}^{(n)}, \widehat{\theta}_{-k}^{(n)} \rangle$, else $m_{-k}^{(n)} = \langle h_{-k}^{(n)}, \widehat{m}_{-k}^{(n)} \rangle$. Note that $\widetilde{b}_{k,0}^{(n)}$ is a probability distribution over the physical state space. We generate $\widetilde{b}_{k,l}^{t-1}$ by recursively sampling N particles from beliefs represented using polynomials at each level of nesting, using the PRIORSAMPLE procedure outlined in the previous section. The particle filtering proceeds by *propagating* each particle forward in time. However, as opposed to traditional particle filtering, this is not a one-step process. In order to perform the propagation, other agent's action must be known. If the model ascribed to the other agent is intentional,

Function I-PARTICLEFILTER($\widetilde{b}_{k,l}^{t-1}, a_k^{t-1}, o_k^t, l > 0$) **returns** $\widetilde{b}_{k,l}^t$

1. $\widetilde{b}_{k,l}^{tmp} \leftarrow \phi$, $\widetilde{b}_{k,l}^t \leftarrow \phi$

   Importance Sampling

2. **for all** $is_k^{(n),t-1} = \langle s^{(n),t-1}, m_{-k}^{(n),t-1} \rangle \in \widetilde{b}_{k,l}^{t-1}$ **do**
3. **if** $m_{-k}^{(n),t-1} \in \Theta_{-k}$ **then**
4. $Pr(A_{-k}|\theta_{-k}^{(n),t-1}) \leftarrow$ APPROXPOLICY($\theta_{-k}^{(n),t-1}, l-1$)
5. Sample $a_{-k}^{t-1} \sim Pr(A_{-k}|\theta_{-k}^{(n),t-1})$
6. **else**
7. Sample $a_{-k}^{t-1} \sim Pr(A_{-k}|m_{-k}^{(n),t-1})$
8. Sample $s^{(n),t} \sim T_k(S^t|a_k^{t-1}, a_{-k}^{t-1}, s^{(n),t-1})$
9. **for all** $o_{-k}^t \in \Omega_{-k}$ **do**
10. **if** $m_{-k}^{(n),t-1} \in \Theta_{-k}$ **then**
11. **if** (l = 1) **then**
12. $b_{-k}^{(n),t} \leftarrow$ LEVEL0BELIEFUPDATE($b_{-k}^{(n),t-1}, a_{-k}^{t-1}, o_{-k}^t$)
13. $\theta_{-k}^{(n),t} \leftarrow \langle b_{-k}^{(n),t}, \widehat{\theta}_{-k}^{(n)} \rangle$
14. $is_k^{(n),t} \leftarrow \langle s^{(n),t}, \theta_{-k}^{(n),t} \rangle$
15. **else**
16. $\widetilde{b}_{-k}^{(n),t} \leftarrow$ I-PARTICLEFILTER($\widetilde{b}_{-k}^{(n),t-1}, a_{-k}^{t-1}, o_{-k}^t, l-1$)
17. $\theta_{-k}^{(n),t} \leftarrow \langle \widetilde{b}_{-k}^{(n),t}, \widehat{\theta}_{-k}^{(n)} \rangle$
18. $is_k^{(n),t} \leftarrow \langle s^{(n),t}, \theta_{-k}^{(n),t} \rangle$
19. **else**
20. $h_{-k}^{(n),t} \leftarrow$ APPEND($h_{-k}^{(n),t-1}, o_{-k}^t$)
21. $m_{-k}^{(n),t} \leftarrow \langle h_{-k}^{(n),t}, \widehat{m}_{-k}^{(n)} \rangle$
22. $is_k^{(n),t} \leftarrow \langle s^{(n),t}, m_{-k}^{(n),t} \rangle$
23. Weight $is_k^{(n),t}$: $w_t^{(n)} = O_{-k}(o_{-k}^t|s^{(n),t}, a_k^{t-1}, a_{-k}^{t-1})$
24. Adjust weight: $w_t^{(n)} = w_t^{(n)} \times O_k(o_k^t|s^{(n),t}, a_k^{t-1}, a_{-k}^{t-1})$
25. $\widetilde{b}_{k,l}^{tmp} \overset{\cup}{\leftarrow} (is_k^{(n),t}, w_t^{(n)})$
26. Normalize all $w_t^{(n)}$ so that $\sum_{n=1}^N w_t^{(n)} = 1$

    Selection

27. Resample with replacement N particles $\{is_k^{(n),t}, n = 1...N\}$ from the set $\widetilde{b}_{k,l}^{tmp}$ according to the importance weights.
28. $\widetilde{b}_{k,l}^t \leftarrow \{is_k^{(n),t}, n = 1...N\}$
29. **return** $\widetilde{b}_{k,l}^t$

end function

Figure 6.4: Interactive particle filtering for approximating the I-POMDP belief update. A nesting of particle filters is used to update all levels of the belief. Also see Fig. 6.6 for a visualization.

then this is obtained by solving the other agent's model (using the algorithm APPROXPOLICY described in Section 6.5) to find a distribution over its actions, from which its action is sampled (line 4 in Fig. 6.4). Additionally, analogously to the exact belief update, for each of the other agent's possible observations, we must

Function LEVEL0BELIEFUPDATE($b_k^{t-1}, a_k^{t-1}, o_k^t$) **returns** b_k^t
1. $Pr(a_{-k}^{t-1}) \leftarrow 1/a_{-k}^{t-1}$
2. **for all** $s^t \in S$ **do**
3. sum $\leftarrow 0$
4. **for all** $s^{t-1} \in S$ **do**
5. $Pr(s^t|s^{t-1}, a_k^{t-1}) \leftarrow 0$
6. **for all** $a_{-k}^{t-1} \in A_{-k}$ **do**
7. $Pr(s^t|s^{t-1}, a_k^{t-1}) \stackrel{+}{\leftarrow} T_k(s^t|s^{t-1}, a_k^{t-1}, a_{-k}^{t-1}) Pr(a_{-k}^{t-1})$
8. sum $\stackrel{+}{\leftarrow} Pr(s^t|s^{t-1}, a_k^{t-1}) b_k^{t-1}(s^{t-1})$
9. $Pr(o_k^t|s^t, a_k^{t-1}) \leftarrow 0$
10. **for all** $a_{-k}^{t-1} \in A_{-k}$ **do**
11. $Pr(o_k^t|s^t, a_k^{t-1}) \stackrel{+}{\leftarrow} O_k(o_k^t|s^t, a_k^{t-1}, a_{-k}^{t-1}) Pr(a_{-k}^{t-1})$
12. $b_k^t(s^t) \leftarrow Pr(o_k^t|s^t, a_k^{t-1}) \times$ sum
13. Normalize the belief, b_k^t
14. **return** b_k^t

end function

Figure 6.5: The level 0 belief update which is similar to the exact POMDP belief update with a noise factor.

update its model (line 9). If its model is intentional, then we must update its belief state. If $l > 1$, updating the other agent's belief requires invoking the interactive particle filter for performing its belief update (lines 16–18). This recursion in depth of the belief nesting terminates when the level of nesting becomes one, and a LEVEL0BELIEFUPDATE described in Fig. 6.5 is performed (lines 12–14).[5] If the model of the other agent is subintentional, then we simply append the observation to its previous observation history. Though the propagation step generates $|\Omega_{-k}|N$ appropriately weighted particles, we *resample* N particles out of these (line 27), using an unbiased resampling scheme. A visualization of our implementation is shown in Fig. 6.6.

6.4.2 Illustration of the I-PF

We illustrate the operation of the I-PF using the multiagent tiger problem introduced in Section 4.5. For the sake of understanding, we restrict j's models to be intentional, assume that i is uncertain only of j's beliefs and not its frame, and consider singly nested beliefs for agent i. According to these beliefs, i knows that j is uninformed about the location of the tiger, and is itself unaware of where the tiger is. We demonstrate the operation of the I-PF for the case when i listens and hears a growl from the left and no creaks. This

[5] If the physical state space is also continuous or very large, then we would replace the level 0 belief update with a traditional particle filter. However, in doing so, we would loose the theoretical bounds given in Section 6.5.1

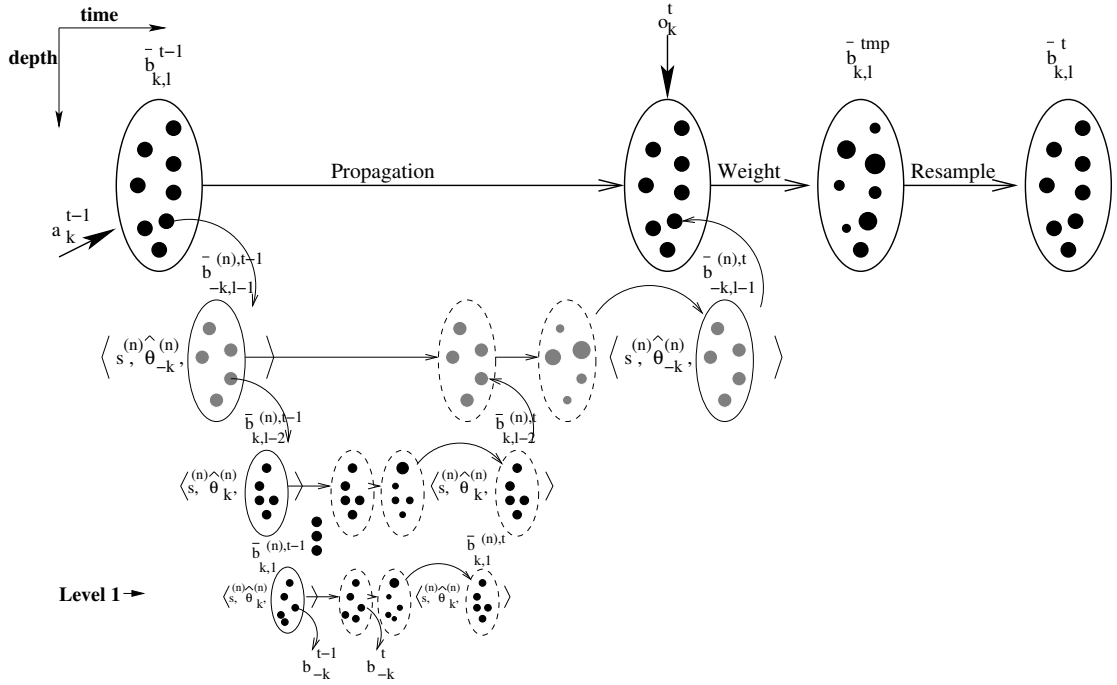

Figure 6.6: An illustration of the nesting in the interactive particle filter. Colors black and gray distinguish filtering for the two agents. Because the propagation step involves updating the other agent's beliefs, we perform particle filtering on its beliefs. The filtering terminates when it reaches the level 1 nesting, where a level 0 belief update is performed for the other agent.

example, is therefore, an approximate implementation of the exact `I-POMDP` belief update shown in Fig. 4.8 of Chapter 4.

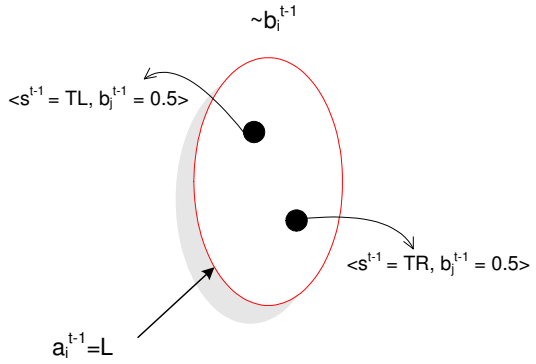

Figure 6.7: Initial sample set of 2 particles that is approximately representative of $b_{i,1}^{t-1}$.

In Fig. 6.7, we show the initial sample set, $\widetilde{b}_{i,1}^{t-1}$, consisting of $N = 2$ particles that is approximately

representative of i's beliefs. As shown, each particle is an interactive state consisting of the tiger's location and j's level 0 belief. Because i knows that j is uninformed, j's level 0 belief is 0.5 in both the particles.

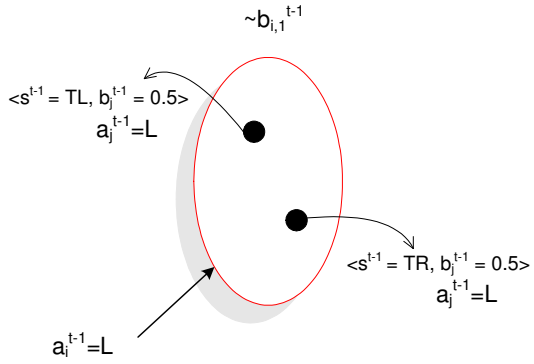

Figure 6.8: The initial sample set with j's optimal action shown for each particle.

As we mentioned before, the *propagation* of the particles from time step $t-1$ to t is a multi-step process. As the first step, we solve j's POMDP to compute its optimal action when its belief is 0.5. j's action is to listen since it does know the location of the tiger. We depict this in Fig. 6.8.

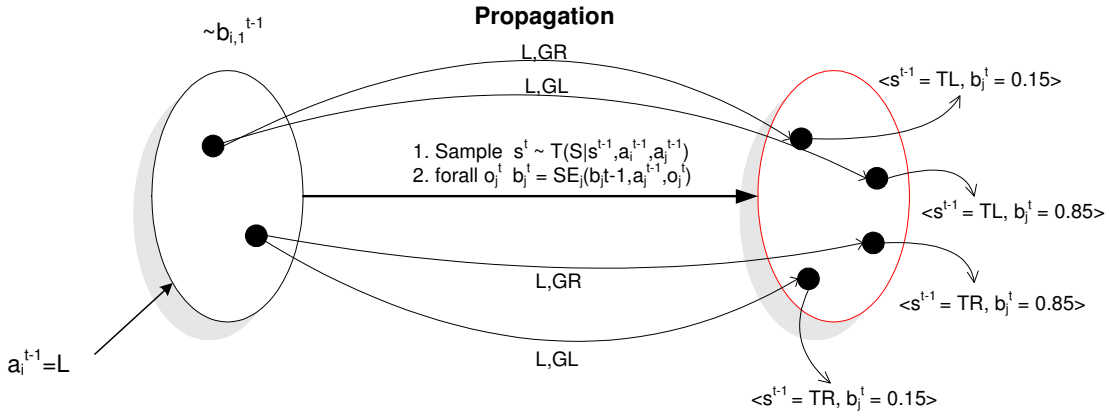

Figure 6.9: The propagation of the particles from time step $t-1$ to time step t. It involves sampling the next physical state and updating j's beliefs by anticipating its observations. Because j may recieve any one of two observations, there are 4 particles in the propagated sample set.

The second step of the propagation is to sample the next physical state for each particle using the transition function. Since both i and j listen, the location of the tiger remains unchanged. Additionally, we must update j's beliefs. We do this by anticipating what j might observe, and updating its belief exactly given its optimal action of listening. Since j could receive one of two possible observations – GL or GR – each particle "splits"

into two. This is shown using the thin arrows going from particles in the initial sample set to the particles in the propagated sample set, in Fig. 6.9. When j hears a GL, its updated belief is 0.85 (that the tiger is behind the left door), otherwise it is 0.15 when it hears a GR.

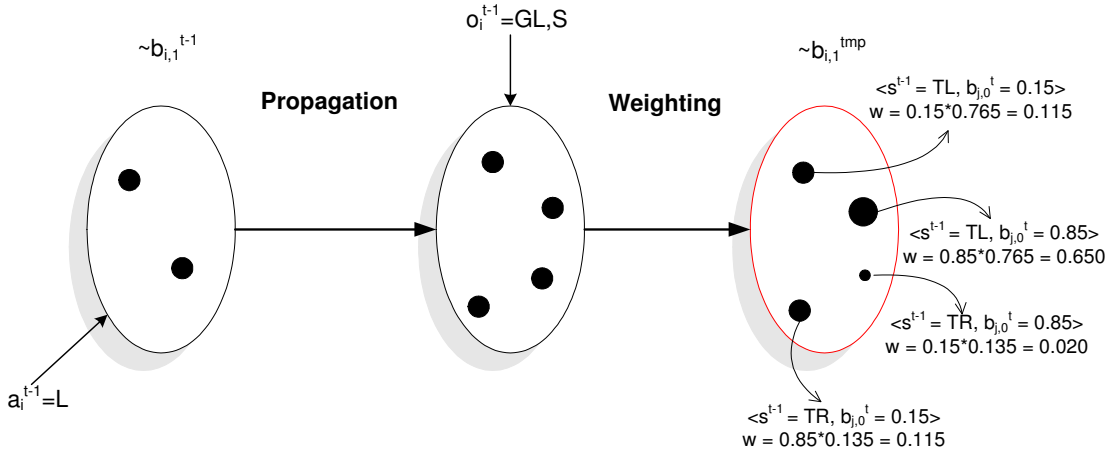

Figure 6.10: The weighting step is a two step process: Each particle is first weighted with the likelihood with which j receives its observations, followed by adjusting this weight using the probability of i making its observation of $\langle$ GL,S $\rangle$. Note that resulting weights as shown are not normalized.

As part of the *weighting*, we will first weight each particle with the probability of j receiving its observations. Thereafter, we will scale this weight with the probability of i observing a growl from the left and no creaks, $\langle$GL,S$\rangle$. To understand the weighting process, let's focus on a single particle. Weighting for the remaining particles is analogous.

We consider the particle on the top right in the sample set, $\widetilde{b}_{i,1}^{tmp}$, shown in Fig. 6.10. j's level 0 belief of 0.85 in this particle is due to j hearing a growl from the left. The probability of j making this observation as given by its observation function, when the tiger is on the left is 0.85. We will adjust this weight with the probability of i receiving $\langle$GL,S$\rangle$ when the tiger is on the left and both agents are listening. This probability as given by i's observation function is 0.765. The final weight attached to this particle is 0.65. Note that the weights as shown in Fig. 6.10 are not normalized. After normalization i's belief that the tiger is on the left is 0.85 (obtained by marginalizing over j's beliefs for particles that have s^t=TL), and 0.15 for tiger on the right.

The final step of the `I-PF` is an unbiased resampling of the particles using the weights as the distribution. To prevent an exponential growth in the number of particles [6], we resample N particles resulting in the sample

[6] After t propagation steps, there will be $N|\Omega_j|^t$ particles in the sample set.

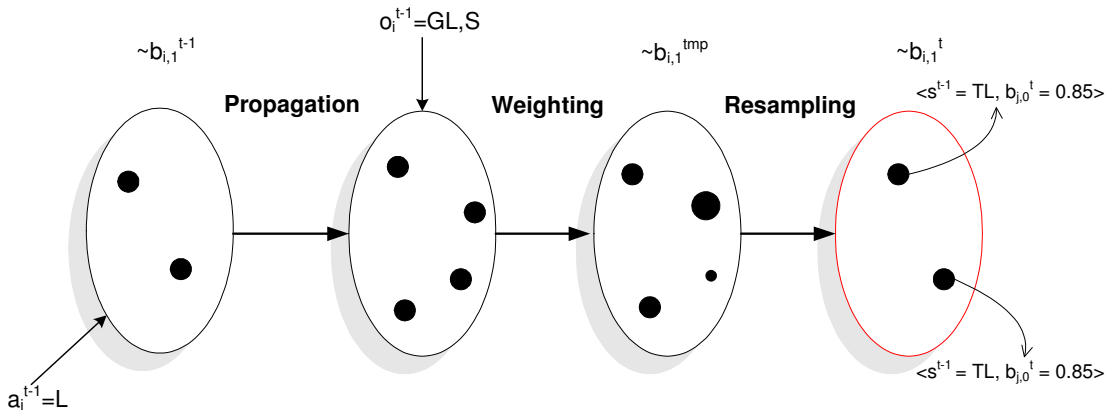

Figure 6.11: The final step is an unbiased resampling using the weights as the distribution.

set, $\widetilde{b}^t_{i,1}$, that is approximately representative of the exact updated belief.

When i's belief is multiply nested, the above mentioned example forms the bottom step of the recursive filtering process.

6.4.3 Performance of the I-PF

As part of our empirical investigation of the performance of the I-PF, we show, using a standard distance metric and visually, that our particle filter approximates the exact state estimation closely. For our analysis, we utilize the two-agent tiger problem, that has two physical states, as described in Section 4.5, and a two-agent version of the machine maintenance problem (MM) (Smallwood & Sondik, 1973), described in detail in Appendix B, that has three physical states. For both these problems, we make the simplifying assumption that models of the other agent are intentional and differ only in their beliefs. We use a numerical integration implementation [7] for the exact filter as the baseline for comparison.

The lineplots in Fig. 6.12 show that the quality of the approximation, as measured by KL-Divergence [8] increases as the number of particles increases, for both the problem domains. As we may expect, level 2 belief approximations require considerably more particles as compared to level 1 approximations, to achieve similar performance. Also, note that the performance of the I-PF remains consistent for both the two-state tiger and the three-state MM problem indicating that our implementation is not affected by the dimensionality

[7] We obtained the points for numerical integration by superimposing a high resolution regular grid on the interactive state space.

[8] When the level of nesting of the beliefs > 1, we compute the average of the KL-Divergences of the lower level beliefs, and add it to the upper level KL-Divergence, which is computed assuming that the lower level beliefs match exactly.

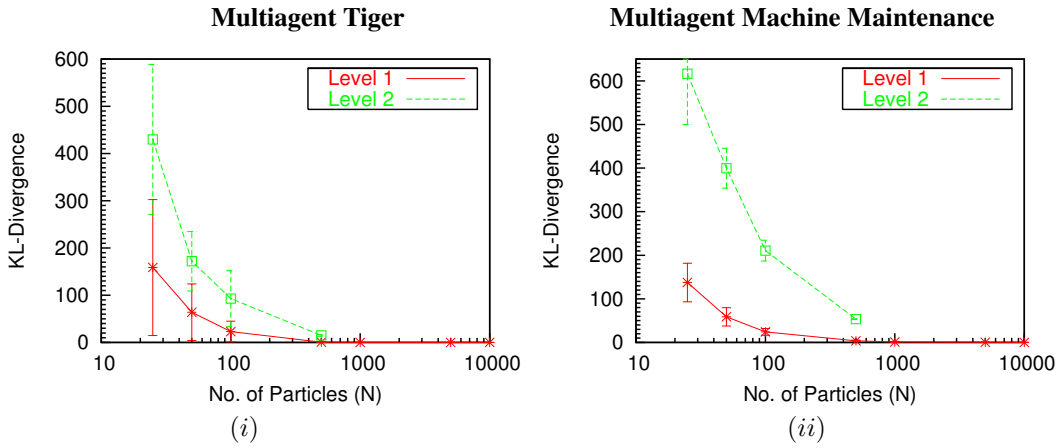

Figure 6.12: Performance of the I-PF as a function of the number of particles, on (i) multiagent tiger problem, (ii) multiagent machine maintenance game.

Belief	Game	Particle Filter N=500	Particle Filter N=1000	Numerical Integration
Level 1	Multiagent Tiger	0.148s ± 0.001s	0.332s ± 0.007s	21.80s ± 0.036s
	Multiagent MM	0.452s ± 0.009s	0.931s ± 0.0146s	1m 18.20s ± 0.45s
Level 2	Multiagent Tiger	2m 23.28s ± 1.1s	11m 41.30s ± 1.52s	51m 12.24s ± 5.66s
	Multiagent MM	1m 37.59s ± 0.17s	8m 27.29s ± 1.65s	151m 29.48s ± 1m 55.73s

Table 6.1: Comparison of the average running times of our numerical integration and particle filter implementations on same platform (*Pentium IV, 1.7GHz, 512MB RAM, Linux).*

of the underlying state space. Each data point in the lineplots is the average of 10 runs of our particle filter. In the case of the tiger problem, the posterior used for comparison is the one that is obtained after agent i listens and hears a growl from the left and no creaks. For the machine maintenance game, the posterior obtained after i manufactures and perceives no defect in the product, is used for comparison. We selected the belief states mentioned in Example 6.1 as the prior level 1 and level 2 beliefs ($d_{max} = 3$) of agent i when playing the tiger problem, and analogously for the machine maintenance game.

A comparison of the run times of the filter implemented using numerical integration and the interactive particle filter is shown in Fig. 6.1. Our particle filtering implementation significantly outperforms the numerical integration based implementation, while providing comparable performance quality. Additionally, the run times of the numerical integration implementation significantly increase when we move from the two-state

tiger problem to the three-state MM problem, in contrast to the particle filter. This is because numerical integration requires more points for larger state spaces to maintain comparable quality (curse of dimensionality). In order to assess the quality of the approximations after successive belief updates, we graphed the p.d.f.s produced by the particle filter and the exact filter. The p.d.f.s arising after each of three filtering steps on the level 1 belief of agent i in the tiger problem, are shown in Fig 6.13. Each approximate p.d.f. is the average of 10 runs of the particle filter which contained 5000 particles, and is estimated using a standard Gaussian kernel. The action/observation sequence followed was $\langle L, GL, S\rangle, \langle L, GL, S\rangle, \langle OR, GL, S\rangle$. As can be seen, our particle filter produces a good approximation of the true densities.

Level 1 Beliefs in the Multiagent Tiger Problem

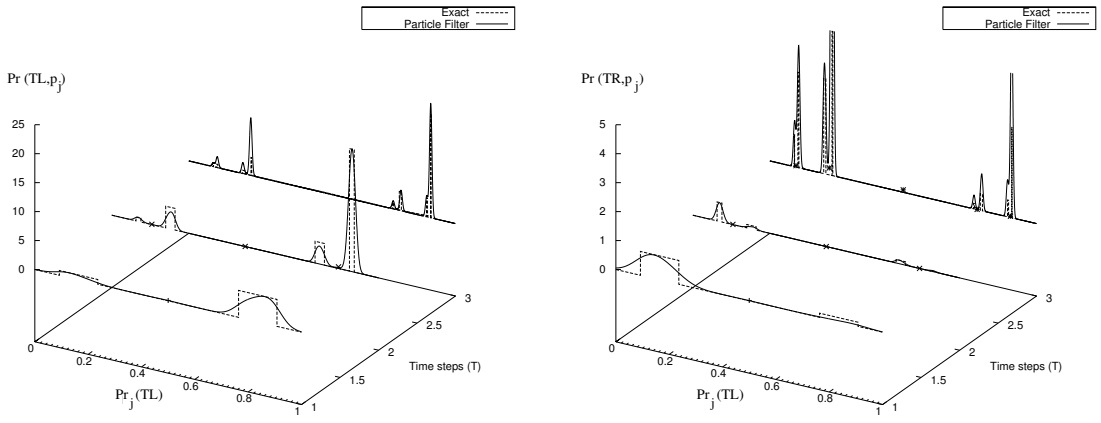

Figure 6.13: The exact and approximate p.d.f.s after successive filtering steps. The peaks of the approximate p.d.f.s align correctly with those of the exact p.d.f.s, and the areas under the approximate and exact p.d.f.s are approximately equal.

6.5 Value Iteration

Because the interactive particle filter represents each belief of agent i, $b_{i,l}$, using a set of N particles, $\widetilde{b}_{i,l}$, a value backup operator which operates on samples is needed. Let $\widetilde{H}$ denote the required backup operator, and $\widetilde{U}$ the approximate value function, then the backup operation, $\widetilde{U}^t = \widetilde{H}\widetilde{U}^{t-1}$, is:

Function APPROXPOLICY($\theta_k, l > 0$) **returns** $\Delta(A_k)$
1. $\widetilde{b}^0_{k,l} \leftarrow \{is^{(n)}_k, n = 1...N | is^{(n)}_k \sim b_{k,l} \in \theta_k\}$
   Reachability Analysis
2. reach(0) $\leftarrow \widetilde{b}^0_{k,l}$
3. **for** $t \leftarrow 1$ **to** T **do**
4. reach(t) $\leftarrow \phi$
5. **for all** $\widetilde{b}^{t-1}_{k,l} \in$ reach$(t-1)$, $a_k \in A_k$, $o_k \in \Omega_k$ **do**
6. reach(t) $\overset{\cup}{\leftarrow}$ I-PARTICLEFILTER$(\widetilde{b}^{t-1}_{k,l}, a_k, o_k, l)$
   Dynamic Programming
7. **for** $t \leftarrow T$ **downto** 0 **do**
8. **for all** $\widetilde{b}^t_{k,l} \in$ reach(t) **do**
9. $\widetilde{U}^{T-t,l}(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle) \leftarrow -\infty$, OPT($\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle$) $\leftarrow \phi$
10. **for all** $a_k \in A_k$ **do**
11. $\widetilde{U}^{T-t}_{a_k}(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle) \leftarrow 0$
12. **for all** $is^{(n),t}_k = \langle s^{(n),t}, m^{(n)}_{-k} \rangle \in \widetilde{b}^t_{k,l}$ **do**
13. **if** $m^{(n)}_{-k}$ is intentional **then**
14. $Pr(A_{-k}|m^{(n)}_{-k}) \leftarrow$ APPROXPOLICY$(\theta^{(n)}_{-k}, l-1)$
15. **for all** $a_{-k} \in A_{-k}$ **do**
16. $\widetilde{U}^{T-t}_{a_k}(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle) \overset{+}{\leftarrow} \frac{1}{N} R(s^{(n),t}, a_k, a_{-k}) Pr(a_{-k}|m^{(n)}_{-k})$
17. **if** $(t < T)$ **then**
18. **for all** $o_k \in \Omega_k$ **do**
19. sum $\leftarrow 0$, $\widetilde{b}^{t+1}_{k,l} \leftarrow$ reach$(t+1)[|\Omega_k| a_k + o_k])$
20. **for all** $is^{(n),t}_k = \langle s^{(n),t}, m^{(n)}_{-k} \rangle \in \widetilde{b}_{k,l}$ **do**
21. **if** $m^{(n)}_{-k}$ is intentional **then**
22. $Pr(A_{-k}|m^{(n)}_{-k}) \leftarrow$ APPROXPOLICY$(\theta^{(n)}_{-k}, l-1)$
23. **for all** $a_{-k} \in A_{-k}$, $s^{t+1} \in S_k$ **do**
24. sum $\overset{+}{\leftarrow} O_k(o_k|s^{t+1}, a_k, a_{-k}) Pr(is^{(n),t+1}|is^{(n),t}, a_k, a_{-k}) Pr(a_{-k}|m^{(n)}_{-k})$
25. $\widetilde{U}^{T-t,l}_{a_k}(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle) \overset{+}{\leftarrow} \gamma \times \frac{1}{N} \times$ sum $\times \widetilde{U}^{T-t-1}(\widetilde{b}^t_{k,l})$
26. **if** $\widetilde{U}^{T-t}_{a_k}(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle) \geq \widetilde{U}^{T-t}(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle))$ **then**
27. **if** $(\widetilde{U}^{T-t}_{a_k}(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle) > \widetilde{U}^{T-t}(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle)$ **then**
28. $\widetilde{U}^{T-t}(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle) \leftarrow \widetilde{U}^{T-t}_{a_k}(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle)$
29. OPT$(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle) \leftarrow \phi$
30. OPT$(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle) \overset{\cup}{\leftarrow} a_k$
31. **for all** $a_k \in A_k$ **do**
32. **if** $(a_k \in$ OPT$(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle)$ **then**
33. $Pr(a_k|\theta_k) \leftarrow \frac{1}{|\text{OPT}(\langle \widetilde{b}^t_{k,l}, \widehat{\theta}_k \rangle)|}$
34. **else**
35. $Pr(a_k|\theta_k) \leftarrow 0$
36. **return** $Pr(A_k|\theta_k)$

end function

Figure 6.14: Algorithm for computing an approximately optimal finite horizon policy tree given a model containing an initial sampled belief. When $l = 0$, the exact POMDP policy tree is computed.

$$\widetilde{U}^t(\langle\widetilde{b}_{i,l},\widehat{\theta}_i\rangle) = \max_{a_i\in A_i}\left\{\frac{1}{N}\sum_{is^{(n)}\in\widetilde{b}_{i,l}} ER_i(is^{(n)},a_i) + \gamma\sum_{o_i\in\Omega_i} Pr(o_i|a_i,\widetilde{b}_{i,l})\widetilde{U}^{t-1}(\langle\text{I-PF}(\widetilde{b}_{i,l},a_i,o_i),\widehat{\theta}_i\rangle)\right\} \tag{6.1}$$

where $ER_i(is^{(n)},a_i) = \sum_{a_j} R_i(s^{(n)},a_i,a_j)Pr(a_j|m_j^{(n)})$, and I-PF denotes the belief update implemented using the interactive particle filter. The set of optimal actions at a given approximate belief, OPT($\langle\widetilde{b}_{i,l},\widehat{\theta}_j\rangle$), is then calculated by returning the actions that have the maximum value:

$$OPT(\langle\widetilde{b}_{i,l},\widehat{\theta}_i\rangle) = \underset{a_i\in A_i}{argmax}\left\{\begin{array}{l}\frac{1}{N}\sum\limits_{is^{(n)}\in\widetilde{b}_{i,l}} ER_i(is^{(n)},a_i) + \gamma\sum\limits_{o_i\in\Omega_i} Pr(o_i|a_i,\widetilde{b}_{i,l}) \\ \times\widetilde{U}^{t-1,l}(\langle\text{I-PF}(\widetilde{b}_{i,l},a_i,o_i),\widehat{\theta}_i\rangle)\end{array}\right\} \tag{6.2}$$

Equations 6.1 and 6.2 are analogous to the Eqs. 4.3 and 4.4 respectively, with exact integration replaced by Monte Carlo integration, and the exact belief update replaced with the interactive particle filter. Note that $\widetilde{H}\rightarrow H$ as $N\rightarrow\infty$. The algorithm for computing an approximately optimal finite horizon policy tree using value iteration when $l>0$ is given in Fig. 6.14. When $l=0$, the algorithm reduces to the POMDP policy tree computation which is carried out exactly.[9] The algorithm consists of the usual two steps: Compute the look ahead reachability tree as part of the reachability analysis (see Section 17.5 of Russell & Norvig, 2003); and perform value backup on the reachability tree.

6.5.1 Convergence and error bounds

The use of randomizing techniques such as particle filters means that value iteration does not necessarily converge. This is because, unlike the exact belief update, posteriors generated by the particle filter with finitely many particles are not guaranteed to be identical for identical input. The non-determinism of the approximate belief update rules out isotonicity and contraction for $\widetilde{H}$ as $N\rightarrow\infty$. [10]

Our inability to guarantee convergence implies that we must approximate an infinite horizon policy with the approximately optimal finite horizon policy tree. Let U^* be the value of the optimal infinite horizon policy, $\widetilde{U}^t$ be the value of the approximate and U^t be the value of the optimal t-horizon policy tree, then the error

[9] For large problems, exact POMDP solutions may be replaced with approximate ones. But in doing so, our error bounds will no longer be applicable.

[10] One may turn particle filters into deterministic belief update operators (de-randomization) by generating several posteriors from the same input. A representative posterior is then formed by taking a convex combination of the different posteriors.

bound (using the supremum norm $||\cdot||$) is, $||U^* - \widetilde{U}^t|| = ||U^* - U^t + U^t - \widetilde{U}^t|| \leq ||U^* - U^t|| + ||U^t - \widetilde{U}^t||$. Note that the first term is bounded by $\gamma^t||U^* - U^0||$. The bound for the second term is calculated below:

$$\begin{aligned}
\mathcal{E}^t &= ||\widetilde{U}^t - U^t|| \\
&= ||\widetilde{H}\widetilde{U}^{t-1} - HU^{t-1}|| \\
&= ||\widetilde{H}\widetilde{U}^{t-1} - H\widetilde{U}^{t-1} + H\widetilde{U}^{t-1} - HU^{t-1}|| \qquad \text{(add zero)} \\
&\leq ||\widetilde{H}\widetilde{U}^{t-1} - H\widetilde{U}^{t-1}|| + ||H\widetilde{U}^{t-1} - HU^{t-1}|| \quad \text{(triangle inequality)} \\
&\leq ||\widetilde{H}\widetilde{U}^{t-1} - H\widetilde{U}^{t-1}|| + \gamma||\widetilde{U}^{t-1} - U^{t-1}|| \qquad \text{(contracting } H\text{)} \\
&\leq ||\widetilde{H}\widetilde{U}^{t-1} - H\widetilde{U}^{t-1}|| + \gamma\mathcal{E}^{t-1}
\end{aligned}$$

We will turn our attention to $||\widetilde{H}\widetilde{U}^{t-1} - H\widetilde{U}^{t-1}||$. In the analysis that follows we focus on level 1 beliefs. Let $\dot{U}^t = H\widetilde{U}^{t-1}$, $\widetilde{U}^t = \widetilde{H}\widetilde{U}^{t-1}$, and $b_{i,1}$ be the singly nested belief where the worst error is made: $b_{i,1} = \underset{b_{i,1} \in B_{i,1}}{\text{argmax}} |\dot{U}^t - \widetilde{U}^t|$. Let $\widetilde{\alpha}$ be the policy tree (alpha vector) that is optimal at $\widetilde{b}_{i,1}$ (the sampled estimate of $b_{i,1}$), and $\dot{\alpha}$ be the policy tree that is optimal at $b_{i,1}$. We will use Chernoff-Hoeffding (C-H) upper bounds (Theorem A.1.4, pg 265 in Alon & Spencer, 2000) [11], a well-known tool for analyzing randomized algorithms, to derive a confidence threshold $1 - \delta$ at which the observed estimate, $\widetilde{U}^t_{\widetilde{\alpha}}$, is within 2ϵ of the true estimate $\dot{U}^t_{\dot{\alpha}}$ (= $E[\dot{\alpha}]$):

$$Pr(\widetilde{U}^t_{\widetilde{\alpha}} > \dot{U}^t_{\dot{\alpha}} + \epsilon) \leq e^{-2N\epsilon^2/(\widetilde{\alpha}_{max} - \widetilde{\alpha}_{min})^2}$$
$$Pr(\widetilde{U}^t_{\widetilde{\alpha}} < \dot{U}^t_{\dot{\alpha}} - \epsilon) \leq e^{-2N\epsilon^2/(\widetilde{\alpha}_{max} - \widetilde{\alpha}_{min})^2}$$

For a confidence probability of atleast $1 - \delta$, the error bound is:

$$\epsilon = \sqrt{\frac{(\widetilde{\alpha}_{max} - \widetilde{\alpha}_{min})^2 ln(2/\delta)}{2N}} \tag{6.3}$$

where $\widetilde{\alpha}_{max} - \widetilde{\alpha}_{min}$ may be loosely upper bounded as $\frac{R_{max} - R_{min}}{1-\gamma}$. Note that Eq. 6.3 can also be used to derive the number of particles, N, for some given δ and ϵ. To get the desired bound, we note that with probability $1 - \delta$ our error bound is 2ϵ and with probability δ the worst possible sub-optimal behavior may result: $||\widetilde{H}\widetilde{U}^{t-1} - H\widetilde{U}^{t-1}|| \leq (1 - \delta)2\epsilon + \delta\frac{R_{max} - R_{min}}{1-\gamma}$. The final error bound now obtains:

[11] At horizon t, samples in $\widetilde{b}_{i,1}$ are i.i.d. However, at horizons $< t$, the samples are generated by the `I-PF` and exhibit limited statistical independence, but independent research (Schmidt, Spiegel, & Srinivasan, 1995) reveals that C-H bounds still apply.

$$\begin{aligned}\mathcal{E}^t &\leq (1-\delta)2\epsilon + \delta\tfrac{R_{max}-R_{min}}{1-\gamma} + \gamma\mathcal{E}^{t-1} \qquad \text{(geometric series)}\\ &= (1-\delta)\tfrac{2\epsilon(1-\gamma^t)}{1-\gamma} + \delta\tfrac{(R_{max}-R_{min})(1-\gamma^t)}{(1-\gamma)^2}\end{aligned}$$

where ϵ is as defined in Eq. 6.3.

Theorem 6.1 (Error Bound). *For a singly nested t-horizon $\texttt{I-POMDP}_{i,1}$, the error introduced by our approximation technique is upper bounded and is given by:*

$$||\widetilde{U}^t - U^t|| \leq (1-\delta)\frac{2\epsilon(1-\gamma^t)}{1-\gamma} + \delta\frac{(R_{max}-R_{min})(1-\gamma^t)}{(1-\gamma)^2}$$

where ϵ is as defined in Eq. 6.3.

At levels of belief nesting greater than one, j's beliefs are also approximately represented using samples. Hence the approximation error is not only due to the sampling, but also due to the possible incorrect prediction of j's actions based on its approximate beliefs. We are currently investigating if it is possible to derive bounds that are useful, that is, tighter than the usual difference between the best and worst possible behavior, for this case.

6.5.2 Computational savings

Since the complexity of solving I-POMDPs is dominated by the complexity of solving the models of other agents we look at the reduction of the number of agent models that must be solved. In an M+1-agent setting with the number of particles bounded by N, each particle in $\widetilde{b}_{k,l}^{t-1}$ of level l may contain M models all of level $l-1$. Solution of each of these level $l-1$ models requires solution of the lower level models recursively. The upper bound on the number of models that are solved is $O((MN)^{l-1})$. Given that there are M level $l-1$ models in a particle, and N such possibly distinct particles, we need to solve $O((MN)^l)$ models. Note that each of these (level 0) models is a POMDP with an initial belief, and is solved exactly. Our upper bound on the number of models is polynomial in M. This can be contrasted with $O((M|\Theta_*|^M)^l)$ models that need to be solved in the exact case, which is exponential in M. Here, amongst the spaces of models of all agents, Θ_* is the largest space. Typically, $N \ll |\Theta_*|^M$, resulting in a substantial reduction in computation.

6.6 Empirical Performance

The goal of our experimental analysis is to demonstrate empirically, (a) the reduction in error with increasing sample complexity, and (b) savings in computation time and space when our approximation technique is used. We use the multiagent tiger problem introduced previously, and a multiagent version of the machine maintenance (MM) problem (Smallwood & Sondik, 1973) (see Appendix B) as test problems. Because the problems are rather simplistic (Tiger: $|S|$=2, $|A_i|$=$|A_j|$=3, $|\Omega_i|$=$|\Omega_j|$=6; MM: $|S|$=3, $|A_i|$=$|A_j|$=4, $|\Omega_i|$=$|\Omega_j|$=2), our results should be considered preliminary.

To demonstrate the reduction in error, we construct performance profiles showing an increase in performance as more computational resources – in this case particles – are allocated to the approximation algorithm. In Figs. 6.15(a) and (c) we show the performance profile curves when agent i's prior belief is the level 1 belief described previously in Example 6.1, and suitably modified for the MM problem. As expected the average rewards for both, horizon 2 and 3 approach the exact expected reward as the number of particles increases. We show the analogous plots for the level 2 belief in Figs. 6.15(b) and (d). In each of these cases the average of the rewards accumulated by i over a 2 and 3 horizon policy tree (computed using the algorithm in Fig. 6.14) while playing against agent j were plotted. To compensate for the randomness in sampling, we generated i's policy tree 10 times independent of each other, and performed 100 runs each time. Within each run, the location of the tiger and j's prior beliefs were sampled according to i's prior belief. j's policy was then computed using the algorithm in Fig 6.14.

Problem	**Error**	$t=2$		$t=3$	
		N=10^2	**N**=10^3	**N**=10^2	**N**=10^3
Multiagent tiger	Obs.	5.61	0	4.39	2.76
	$\mathcal{E}^t$	108.38	48.56	207.78	86.09
Multiagent MM	Obs.	0.28	0.23	0.46	0.40
	$\mathcal{E}^t$	4.58	2.05	8.79	3.64

Table 6.2: Comparison of the worst case observed errors and the theoretical error bounds.

In Table 6.2, we compare the worst observed error – difference between the exact expected reward and the observed expected reward – with the theoretical worst case error bound (δ=0.1,γ=0.9) from Section 6.5.1, for horizons 2 and 3. The difference between the best and the worst possible behavior for the tiger problem for $t = 2$ is 209.00, and for $t = 3$ is 298.1. For the multiagent MM problem, the differences are 8.84 and 12.61, respectively. The theoretical error bounds appear loose due to the worst-case nature of our analysis but (expectedly) are tighter than the worst bounds, and reduce as the number of particles increases.

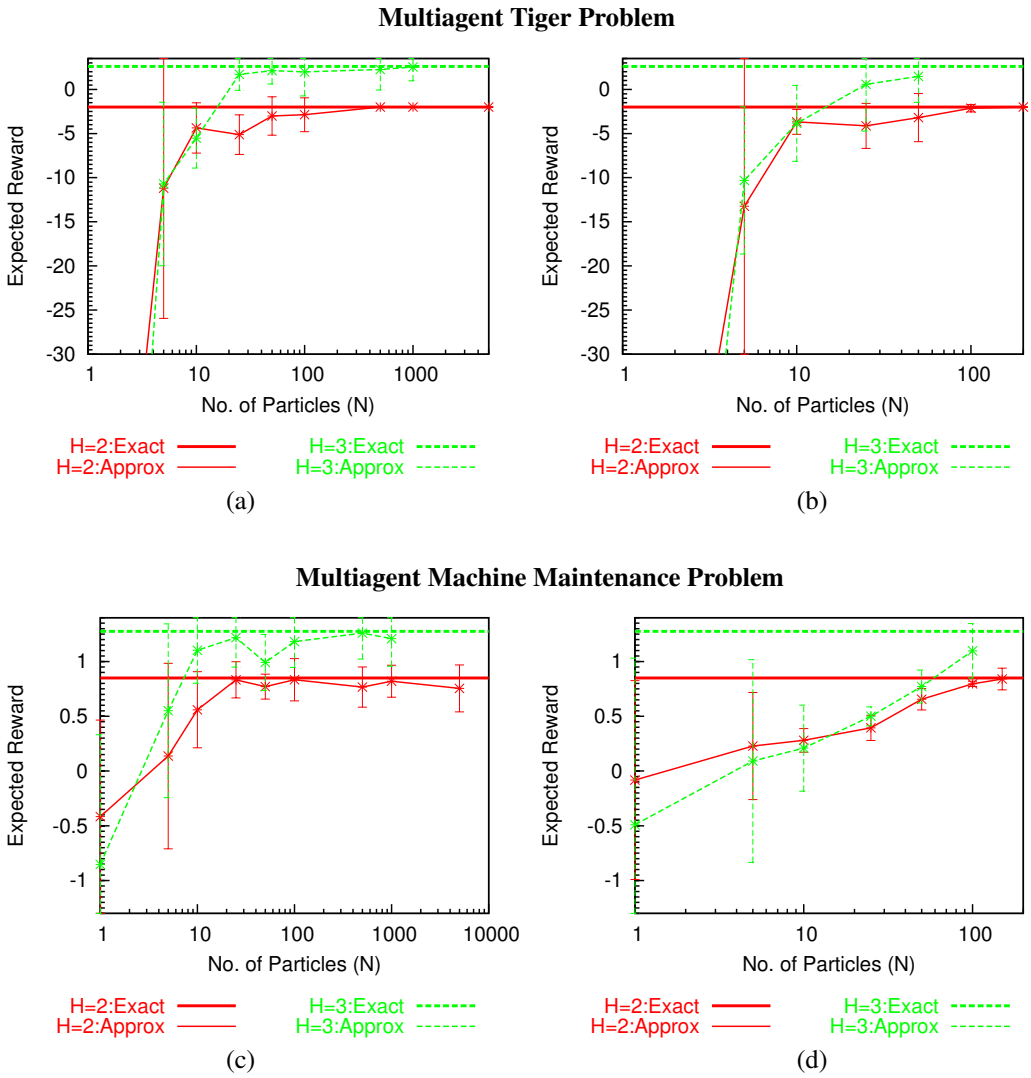

Figure 6.15: Performance profiles: The multiagent tiger problem using the (a) level 1, and (b) level 2 belief as the prior for agent i. The multiagent MM using the (c) level 1, and (d) level 2 belief as i's prior.

Table 6.3 compares the average run times of our sample-based approach (SB) with the exact approach, for computing policy trees of different horizons starting from the level 1 belief. The values of the policy trees generated by the two approaches were similar. The run times demonstrate the dominant impact of the curse of dimensionality on the exact method as shown by the higher run times for the MM problem in comparison to the tiger problem. Our sample based implementation is immune to this curse, but is affected by the curse of history, as illustrated by the higher run times for the tiger problem (branching factor = 18) compared to the MM problem (branching factor = 8).

Problem	Method	Run times			
		$t = 2$	$t = 3$	$t = 4$	$t = 5$
Multiagent tiger	Exact	37.84s $\pm$ 0.6s	11m 22.25s $\pm$ 1.34s	*	*
	SB	1.44s $\pm$ 0.05s	1m 44.29s $\pm$ 0.6s	19m 16.88s $\pm$ 17.5s	*
Multiagent MM	Exact	5m 26.57s $\pm$ 0.07s	20m 45.69s $\pm$ 0.29s	*	*
	SB	5.75s $\pm$ 0.01s	34.52s $\pm$ 0.01s	3m 24.9s $\pm$ 0.04s	17m 58.39s $\pm$ 0.57s

Table 6.3: Run times on a Pentium IV 2.0 GHz, 2.0GB RAM and Linux. * = program ran out of memory.

6.7 Sampling the Look Ahead Reachability Tree

In order to address the curse of dimensionality, we took recourse to a sampling based method – the interactive particle filter – that is typically immune to the dimensions of the underlying state space. Though we successfully addressed the problem of dimensionality, we were unable to generate solutions for large horizons. The main reason for this is the exponential growth of the look ahead reachability tree with increasing horizons; we referred to this as the curse of history. At some time step t, there could be $(|A_i||\Omega_i|)^{t-1}$ reachable beliefs states of the agent i. For example, in the multiagent tiger problem, at the second time step there could be 18 possible belief states, 324 of them at the third time step, and more than 0.1 million at the fifth time step.

To mitigate the curse of history, we reduce the branching factor of the look ahead reachability tree by sampling from the possible observations that the agent may receive. While this approach does not completely address the curse of history, it beats back the impact of this curse substantially. On performing an action, the agent propagates its belief and uses the propagated belief to arrive at a distribution over its observations, which is then used for sampling. In other words, $o_i^t \sim Pr(\Omega_i|a_i^{t-1}, \widetilde{b}_{i,l}^{t-1})$. Of course, in the process we may build a partial reachability tree, and therefore obtain a partial policy tree. For observations that occur which were not sampled (the probability of such observations will be low), we pick a policy tree at random, out of the prescribed policy trees for sampled observations. Let us label this approach as *reachability tree sampling* (`RTS`). `RTS` shares its conceptual underpinnings with the exploration models of PBVI (Pineau et al., 2003b), but differs in that our method is applicable to online policy tree generation for `I-POMDPs`, compared to PBVI's use in offline policy generation for POMDPs.

6.7.1 Computational Savings

Let us consider the computational savings that result from sampling the look ahead reachability tree. If we are sampling N_{Ω_i} observations within the reachability tree, then at some time step t, we will obtain $(|A_i||N_{\Omega_i}|)^{t-1}$ possible belief states, assuming the worst case occurs and we end up sampling $N_{\Omega_i} < |\Omega_i|$ distinct observations. Typically, as our experiments demonstrate, the number of distinct sampled observations is less than $|\Omega_i|$, resulting in significant computational savings.

As an illustration of the computational savings we compare the run times of computing the policy tree for the multiagent tiger problem. We compare value iteration in which the reachability tree is sampled (SB-RTS) with complete value iteration and no reachability tree sampling (SB-No-RTS), the algorithm for which is given in Fig. 6.14. For SB-RTS, we sampled three times from the observation distribution upto the sixth horizon and two times thereafter. For both the algorithms, we used a similar number of particles. Not only does the SB-RTS compute the policy faster, we were able to compute it upto eight time horizons. When compared with the SB-No-RTS, our results demonstrate that the approach of sampling the reachability tree yields significant computational savings.

Method	**Run times**						
	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$
SB-No-RTS	1.44s $\pm$ 0.05s	1m 44.29s $\pm$ 0.6s	19m 16.88s $\pm$ 17.5s	*	*	*	*
SB-RTS	0.10s $\pm$ 0.003s	0.923s $\pm$ 0.003s	7.307s $\pm$ 0.71s	41.73s $\pm$ 3.49s	2m 12.64s $\pm$ 14.32s	8m 15.55s $\pm$ 56.11s	17m 9.83s $\pm$ 0.95s

Table 6.4: Run times on a Pentium IV 2.0 GHz, 2.0GB RAM and Linux. * = program ran out of memory.

6.7.2 Empirical Performance

We present the performance profiles in Fig. 6.16 for the multiagent tiger problem when partial look ahead reachability trees are built by sampling the observations. We plot the average reward accumulated by i over 10 independent trials consisting of 100 runs each, as the number of the observation samples, N_{Ω_i} are gradually increased. Within each run, the location of the tiger and j's prior beliefs were sampled according to i's prior level 1 belief. Since we have combined `RTS` with the `I-PF`, in addition to varying N_{Ω_i}, we also vary the number of particles, N_p, employed to approximate the beliefs. As expected of performance profiles, the expected reward initially increases sharply, before flattening out as N_{Ω_i} becomes large and the

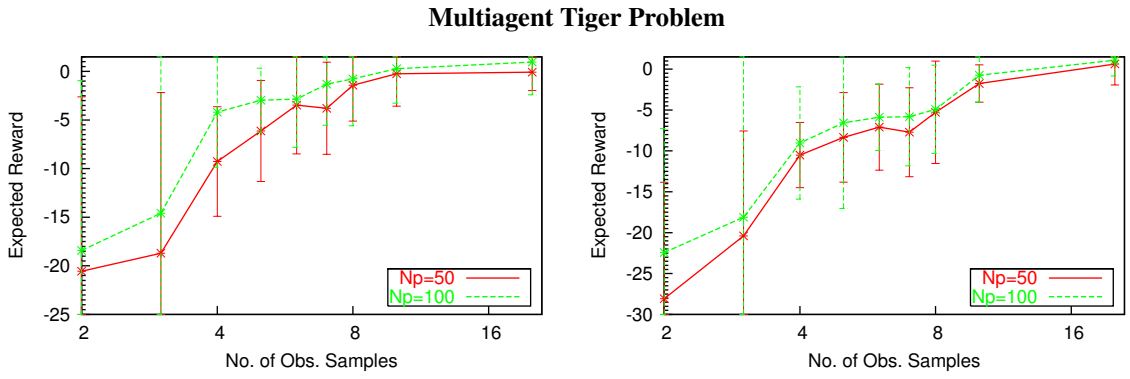

Figure 6.16: Performance profiles for the multiagent tiger problem for (a) horizon 3, and for (b) horizon 4 when the look ahead tree is built by sampling observations.

sampled observation distribution reaches the true one. Reflecting intuition, the plots for $N_p = 100$ exhibit better expected rewards than those for $N_p = 50$. We also obtained the average reward over a similar number of trials when a random policy (null hypothesis) is used for i. For horizon 3, the random policy gathered an average reward of -84.785 ($\pm$ 37.9847), and -108.5 ($\pm$ 41.56) for horizon 4. Even for a small number of observation samples, RTS does significantly better than the random policy thereby demonstrating the usefulness of partial tree expansion.

We observed that the empirical expected reward is close to the exact expected reward when only a few distinct observations were sampled while building the reachability tree. This observation when combined with the tremendous computational savings demonstrated in Section 6.7.1 indicate that our approximation approach is viable. Additionally, by varying the parameters N_p and N_{Ω_i}, we can flexibly control the effects of both the curses on the solutions. An interesting line of future work is to investigate the interplay of these parameters.

6.8 Summary

We described a randomized method for obtaining online approximate solutions to I-POMDPs based on a novel extension of particle filtering to multiagent settings. The extension is not straightforward because we are confronted with an interactive belief hierarchy when dealing with multiagent settings. We proposed the interactive particle filter which descends the levels of interactive belief hierarchies and samples and propagates beliefs at each level. The interactive particle filter is able to deal with the belief space dimensionality,

but it does not address the policy space complexity. Though our technique is not guaranteed to converge asymptotically, we established useful error bounds for level 1 nested I-POMDPs. We provided performance profiles for the multiagent tiger problem and the machine maintenance problem. They show that the filter saves on computation over the space of models but it does not scale (usefully) to large values of time horizons and needs to be combined with methods that deal with the curse of history. In order to reduce the impact of the curse of history, we proposed sampling observations while constructing the look ahead reachability tree during the reachability analysis phase of value iteration. This effectively reduces the branching factor of the tree and allows computation of solutions for larger horizons.

6.9 Contributions

Bounded approximation technique: We proposed a randomized method to compute online approximately optimal plans for I-POMDPs. Our method addresses the curse of dimensionality afflicting I-POMDPs, by utilizing a Monte Carlo sampling based approach that is typically immune to the cardinality of the underlying state space. We also established useful bounds on the approximation error for singly nested I-POMDPs.

Interactive particle filter: We introduced the interactive particle filter that extends the basic bootstrap filter to the multiagent setting. Mirroring the hierarchical nature of the interactive beliefs, the interactive particle filter descends through the levels of nesting, and samples and propagates beliefs at each level. The interactive particle filter reduces to the traditional particle filter when there is just one agent. Our empirical results demonstrate that the interactive particle filter closely approximates the exact belief update.

I-PF+RTS: For increasing horizons, the exponential growth of the reachability tree becomes the main bottleneck. We combined the interactive particle filter with the sampling of observations while building the reachability tree. The resulting technique, I-PF+RTS, addresses the curse of dimensionality and beats back curse of history. As a result, we are able to compute solutions for larger horizons.

Anytime algorithm: Our approximation technique is anytime: the quality of the approximation increases as more computational resources – particles and observation samples – are allocated to the method. In support of this, we generated performance profiles for two simple test problems.

6.10 Future Work

While we bounded the approximation error for singly nested I-POMDPs, the same approach cannot be used to bound the error for I-POMDPs nested to levels greater than one. Therefore, another research issue is to investigate whether the approximation error for multiply nested I-POMDPs can be usefully bounded. A method to further scale the approximation technique is to pick a subset of actions in addition to sampling the observations while building the reachability tree. This further dampens the exponential growth of the reachability tree with increasing horizons, and permits solutions for large horizons. However, this approach must be used cautiously – we do not want to leave out critical actions from the policy. Finally, specific approaches to speeding up the computation remain to be explored. For example, can we assign monotonically decreasing number of particles to represent beliefs nested at deeper levels exploiting the common sense notion that beliefs nested at deeper levels are less likely to influence the optimal policy?

Chapter 7

SUBJECTIVE EQUILIBRIA IN I-POMDPS: THEORY AND COMPUTATIONAL LIMITATIONS

We theoretically analyze the interactions taking place between agents participating in the infinite horizon partially observable stochastic game (POSG) as formalized within the I-POMDP framework. As we mentioned before, I-POMDPs represent and solve a POSG from the perspective of an agent playing the game. We consider the setting in which an agent may be unaware of other agents' behavioral strategies, it is uncertain about their observations, and it may be unable to perfectly observe other agents' actions. In accordance with Bayesian decision theory, the agent maintains and updates its belief about the physical state as well as the strategies of the other agents, and its decisions are best responses to its beliefs.

Under the assumption of compatibility of agents' prior beliefs about future observations with the true distribution induced by the actual strategies of all agents, we show that for agents modeled within the I-POMDP framework, the following properties hold: (i) the agents' beliefs about the future observation paths of the game coincide in the limit with the true distribution over the future, and (ii) the agents' beliefs about the opponents' strategies do not change in the limit. Strategies that are best responses to beliefs with these properties are said to be in *subjective equilibrium*, which is stable with respect to learning and optimization. Strategies in subjective equilibrium need not necessarily also be in Nash equilibrium, though the converse is always true.

Our results in this chapter generalize prior results. Specifically, we theoretically show the asymptotic

existence of subjective equilibrium in a general and realistic multiagent setting; there is state and action outcome uncertainty and imperfect observations of others' actions. We note that the I-POMDP belief update plays a key role in making the generalization possible. Further, we address the open research problem posed in (Kalai & Lehrer, 1993a) regarding the existence of subjective equilibrium in POSGs. We also draw a parallel with works in multiagent learning (Hu & Wellman, 1998; Bowling & Veloso, 2002)(also see Section 3.3 of Chapter 3) that show convergence of play to Nash equilibrium. However, our results differ in that we assume that the state and others' actions are partially observable, and the plan is computed offline using a given model of the environment. Finally, we comment on the difficulties in achieving subjective equilibria in I-POMDPs when a computational perspective is adopted. The difficulties arise because of obstacles in satisfying the truth compatibility condition, in practice. This potentially negative result – the possible inability of I-POMDPs to reach the subjective equilibrium in practice – also calls into question the role of equilibrium in multiagent planning when a decision-theoretic viewpoint is adopted.

The rest of this chapter is structured in the following manner. In the next section, we briefly review the previous work related to ours. In Section 7.2, we review the I-POMDP belief update focusing on the setting where general models are ascribed to the other agent. In Section 7.3, we introduce the concept of a subjective equilibrium and theoretically prove that the strategy profile of agents playing a POSG within the I-POMDP framework, in the limit, is in subjective equilibrium. In Section 7.4, we remark on the computational infeasibility of arriving at this equilibrium. We then summarize this chapter in Section 7.5, and give the contributions of our work in Section 7.6. We conclude this chapter with open research directions in Section 7.7.

7.1 Related Work

In prior work, Kalai and Lehrer (1993a, 1993b) (also see Section 3.2.2 of Chapter 3) have shown that the strategies of agents engaged in infinitely repeated games with discounted payoffs, who are unaware of others' strategies, and under the assumptions of perfect observability of others' actions (perfect monitoring) and truth compatibility of prior beliefs will converge to a subjective equilibrium. Hahn (Hahn, 1973) introduced the concept of a *conjectural equilibrium* in economies where the signals generated by the economy do not cause changes in the agents' theories, nor do they induce changes in the agents' policies. Fudenberg and Levine (1993) consider a general model of finitely repeated extensive form games wherein strategies of opponents may be correlated (unlike Kalai & Lehrer, 1993a, where strategies are assumed independent),

and show that behavior of agents that maintain beliefs and optimize according to their beliefs, converges to a *self-confirming equilibrium*. There is a strong link between the subjective equilibrium and its objective counterpart – the Nash equilibrium. Specifically, under the assumption of perfect monitoring, both (Kalai & Lehrer, 1993a) and (Fudenberg & Levine, 1993) show that the strategy profile in subjective and self-confirming equilibrium induce a distribution over the future action paths that coincides with the distribution induced by a set of strategies in Nash equilibrium. In other words, the continuation path of the game would asymptotically resemble that of a Nash equilibrium. Of course, this does not imply that strategies in subjective equilibrium are also in Nash equilibrium; however, the converse is always true. Work of a similar vein is reported in (Jordan, 1995). It assumes agents have a common prior over the possible types of agents engaged in a repeated game, and shows that the sequence of Bayesian-Nash equilibrium beliefs of agents converges to a Nash equilibrium.

7.2 Review: Bayesian Belief Update in I-POMDPs

In order to act rationally, agents within the I-POMDP framework continually update their beliefs over the physical states and other agents' models on performing an action and receiving an observation. We described the belief update process in detail in Section 4.3.2, but gave the explicit formulations for intentional models. Here, we give the equations for the general model, and utilize them later for our results.

Recall that $is \in IS_i = S \times M_j$, where $m_j \in M_j$ and $m_j = \langle h_j, O_j, f_j \rangle$. f_j is agent j's function [1], $f_j : H_j \rightarrow \Delta(A_j)$, assumed computable [2], which maps possible histories of j's observations to distributions over its actions. h_j is an element of H_j, and O_j is a function specifying the way the environment is supplying the agent with its input. For convenience, we may write model m_j as $m_j = \langle h_j, \widehat{m}_j \rangle$, where $\widehat{m}_j$ consists of f_j and O_j. We assume without loss of generality that the models of the other agent are not directly observable nor manipulable. We decompose the belief update process into two steps:

[1] Note that an agent function is similar to a behavioral strategy in game theory parlance.
[2] We assume computability in the Turing machine sense: f_j is a total recursive function.

- *Prediction:* When an agent, say i, with a previous belief, b_i^{t-1}, performs a control action a_i^{t-1} and if the other agent performs its action a_j^{t-1}, the predicted belief state is:

$$Pr(is^t|a_i^{t-1}, a_j^{t-1}, b_i^{t-1}) = \sum_{IS^{t-1}:\widehat{m}_j^{t-1}=\widehat{m}_j^t} b_i^{t-1}(is) Pr(a_j^{t-1}|m_j^{t-1}) T(s^{t-1}, a_i^{t-1}, a_j^{t-1}, s^t) \\ \times \sum_{o_j^t} O_j(s^t, a_i^{t-1}, a_j^{t-1}, o_j^t) \delta_K(\text{APPEND}(h_j^{t-1}, o_j^t) - h_j^t)$$

  where δ_K is the Kronecker delta function, and APPEND$(\cdot,\cdot)$ returns a string in which the second argument is appended to the first.

- *Correction:* When agent i perceives an observation, o_i^t, the intermediate belief state $Pr(\cdot|a_i^{t-1}, a_j^{t-1}, b_i^{t-1})$, is corrected according to:

$$Pr(is^t|o_i^t, a_i^{t-1}, b_i^{t-1}) = \beta \sum_{a_j^{t-1}} O_i(s^t, a_i^{t-1}, a_j^{t-1}, o_i^t) Pr(is^t|a_i^{t-1}, a_j^{t-1}, b_i^{t-1})$$

  where β is the normalizing constant.

To act, the agent optimizes its beliefs using Eqs. 4.3 and 4.4.

7.3 Subjective Equilibrium in I-POMDPs

In the two-agent I-POMDP framework presented in Section 4.3 of Chapter 4, each agent computes the discounted infinite horizon policy tree (strategy) which is the subjective best response of the agent to its belief. During each step of game play, the agent starting with a prior belief revises it in light of the new information using the Bayesian belief update process outlined in Section 7.2, and computes the optimal strategy given its beliefs. The latter step is equivalent to using its observation history to index into its policy tree (computed offline using the process given in Section 4.3.3) [3], to compute the best response future strategy.

Before we analyze the play of agents, we briefly introduce the requisite background concepts, which we will reference later.

[3] In the infinite horizon case, convergence of value iteration allows us to conveniently represent the policy tree as a finite state machine

7.3.1 Background: Stochastic Processes, Martingales, and Bayesian Learning

A stochastic process is a sequence of random variables, $\{X_t\}, t = 0, 1, \ldots$, whose values are realized one at a time. Well-known examples of stochastic processes are Markov chains, as well as sequences of beliefs updated using the Bayesian update. Bayesian learning turns out to exhibit an additional property that classifies it as a special type of stochastic process, called a Martingale.

A Martingale is a stochastic process that, for any observation history up to time t, h^t, exhibits the property that for all $l \geq t$:

$$E[X_l|h^t] = X_t$$

Consequently, for all future time points $l \geq t$ the expected change, $E[X_l - X_t|h^t] = 0$. A sequence of an agent's beliefs updated using Bayesian learning is known to be a Martingale. Intuitively, this means that the agent's current estimate of the state is equal to what the agent expects its future estimates of the state will be, based on its current observation history. Because the Martingale property of Bayesian learning is central to our results, we sketch a formal proof below.

Let an agent, say i's, initial belief about some state, $\xi \in \Xi$, be $X_0 = Pr_i(\xi)$. The agent receives an observation, o_i, in the future according to a distribution ϕ_i that depends on ξ. Let the revised future belief be $X_1 = Pr_i(\xi|o_i)$. By Bayes theorem, $Pr(\xi|o_i) = \frac{\phi_i(o_i|\xi)Pr_i(\xi)}{Pr(o_i)}$. We will show that $E[Pr_i(\xi|o_i)] = Pr_i(\xi)$, where the expectation is over the future observations:

$$\begin{aligned}
E[Pr_i(\xi|o_i)] &= \textstyle\sum_{o_i} Pr_i(\xi|o_i)Pr_i(o_i) \\
&= \textstyle\sum_{o_i} \frac{\phi_i(o_i|\xi)Pr_i(\xi)}{Pr_i(o_i)} Pr_i(o_i) \\
&= \textstyle\sum_{o_i} \phi_i(o_i|\xi)Pr_i(\xi) \\
&= Pr_i(\xi) \textstyle\sum_{o_i} \phi_i(o_i|\xi) \\
&= Pr_i(\xi) \\
&= X_0
\end{aligned}$$

The above result extends immediately to observation histories of any length t. Formally, $E[X_{t+1}|h^t] = X_t$, and from the law of conditional expectations, $E[X_l|h^t] = X_t, \;\; l \geq t$. Therefore, Bayesian learning is a Martingale.

All Martingales share the following convergence property:

Theorem 7.1 (Martingale Convergence Theorem (§4 of Chapter 7 in Doob, 1953)). *If $\{X_t\}, t = 0, 1, \ldots$ is a Martingale with $E[|X_t|^2] < U < \infty$ for some U and all t, then the sequence of random variables, $\{X_t\}$ converges with probability 1 to some X_∞ in mean-square.*

7.3.2 Subjective Equilibrium

We investigate the asymptotic behavior of agents playing an infinite horizon POSG as formalized by I-POMDPs, in which each agent learns and optimizes. Specifically, each agent starts with a prior belief which is revised on performing an action and receipt of sensory information, followed by computing the strategy which optimizes its beliefs. In the context of I-POMDPs, each agent uses its prior beliefs to index into its policy (computed offline using Equations 4.3 and 4.4) resulting in the policy tree that will form its behavior strategy.

Sequential behavior of agents in the I-POMDP framework may be represented using their observation histories. For an agent, say i, let o_i^t be its observation at time step t. Let $o^t = [o_i^t, o_j^t]$. An observation history of the game is a sequence, $h = \{o^t\}, t = 1, 2, \ldots$. The set of all histories is, $H = \bigcup_{t=1}^{\infty} \Omega^t$ where $\Omega^t = \Pi_1^t(\Omega_i \times \Omega_j)$. The set of observation histories upto time t is, $H^t = \Pi_1^t(\Omega_i \times \Omega_j)$, and the set of future observation paths from time t onwards is, $H_t = \Pi_t^\infty(\Omega_i \times \Omega_j)$.

Example 7.1. *We use the multiagent tiger problem described in Section 4.5 of Chapter 4 as an illustrative example. Briefly, the game consists of two doors, behind one is a tiger and behind the other is some gold, and two agents, i and j. The agents are unaware of where the tiger is (TL or TR), and each can either open any one of two doors, or listen(OL, OR, or L). A tiger emits a growl periodically, which reveals its position behind a door (GL or GR) but only with some certainty. Additionally, each agent can also hear a creak with some certainty, if the other agent opens a door (CL, CR, or S). We will assume that neither agent can perceive other's observations nor actions. The game is not cooperative since either i or j may open a door, thereby reseting the location of the tiger, and rendering any information collected by the other agent about the tiger's location useless to it. Example histories in the multiagent tiger problem are shown in Fig. 7.1.*

In the I-POMDP framework, each agent's belief over the physical state and others' candidate models, together with the agent's perfect information regarding its own model, induces a *predictive* probability distribution over the future observation paths. Because these distributions play a critical role in our analysis, we

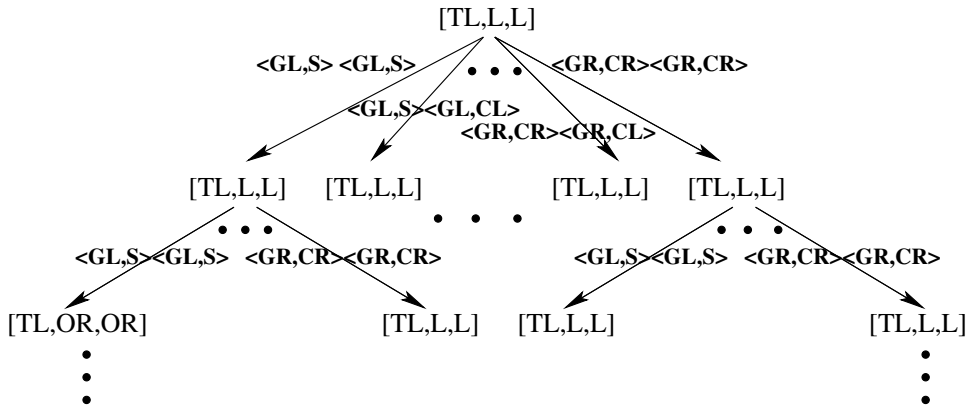

Figure 7.1: Joint observation histories in the infinite horizon multiagent tiger problem. The nodes represent the physical state of the game and play of agents, while the edges are labelled with the possible observations. This example starts with the tiger on the left and each agent listening. Each agent may receive one of six observations (labels on the arrows), and performs an action that optimizes its resulting belief.

represent them mathematically using a collection of probability measures:

$$\{\mu_k\}, k = 0, i, j$$

defined over the space $M \times H$, where $M = M_i \times M_j$ and H is as defined previously, such that:

1. μ_0 is the objective true distribution over models of each agent and the observation histories,
2. $proj_{M_k}\ \mu_k = proj_{M_k}\ \mu_0 = \delta_{m_k} \quad k = i, j$
3. $proj_{M_{-k}}\ \mu_k = proj_{M_{-k}}\ b_k^0 \quad k = i, j$

Here, condition 1 is self-explanatory and condition 2 states that each agent knows its own model (δ_{m_k} is the Kronecker delta function). Condition 3 states that the probability measures, μ_i and μ_j, "contain" i and j's prior beliefs over other's models, respectively. Additionally, $proj_H\ \mu_0$ gives the true distribution over the histories as induced by the initial strategy profile, and $proj_H\ \mu_k$ for $k = i, j$ gives the predictive probability distribution for each agent over the histories at the start of the game. [4]

If the actual sequence of observations in the game does not proceed along a history that is assigned some positive predictive probability by an agent, then the agent's observations would contradict its beliefs and

[4]Following (Nyarko, 1997; Jordan, 1995) the unconditional measure μ_k may be seen as a prior before an agent knows its own model, and μ_k along with the conditions as an *interim* prior once an agent knows its own model.

the Bayesian update would not be possible. Clearly, it is desirable for each agent's initial belief to assign a non-zero probability to each possible observation history; this is called the truth compatibility condition. To formalize this condition we need a notion of absolute continuity of two probability measures.

Definition 7.1 (Absolute Continuity). *A probability measure p_1 is absolutely continuous with p_2, denoted as $p_1 \ll p_2$, if $p_2(E) = 0$ implies $p_1(E) = 0$, for any measurable set E.*

We will utilize the absolute continuity as defined above to formalize the truth compatibility condition, which we call the absolute continuity condition.

Condition 7.1 (Absolute Continuity Condition (ACC)). *ACC holds for any agent $k = i, j$ if $proj_H\ \mu_0 \ll proj_H\ \mu_k$.*

Condition 7.1 states that the probability distribution induced by an agent's initial belief on future observation paths should not rule out positive probability events according to the real probability distribution on the paths. A sure way to satisfy ACC is for each agent's initial belief to have a "grain of truth" – assign a non-zero probability to the true model of the other agent. Since an agent has no way of knowing the true model of its opponent from beforehand, it must assign a non-zero probability to each candidate model of the other agent.

Truth compatible beliefs of an agent that performs Bayesian learning tend to converge in the limit to the opponent model(s) that most likely generates the observations of the agent. In the context of the I-POMDP framework, an agent's belief updated using the process outlined in Section 7.2, will converge in the limit. Formally:

Theorem 7.2 (Bayesian Learning in I-POMDPs). *For an agent in the I-POMDP framework, if its initial belief satisfies the ACC, its posterior beliefs will converge with probability 1.*

Proof. As we proved before, Bayesian learning is a Martingale. In Section 7.3.1, set the state space $\Xi = IS_i$, and the observation function $\phi_i = O_i$. Noting that the I-POMDP belief update is Bayesian, its Martingale property follows from applying the proof outlined in Section 7.3.1 appropriately. In order to apply Theorem 7.1 to the I-POMDP belief update, set $X_t = b_i^t$ where b_i^t is agent i's belief at some time t. We must first

show that $E[|b_i^t|^2]$ is bounded:

$$\begin{aligned} E[|b_i^t|^2] &= \sum_{k=1}^{(|A_i||\Omega_i|)^t} |b_i^t = \widehat{b}_i^k|^2 Pr(\widehat{b}_i^k) \\ &= \sum_{k=1}^{(|A_i||\Omega_i|)^t} \sum_{IS^t} \widehat{b}_i^k(is)^2 Pr(\widehat{b}_i^k) \quad (L_2 \text{ norm}) \\ &\leq \sum_{k=1}^{(|A_i||\Omega_i|)^t} 1 \cdot Pr(\widehat{b}_i^k) \quad (\textstyle\sum_x p(x)^2 \leq 1) \\ &= 1 \end{aligned}$$

Theorem 7.2 now follows from a straightforward application of Theorem 7.1. □

The above result does not imply that an agent's belief converges to the true model of the other agent. This is due to the possible presence of *observationally equivalent* models of the other agent. Given agent i's model, all models of j that induce identical distributions over all possible future observation paths are said to be observationally equivalent. When a particular observation history obtains, agent i is unable to distinguish between the observationally equivalent models of j. In other words, observationally equivalent models generate distinct behaviors for histories which are never observed. As an aside, if for some model of i, all models of j induce distributions over the paths that are mutually singular [5], then Bayesian learning is consistent, and will converge to a point mass.

Example 7.2. *For an example of observationally equivalent models, consider a version of the multiagent tiger game in which the tiger persists behind its original door once any door has been opened. Additionally, i has superior observation capabilities compared to j, and each agent is able to perfectly observe other's actions but observes the growls imperfectly. Let i's utility dictate that it will not open any doors until it's 100% certain of the tiger's location. The corresponding strategy for i is to listen for an infinite number of time steps, and then open the door. Suppose that as a best response to its belief, j were to adopt a strategy in which it would listen for an infinite number of steps, but if at any time i opened a door, it would also open the same door at the next time step (because the tiger persists) and then continue opening the same door. The true distribution assigns a probability 1 to the histories $\{[\langle GL|GR, S\rangle, \langle GL|GR, S\rangle]\}_1^\infty$. Instead of the above mentioned strategy if j were to adopt a follow-the-leader strategy, i.e. j performs the action which i did in the previous time step, then the true distribution would again assign probability 1 to the previously*

[5] A pair of probability measures, p_1 and p_2, are mutually singular, $p_1 \perp p_2$, if there exist a disjoint pair of measurable sets, A and B, such that $p_1(E \cap A) = p_2(E \cap B) = 0$ for any measurable E.

mentioned histories. The two different strategies of j turn out to be observationally equivalent for i; however they differ for observation paths that are not part of the game.

An immediate consequence of the convergence of Bayesian learning is that the predictive distribution over the future observation paths induced by each agent's belief after a finite sequence of observations h_k^t, $proj_{H_t}\ \mu_k(\cdot|h_k^t),\ k = i, j$ becomes arbitrary close to the true distribution, $proj_{H_t}\ \mu_0(\cdot|h^t)$, for a finite t, and converges uniformly in the limit. This is an important result, because it establishes that no matter what the initial beliefs of the agents are, provided that these beliefs are truth compatible, the agents' opinions (about the future) will merge and correctly predict the true future in the limit. This result was first noted in (Blackwell & Dubins, 1962); we present the theorem below and refer the reader to the paper for its proof.

Theorem 7.3 ((Blackwell & Dubins, 1962)). *Suppose that P is a predictive probability on X, and Q is absolutely continuous w.r.t. P. Then for each conditional distribution $P^t(x_1, \ldots, x_t)$ of the future given the past w.r.t. P, there exists a conditional distribution $Q^t(x_1, \ldots, x_t)$ of the future given the past w.r.t. Q such that, $||P^t(x_1, \ldots, x_t) - Q^t(x_1, \ldots, x_t)|| \underset{t\to\infty}{\to} 0$ with Q-probability 1.*

We use Theorem 7.3 to establish predictive convergence within the `I-POMDP` framework.

Theorem 7.4 (ϵ-Predictive Convergence in `I-POMDPs`). *For all agents in the `I-POMDP` framework, if their initial beliefs satisfy the ACC, then for every $\epsilon > 0$, there exists a finite T which is a function of ϵ, such that for all $t \geq T$ and with μ_0-probability 1,*

$$||proj_{H_t}\ \mu_0(\cdot|h^t) - proj_{H_t}\ \mu_k(\cdot|h_k^t)|| \leq \epsilon$$

for $k = i, j$.

Proof. Referring to Theorem 7.3, let $X = H$. We observe that $proj_H\ \mu_0$ and $proj_H\ \mu_k$ for $k = i, j$ are predictive as defined in (Blackwell & Dubins, 1962). Set $Q = proj_H\ \mu_0$, and $P = proj_H\ \mu_k$. Subsequently, $Q^t = proj_{H_t}\ \mu_0(\cdot|h^t)$, and $P^t = proj_{H_t}\ \mu_k(\cdot|h_k^t)$. Theorem 7.4 then follows immediately from a straightforward application of Theorem 7.3. □

We have shown that for a POSG modeled using the `I-POMDP` formalism, the players' beliefs over opponent's models converge in the limit if they satisfy the ACC property. However, the limit beliefs may be

incorrect, due to the inability of agents to distinguish between observationally equivalent models of the opponent on the basis of their observation history. Nevertheless, their beliefs over the future paths come arbitrary close, and remain close, to the true distribution over the future, after a finite amount of time. Further observations will only confirm their beliefs about the truth [6], and will not alter their beliefs. We capture this notion using the concept of a subjective equilibrium (Kalai & Lehrer, 1993a), defined as follows:

Definition 7.2 (Subjective ϵ-Equilibrium). *Let $b_k^t, \ \ k = i, j$ be the agents' beliefs at some time t. A pair of policy trees, $\pi^* = [\pi_i^*, \pi_j^*]$ is a subjective ϵ-equilibrium if,*

1. $\pi_i^* = OPT(b_i^t), \pi_j^* = OPT(b_j^t)$
2. $||proj_{H_t}\ \mu_0(\cdot|h^t) - proj_{H_t}\ \mu_k(\cdot|h_k^t)|| \leq \epsilon, \ \ k = i, j$ *with a μ_0-probability 1.*

When ϵ = 0, subjective equilibrium obtains. Condition 1 of subjective ϵ-equilibrium states that the agents are subjectively rational, i.e. their strategies are best responses to their beliefs. As we mentioned before, these strategies are the policy trees computed using Equations 4.3 and 4.4. The second condition states that the agents' beliefs have attained ϵ-predictive convergence. In other words, a strategy profile is in subjective ϵ-equilibrium when the strategies are best responses to agents' beliefs that have attained ϵ-predictive convergence. Note that the beliefs, b_i^t and b_j^t, are "contained" in the measures $\mu_i(\cdot|h_i^t)$ and $\mu_j(\cdot|h_j^t)$, respectively.

We now establish the main result of this chapter, which is that behavior strategies of agents playing a POSG within the I-POMDP framework, attain subjective ϵ-equilibrium in finite time, and subjective equilibrium in the limit. The following corollary gives our result:

Corollary 7.1 (Convergence to Subjective Equilibrium in I-POMDPs). *Let $\pi = [\pi_i, \pi_j]$ be the strategies of agents i, and j respectively, within the I-POMDP formalism. Let b_i^0, and b_j^0 be their initial beliefs. If the following conditions are met,*

1. $\pi_i = OPT(b_i^0), \pi_j = OPT(b_j^0)$
2. $proj_H\ \mu_0 \ll proj_H\ \mu_k, \ \ k = i, j$ *(ACC)*

then for any $\epsilon > 0$, and for all μ_0-positive probability histories, there exists some finite time step T which is a function of ϵ, such that for all $t \geq T$, the strategy profile, $\pi^* = [\pi_i^*, \pi_j^*]$ is a subjective ϵ-equilibrium where,

[6] Hence these beliefs are sometimes called self-confirming.

- *b_i^t and b_j^t are the agents' beliefs at time t*
- $\pi_i^* = OPT(b_i^t), \pi_j^* = OPT(b_j^t)$

Proof. Corollary 7.1 follows in part from Theorem 7.4, and in part from noting that agents' strategies in the I-POMDP framework are best responses to their posterior beliefs at each time step, and that the beliefs are updated using their observation history. □

Strategy profiles in subjective ϵ-equilibrium for arbitrarily small $\epsilon \geq 0$ are stable. Specifically, further play will bring agents' beliefs over the future closer to the truth statistically, and the corresponding strategy profiles will remain in the subjective ϵ-equilibrium. Note that ACC is a sufficient condition, but not a necessary one. An example setting in which even though ACC is violated, yet subjective ϵ-equilibrium still results is given in (Kalai & Lehrer, 1993a).

7.4 Computational Limitations of Our Results

Recall that in Section 7.2, we made the assumption that agent models (strategies) are computable. This restricts the space of possible strategies to be countable. However, as observed in (Nachbar & Zame, 1996), there exist computable strategies for which no exact best response strategy is computable, and even when computable best responses do exist, the decision procedure of computing these best responses may not be computable. Consequences of these negative results lead to a subtle tension between learning and optimization within the I-POMDP framework. Specifically, if agents' exact best response strategies are not computable, then their beliefs fail to account for such strategies of others, thereby violating the mutual grain of truth assumption. This presents a serious impediment to satisfying ACC and thereby obtaining predictive convergence, in practice. On the other hand, if we posit that best responses be computable, then the corresponding prior beliefs may be unrealistic – for example, they may not assign non-zero probability to all possible strategies of others. Nachbar (1997) makes an argument along similar lines in the context of repeated games using the notion of a conventional set of strategies (analogous to the computable set in our setting) attributed to each agent.[7] We believe that these implausibility issues are a direct implication of Binmore's claim in (1990) that perfect rationality is an unattainable ideal. Binmore proves that a Turing machine cannot always predict truthfully the behavior of an opponent Turing machine (given its complete description) and

[7] Also see (Nachbar, 1997) for a simple illustration of our argument using the game of Matching Pennies.

optimize simultaneously. His claim rests on a particular construction of a two-agent game in which a supposedly rational Turing machine when required to compute the best response is unable to predict truthfully, and when required to predict truthfully is unable to terminate its computations and optimize.

Manifestations of the computational obstacles mentioned above are evident in the I-POMDP framework in a more straightforward manner. In Section 4.4 of Chapter 4, we introduced finitely nested I-POMDPs as computable approximations of I-POMDPs. Within the finitely nested I-POMDP framework, we prove the impossibility of all agents simultaneously satisfying the *grain of truth* assumption. Recall that the grain of truth assumption required assigning a non-zero probability to the true model of the other agent. Beliefs of agents that exhibited a grain of truth also satisfied the ACC, though the converse is not necessarily true [8].

Theorem 7.5 (Impossibility Result). *Within the finitely nested I-POMDP framework, all of the agents' beliefs cannot simultaneously satisfy the grain of truth assumption.*

Proof. In keeping with the spirit of this thesis, we will consider two agents, i and j. Let agent i's strategy level be l_i, and j's strategy level be l_j. We consider the following three cases that are exhaustive:

Case 1: $l_i = l_j$. For agent i, if its strategy level is l_i, then by construction, it considers models of j that have strategy level at most $l_i - 1$. Analogously, if j's strategy level is l_j, then i's models have strategy level at most $l_j - 1$. Because $l_i = l_j$, neither i's nor j's beliefs can account for the true model of the other, and therefore fail to satisfy the grain of truth assumption.

Case 2: $l_i > l_j$. When i's strategy level is l_i, it considers models of j that have a strategy level at most $l_i - 1$. Therefore, i's beliefs that assign non-zero probability to every model of j satisfy the grain of truth assumption. For agent j, because its strategy level is l_j, i's models are ascribed a strategy level of at most $l_j - 1$. Since $l_i > l_j$, j's beliefs cannot satisfy the grain of truth assumption.

Case 3: $l_i < l_j$. Proof for this case is analogous to Case 2; i's beliefs cannot satisfy the grain of truth assumption, while j's can.

For each of the three cases listed above, i's and j's beliefs cannot simultaneously satisfy the grain of truth assumption when formalized using finitely nested I-POMDPs. □

[8] The converse is not true because the grain of truth assumption is stronger than the ACC.

We stress that the impossibility of satisfying the grain of truth assumption does not imply that agents within the finitely nested I−POMDP framework cannot satisfy the ACC – there may exist lower strategy level models that turn out to be observationally equivalent to the higher level models for the observation path of the game. However, because the grain of truth assumption is a realistic way of satisfying the ACC, Theorem 7.5 does indicate the practical difficulties in achieving ACC and therefore equilibria.

Though the above mentioned negative results are existential, they serve to show that it may be problematic to fulfill the assumptions laid out in our analysis – the ACC – in practice. Nevertheless, there may be ways to overcome these limitations. One interesting direction is to replace exact optimization with approximate optimization. Specifically, rather than computing the exact best response to its subjective belief, an agent may compute an ϵ-best response [9] that is guaranteed to be always computable. However, strategies that are ϵ-best responses may differ considerably from strategies that are exact best responses. Consequently, the effect of ϵ-optimality on predictive convergence remains an open question.

7.5 Summary

We analyzed the play of agents engaged in a partially observable stochastic game formalized using the interactive POMDP framework. In particular, we considered subjectively rational agents who may not know others' strategies. Therefore, they maintain beliefs over the physical state and models of other agents and optimize with respect to their beliefs. We have also shown how such agents update their beliefs on performing actions and receiving observations, and compute best responses to their beliefs. Within this framework, we proved that if agents' beliefs satisfy a truth compatibility condition, then strategies of agents that learn and optimize converge to the subjective equilibrium in the limit, and subjective ϵ-equilibrium for arbitrarily small $\epsilon > 0$ in finite time.

We pointed out that attempts to practically validate these theoretical results could run into obstacles. One problem is the inherent difficulty in perfect optimization and simultaneous prediction. As an example, we showed the difficulty in achieving the equilibrium in finitely nested I-POMDPs that are computable approximations of I−POMDPs. One may be forced to resort to ϵ-optimality. Whether any form of equilibrium obtains when the players are bounded rational is a topic of future work.

[9] One way to compute an ϵ-best response is to consider finite horizons for maximization, rather than infinite.

7.6 Contributions

Existence of equilibria in I-POMDPs: Though we adopted a decision-theoretic solution concept (not based on equilibria), we proved the asymptotic existence of (subjective) equilibria in I-POMDPs. Our result demonstrates the role of equilibria as fixed points of play within decision-theoretic frameworks for multiagent settings.

Generalization of prior results: While the concept of subjective equilibrium is not novel, we believe that our results complement and generalize the existing results in the game theory literature. Specifically, using the I-POMDP framework for learning and optimizing, we have shown the existence of equilibria in a POSG, in which, additionally, the assumption of perfect monitoring has been relaxed. The POSGs provide a more realistic setting than repeated games, in which existence of equilibria was known previously.

Obstacles in equilibration within finitely nested I-POMDPs: We commented on and showed the difficulty in satisfying the sufficiency conditions for achieving subjective equilibrium, in practice. The computational obstacles call into the question the role of equilibria in multiagent planning when a decision-theoretic viewpoint is adopted. They also suggest bounded rationality as an important topic for future research.

7.7 Future Work

The computational complications arising out of simultaneous prediction and perfect optimization motivate us to adopt a solution approach that takes into account the bounded rationality of practical agents. Though work exists in game theory that addresses bounded rationality (Rubinstein, 1998), it is not from the perspective of sequential decision-making. Consequently, computational models of boundedly rational agents that optimize and predict form an interesting line of future work. Another related line of research is to then investigate whether our results of asymptotic convergence to equilibria hold for bounded rational agents, and if so under what additional conditions.

Chapter 8

CONCLUSION

INTELLIGENT decision making is a characteristic trait of human behavior. Humans usually co-habit with others in societies; therefore they must make decisions keeping in mind how their actions will affect others, and how others' decisions will affect them. Sometimes, this involves reasoning about the state of mind of others, others' reasoning about others' states of minds, and so on. Humans are also primarily self-interested – they act to advance their own goals or preferences. In cooperative societies, we realize that it is in our best interest to promote the welfare of others, while in non-cooperative societies, the opposite is true.

In this thesis, we presented a computational framework, called the interactive POMDP (I-POMDP), that models the decision making situation of an agent co-habiting a cooperative or non-cooperative multiagent setting. We presented exact and approximation algorithms that enable an agent to plan sequentially, over the long term and strategically, within the I-POMDP framework. Similar to human behavior, our algorithms maintain beliefs and reason with a nested belief system. Applications of the I-POMDP framework are significant: I-POMDPs may be used to control planetary rovers in their exploration missions, plan a long term patient treatment therapy in the context of other interacting treatments, coordinate troop movements in battlefields, and provide formal explanations for social behaviors such as *follow the leader*.

In the remainder of this chapter, we summarize this thesis, and outline avenues of future work. In particular, we briefly review the I-POMDP framework in Section 8.1, focusing on its interdisciplinary nature. We then summarize the particle filtering based approximation technique, in Section 8.2. In Section 8.3, we comment on the existence of equilibria in I-POMDPs, and its role in multiagent planning. In Section 8.4, we outline several avenues of future work in some detail.

8.1 I-POMDP: An Interdisciplinary Approach to Multiagent Planning

We proposed I-POMDP, a new framework for autonomous rational planning in multiagent environments. The framework is applicable to agents that locally compute what actions they must execute in order to optimize their preferences given what they believe, while interacting with other agents. The preferences of others may conflict or agree with those of the agent. I-POMDPs combine the decision-theoretic framework of POMDPs with elements from game theory. Specifically, they generalize Bayesian games by relaxing the assumptions of common knowledge of prior beliefs; they extend repeated games to a sequential setting; and they generalize stochastic games to partially observable environments. Much of game theory uses Nash equilibrium as a solution concept. However, Nash equilibrium suffers from the limitations of being non-unique (there could be many Nash equilibria for a game) and incomplete (Nash equilibrium does not prescribe what an agent should do if others do not follow their part of the equilibrium). I-POMDPs, by adopting a decision-theoretic solution approach based on best response to anticipated actions of others, do not suffer from these limitations. On the decision-theoretic side, they generalize POMDPs, traditionally used for single agent planning, to multiagent settings.

I-POMDPs expand the traditional physical state space of POMDPs to include models of other agents. These models may either be the sophisticated *intentional* models (analogous to types as used in Bayesian games) that include the agent's beliefs, capabilities, and preferences, or the *subintentional* models that are simply mappings from the agent's observation history to a probability distribution over its actions, coupled with its observation history. An example of a subintentional model is a finite state machine. When the other agents' models are intentional, maintaining a belief over the expanded state space results in beliefs over others' beliefs over others, and so on. Since operations such as the belief update on the infinitely nested beliefs are not computable in general, we defined *finitely nested* belief systems as computable approximations of the infinitely nested ones. Similar interactive belief systems have been studied before, in game theory and in theoretical computer science. Our contribution is a novel method for updating an agent's beliefs within the interactive belief system after the agent acts and receives an observation (Proposition 4.2), allowing its use for decision making. The I-POMDP belief update is a conditional update depending on whether the other agent's model is intentional or subintentional. If it is intentional, the update proceeds by anticipating the other agent's action(s) by solving its model, tracking its possible observations, and updating its beliefs. Because the beliefs are nested, the belief update recurses through each level of the belief nesting, until level 0 is reached.

Multiagent Tiger Game Versions	Singly nested beliefs of i			
	i believes j is uninformed	i believes j is informed	i believes j is partly informed	i is uninformed of j's beliefs
NON-COOPERATIVE				
ENEMY	1, 2	1, 2	1, 2	–
NEUTRAL	1, 2	1, 2	1, 2	1, 2, 3
COOPERATIVE				
FRIEND	1, 2	1, 2	1, 2	–
TEAM	1, 2	1, 2	1, 2	–

Table 8.1: A quick summary of the solutions to the multiagent tiger problems that appeared in this thesis. Our solutions are conditioned on the shape of i's beliefs over j's. The numbers in each cell indicate the horizon of the solutions.

If the other agent's model is subintentional, we anticipate the other agent's action(s) by solving its model, and append to its previous observation history, each of its possible observations, weighted with the likelihood of the agent receiving the observation. In Proposition 4.1, we showed that the I-POMDP belief update is a sufficient statistic for the agent's past observation history, thereby paving the way for solving I-POMDPs. For the finitely nested I-POMDP framework, we proved in Theorem 4.1, that value iteration converges to a unique fixed-point, and the value function is always piece-wise linear and convex (Theorem 4.2). Both these properties are analogous to those for POMDPs, and make it possible to compute solutions for I-POMDPs.

We extended the single agent tiger problem traditionally used to illustrate POMDPs, to the multiagent setting. Using the multiagent tiger problem, we illustrated the I-POMDP framework by first demonstrating its superior performance in comparison to the simple approach of using POMDPs in multiagent settings (by treating the other agent as static noise in the environment). Second, we showed value functions and policies for several non-cooperative and cooperative versions of the multiagent tiger problem within the I-POMDP framework. See Table 8.1 for a summary of the multiagent tiger problems that we solved.

The complexity of I-POMDPs restricts their application to all but the simplest settings. However, even in simple problems, we demonstrated intuitive anthropomorphic social behaviors: we showed the emergence of a *follow the leader* behavior in a setting of two agents in which one agent possesses an ability that is superior to that of the other, thereby assuming a leadership role. Additionally, we empirically demonstrated the simple insight that in cooperative settings it is beneficial to have friends that are informed rather than uninformed about the state of the situation. However, in non-cooperative settings, the opposite is true for the adversaries.

8.2 Approximation Methods

The benefit derived from modeling others' beliefs comes at a price: Solving finitely nested I-POMDPs is a PSPACE-Complete problem. Therefore, approximation techniques that trade off computations with quality of the solution are critically needed, if we are to move beyond toy applications. Analogously to POMDPs, I-POMDPs are afflicted with two sources of complexity: the *curse of dimensionality* of the belief space, and the *curse of history* due to the complexity of the policy space. While these curses also affect POMDPs, the complexity of the belief space is greater for I-POMDPs; they include beliefs about the physical environment, and possibly about other agents' beliefs, and their beliefs about others, and so on.

To address the belief space dimensionality problem, we took recourse to sampling methods which are typically immune to the high dimensionality of the underlying state space. We introduced a polynomial-based representation language for interactive beliefs to allow nested sampling. We adapted the basic particle filtering algorithm – the bootstrap filter – to the multiagent setting, resulting in an interactive version of the particle filter. Mirroring the hierarchical nature of interactive beliefs, the *interactive particle filter* (I-PF), samples and propagates particles on each level of the nested belief. We combined the I-PF with value iteration to compute approximately optimal solutions for I-POMDPs. Using two simple test problems, namely the multiagent tiger problem and the multiagent machine maintenance problem, we gave a preliminary indication of the favorable performance of our approximation method. We also derived bounds on the approximation error introduced by the randomized algorithm (Theorem 6.1), and commented on the computational savings.

While the I-PF does successfully alleviate the curse of dimensionality, we are unable to compute solutions beyond a few time horizons. Therefore, we combined the I-PF with a method to beat back the curse of history. Instead of including all possible reachable beliefs at each step in the look ahead reachability tree generated during value iteration, we include only a subset of the likely reachable beliefs. We sample from the observation space to generate this subset. This effectively reduces the branching factor of the reachability tree – the main source of complexity when we scale to larger horizons. The net result is a scalable anytime approximation method that addresses the curse of dimensionality and reduces the impact of the curse of history in I-POMDPs.

8.3 Equilibria in I-POMDPs

Agents within the I-POMDP framework update their beliefs over models of other agents after they act and receive sensory information. Using the Martingale property of the Bayesian belief update, and under the condition of truth compatibility of the prior beliefs – the *absolute continuity condition* – we showed in Theorem 7.2 that an agent's belief will converge uniformly in the limit. A natural result of this convergence is that the distribution over the future joint observations induced by the agent's belief after finite time will become arbitrarily close to the true distribution induced by the actual strategies, and coincide with it in the limit (Corollary 7.1). Strategies that are best responses to beliefs that are consistent with others' actual behaviors (though not necessarily converged to others' true strategies) and the state of the game are said to be in *subjective equilibrium*. The equilibrium is stable because additional observations will only reinforce their beliefs about others' behaviors and the state of the game. This result generalizes a similar result for repeated games to partially observable stochastic games as modeled within I-POMDPs.

While we theoretically proved the existence of equilibrium as a fixed point of play for agents within the I-POMDP framework, realizing it in practice is a different matter. When we stipulate that all strategies be computable, the task of simultaneous prediction and exact optimization may become impossible. Theorem 7.5 exemplified the difficulty by showing that within the finitely nested I-POMDP framework, it is impossible for all the agents to simultaneously satisfy the grain of truth assumption. Inability to satisfy the grain of truth assumption implies that we must find other (non-intuitive) ways to satisfy the absolute continuity condition.

The difficulty in achieving equilibrium computationally calls into question the role of equilibrium in multiagent planning. Within the multiagent learning community, researchers are questioning the relevance of Nash equilibria as a solution paradigm, because to achieve it requires unrealistic conditions on the behaviors of the learning algorithms. Similarly, the limitations of Nash equilibria such as non-uniqueness and incompleteness make it unsuitable as a solution concept for planning. This points to the normative decision theoretic approach as being more practical.

8.4 Future Work

Much of the work described in this thesis is foundational. We introduced a new framework for planning in multiagent settings, and analyzed its working using simple toy problems. To address its enormous computational complexity we developed the first approximation techniques based on sampling. We also analyzed the play of agents within this framework, establishing the theoretical existence of equilibria as a fixed point, but realizing the potential computational obstacles in reaching the fixed point. Future work involves finding new exact and approximate methods that in addition to reducing the complexity, give tighter bounds on approximation errors. Additionally, we are also interested in large scale realistic applications that will bring I-POMDPs into mainstream thinking. We outline a few of the directions of future research in some detail.

8.4.1 Lossless Compression of the Interactive State Space

The interactive state space includes not only the states of the physical environment but also the models of the other agents. Some of these models may be intentional and include the beliefs, capabilities as well as preferences of the agents, while others may be subintentional. It is possible to define an *equivalence* relation on the space of the models which will partition it into a collection of equivalence classes. All models within an equivalence class when solved generate identical policy trees. As an example, let models of the other agent be level 0 intentional models or POMDPs that differ only in the beliefs. Then the partition of the belief space induced by the value function (obtained from solving the POMDP) is a collection of equivalence classes. If the number of actions and observations are finite, then the number of equivalence classes are also finite.

The new interactive state space, $\tilde{IS}_{i,l}$ is a combination of the physical state space and the equivalence classes. The compression of the original interactive state space into the new one is lossless: the value function over the new belief space and therefore the optimal policy remain unchanged. The theorem below captures this result. Note that the I-POMDP and the beliefs are of strategy level l, but for the sake of clarity we do not indicate it explicitly.

Theorem 8.1. *For a finitely nested I-POMDP$_i$, define a mapping $\mathcal{CP} : \Delta(IS_i) \rightarrow \Delta(\tilde{IS}_i)$ such that,*

$$\tilde{b}_i(s, c_{j,k}) = \int_{b_j \in c_{j,k}} b_i(s, b_j) db_j \tag{8.1}$$

where $b_i \in \Delta(IS_i)$, $\tilde{b}_i \in \Delta(\tilde{IS}_i)$, and $c_{j,k}$ is the k^{th} equivalence class of j's models. Then the mapping $\mathcal{CP}$ is value preserving.

Proof by induction. Let b_i be an arbitrary belief of agent i. Let $EC_j = \{c_{j,1}, c_{j,2}, ..., c_{j,n}\}$ be the collection of equivalence classes of agent j's belief. Each class $c_{j,k}$ is a set of beliefs of j such that the action a_j^k is optimal for each belief. Thus $\forall_{b_j \in c_{j,k}} ER(s, b_j, a_i) = R(s, a_i, a_j^k)$, because a_j^k is optimal for all $b_j \in c_{j,k}$.

Basis Step: We show that the horizon 1 value remains unchanged when i's original belief is replaced by its belief over the equivalence classes.

$$
\begin{aligned}
&Q^1(b_i, a_i) = \textstyle\int_{is_i} b_i(is_i) ER(is_i, a_i) = \sum_s \int_{b_j} b_i(s, b_j) ER(s, b_j, a_i) \\
&= \textstyle\sum_s \left\{ \int_{b_j \in c_{j,1}} b_i(s, b_j) ER(s, b_j, a_i) + ... + \int_{b_j \in c_{j,n}} b_i(s, b_j) ER(s, b_j, a_i) \right\} \\
&= \textstyle\sum_s \left\{ \int_{b_j \in c_{j,1}} b_i(s, b_j) R(s, a_i, a_j^1) + ... + \int_{b_j \in c_{j,n}} b_i(s, b_j) R(s, a_i, a_j^n) \right\} \\
&= \textstyle\sum_s \left\{ R(s, a_i, a_j^1) \int_{b_j \in c_{j,1}} b_i(s, b_j) + ... + R(s, a_i, a_j^n) \int_{b_j \in c_{j,n}} b_i(s, b_j) \right\} \\
&= \textstyle\sum_s \left\{ R(s, a_i, a_{j,1}) \tilde{b}_i(s, c_{j,1}) + ... + R(s, a_i, a_{j,n}) \tilde{b}_i(s, c_{j,n}) \right\} && \text{(using Eq. 8.1)} \\
&= \textstyle\sum_{s,k} \tilde{b}_i(s, c_{j,k}) R(s, a_i, a_{j,k}) = \tilde{Q}^1(\tilde{b}_i, a_i)
\end{aligned}
$$

Because the Q values remain unchanged, maximizing over them will also yield identical values.

Inductive Hypothesis: Let us assume that $\forall a_i, b_i \; Q^N(b_i, a_i) = \tilde{Q}^N(\tilde{b}_i, a_i)$ where $\tilde{b}_i$ is related to b_i using Eq. 8.1. Because the Q values are identical, the N horizon value function also remains unchanged.

Inductive Proof:

$$
\begin{aligned}
&Q^{N+1}(b_i, a_i) = \textstyle\int_{is_i} b_i(is_i) ER(is_i, a_i) + \gamma \sum_{o_i} Pr(o_i | b_i, a_i) V^N(SE(b_i, a_i, o_i)) \\
&= Q^1(b_i, a_i) + \gamma \textstyle\sum_{o_i, is_i} Pr(o_i | is_i, a_i) b_i(is_i) V^N(SE(b_i, a_i, o_i)) \\
&= \tilde{Q}^1(\tilde{b}_i, a_i) + \gamma \textstyle\sum_{o_i, is_i, a_j} Pr(o_i | is_i, a_i, a_j) Pr(a_j | b_j) b_i(is_i) V^N(SE(b_i, a_i, o_i)) && \text{(Basis step)} \\
&= \tilde{Q}^1(\tilde{b}_i, a_i) + \gamma \textstyle\sum_{o_i} \sum_s \int_{b_j} \sum_{a_j} Pr(o_i | is_i, a_i, a_j) Pr(a_j | b_j) b_i(s, b_j) V^N(SE(b_i, a_i, o_i)) \\
&= \tilde{Q}^1(\tilde{b}_i, a_i) + \gamma \textstyle\sum_{o_i} \sum_{s,k} \int_{b_j \in c_{j,k}} \sum_{a_j} Pr(o_i | is_i, a_i, a_j) Pr(a_j | b_j) b_i(s, b_j) V^N(SE(b_i, a_i, o_i))
\end{aligned}
$$

Using the BNM and BNO assumption:

$$
\begin{aligned}
&Q^{N+1}(b_i, a_i) = \tilde{Q}^1(\tilde{b}_i, a_i) + \gamma \textstyle\sum_{o_i} \sum_{s,k} \int_{b_j \in c_{j,k}} Pr(o_i | s, a_i, a_j^k) b_i(s, b_j) V^N(SE(b_i, a_i, o_i)) \\
&= \tilde{Q}^1(\tilde{b}_i, a_i) + \gamma \textstyle\sum_{o_i} \sum_{s,k} Pr(o_i | s, a_i, a_j^k) \int_{b_j \in c_{j,k}} b_i(s, b_j) V^N(SE(b_i, a_i, o_i)) && \text{(using Eq. 8.1)} \\
&= \tilde{Q}^1(\tilde{b}_i, a_i) + \gamma \textstyle\sum_{o_i} \sum_{s,k} Pr(o_i | s, a_i, a_j^k) \tilde{b}_i(s, c_{j,k}) V^N(SE(b_i, a_i, o_i))
\end{aligned}
$$

Note that $Pr(o_i|s, a_i, c_{j,k}) = \sum_{a_j} Pr(o_i|s, a_i, a_j, c_{j,k}) Pr(a_j|c_{j,k}) = Pr(o_i|s, a_i, a_{j,k})$

$= \tilde{Q}^1(\tilde{b}_i, a_i) + \gamma \sum_{o_i} \sum_{s,k} Pr(o_i|s, a_i, c_{j,k}) \tilde{b}_i(s, c_{j,k}) \tilde{V}^N(SE(\tilde{b}_i, a_i, o_i))$ (Inductive Hypothesis)

$= \tilde{Q}^1(\tilde{b}_i, a_i) + \gamma \sum_{o_i} Pr(o_i|a_i, \tilde{b}_i) \tilde{V}^N(SE(\tilde{b}_i, a_i, o_i))$

$= \tilde{Q}^{N+1}(\tilde{b}_i, a_i)$

Because the Q values are identical under the mapping, the values of the compressed beliefs remain unchanged. We have assumed in the proof that agent j's policies are deterministic, i.e., for every partition $c_{j,k}$ there is one optimal action $a_{j,k}$. The proof extends in a straightforward manner when there is more than one optimal action for a class. □

Because the compressed interactive state space $\tilde{IS}_i$ is of less dimension than the original interactive state space, it is possible to visualize complete solutions of I-POMDPs, in constrast to solutions for specific beliefs.

8.4.2 Other Approximation Methods

Since methods for solving I-POMDPs are conceptually similar to those for solving POMDPs, we can leverage the variety of POMDP approximation techniques to approximate I-POMDPs. While the sample based approximation method introduced in this thesis proved to be scalable and delivered good performance, the approximation bounds were loose. Therefore, new approximation techniques in addition to addressing both the sources of complexity in order to be viable, must provide tighter error bounds.

Promising approaches for addressing the curse of dimensionality include statistically identifying those physical states and models of other agents that are most relevant from the perspective of making decisions (for example, see Roy & Gordon, 2002), and finding a threshold level for nested beliefs beyond which the additional levels of beliefs do not significantly affect the behavior. Approaches that may address the curse of history include utilizing bounded finite state controllers as policies (for example, see Poupart & Boutilier, 2004), and other innovative methods to prune the look ahead reachability tree. Deriving tight error bounds would be a key requirement for any approximation method.

8.4.3 Multiagent Planning with Bounded Rational Agents

There is growing theoretical evidence that perfect rationality is an unattainable ideal (Binmore, 1990; Rubinstein, 1998). The limits on perfect rationality arise due to the boundedness of time and space, and our

choice of computability models [1]. Researchers are therefore turning their attention to bounded rationality and ways to represent it (for example, see Rubinstein, 1998).

Within the I−POMDP framework, the computational obstacle in reaching the subjective equilibrium is an implication of the inability of agents to be perfectly rational. Consequently, we must look at models that compactly represent the boundedness of the resources available to the agents and the approximate ϵ-optimization performed by practical agents. Such models may simply be the intentional models augmented with additional parameters that capture the limited resources. Investigating whether any type of equilibria results when agents are bounded rational is another interesting line of future work.

[1] The inability of Turing machines to decide the halting problem is at the heart of several arguments against perfect rationality.

APPENDICES

Appendix A Proofs of Theorems

Proof of Propositions 4.1 and 4.2. We start with Proposition 4.2, by applying the Bayes Theorem:

$$b_i^t(is^t) = Pr(is^t|o_i^t, a_i^{t-1}, b_i^{t-1}) = \frac{Pr(is^t, o_i^t|a_i^{t-1}, b_i^{t-1})}{Pr(o_i^t|a_i^{t-1}, b_i^{t-1})}$$

$$\begin{aligned}
&= \beta \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) Pr(is^t, o_i^t|a_i^{t-1}, is^{t-1}) \\
&= \beta \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \sum_{a_j^{t-1}} Pr(is^t, o_i^t|a_i^{t-1}, a_j^{t-1}, is^{t-1}) Pr(a_j^{t-1}|a_i^{t-1}, is^{t-1}) \\
&= \beta \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \sum_{a_j^{t-1}} Pr(is^t, o_i^t|a_i^{t-1}, a_j^{t-1}, is^{t-1}) Pr(a_j^{t-1}|is^{t-1}) \\
&= \beta \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \sum_{a_j^{t-1}} Pr(a_j^{t-1}|m_j^{t-1}) Pr(o_t^i|is^t, a^{t-1}, is^{t-1}) Pr(is^t|a^{t-1}, is^{t-1}) \\
&= \beta \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \sum_{a_j^{t-1}} Pr(a_j^{t-1}|m_j^{t-1}) Pr(o_t^i|is^t, a^{t-1}) Pr(is^t|a^{t-1}, is^{t-1}) \\
&= \beta \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \sum_{a_j^{t-1}} Pr(a_j^{t-1}|m_j^{t-1}) O_i(s^t, a^{t-1}, o_i^t) Pr(is^t|a^{t-1}, is^{t-1})
\end{aligned} \tag{A-1}$$

To simplify the term $Pr(is^t|a^{t-1}, is^{t-1})$ let us substitute the interactive state is^t with its components. When m_j in the interactive states is intentional: $is^t = (s^t, \theta_j^t) = (s^t, b_j^t, \widehat{\theta}_j^t)$.

$$\begin{aligned}
Pr(is^t|a^{t-1}, is^{t-1}) &= Pr(s^t, b_j^t, \widehat{\theta}_j^t|a^{t-1}, is^{t-1}) \\
&= Pr(b_j^t|s^t, \widehat{\theta}_j^t, a^{t-1}, is^{t-1}) Pr(s^t, \widehat{\theta}_j^t|a^{t-1}, is^{t-1}) \\
&= Pr(b_j^t|s^t, \widehat{\theta}_j^t, a^{t-1}, is^{t-1}) Pr(\widehat{\theta}_j^t|s^t, a^{t-1}, is^{t-1}) Pr(s^t|a^{t-1}, is^{t-1}) \\
&= Pr(b_j^t|s^t, \widehat{\theta}_j^t, a^{t-1}, is^{t-1}) I(\widehat{\theta}_j^{t-1}, \widehat{\theta}_j^t) T_i(s^{t-1}, a^{t-1}, s^t)
\end{aligned} \tag{A-2}$$

When m_j is subintentional: $is^t = (s^t, m_j^t) = (s^t, h_j^t, \widehat{m}_j^t)$.

$$\begin{aligned} Pr(is^t|a^{t-1}, is^{t-1}) &= Pr(s^t, h_j^t, \widehat{m}_j^t|a^{t-1}, is^{t-1}) \\ &= Pr(h_j^t|s^t, \widehat{m}_j^t, a^{t-1}, is^{t-1}) Pr(s^t, \widehat{m}_j^t|a^{t-1}, is^{t-1}) \\ &= Pr(h_j^t|s^t, \widehat{m}_j^t, a^{t-1}, is^{t-1}) Pr(\widehat{\theta}_j^t|s^t, a^{t-1}, is^{t-1}) Pr(s^t|a^{t-1}, is^{t-1}) \\ &= Pr(h_j^t|s^t, \widehat{m}_j^t, a^{t-1}, is^{t-1}) I(\widehat{m}_j^{t-1}, \widehat{m}_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \end{aligned} \quad \text{(A-2')}$$

The joint action pair, a^{t-1}, may change the physical state. The third term on the right-hand side of Eqs. A-2 and A-2' above captures this transition. We utilized the MNM assumption to replace the second terms of the equations with boolean identity functions, $I(\widehat{\theta}_j^{t-1}, \widehat{\theta}_j^t)$ and $I(\widehat{m}_j^{t-1}, \widehat{m}_j^t)$ respectively, which equal 1 if the two frames are identical, and 0 otherwise. Let us turn our attention to the first terms. If m_j in is^t and is^{t-1} is intentional:

$$\begin{aligned} Pr(b_j^t|s^t, \widehat{\theta}_j^t, a^{t-1}, is^{t-1}) &= \textstyle\sum_{o_j^t} Pr(b_j^t|s^t, \widehat{\theta}_j^t, a^{t-1}, is^{t-1}, o_j^t) Pr(o_j^t|s^t, \widehat{\theta}_j^t, a^{t-1}, is^{t-1}) \\ &= \textstyle\sum_{o_j^t} Pr(b_j^t|s^t, \widehat{\theta}_j^t, a^{t-1}, is^{t-1}, o_j^t) Pr(o_j^t|s^t, \widehat{\theta}_j^t, a^{t-1}) \\ &= \textstyle\sum_{o_j^t} \tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t) O_j(s_t, a^{t-1}, o_j^t) \end{aligned} \quad \text{(A-3)}$$

Else if it is subintentional:

$$\begin{aligned} Pr(h_j^t|s^t, \widehat{m}_j^t, a^{t-1}, is^{t-1}) &= \textstyle\sum_{o_j^t} Pr(h_j^t|s^t, \widehat{m}_j^t, a^{t-1}, is^{t-1}, o_j^t) Pr(o_j^t|s^t, \widehat{m}_j^t, a^{t-1}, is^{t-1}) \\ &= \textstyle\sum_{o_j^t} Pr(h_j^t|s^t, \widehat{m}_j^t, a^{t-1}, is^{t-1}, o_j^t) Pr(o_j^t|s^t, \widehat{m}_j^t, a^{t-1}) \\ &= \textstyle\sum_{o_j^t} \delta_K(\text{APPEND}(h_j^{t-1}, o_j^t) - h_j^t) O_j(s_t, a^{t-1}, o_j^t) \end{aligned} \quad \text{(A-3')}$$

In Eq. A-3, the first term on the right-hand side is 1 if agent j's belief update, $SE_{\theta_j}(b_j^{t-1}, a_j^{t-1}, o_j^t)$ generates a belief state equal to b_j^t. Similarly, in Eq. A-3', the first term is 1 if appending the o_j^t to h_j^{t-1} results in h_j^t. δ_K is the Kronecker delta function. In the second terms on the right-hand side of the equations, the MNO assumption makes it possible to replace $Pr(o_j^t|s^t, \widehat{\theta}_j^t, a^{t-1})$ with $O_j(s^t, a^{t-1}, o_j^t)$, and $Pr(o_j^t|s^t, \widehat{m}_j^t, a^{t-1})$ with $O_j(s^t, a^{t-1}, o_j^t)$ respectively.

Let us now substitute Eq. A-3 into Eq. A-2.

$$Pr(is^t|a^{t-1}, is^{t-1}) = \textstyle\sum_{o_j^t} \tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t) O_j(s^t, a^{t-1}, o_j^t) I(\widehat{\theta}_j^{t-1}, \widehat{\theta}_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \quad \text{(A-4)}$$

Substituting Eq. A-3' into Eq. A-2' we get,

$$Pr(is^t|a^{t-1}, is^{t-1}) = \sum_{o_j^t} \delta_K(\text{APPEND}(h_j^{t-1}, o_j^t), h_j^t) O_j(s^t, a^{t-1}, o_j^t) I(\widehat{m}_j^{t-1}, \widehat{m}_j^t) \times T_i(s^{t-1}, a^{t-1}, s^t) \quad \text{(A-4')}$$

Replacing Eq. A-4 into Eq. A-1 we get:

$$b_i^t(is^t) = \beta \sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \sum_{a_j^{t-1}} Pr(a_j^{t-1}|\theta_j^{t-1}) O_i(s^t, a^{t-1}, o_i^t) \sum_{o_j^t} \tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t) \times O_j(s^t, a^{t-1}, o_j^t) I(\widehat{\theta}_j^{t-1}, \widehat{\theta}_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \quad \text{(A-5)}$$

Similarly, replacing Eq. A-4' into Eq. A-1 we get:

$$b_i^t(is^t) = \beta \sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \sum_{a_j^{t-1}} Pr(a_j^{t-1}|m_j^{t-1}) O_i(s^t, a^{t-1}, o_i^t) \times \sum_{o_j^t} \delta_K(\text{APPEND}(h_j^{t-1}, o_j^t) - h_j^t) O_j(s^t, a^{t-1}, o_j^t) I(\widehat{m}_j^{t-1}, \widehat{m}_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \quad \text{(A-5')}$$

We arrive at the final expressions for the belief update by removing the terms $I(\widehat{\theta}_j^{t-1}, \widehat{\theta}_j^t)$ and $I(\widehat{m}_j^{t-1}, \widehat{m}_j^t)$ and changing the scope of the first summations.

When m_j in the interactive states is intentional:

$$b_i^t(is^t) = \beta \sum_{is^{t-1}: \widehat{m}_j^{t-1} = \widehat{\theta}_j^t} b_i^{t-1}(is^{t-1}) \sum_{a_j^{t-1}} Pr(a_j^{t-1}|\theta_j^{t-1}) O_i(s^t, a^{t-1}, o_i^t) \times \sum_{o_j^t} \tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t) O_j(s^t, a^{t-1}, o_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \quad \text{(A-6)}$$

Else, if it is subintentional:

$$b_i^t(is^t) = \beta \sum_{is^{t-1}: \widehat{m}_j^{t-1} = \widehat{m}_j^t} b_i^{t-1}(is^{t-1}) \sum_{a_j^{t-1}} Pr(a_j^{t-1}|m_j^{t-1}) O_i(s^t, a^{t-1}, o_i^t) \times \sum_{o_j^t} \delta_K(\text{APPEND}(h_j^{t-1}, o_j^t) - h_j^t) O_j(s^t, a^{t-1}, o_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \quad \text{(A-7)}$$

Since proposition 2 expresses the belief $b_i^t(is^t)$ in terms of parameters of the previous time step only, Proposition 1 holds as well. □

Before we present the proof of Theorem 4.1 we note that the Equation 4.3, which defines value iteration in I-POMDPs, can be rewritten in the following form, $U^n = HU^{n-1}$. Here, $H : B(\Theta_i) \rightarrow B(\Theta_i)$ is a

backup operator, and is defined as,

$$HU^{n-1}(\theta_i) = \max_{a_i \in A_i} h(\theta_i, a_i, U^{n-1})$$

where $h : \Theta_i \times A_i \times B(\Theta_i) \rightarrow \mathbb{R}$ is,

$$h(\theta_i, a_i, U) \quad = \sum_{is} b_i(is) ER_i(is, a_i) + \gamma \sum_{o \in \Omega_i} Pr(o_i | a_i, b_i) U(\langle SE_{\theta_i}(b_i, a_i, o_i), \hat{\theta}_i \rangle)$$

and where $B(\Theta_i)$ is the set of all bounded value functions U. Lemmas 1 and 2 establish important properties of the backup operator. Proof of Lemma 1 is given below, and proof of Lemma 2 follows thereafter.

Proof of Lemma 4.1. Select arbitrary value functions V and U such that $V(\theta_{i,l}) \leq U(\theta_{i,l}) \ \ \forall \theta_{i,l} \in \Theta_{i,l}$. Let $\theta_{i,l}$ be an arbitrary type of agent i.

$$\begin{aligned}
HV(\theta_{i,l}) &= \max_{a_i \in A_i} \left\{ \sum_{is} b_i(is) ER_i(is, a_i) + \gamma \sum_{o \in \Omega_i} Pr(o_i | a_i, b_i) V(\langle SE_{\theta_{i,l}}(b_i, a_i, o_i), \hat{\theta}_i \rangle) \right\} \\
&= \sum_{is} b_i(is) ER_i(is, a_i^*) + \gamma \sum_{o \in \Omega_i} Pr(o_i | a_i^*, b_i) V(\langle SE_{\theta_{i,l}}(b_i, a_i^*, o_i), \hat{\theta}_i \rangle) \\
&\leq \sum_{is} b_i(is) ER_i(is, a_i^*) + \gamma \sum_{o \in \Omega_i} Pr(o_i | a_i^*, b_i) U(\langle SE_{\theta_{i,l}}(b_i, a_i^*, o_i), \hat{\theta}_i \rangle) \\
&\leq \max_{a_i \in A_i} \left\{ \sum_{is} b_i(is) ER_i(is, a_i) + \gamma \sum_{o \in \Omega_i} Pr(o_i | a_i, b_i) U(\langle SE_{\theta_{i,l}}(b_i, a_i, o_i), \hat{\theta}_i \rangle) \right\} \\
&= HU(\theta_{i,l})
\end{aligned}$$

Since $\theta_{i,l}$ is arbitrary, $HV \leq HU$. □

Proof of Lemma 4.2. Assume two arbitrary well defined value functions V and U such that $V \leq U$. From Lemma 4.1 it follows that $HV \leq HU$. Let $\theta_{i,l}$ be an arbitrary type of agent i. Also, let a_i^* be the action that optimizes $HU(\theta_{i,l})$.

$$
\begin{aligned}
0 \quad & \leq HU(\theta_{i,l}) - HV(\theta_{i,l}) \\
& = \max_{a_i \in A_i} \left\{ sum_{is} b_i(is) ER_i(is, a_i) + \gamma \textstyle\sum_{o \in \Omega_i} Pr(o_i|a_i, b_i) U(SE_{\theta_{i,l}}(b_i, a_i, o_i), \langle \hat{\theta}_i \rangle) \right\} - \\
& \quad \max_{a_i \in A_i} \left\{ \textstyle\sum_{is} b_i(is) ER_i(is, a_i) + \gamma \sum_{o \in \Omega_i} Pr(o_i|a_i, b_i) V(SE_{\theta_{i,l}}(b_i, a_i, o_i), \langle \hat{\theta}_i \rangle) \right\} \\
& \leq \textstyle\sum_{is} b_i(is) ER_i(is, a_i^*) + \gamma \sum_{o \in \Omega_i} Pr(o_i|a_i^*, b_i) U(SE_{\theta_{i,l}}(b_i, a_i^*, o_i), \langle \hat{\theta}_i \rangle) - \\
& \quad \textstyle\sum_{is} b_i(is) ER_i(is, a_i^*) - \gamma \sum_{o \in \Omega_i} Pr(o_i|a_i^*, b_i) V(SE_{\theta_{i,l}}(b_i, a_i^*, o_i), \langle \hat{\theta}_i \rangle) \\
& = \gamma \textstyle\sum_{o \in \Omega_i} Pr(o_i|a_i^*, b_i) U(SE_{\theta_{i,l}}(b_i, a_i^*, o_i), \langle \hat{\theta}_i \rangle) - \gamma \sum_{o \in \Omega_i} Pr(o_i|a_i^*, b_i) V(SE_{\theta_{i,l}}(b_i, a_i^*, o_i), \langle \hat{\theta}_i \rangle) \\
& = \gamma \textstyle\sum_{o \in \Omega_i} Pr(o_i|a_i^*, b_i) \left[U(SE_{\theta_{i,l}}(b_i, a_i^*, o_i), \langle \hat{\theta}_i \rangle) - V(SE_{\theta_{i,l}}(b_i, a_i^*, o_i), \langle \hat{\theta}_i \rangle) \right] \\
& \leq \gamma \textstyle\sum_{o \in \Omega_i} Pr(o_i|a_i^*, b_i) ||U - V|| \\
& = \gamma ||U - V||
\end{aligned}
$$

As the supremum norm is symmetrical, a similar result can be derived for $HV(\theta_{i,l}) - HU(\theta_{i,l})$. Since $\theta_{i,l}$ is arbitrary, the Contraction property follows, i.e. $||HV - HU|| \leq ||V - U||$. □

Lemmas 4.1 and 4.2 provide the stepping stones for proving Theorem 4.1. Proof of Theorem 4.1 follows from a straightforward application of the Banach Fixed-point theorem. Of course, we must first show that the normed space $(B(\Theta_i), ||\cdot||)$ is a Banach space. The proof for this is identical to that of Theorem 2.1 in Chapter 2.

We state the Banach Fixed-point theorem (Stokey & E., 1989) below:

Theorem A.1 (Banach Fixed-point Theorem). *If (S, ρ) is a complete metric space and $T : S \rightarrow S$ is a contraction mapping with modulus γ, then*

1. *T has exactly one fixed point U^* in S, and*
2. *The sequence $\{U^n\}$ converges to U^*.*

Proof of Theorem 4.1 follows.

Proof of Theorem 4.1. The normed space $(B(\Theta_i), ||\cdot||)$ is complete w.r.t the metric induced by the supremum norm. Lemma 2 establishes the contraction property of the backup operator, H. Using Theorem A.1, and substituting T with H, convergence of value iteration in I-POMDPs to a unique fixed point is established. □

We go on to the piecewise linearity and convexity (PWLC) property of the value function. We follow the outlines of the analogous proof for POMDPs in Section 2.1.2 of Chapter 2.

Let $\alpha : IS \rightarrow \mathbb{R}$ be a real-valued and bounded function. Let the space of such real-valued bounded functions be $B(IS)$. We will now define an inner product.

Definition A.1 (Inner product). *Define the inner product,* $\langle \cdot, \cdot \rangle : B(IS) \times \Delta(IS) \rightarrow \mathbb{R}$*, by*

$$\langle \alpha, b_i \rangle = \sum_{is} b_i(is)\alpha(is)$$

The next lemma establishes the bilinearity of the inner product defined above.

Lemma A.1 (Bilinearity). *For any* $s, t \in \mathbb{R}$*,* $f, g \in B(IS)$*, and* $b, \lambda \in \Delta(IS)$ *the following equalities hold:*

$$\langle sf + tg, b \rangle = s\langle f, b \rangle + t\langle g, b \rangle$$
$$\langle f, sb + t\lambda \rangle = s\langle f, b \rangle + t\langle f, \lambda \rangle$$

We are now ready to give the proof of Theorem 4.2. Theorem A.2 restates Theorem 4.2 mathematically, and its proof follows thereafter.

Theorem A.2 (PWLC). *The value function,* U^n*, in finitely nested* $\texttt{I-POMDP}_{i,l}$ *is piece-wise linear and convex (PWLC). Mathematically,*

$$U^n(\theta_{i,l}) = \max_{\alpha^n} \sum_{is} b_i(is)\alpha^n(is) \qquad n = 1, 2, ...$$

Proof of Theorem A.2. **Basis Step:** $n = 1$

From Bellman's Dynamic Programming equation,

$$U^1(\theta_i) = \max_{a_i} \sum_{is} b_i(is)ER(is, a_i) \tag{A-8}$$

where $ER_i(is, a_i) = \sum_{a_j} R(is, a_i, a_j) Pr(a_j|m_j)$. Here, $ER_i(\cdot)$ represents the expectation of R w.r.t. agent j's actions. Eq. A-8 represents an inner product and using Lemma A.1, the inner product is linear in b_i. By selecting the maximum of a set of linear vectors, we obtain a PWLC horizon 1 value function.[2]

Inductive Hypothesis: Suppose that $U^{n-1}(\theta_{i,l})$ is PWLC. Formally we have,

$$\begin{aligned} U^{n-1}(\theta_{i,l}) &= \max_{\alpha^{n-1}} \textstyle\sum_{is} b_i(is)\alpha^{n-1}(is) \\ &= \max_{\dot{\alpha}^{n-1},\, \ddot{\alpha}^{n-1}} \left\{ \textstyle\sum_{is:m_j \in IM_j} b_i(is)\dot{\alpha}^{n-1}(is) + \sum_{is:m_j \in SM_j} b_i(is)\ddot{\alpha}^{n-1}(is) \right\} \end{aligned} \tag{A-9}$$

Inductive Proof: To show that $U^n(\theta_{i,l})$ is PWLC.

$$U^n(\theta_{i,l}) = \max_{a_i^{t-1}} \left\{ \sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) ER_i(is^{t-1}, a_i^{t-1}) + \gamma \sum_{o_i^t} Pr(o_i^t|a_i^{t-1}, b_i^{t-1}) U^{n-1}(\theta_{i,l}) \right\}$$

From the inductive hypothesis:

$$U^n(\theta_{i,l}) = \max_{a_i^{t-1}} \left\{ \begin{array}{l} \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) ER_i(is^{t-1}, a_i^{t-1}) \\ \quad + \gamma \textstyle\sum_{o_i^t} Pr(o_i^t|a_i^{t-1}, b_i^{t-1}) \max_{\alpha^{n-1} \in \Gamma^{n-1}} \sum_{is^t} b_i^t(is^t)\alpha^{n-1}(is^t) \end{array} \right\}$$

Let $l(b_i^{t-1}, a_i^{t-1}, o_i^t)$ be the index of the alpha vector that maximizes the value at $b_i^t = SE(b_i^{t-1}, a_i^{t-1}, o_i^t)$. Then

$$U^n(\theta_{i,l}) = \max_{a_i^{t-1}} \left\{ \begin{array}{l} \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) ER_i(is^{t-1}, a_i^{t-1}) \\ \quad + \gamma \textstyle\sum_{o_i^t} Pr(o_i^t|a_i^{t-1}, b_i^{t-1}) \sum_{is^t} b_i^t(is^t) \alpha^{n-1}_{l(b_i^{t-1}, a_i^{t-1}, o_i^t)} \end{array} \right\}$$

From the second equation in the inductive hypothesis:

$$U^n(\theta_{i,l}) = \max_{a_i^{t-1}} \left\{ \begin{array}{l} \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) ER_i(is^{t-1}, a_i^{t-1}) + \gamma \sum_{o_i^t} Pr(o_i^t|a_i^{t-1}, b_i^{t-1}) \\ \quad \times \left\{ \textstyle\sum_{is^t:m_j^t \in IM_j} b_i^t(is^t) \dot{\alpha}^{n-1}_{l(b_i^{t-1}, a_i^{t-1}, o_i^t)} + \sum_{is^t:m_j^t \in SM_j} b_i^t(is^t) \ddot{\alpha}^{n-1}_{l(b_i^{t-1}, a_i^{t-1}, o_i^t)} \right\} \end{array} \right\}$$

[2] If $|S| = 2$, then the value function is composed of a set of lines, otherwise it is composed of a set of hyperplanes.

Substituting b_i^t with the appropriate belief updates from Eqs. A-5 and A-5' we get:

$$
\begin{aligned}
U^n(\theta_{i,l}) \quad &= \max_{a_i^{t-1}} \Bigg\{ \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) ER_i(is^{t-1}, a_i^{t-1}) + \gamma \sum_{o_i^t} Pr(o_i^t | a_i^{t-1}, b_i^{t-1}) \\
&\times \beta \Bigg[\textstyle\sum_{is^t : m_j^t \in IM_j} \sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \Bigg\{ \sum_{a_j^{t-1}} Pr(a_j^{t-1} | \theta_j^{t-1}) \Bigg[O_i(s^t, a^{t-1}, o_i^t) \\
&\times \textstyle\sum_{o_j^t} O_j^t(s^t, a^{t-1}, o_j^t) \Bigg\{ \tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t) I(\widehat{\theta}_j^{t-1}, \widehat{\theta}_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \Bigg\} \Bigg] \Bigg\} \\
&\times \dot{\alpha}^{n-1}_{l(b_i^{t-1}, a_i^{t-1}, o_i^t)}(is^t) \\
&+ \textstyle\sum_{is^t : m_j^t \in SM_j} \sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \Bigg\{ \sum_{a_j^{t-1}} Pr(a_j^{t-1} | m_j^{t-1}) \Bigg[O_i(s^t, a^{t-1}, o_i^t) \\
&\times \textstyle\sum_{o_j^t} O_j^t(s^t, a^{t-1}, o_j^t) \Bigg\{ \delta_K(\text{APPEND}(h_j^{t-1}, o_j^t) - h_j^t) I(\widehat{m}_j^{t-1}, \widehat{m}_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \Bigg\} \Bigg] \Bigg\} \\
&\times \ddot{\alpha}^{n-1}_{l(b_i^{t-1}, a_i^{t-1}, o_i^t)}(is^t) \Bigg] \Bigg\} \\
&= \max_{a_i^{t-1}} \Bigg\{ \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) ER_i(is^{t-1}, a_i^{t-1}) + \gamma \sum_{o_i^t} \Bigg[\sum_{is^t : m_j^t \in IM_j} \\
&\times \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \Bigg\{ \sum_{a_j^{t-1}} Pr(a_j^{t-1} | \theta_j^{t-1}) \Bigg[O_i(s^t, a^{t-1}, o_i^t) \\
&\times \textstyle\sum_{o_j^t} O_j^t(s^t, a^{t-1}, o_j^t) \Bigg\{ \tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t) I(\widehat{\theta}_j^{t-1}, \widehat{\theta}_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \Bigg\} \Bigg] \Bigg\} \\
&\times \dot{\alpha}^{n-1}_{l(b_i^{t-1}, a_i^{t-1}, o_i^t)}(is^t) \\
&+ \textstyle\sum_{is^t : m_j^t \in SM_j} \sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \Bigg\{ \sum_{a_j^{t-1}} Pr(a_j^{t-1} | m_j^{t-1}) \Bigg[O_i(s^t, a^{t-1}, o_i^t) \\
&\times \textstyle\sum_{o_j^t} O_j^t(s^t, a^{t-1}, o_j^t) \Bigg\{ \delta_K(\text{APPEND}(h_j^{t-1}, o_j^t) - h_j^t) I(\widehat{m}_j^{t-1}, \widehat{m}_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \Bigg\} \Bigg] \Bigg\} \\
&\times \ddot{\alpha}^{n-1}_{l(b_i^{t-1}, a_i^{t-1}, o_i^t)}(is^t) \Bigg] \Bigg\}
\end{aligned}
$$

Rearranging the terms of the equation:

$$\begin{aligned}
U^n(\theta_{i,l}) &= \max_{a_i^{t-1}} \Bigg\{ \textstyle\sum_{is^{t-1}:m_j^{t-1}\in IM_j} b_i^{t-1}(is^{t-1}) \Bigg\{ ER_i(is^{t-1}, a_i^{t-1}) + \gamma \textstyle\sum_{o_i^t} \sum_{is^t:m_j^t\in IM_j} \\
&\times \Bigg\{ \textstyle\sum_{a_j^{t-1}} Pr(a_j^{t-1}|\theta_j^{t-1}) \Bigg[O_i(s^t, a^{t-1}, o_i^t) \textstyle\sum_{o_j^t} O_j^t(s^t, a^{t-1}, o_j^t) \\
&\times \Bigg\{ \tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t) I(\widehat{\theta}_j^{t-1}, \widehat{\theta}_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \Bigg\} \Bigg] \Bigg\} \dot{\alpha}^{n-1}_{l(b_i^{t-1}, a_i^{t-1}, o_i^t)}(is^t) \Bigg\} \\
&+ \textstyle\sum_{is^{t-1}:m_j^{t-1}\in SM_j} b_i^{t-1}(is^{t-1}) \Bigg\{ ER_i(is^{t-1}, a_i^{t-1}) + \gamma \textstyle\sum_{o_i^t} \sum_{is^t:m_j^t\in SM_j} \sum_{o_i^t} \\
&\times \Bigg\{ \textstyle\sum_{a_j^{t-1}} Pr(a_j^{t-1}|m_j^{t-1}) \Bigg[O_i(s^t, a^{t-1}, o_i^t) \textstyle\sum_{o_j^t} O_j^t(s^t, a^{t-1}, o_j^t) \\
&\times \Bigg\{ \delta_K(\text{APPEND}(h_j^{t-1}, o_j^t) - h_j^t) I(\widehat{m}_j^{t-1}, \widehat{m}_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \Bigg\} \Bigg] \Bigg\} \ddot{\alpha}^{n-1}_{l(b_i^{t-1}, a_i^{t-1}, o_i^t)}(is^t) \Bigg\} \Bigg\} \\
&= \max_{a_i^{t-1}} \Bigg\{ \textstyle\sum_{is^{t-1}:m_j^{t-1}\in IM_j} b_i^{t-1}(is^{t-1}) \dot{\alpha}^n_{a_i}(is^{t-1}) \\
&+ \textstyle\sum_{is^{t-1}:m_j^{t-1}\in SM_j} b_i^{t-1}(is^{t-1}) \ddot{\alpha}^n_{a_i}(is^{t-1}) \Bigg\}
\end{aligned}$$

Therefore,

$$\begin{aligned}
U^n(\theta_{i,l}) &= \max_{\dot{\alpha}^n,\, \ddot{\alpha}^n} \Bigg\{ \textstyle\sum_{is^{t-1}:m_j^{t-1}\in IM_j} b_i^{t-1}(is^{t-1}) \dot{\alpha}^n(is^{t-1}) \\
&+ \textstyle\sum_{is^{t-1}:m_j^{t-1}\in SM_j} b_i^{t-1}(is^{t-1}) \ddot{\alpha}^n(is^{t-1}) \Bigg\} \\
&= \max_{\alpha^n} \textstyle\sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \alpha^n(is^{t-1}) = \max_{\alpha^n} \langle b_i^{t-1}, \alpha^n \rangle
\end{aligned} \tag{A-10}$$

where, if m_j^{t-1} in is^{t-1} is intentional then $\alpha^n = \dot{\alpha}^n$:

$$\begin{aligned}
\dot{\alpha}^n(is^{t-1}) &= ER_i(is^{t-1}, a_i^{t-1}) + \gamma \textstyle\sum_{o_i^t} \sum_{is^t:m_j^t\in IM_j} \Bigg\{ \textstyle\sum_{a_j^{t-1}} Pr(a_j^{t-1}|\theta_j^{t-1}) \Bigg[O_i(is^t, a^{t-1}, o_i^t) \\
&\times \textstyle\sum_{o_j^t} O_j^t(is_j^t, a^{t-1}, o_j^t) \Bigg\{ \tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t) I(\widehat{\theta}_j^{t-1}, \widehat{\theta}_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \Bigg\} \Bigg] \Bigg\} \\
&\times \alpha^{n-1}_{l(b_i^{t-1}, a_i^{t-1}, o_i^t)}(is^t)
\end{aligned}$$

and, if m_j^{t-1} is subintentional then $\alpha^n = \ddot{\alpha}^n$:

$$\begin{aligned}\ddot{\alpha}^n(is^{t-1}) &= ER_i(is^{t-1}, a_i^{t-1}) + \gamma \sum_{o_i^t} \sum_{is^t : m_j^t \in SM_j} \left\{ \sum_{a_j^{t-1}} Pr(a_j^{t-1}|\theta_j^{t-1}) \Big[O_i(is^t, a^{t-1}, o_i^t) \right. \\ &\quad \times \sum_{o_j^t} O_j^t(is_j^t, a^{t-1}, o_j^t) \Big\{ \delta_K(\text{APPEND}(h_j^{t-1}, o_j^t) - h_j^t) I(\widehat{m}_j^{t-1}, \widehat{m}_j^t) T_i(s^{t-1}, a^{t-1}, s^t) \Big\} \Big] \Big\} \\ &\quad \times \alpha^{n-1}_{l(b_i^{t-1}, a_i^{t-1}, o_i^t)}(is^t)\end{aligned}$$

Eq. A-10 is an **inner product** and using Lemma A.1, the value function is linear in b_i^{t-1}. Furthermore, maximizing over a set of linear vectors (which are either lines or hyperplanes) produces a piecewise linear and convex value function. □

Appendix B Multiagent Machine Maintenance Problem

We extend the traditional single agent version of the machine maintenance problem (Smallwood & Sondik, 1973) to the two-agent purely cooperative version. We increase the non-determinism of the original problem to make it more realistic. This has the beneficial effect of producing a rich policy structure.

- **Physical state space:** S = {0-fail, 1-fail, 2-fail}
- **Action space:** $A = A_i \times A_j$ where $A_i = A_j$ = {M, E, I, R}
- **Observation space:** $\Omega_i = \Omega_j$ = {not-defective, defective}
- **Transition function:** $T_i : S \times A \times S \rightarrow [0, 1]$

$\langle a_i, a_j \rangle$	**State**	**0-fail**	**1-fail**	**2-fail**
⟨M/E,M/E⟩	0-fail	0.81	0.18	0.01
⟨M/E,M/E⟩	1-fail	0.0	0.9	0.1
⟨M/E,M/E⟩	2-fail	0.0	0.0	1.0
⟨M,I/R⟩	0-fail	1.0	0.0	0.0
⟨M,I/R⟩	1-fail	0.95	0.05	0.0
⟨M,I/R⟩	2-fail	0.95	0.0	0.05
⟨E,I/R⟩	0-fail	1.0	0.0	0.0
⟨E,I/R⟩	1-fail	0.95	0.05	0.0
⟨E,I/R⟩	2-fail	0.95	0.0	0.05
⟨I/R,*⟩	0-fail	1.0	0.0	0.0
⟨I/R,*⟩	1-fail	0.95	0.05	0.0
⟨I/R,*⟩	2-fail	0.95	0.0	0.05

Table B-1: $T_i = T_j$

- **Observation function:** $O_i : S \times A \times \Omega_i \rightarrow [0, 1]$

$\langle a_i, a_j \rangle$	**State**	**not-defective**	**defective**
⟨M,M/E⟩	*	0.5	0.5
⟨M,I/R⟩	*	0.95	0.05
⟨E,M/E⟩	0-fail	0.75	0.25
⟨E,M/E⟩	1-fail	0.5	0.5
⟨E,M/E⟩	2-fail	0.25	0.75
⟨E,I/R⟩	*	0.95	0.05
⟨I/R,*⟩	*	0.95	0.05

$\langle a_i, a_j \rangle$	**State**	**not-defective**	**defective**
⟨M/E,M⟩	*	0.5	0.5
⟨I/R,M⟩	*	0.95	0.05
⟨M/E,E⟩	0-fail	0.75	0.25
⟨M/E,E⟩	1-fail	0.5	0.5
⟨M/E,E⟩	2-fail	0.25	0.75
⟨I/R,E⟩	*	0.95	0.05
⟨*,I/R⟩	*	0.95	0.05

Table B-2: Observation functions for agents i and j.

- **Reward function:** $R_i : S \times A \rightarrow \mathbb{R}$

$\langle a_i, a_j \rangle$	**0-fail**	**1-fail**	**2-fail**
⟨M,M⟩	1.805	0.95	0.5
⟨M,E⟩	1.555	0.7	0.25
⟨M,I⟩	0.4025	-1.025	-2.25
⟨M,R⟩	-1.0975	-1.525	-1.75
⟨E,M⟩	1.5555	0.7	0.25
⟨E,E⟩	1.305	0.45	0.0
⟨E,I⟩	0.1525	-1.275	-2.5
⟨E,R⟩	-1.3475	-1.775	-2.0
⟨I,M⟩	0.4025	-1.025	-2.25
⟨I,E⟩	0.1525	-1.275	-2.5
⟨I,I⟩	-1.0	-3.00	-5.00
⟨I,R⟩	-2.5	-3.5	-4.5
⟨R,M⟩	-1.0975	-1.525	-1.75
⟨R,E⟩	-1.3475	-1.775	-2.0
⟨R,I⟩	-2.5	-3.5	-4.5
⟨R,R⟩	-4	-4	-4

$\langle a_i, a_j \rangle$	**0-fail**	**1-fail**	**2-fail**
⟨M,M⟩	1.805	0.95	0.5
⟨M,E⟩	1.555	0.7	0.25
⟨M,I⟩	0.4025	-1.025	-2.25
⟨M,R⟩	-1.0975	-1.525	-1.75
⟨E,M⟩	1.555	0.7	0.25
⟨E,E⟩	1.305	0.45	0.0
⟨E,I⟩	0.1525	-1.275	-2.5
⟨E,R⟩	-1.3475	-1.775	-2.0
⟨I,M⟩	0.4025	-1.025	-2.25
⟨I,E⟩	0.1525	-1.275	-2.5
⟨I,I⟩	-1.0	-3.00	-5.00
⟨I,R⟩	-2.5	-3.5	-4.5
⟨R,M⟩	-1.0975	-1.525	-1.75
⟨R,E⟩	-1.3475	-1.775	-2.0
⟨R,I⟩	-2.5	-3.5	-4.5
⟨R,R⟩	-4	-4	-4

Table B-3: Reward functions for agents i and j.

CITED LITERATURE

Aberdeen, D. (2003). A survey of approximate methods for solving partially observable markov decision processes. Technical report, National ICT Australia.

Aliprantis, C. D., & Burkinshaw, O. (1998). *Principles of Real Analysis*. Academic Press.

Alon, N., & Spencer, J. (2000). *The Probabilistic Method*. John Wiley and Sons.

Ambruster, W., & Boge, W. (1979). *Bayesian Game Theory*. North Holland.

Aumann, R., & Brandenburger, A. (1995). Epistemic conditions for nash equilibrium. *Econometrica*, *63*(5), 1161 – 1180.

Aumann, R. J. (1999). Interactive epistemology i: Knowledge. *International Journal of Game Theory*, *28*, 263–300.

Aumann, R. J., & Heifetz, A. (2002). *Handbook of Game Theory with Economic Applications*, Vol. 3. Elsevier Science.

Battigalli, P. (1996). Hierarchies of conditional beliefs and interactive epistemology in dynamic games..

Bernstein, D. S., Givan, R., Immerman, N., & Zilberstein, S. (2002). The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, *27*(4), 819–840.

Bertsekas, D. (1995). *Dynamic Programming and optimal control*. Athena Scientific.

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

Blackwell, D., & Dubins, L. (1962). Merging of opinions with increasing information. *Annals of Mathematical Statistics*, *33*(3), 882–886.

Boutilier, C. (1999). Sequential optimality and coordination in multiagent systems. In *Sixteenth International Joint Conference on Artificial Intelligence*, pp. 478–485.

Boutilier, C., & Poole, D. (1996). Computing optimal policies for partially observable decision processes using compact representations. In *AAAI*.

Bowling, M., & Veloso, M. (2002). Multiagent learning using a variable learning rate. *Artificial Intelligence Journal*, *136*, 215–250.

Brafman, R. I. (1997). A heuristic variable grid solution method for pomdps. In *AAAI*.

Brandenburger, A. (2002). The power of paradox: Some recent developments in interactive epistemology..

Brandenburger, A., & Dekel, E. (1993). Hierarchies of beliefs and common knowledge. *Journal of Economic Theory*, *59*, 189–198.

Cassandra, A. R., Kaelbling, L. P., & Littman, M. L. (1994). Acting optimally in partially observable stochastic domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, WA.

Cassandra, A. R., Littman, M. L., & Zhang, N. L. (1997). Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In *Uncertainty in Artificial Intelligence*, Rhode Island, Providence.

Claus, C., & Boutilier, C. (1997). The dynamics of reinforcement learning in cooperative multiagent systems. In *Workshop on Learning in Multiagent Systems AAAI-97*.

Crisan, D., & Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, *50*(3), 736–746.

Dennett, D. (1986). *Intentional Systems*. Brainstorms. MIT Press.

Doob, J. L. (1953). *Stochastic Processes*. John Wiley and Sons.

Doshi, P., & Gmytrasiewicz, P. J. (2005a). Approximating state estimation in multiagent settings using particle filters. In *Autonomous Agents and Multi-agent Systems (AAMAS)*.

Doshi, P., & Gmytrasiewicz, P. J. (2005b). A particle filtering based approach to approximating interactive pomdps. In *National Conference on AI (AAAI)*.

Doucet, A., Freitas, N. D., & Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer Verlag.

Fagin, R., Geanakoplos, J., Halpern, J., & Vardi, M. (1999). The hierarhical approach to modeling knowledge and common knowledge. *International Journal of Game Theory*, *28*.

Fagin, R., Halpern, J., Moses, Y., & Vardi, M. (1995). *Reasoning about Knowledge*. MIT Press.

Fox, D., Burgard, W., Kruppa, H., & Thrun, S. (2000). A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots on Heterogenous Multi-Robot Systems*, *8*(3).

Fudenberg, D., & Levine, D. (1993). Self-confirming equilibrium. *Econometrica*, *61*, 523–545.

Fudenberg, D., & Levine, D. (1997). *Theory of Learning in Games*. MIT Press.

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

Gal, Y., & Pfeffer, A. (2003). A language for modeling agents' decision making processes in games. In *AAMAS*, pp. 265–272, Melbourne, Australia.

Geweke, J. (1989). Bayesian inference in econometric models using monte carlo integration. *Econometrica*, *57*, 1317–1339.

Gmytrasiewicz, P., & Doshi, P. (2004). Interactive pomdps: Properties and preliminary results. In *AAMAS*, pp. 1374–1375, NYC, NY.

Gmytrasiewicz, P., & Durfee, E. (2000). Rational coordination in multi-agent environments. *Autonomous Agents and Multiagent Systems Journal*, *3*(4), 319–350.

Gmytrasiewicz, P. J., & Doshi, P. (2004). A framework for sequential planning in multi-agent settings. In *AMAI*, Ft. Lauderdale, Florida.

Gmytrasiewicz, P. J., & Doshi, P. (2005). A framework for sequential planning in multi-agent settings. *JAIR*, *23*.

Gordon, N., Salmond, D., & Smith, A. (1993). Novel approach to non-linear/non-gaussian bayesian state estimation. *IEEE Proceedings-F*, *140*(2), 107–113.

Hahn, F. (1973). *On the Notion of Equilibrium in Economics: An Inaugural Lecture*. Cambridge University Press.

Hansen, E. (1998). Solving pomdps by searching in policy space. In *Uncertainty in AI*.

Hansen, E., & Feng, Z. (2000). Dynamic programming for pomdps using a factored state representation. In *AI Planning and Scheduling*.

Harsanyi, J., & Selten, R. (1988). *A General Theory of Equilibrium Selection in Games*. Cambridge: MIT Press.

Harsanyi, J. C. (1967). Games with incomplete information played by 'bayesian' players. *Mgmt. Science*, *14*(3), 159–182.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, *57*, 97–109.

Hauskrecht, M. (1997). *Planning and control in stochastic domains with imperfect information*. Ph.D. thesis, MIT.

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

Heifetz, A., & Samet, D. (1998). Topology-free typology of beliefs. *Journal of Economic Theory*, *82*, 324–341.

Hu, J., & Wellman, M. P. (1998). Multiagent reinforcement learning: Theoretical framework and an algorithm. In *15th Intl Conference on Machine Learning*, pp. 242–250.

Jordan, J. S. (1995). Bayesian learning in repeated games. *Games and Economic Behavior*, *1*, 8–20.

Kadane, J., & Larkey, P. (1982). Subjective probability and the theory of games. *Management Science*, *28*(2), 113–120.

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

Kalai, E., & Lehrer, E. (1993a). Rational learning leads to nash equilibrium. *Econometrica*, *61*(5), 1019–1045.

Kalai, E., & Lehrer, E. (1993b). Subjective equilibrium in repeated games. *Econometrica*, *61*(5), 1231–1240.

Koller, D., & Milch, B. (2001). Multi-agent influence diagrams for representing and solving games. In *17th Intl Joint Conf on AI*, pp. 1027–1034.

Kramer, S. C., & Sorenson, H. (1988). Recursive bayesian estimation using piecewise constant approximations. *Automatica*, *24*, 789–801.

Li, M., & Vitanyi, P. (1997). *An Introduction to Kolmogorov Complexity and its Applications*. Springer.

Littman, M. (1994). Markov games as a framework for multiagent reinforcement learning. In *International Conference on Machine Learning*.

Littman, M., Cassandra, A., & Kaelbling, L. (1995a). Learning policies for partially observable environments: Scaling up.. In *International Conference on Machine Learning (ICML)*.

Littman, M. L., Dean, T. L., & Kaelbling, L. P. (1995b). On the complexity of solving Markov decision problems. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*.

Lovejoy, W. S. (1991). Computationally feasible bounds for partially observed markov decision processes. *Operations Research*, *39*(1), 162–175.

Madani, O., Hanks, S., & Condon, A. (2003). On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, *147*, 5–34.

Mertens, J., & Zamir, S. (1985). Formulation of bayesian analysis for games with incomplete information. *International Journal of Game Theory*, *14*, 1–29.

Monahan, G. E. (1982). A survey of partially observable markov decision processes: Theory, models, and algorithms. *Management Science*, *28*(1), 1–16.

Myerson, R. B. (1991). *Game Theory: Analysis of Conflict*. Harvard University Press.

Nachbar, J. H. (1997). Prediction, optimization, and learning in repeated games. *Econometrica*, *65*(2), 275–309.

Nachbar, J. H., & Zame, W. R. (1996). Non-computable strategies and discounted repeated games. *Economic Theory*, *8*, 103–122.

Nair, R., Tambe, M., Yokoo, M., Pynadath, D., & Marsella, S. (2003). Taming decentralized pomdps : Towards efficient policy computation for multiagent settings. In *International Joint Conference on AI*.

Noh, S., & Gmytrasiewicz, P. (2001). Identifying the scope of modeling for time-critical multiagent decision-making. In *7th Intl Conf. on AI*, pp. 1043–1048.

Nyarko, Y. (1997). Convergence in economic models with bayesian hierarchies of beliefs. *Journal of Economic Theory*, *74*, 266–296.

Ooi, J. M., & G.W.Wornell (1996). Decentralized control of a multiple broadcast channel. In *35th Conference on Decision and Control*.

Ortiz, L., & Kaelbling, L. (2000). Sampling methods for action selection in influence diagrams. In *AAAI*, pp. 378–385, Austin, TX.

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

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

Papadimitriou, C., & Tsitsiklis, J. (1987b). The complexity of markov decision processes. *Mathematical Journal of Operations Research*, *12*(3), 441–450.

Pineau, J., Roy, N., & Thrun, S. (2001). A hierarchical approach to pomdp planning and execution. In *Workshop on Hierarchy and Memory in Reinforcement Learning (ICML 2001)*.

Pineau, J., Gordon, G., & Thrun, S. (2003a). Applying metric trees for belief-point pomdps. In *NIPS*.

Pineau, J., Gordon, G., & Thrun, S. (2003b). Point-based value iteration: An anytime algorithm for pomdps. In *International Joint Conference on AI (IJCAI)*.

Poupart, P., & Boutilier, C. (2003). Value-directed compression in pomdps. In *Neural Information Processing Systems*.

Poupart, P., & Boutilier, C. (2004). Vdcbpi: An approximate algorithm scalable for large-scale pomdps. In *NIPS*.

Poupart, P., Ortiz, L., & Boutilier, C. (2001). Value-directed sampling methods for monitoring pomdps. In *UAI*, pp. 453–461, Seattle, USA.

Powers, R., & Shoham, Y. (2005). New criteria and a new algorithm for learning in multi-agent systems. In *NIPS*, Vancouver, Canada.

Puterman, M. L. (1994). *Markov Decision Processes:Discrete Stochastic Dynamic Programming*. Wiley series in probability and mathematical statistics. Wiley-Interscience.

Roy, N., & Gordon, G. (2002). Exponential family pca for belief compression. In *Neural Information Processing Systems (NIPS)*.

Rubinstein, A. (1998). *Modeling Bounded Rationality*. Cambridge: MIT Press.

Russell, S., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach (Second Edition)*. Prentice Hall.

Schmidt, J., Spiegel, A., & Srinivasan, A. (1995). Chernoff-hoeffding boundings for applications with limited independence. *SIAM Journal on Discrete Mathematics*, *8*, 223–250.

Shoham, Y., & Lleyton-Brown, K. (2002). Introduction to multi-agent systems. In preparation: Available at http://ai.stanford.edu/people/shoham.

Shoham, Y., Powers, R., & Grenager, T. (2003). Multi-agent reinforcement learning: A critical survey. Tech. rep., Stanford University.

Smallwood, R., & Sondik, E. (1973). The optimal control of pomdps over a finite horizon. *Op. Res.*, *21*, 1071–1088.

Sorenson, H. W., & Alspach, D. L. (1971). Recursive bayesian estimation using gaussian sums. *Automatica*, *7*, 465–479.

Sorenson, H. W. (Ed.). (1985). *Kalman Filtering: Theory and Application*. IEEE Press, New York.

Stokey, N. L., & E., L. R. (1989). *Recursive Methods in Economic Dynamics*. Harvard Univ. Press.

Sutton, R., & Barto, A. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Tatman, J. A., & Shachter, R. D. (1990). Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man, and Cybernetics*, *20*(2), 365–379.

Tesauro, G. (2003). Extending q-learning to general adaptive multi-agent systems. In *Neural Information Processing Systems*.

Thrun, S. (2000). Monte carlo pomdps. In *NIPS 12*, pp. 1064–1070.

Tsitsiklis, J., & Roy, B. V. (1996). Feature-based methods for large scale dynamic programming. *Machine Learning*, *22*, 59–94.

Von Neumann, J., & Morgenstern, O. (1953). *Theory of Games and Economic Behavior* (third edition). Princeton University Press, Princeton.

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards*. Ph.D. thesis, Cambridge University.

Zhou, R., & Hansen, E. (2001). An improved grid-based approximation algorithm for pomdps. In *International Joint Conference on AI (IJCAI)*.

VITA

NAME: Prashant J. Doshi

EDUCATION:
Ph.D. in Computer Science, 2005
University of Illinois at Chicago, Chicago, IL

M.S. in Computer Science, 2001
Drexel University, Philadelphia, PA

B.E. in Computer Engineering, 1999
University of Mumbai, India

HONORS:
University Fellowship, 2004 – 2005
University of Illinois at Chicago, Chicago, IL

Best Research Poster Award, 2003
IBM T. J. Watson Research Center, Hawthorne, NY

Dean's Fellowship, 1999–2000
Drexel University, Philadelphia, PA

WORK EXPERIENCE:
Research Assistant, 2002–2004
University of Illinois at Chicago, Chicago, IL

Research Intern, 2002, 2003
IBM T. J. Watson Research Center, Hawthorne, NY

Teaching Assistant, 1999 – 2002
University of Illinois at Chicago, Chicago, IL
Drexel University, Philadelphia, PA

PUBLICATIONS:
Journals
1. Gmytrasiewicz, P. J., & Doshi, P. (2005). A framework for sequential planning in multiagent settings. *Journal of AI Research (JAIR)*, Vol 23.

2. Doshi, P., Goodwin, R., Akkiraju, R., & Verma, K. (2005). Dynamic workflow composition using markov decision processes. *International Journal of Web Services Research (JWSR)*, Vol 2(1): 1–17.

3. Clarke, J., Trooskin, S., Doshi, P., Greenwald, L., & Mode, C. (2002). Time to laparotomy from intra-abdominal bleeding from trauma does affect survival rates upto 90 minutes. *The Journal of Trauma: Injury, Infection, and Critical Care*, Vol 52(3): 420–425.

Conferences

1. Doshi, P., & Gmytrasiewicz, P. J. (2005). A particle filtering based approximation method for interactive POMDPs. *National Conference on AI (AAAI)*, Pittsburgh, July 9-13.

2. Doshi, P., & Gmytrasiewicz, P. J. (2005). Approximating state estimation in multiagent settings using particle filters. *International Autonomous Agents and Multiagent Systems Conference (AAMAS)*, Utrecht, Netherlands, July 25-29.

3. Gmytrasiewicz, P. J., & Doshi, P. (2004). Interactive POMDPs: Properties and preliminary results. *International Autonomous Agents and Multiagent Systems Conference (AAMAS)*, poster, pp. 1374–1375, New York, NY, July 19-23.

4. Doshi, P., Goodwin, R., Akkiraju, R., & Verma, K. (2004). Dynamic workflow composition using markov decision processes. *International Conference on Web Services (ICWS)*, pp. 576–582, San Diego, CA, July 6-9.

5. Gmytrasiewicz, P. J., & Doshi, P. (2004). A framework for sequential planning in multiagent settings. AI&M9-2004, *International Symposium on AI & Math (AMAI)*, Ft. Lauderdale, Jan 4-6.

6. Doshi, P., Greenwald, L., & Clarke, J. (2003). "Using bayesian networks for cleansing trauma data. *International FLAIRS Conference*, pp. 72–76, St. Augustine, FL, May 12-14.

Selected Workshops and Symposiums

1. Doshi, P., & Gmytrasiewicz, P. J. (2005). Subjective equilibrium in interactive POMDPs: Theory and computational limitations. *Game Theory and Decision Theory Workshop (GTDT)*, IJCAI, Edinburgh, Scotland, July 31.

2. Doshi, P. (2004). A framework for optimal sequential planning in multiagent settings", *AAAI/SIGART Doctoral Consortium*, AAAI, San Jose, July 25-26.

3. Verma, K., Akkiraju, R., Goodwin, R., Doshi, P., & Lee, J. (2004). On accommodating inter-service dependencies in web process flow composition. *AAAI Spring Symposium on Semantic Web Services, Stanford*, CA, March 22-24.