APPROXIMATE MODEL EQUIVALENCE FOR INTERACTIVE DYNAMIC INFLUENCE DIAGRAMS

by

MUTHUKUMARAN CHANDRASEKARAN

(Under the direction of Prashant Doshi)

ABSTRACT

Interactive dynamic influence diagrams (I-DIDs) graphically visualize a sequential decision problem for uncertain settings where multiple agents interact not only amongst themselves but also with the environment that they are in. Algorithms currently available for solving these I-DIDs face the issue of an exponentially growing candidate model space ascribed to the other agents, over time. One such algorithm identifies and prunes behaviorally equivalent models and replaces them with a representative thereby reducing the model space. We seek to further reduce the complexity by additionally pruning models that are approximately subjectively equivalent. Toward this, we define subjective equivalence in terms of the distribution over the subject agent's future action-observation paths, and introduce the notion of ϵ-subjective equivalence. We present a new approximation technique that uses our new definition of subjective equivalence to reduce the candidate model space by pruning models that are ϵ-subjectively equivalent with representative ones.

INDEX WORDS: Distributed Artificial Intelligence, Multiagent Systems, Decision making, Interactive Dynamic Influence Diagrams, Agent modeling, Behavioral equivalence, Subjective equivalence

APPROXIMATE MODEL EQUIVALENCE FOR INTERACTIVE DYNAMIC INFLUENCE DIAGRAMS

by

MUTHUKUMARAN CHANDRASEKARAN

B.Tech., SASTRA University, 2007

A Thesis Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2010

APPROXIMATE MODEL EQUIVALENCE FOR INTERACTIVE DYNAMIC INFLUENCE DIAGRAMS

by

MUTHUKUMARAN CHANDRASEKARAN

Approved:

Major Professor: Prashant Doshi

Committee: Khaled Rasheed
Walter D. Potter

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
May 2010

DEDICATION

To Chandrasekaran and Brinda, my loving parents, Hariharan, my supportive brother, and friends.

ACKNOWLEDGMENTS

First, I would like to thank my major professor, Dr. Prashant Doshi, for his supervision, advice and guidance. I am indebted to him for having faith in me during difficult times and for all his encouragement.

I am grateful to Dr. Yifeng Zeng for his advice and crucial contributions, which made him the backbone of this research. I would like to particularly thank him for patiently answering all my sometimes−basic questions. I hope to continue collaborating with him in the future.

I would like to express my sincere gratitude to Dr. Khaled Rasheed for being my graduate advisor and helping me make important decisions when it mattered the most.

I would like to thank Nithya Vembu for helping me proofread my thesis and for sitting through infinite dry runs of my defense presentation.

I would also like to thank my brother, Hariharan, for his valuable advice on efficient programming.

Last but not the least, I would like to thank Ekhlas, Xia, Ananta, Tom, Matt and others at the Institute for Artificial Intelligence at the University of Georgia for indulging in very useful discussions which benifited me in more ways than they actually know.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Decisions in the real world often need to be made under conditions of uncertainty. Here, the decision maker has to choose among alternatives (that may have one of several consequences) where each of these alternatives is associated with a probability distribution that is known. There has been much advancement in this field in recent years. Researchers have realized the need to develop strategies that enhance the ability to deal with uncertain information in a straight forward natural way which will in turn improve the quality of planning, enable more rational responses to unexpected events, and allow a better understanding of available options. These enhancements will enable people and machines to make better decisions in less time and with lower costs. This growth in interest for developing algorithms/strategies to handle such uncertain scenarios was motivated by a large number of applications in various fields such as computer science, business, engineering, etc.

Decision theory offers two main approaches for handling conditions of uncertainty. The first exploits criteria of choice developed in a broader context by game theory [3, 7, 16, 17], for example the *min-max* strategy, where an alternative is chosen such that the worst possible consequence of the chosen alternative is utilized. The second approach is to model uncertainty by using subjective probabilities, based on analysis of previous decisions made in similar circumstances. Utility theory [4] helps in understanding the value of a choice. There are three traditions in utility theory. One attempts to describe people's utility functions and is called the descriptive approach. Another attempts to use utility in the construction of a rational model of decision making and is called the normative approach. The third attempts to bridge the descriptive and normative approaches by considering the limitations people have with the normative goal they would like to reach; this is

called the prescriptive approach. Our research is aimed at constructing rational models for decision making, or in other words, we focus on developing new and improved normative approaches (how humans should take decisions) for situations where decisions have to be made under conditions of uncertainty. The decisions made using these rational models would be more reliable in a given scenario since they would be the most rational of all the decisions that could be made. *Interactive Partially Observable Markov Decision Processes (I-POMDPs)* [20] provide a framework for planning in multi-agent settings in complex problem domains with *partially observable* (uncertain) environments that include either *cooperative* or *competitive* participating agents. The domains have very few restrictions as opposed to other approaches that restrict their problem domains, in part, to reduce complexity (Decentralized POMDPs [21, 39, 43] work in cooperative multi-agent settings only). However, as expected, these benefits come with a cost; they involve complex time consuming computations for arriving at a solution. *Interactive dynamic influence diagrams (I-DIDs)* [34] are the graphical counterparts of I-POMDPs. Hence, their computational complexity is comparable to that of I-POMDPs. However, since they offer an intuitive way to not only identify but also *display* the essential elements, including decisions, uncertainties, and objectives, and how they influence each other, they represent a more intuitive framework to model the decision problem.

The purpose of this thesis is to develop a new approximation technique for solving I-DIDs in order to improve the quality of the solution and make it more scalable in terms of the number of horizons (span of time ahead in the planning sequence) it can plan for.

Generally, the quality of the solution or the limit on scalability is influenced by the curses of history and dimensionality (which are explained in detail later in this chapter). There exists an infinite number of models in the model space of the other agent, some of which predict identical behaviors for the subject agent. Hence, the model space can be losslessly reduced considerably by replacing such models, termed as *behaviorally equivalent* models [37], by a representative model in an attempt to mitigate the curse. In this work, we aim to further reduce this model space by additionally pruning models that are approximately *subjectively equivalent*. To facilitate this, we first define subjective equivalence as a group of models of the other agent that induce a similar distri-

bution over the subject agent's future *action-observation* paths. Using this definition, we introduce the notion of ϵ-subjective equivalence as the group of candidate models that induce distributions over the paths, which are within $\epsilon \geq 0$ apart. Intuitively, this will result in fewer number of equivalence classes than behavioral equivalence. If we pick a single model as the representative for each class, we will end up with fewer number of models than the approaches that use exact behavioral equivalence.

Our algorithm begins by selecting a model at random from the other agents' model space and grouping together all the models that are ϵ-subjectively equivalent with it. This process is repeated until all the models have been grouped. The models that were picked (the representative models) are retained in the model set and the rest are pruned after their probability masses have been transferred to the representatives. Our new definition is such that it allows us to measure the degree of equivalence. Hence if $\epsilon = 0$, our approach identifies exact subjective equivalence and the model set contains only subjectively distinct models and as we increase ϵ, the degree of approximation increases. Our approach provides a unique opportunity to bound the error that arises in the optimality of the solution of the subject agent. We also experimentally evaluate our approach on I-DIDs formulated for benchmark problem domains and show significant qualitative improvement. However, this improvement comes with the cost of increased time complexity of computing ϵ-subjective equivalence of models. *Chapter 5* will provide a more in-depth discussion of the proposed algorithm.

1.1 Relevance to Artificial Intelligence

Artificial Intelligence is the field that strives to program software agents that exhibit intelligence. The word *artificial* means something that can be built and the word *intelligence* describes a property of the mind that encompasses many abilities, such as the capacities to reason, to plan, to solve problems, to think abstractly, and to comprehend ideas. Thus, in order to create an AI agent, we end up with four possible goals [38]:

1. Systems that *think like humans* also known as *cognitive modeling* which focuses on reasoning like humans and the human framework.

2. Systems that *think rationally* or in other words systems that are governed by the *laws of thought* which focuses on reasoning and a general concept of intelligence.

3. Systems that *act like humans* or in other words systems that pass the *Turing test* [45] where the focus is on behavior of humans and the human framework.

4. Systems that *act rationally*, also called *rational agents* that focus on behavior and a general concept of intelligence.

Our research caters to the fourth goal of AI stated above. *Rationality* is an idealized concept of intelligence, which means "doing the right thing". We will only deal with creating algorithms for modeling intelligent *software* agents. For convenience sake, throughout this paper, we will refer to *intelligent software agents* as just *agents* or *intelligent agents* unless it is mentioned otherwise.

1.2 Intelligent Agents

An *intelligent agent* is an entity that *observes* its environment through sensors and *acts* intelligently upon that environment through actuators. A human agent has eyes, ears and other organs for sensors, and mouth, hands, legs and other body parts for actuators. Similarly, a software agent receives keyboard inputs, and files as sensory input and acts on the environment by displaying the output on the screen or writing files. A rational agent selects an action that maximizes its performance measure given all the information it has regarding the nature of the environment and the percepts it receives from it. These environments are characterized along several dimensions – They can be fully or partially observable, deterministic or stochastic, episodic or sequential, static or dynamic, discrete or continuous, and single-agent or multi-agent. Environments that are fully observable, deterministic, and static are less common in nature when compared to those that are partially observable (allow for uncertainty in observations), and dynamic in the real world. Thus, for correct modeling of many real world problems, the method to use must account for possible

actions with stochastic effects and for noisy measurements. When the environment exhibits these properties, the planning task becomes a non-trivial problem. Solving these problems is a complex, and time-consuming procedure. Hence, the need for better and efficient algorithms to solve them becomes prominent.

1.3 Rational Decision Making

The decision-making process is similar to a problem solving process which is often time consuming, and context dependent. For example, consider the problem of a robot navigating in a large office building. The robot can move from hallway intersection to intersection and can make local observations of its world. Its actions are not completely reliable, however. Sometimes when it intends to move, it stays where it is, or goes too far; sometimes when it intends to turn, it overshoots. It has similar problems with the observations it makes. The point here is that, machines that are autonomous are not completely reliable because of various factors like sensor malfunction, power shut down, or even lack of adequate data or information regarding the environment it is in. Hence, these agents are faced with the problem of partially observable environments [38]. Hence, accurate analysis of the environment and rational decision-making become extremely difficult and it is interesting to see how these agents handle such scenarios.

So researchers were faced with their next challenge in the rational decision making process; making the agent understand what a good or a bad decision is. They came up with a solution. If an agent was going to make decisions by itself, it required some metric that it could use to differentiate good and bad choices. This was another spot of bother because even humans often find it difficult to articulate the difference between good and bad choices. Nevertheless, the researchers had to articulate these differences in order to provide the agents with options to choose from. Hence, they assumed that each state had an associated reward for performing each possible action or a decision choice in that state. Rewards are a way of assigning values to different states of the environment. Given these values, the agent attempts to make the decision that it knows has a greater expected value.

The next problem encountered by researchers; what if planning had to be done for the future? For the sake of convenience, time was assumed to pass in discrete increments and the agent had to choose some action to perform at each tick of the clock (it could also choose to do nothing). Say, planning had to be done for two time steps in advance. Decisions had to be made taking into consideration factors like the future and expected rewards. In order to better understand why it is important, consider the following scenario.

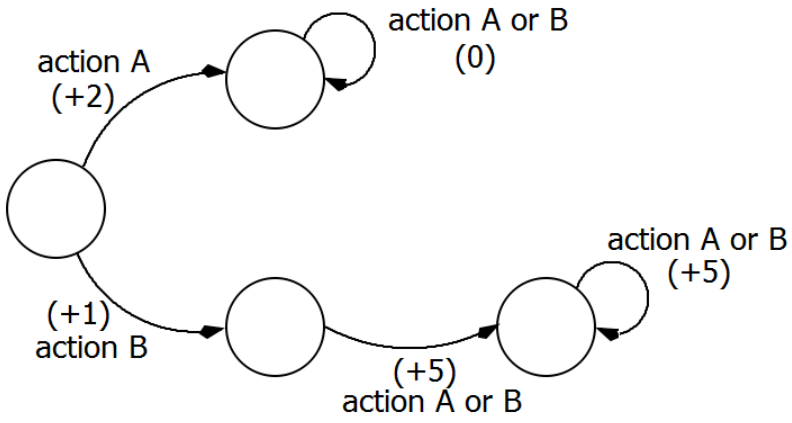


Figure 1.1: Sample environment to show why one step greedy strategy is undesirable

As it can be seen from the above figure, if the agent had chosen to perform the action A (higher immediate reward), a one step greedy strategy, it would not have ended up with a reward as high as it would have had it chosen to consider two time steps in advance and made its decision to go with action B. So this situation shows an example in which the agent would probably want to take into account the rewards it might receive in the future, and not just immediate rewards.

The next challenge for researchers was to tackle the problem when the agent had infinitely many sequential decisions to make. Hence, Puterman et. al. formalized this as the infinite horizon problem [25, 35]. Finite and infinite horizon problems are mentioned in greater detail in [25].

Formally, a model can be created for an agent consisting of a finite set of states, a finite set of actions and a reward structure defined for each action-state pair. The set of states are the different locations in which the agent can be in the environment, the set of actions are the things that the agent can do, and the reward structure for each action-state pair is the agent's desirability for being in the particular state after performing a particular action. For the robot navigation problem, the

states can be viewed as the location of the robot in the environment. The actions are the things that it can do such as move forward, move left, move right, and move backwards, and associated with each action is an immediate reward for being in a particular state. For example, if there was a pit directly in front of the robot and the robot did not know how to climb out of the pit, then the reward for moving forward would be less compared to a safe area within its reach. However, the real difficulties lie in precisely that; making machines act/think rationally.

1.4 MARKOV DECISION PROCESSES

In a *Markov decision process* (MDP) model [35, 38], the agent knows its current state (fully observable environment). Markov decision processes (MDPs) provide a framework to optimize the action sequence of the modeled agent under these environments. A Markov decision process is defined by a tuple $<S, A, T, R>$, where S is the set of the states in the planning problem; A is the set of possible actions of the agent; T is the transition function that specifies the probability to rerach state s' from state s given action a where, $\{s, s'\} \in S$ and $a \in A$; and R is the reward function that specifies the reward the agent gets for performing action a when the world is in the s state. It is important to understand that while MDP solution techniques are able to solve large state space problems, the assumptions of classical planning (mainly the full observability assumption) make them unsuitable for most complex real world applications.

However, if a participating agent cannot directly observe the underlying environmental state but instead, infers a distribution over the state based on a model of the world and some local observations, or in other words, if the environment is partially observable, then such a model is known as *Partially Observable Markov Decision Process* (POMDP) [2, 6, 8, 15, 23, 28, 33]. POMDP is a generalization of the Markov decision process. The POMDP framework is general enough to model a variety of real-world sequential decision processes. A POMDP is a belief-state MDP; we have a set of states, a set of actions, transitions and immediate rewards. The actions' effects on the state in a POMDP is exactly the same as in an MDP. The only difference is in whether or not we can observe the current state of the process. In a POMDP we add a set of observations to the

model. So instead of directly observing the current state, the state gives us an observation which provides a hint about what state the agent is in. The observations can be probabilistic; so we also specify an observation function. This observation function simply tells us the probability of each observation for each state in the model. We can also have the observation likelihood depend on the action. Formally, a POMDP is defined by a tuple $<S, A, \Omega, T, O, R>$ where S is a finite set of states, A is a finite set of actions, Ω is a finite set of observations, T is the transition function that specifies the probabilities to go from state s to state s' given action a, where, $s, s' \in S$ and $a \in A$; O is the observation function and R is the reward function, that specifies the reward the agent gets for performing action a when the world is in the s state. POMDPs, when generalized to multi-agent settings [25, 41] by including other agents' computable models in the state space along with the physical environment, are known as *Interactive partially observable Markov decision processes* (I-POMDP) [5, 10, 20]. They provide a framework for sequential decision making in partially observable multi-agent environments. This framework will be discussed in *Chapter 2*.

1.5 Graphical Models

An *influence diagram* (ID) [24, 40, 31] is a simple visual representation of a decision problem. Influence diagrams offer an intuitive way to identify and display the essential elements, including decisions, uncertainty, and objectives, and how they influence each other. Solving an ID unrolled over many time slices is called a *Dynamic ID* (DID). DIDs may be viewed as structural representations of POMDPs.

Interactive dynamic influence diagrams (I-DID) [14, 34] are graphical counterparts of interactive POMDPs (I-POMDPs) [20]. I-DIDs are consise in their representation of the problem of how an agent should act in uncertain multi-agent environments. They generalize DIDs [44], which are graphical representations of POMDPs, to multi-agent settings analogously to how I-POMDPs generalize POMDPs. These graphical models will be explained in greater detail in the *Chapter 2*.

1.6 CURSES OF DIMENSIONALITY AND HISTORY

The curse of dimensionality is the problem caused by increase in size of the state space due to the exponential increase in the number of models of the other agent, over time. This results in an increase in the number of dimensions of the belief simplex. Since there exists limitations in the CPU speed and memory available to us, it leads to large computational costs in terms of the time needed to solve each of these models in the model space. This is further complicatedif other agents are modeling other as well (nested modeling). Additionally, in order to properly model the other agents, agents keep track of the evolution of the models over time. Since, the number of models increases exponentially over time, these frameworks suffer from the curse of history. Factors contributing to these curses are enumerated below.

- The initial number of *candidate models* for the other agents: The greater the initial models considered, better are the chances of finding the exact model of the other agent and greater the computational cost as more models have to be solved. This problem contributes to the curse of dimensionality.

- The number of *horizons* (look ahead steps): At time step t, there could be $|\mathcal{M}_j^0|(|A_j||\Omega_j|)^t$ many models of the other agent j, where $|\mathcal{M}_j^0|$ is the number of models considered initially, $|A_j|$ is the number of possible actions for j, and $|\Omega_j|$ is the number of possible observations for j. As it can be seen, the number of models that have to be solved increase exponentially with increase in the number of horizon considered (t).

- The number of *strategy levels* (nested modeling): Nested modeling further contributes to the curse of dimensionality and hence to the complexity because the solution of each of the models at level $l-1$ requires solving the lower level $l-2$ models and so on recursively up to *level* 0.

Hence, good techniques that mitigate these curses to the greatest extent possible will enable a wider range of applications in larger problem domains. Our approach will introduce another factor

contributing to the curse of dimensionality. This factor comes as a cost while attempting to further reduce the size of the model space. We will discuss this issue in greater detail in the later chapters.

1.7 Claims and Contributions

In the previous section we provided some basic concepts that underlie the study of multi-agent decision making. This section enumerates our claims and contributions to the field.

- The primary focus of this thesis is the development of an approximate solution for interactive dynamic influence diagrams that helps in improving the quality of the solution.
- Algorithms for solving I-DIDs face the challenge of an exponentially growing space of candidate models ascribed to other agents, over time. Previous methods pruned the behaviorally equivalent models to identify the minimal model set. We mitigate the curse of dimensionality by further reducing the candidate model space by additionally pruning models that are approximately subjectively equivalent and replacing them with representatives.
- We define subjective equivalence in terms of the distribution over the subject agent's future action-observation paths. While rigorous, it has the additional advantage that it permits us to measure the degree to which the candidate models of the other agent are subjectively equivalent. We use symmetric Kullback Leibler (KL) divergence as the metric to measure this degree.
- We introduce the notion of ϵ-subjective equivalence as a way to approximate subjective equivalence.
- We also propose that our ϵ-subjective equivalence approach results in at most one model for each equivalence class after pruning which results in better solutions given the number of models ascribed and quality when compared to the *model clustering* approach by Zeng et al. [46] and other exact algorithms that utilize the behavioral equivalence approach.

- We theoretically analyze the error introduced by this approach in the optimality of the subject agent's solution and also discuss its advantages over the *model clustering* approach.

- We empirically evaluate the performance of our approximation technique on benchmark problem domains such as the multi-agent tiger problem and the multi-agent machine maintenance problem and compare the results with previous exact and approximation techniques including the *discriminative model update* approach by Doshi et al. [12]. We show significant improvement in performance, although with limitations.

1.8 Structure of this Work

Due to the nature of this research topic, it is necessary to perform a large literature review to get a hold of the issues and facts about the sequential decision problems that are solved using I-DIDs. It is therefore necessary to present a significant amount of background information to the reader so that the foundation is laid and an understanding of the key issues involving this research are easier to acquire. We thus, outline the structure of this thesis as follows in order to have a proper flow in understanding.

In *this chapter*, the focus is to give a very broad idea of the context of the research area, introduce a few general concepts, and give a basic outline of our contributions to the field.

In *Chapter 2*, we briefly review the framework of finitely nested Interactive POMDPs which provides the mathematical foundations for graphical models formalized by influence diagrams applied to multiagent settings. We will also introduce the readers to IDs and dynamic IDs which can be viewed as structurd representations for POMDPs. We will also provide a detailed description of Interactive IDs and their extensions to dynamic settings - I-DIDs. Exact algorithms to solve I-DIDs will also be discussed in detail.

In *Chapter 3*, we survey different implementations of I-DIDs and review their pros and cons, keeping in mind that some of these previous approaches, both exact and approximate, may be applicable in our proposed method. We introduce the readers to the initial concept of behavioral

equivalence and discuss why its definition makes it difficult to define an approximate BE measure and also discuss exact and approximate algorithms developed for solving I-DIDs in the past.

In *Chapter 4*, we define subjective equivalence in terms of the distribution over future action-observation paths. In addition to being rigorous, the definition of subjective equivalence has the additional advantage of providing a way to measure the degree to which the models are subjectively equivalent. We also derive an equation that computes the distribution of the future action-observation paths which lays the foundation of our proposed approximation technique.

In *Chapter 5*, we define the notion of ϵ-subjective equivalence, and introduce our new and improved approximation technique.

In *Chapter 6*, we provide a detailed description of the problem domains in which our technique was applied. The reward, observation, and transition functions for each of these application domains will be presented. Also, we illustratively show how I-DIDs were applied in these problem domains.

In *Chapter 7*, we present empirical evaluations of the proposed method. We take the two problems from the literature; the multiagent tiger problem, and the multiagent machine maintenance problem and perform simulations to measure the time needed to achieve different levels of performance and their average rewards. We compare our results with the other exact and approximation methods available for solving I-DIDs.

In *Chapter 8*, we mention the computational advantages due to our proposed approximation technique and also attempt to bound the error due to the approximation. We also theoretically analyze our method's savings with respect to the model clustering approximation technique.

In *Chapter 9*, we summarize our contributions, claims and results from the theoretical and experimental evaluations and also provide some ideas to further improve on our approximation method for solving I-DIDs.

CHAPTER 2

BACKGROUND

Interactive POMDPs [20] generalize POMDPs and provide a mathematical framework for solving sequential decision problems in multi-agent settings. They lay the foundation for graphical models which visually represent the decision problem. These graphical models are formalized by *influence diagrams* (IDs) [24]. In this chapter we will briefly review the I-POMDP framework. *Influence Diagrams* and *Dynamic Influence Diagrams* (DIDs) will also be discussed in some detail. We will also provide a detailed description of *Interactive Influence Diagrams* (I-IDs) and their extension to dynamic settings - *Interactive Dynamic Influence Diagrams* (I-DIDs) and methods to solve them. Just as DIDs can viewed as the structured counterparts for POMDPs, I-DIDs can be viewed as the structured counterparts for I-POMDPs.

2.1 INTERACTIVE POMDP (I-POMDP) FRAMEWORK

In Chapter 1, we introduced POMDPs as a framework to solve sequential decision problems where the subject agent is assumed to act alone in the environment. However, the real world consists of many scenarios where the agent may not be alone. It must interact not only with the environment, but also with other agents. These other agents could be either cooperating or competing with the subject agent. They could also just be neutral in their approach to achieve a particular task. All the different combinations of the information about the agents such as their beliefs, capabilities, and preferences are represented as models of the agent. So each agent has beliefs about not only the environment but also the other agent's models and their respective beliefs. All this information is included in the state space - called *the interactive state space*.

For the sake of simplicity, I-POMDPs are usually presented assuming *intentional* agents, similar to those used in Bayesian games [22, 29, 32] though the framework can be extended to any kind of model. Also, we will consider just two agents - *i*, and *j* interacting in a common environment. All results can be scaled to three or more agents.

Mathematically, the interaction can be formalized using the I-POMDP framework as follows.

Definition 1 (I-POMDP$_{i,l}$)**.** *A finitely nested I-POMDP of agent i with a strategy level l is*

$$\textit{I-POMDP}_{i,l} = < IS_{i,l}, A, T_i, \Omega_i, O_i, R_i >$$

where:

1. $IS_{i,l}$ is a set of interactive states defined as, $IS_{i,l}$ = S $\times$ $M_{j,l-1}$, where $M_{j,l-1}$ = $\Theta_{j,l-1} \cup SM_j$, for $l \geq 1$, and $IS_{i,0}$ = S, where S is the set of states of the physical environment. $\Theta_{j,l-1}$ is the set of *computable intentional models* of agent *j* . The remaining set of models, SM_j, is the set of *subintentional models* of *j* ;

2. *A* = $A_i \times A_j$, is the set of joint actions of all agents in the environment;

3. Given the *Model Non-Manipulability Assumption (MNM)* that an agent's actions do not change other agents' model directly, T_i is a transition function, T_i : S $\times$ A $\times$ S $\rightarrow$ [0, 1]. It reflects the possibly uncertain effects of the joint actions on the physical states of the environment;

4. Ω_i is the set of observations of agent *i* ;

5. Given the *Model Non-Observability Assumption (MNO)* that an agent cannot observe other agents' model directly, O_i is an observation function, O_i : S $\times$ A $\times$ $\Omega_i \rightarrow$ [0, 1]. It describes how likely it is for agent *i* to receive the observations given the physical state and joint actions;

6. R_i is a reward function, R_i : $IS_i \times$ A $\rightarrow \Re$. It describes agent *i*'s preferences over its interactive states and joint actions, though usually only the physical states and actions matter.

Intentional models ascribe to the other agent beliefs, preferences and rationality in action selection and are analogous to types as used in game theory [7, 17]. Each intentional model, $\theta_{j,l-1}$ = $< b_{j,l-1}, \hat{\theta}_j >$, where $b_{j,l-1}$ is agent *j* 's belief at level *l* - 1, and the frame, $\hat{\theta}_j$ = <*A*, T_j, Ω_j, O_j, R_j, OC_j>. Here, *j* is assumed Bayes rational and OC_j is *j* 's optimality criterion. A subintentional model is a triple, sm_j = $< h_j, O_j, f_j >$, 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 gives the probability with which *j* receives its input. We refer the reader to [20] for details regarding the belief update and the value iteration in I-POMDPs. In this thesis, we restrict our attention to intentional models only.

2.2 Influence Diagrams (IDs)

In this section we briefly describe influence diagrams (IDs) followed by their extensions to dynamic settings, DIDs, and refer the reader to [9, 24] for more details. An *influence diagram* (ID) (also called a decision network) is a compact graphical and mathematical representation of a decision problem. It is a generalization of a Bayesian network, in which both probabilistic inference problems and decision making problems can be modeled and solved. An influence diagram can be used to visualize the probabilistic dependencies in a decision analysis and to specify the states of information for which independencies exist. IDs are the graphical counterparts of POMDPs. Their graphical representation of the problem enables ease of use and provides an edge over their non-graphical counterparts. The first complete algorithm for evaluating an influence diagram was developed by Shachter in 1986 [40].

2.2.1 Syntax

An ID has three types of nodes and three types of arcs (or arrow) between these nodes. See the Fig. 2.1 below. We observe that an ID augments a Bayesian network with decision and utility nodes.

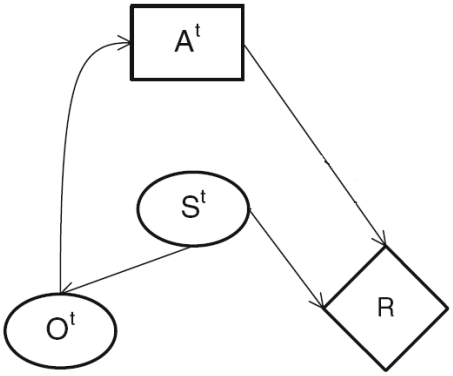


Figure 2.1: A simple influence diagram (ID) representing the decision-making problem of an agent. The oval nodes representing the state (S) and the observation (Ω) reflected in the observation function, O, are the chance nodes. The rectangle is the decision node (A) and the diamond is the reward/utility function (R). Influences (links) connect nodes and represent the relationship between nodes.

TYPES OF NODES

1. *Decision node* (corresponding to each decision to be made) is drawn as a rectangle. It represents points where the decision making agent has a choice of actions.

2. *Chance node* (corresponding to uncertainty to be modeled) is drawn as an oval. These represent random variables, just as they do in Bayes nets. The agent could be uncertain about various things because of the partial observability faced in real world problems. Each chance node has a conditional distribution associated with it that is indexed by the state of the parent nodes.

3. *Utility node* (corresponding to a utility function) is drawn as a diamond (or an octagon). The utility node has all the variables that directly affect the utility, as parents. This description could be just a tabulation of the function or a mathematical function.

TYPES OF ARCS/ARROWS

1. *Functional arcs* (ending in utility node) indicate that one of the components of additively separable utility function is a function of all the nodes at their tails.

2. *Conditional arcs* (ending in chance node) indicate that the uncertainty at their heads is probabilistically conditioned on all the nodes at their tails.

3. *Informational arcs* (ending in decision node) indicate that the decision at their heads is made with the outcome of all the nodes at their tails known beforehand.

2.2.2 EVALUATING INFLUENCE DIAGRAMS

The solution of the influence diagram is the action that is chosen to be performed for each possible setting. This decision is made in the decision node. Once the decision node is set, it behaves just like a chance node that has been set as an evidence variable. The algorithm outline for evaluating the influence diagram is as follows.

1. Set the evidence in the variables for the current state.

2. For each possible value of the decision node;

   (a) Set the decision node to that value.

   (b) Calculate the posterior probabilities for the parent nodes of the utility node, using a standard probabilistic inference algorithm.

   (c) Calculate the resulting utility for the action.

3. Return the action with the highest utility.

2.3 DYNAMIC INFLUENCE DIAGRAMS (DIDS)

IDs can be extended to dynamic settings by unrolling them over as many time slices as the number of horizon. These are known as Dynamic Influence Diagrams (DIDs) [38] shown in Fig. 2.2.

Solving DIDs is similar to solving IDs except now we will have multiple conditional sequences of actions each associated with a value of performing the respective sequence, with the best sequence being the one with the largest value. Dynamic IDs provide a concise and structured representation for large POMDPs [38] expanded over multiple time slices. Hence they can also be used as inputs for any POMDP algorithm.

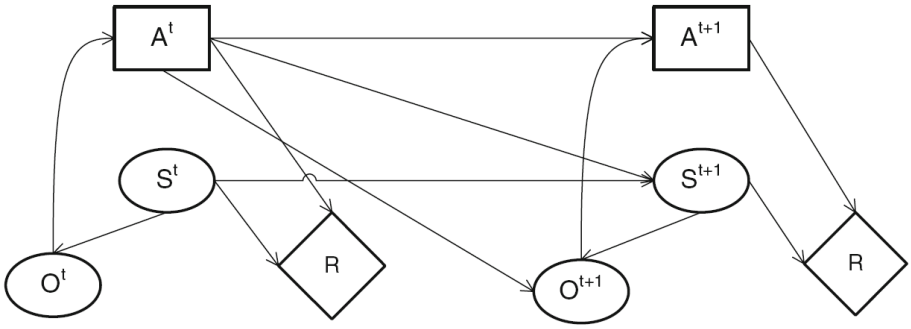


Figure 2.2: A two time-slice/horizon dynamic influence diagram (DID) representing the decision-making problem of an agent. Here, the influences (links) connect nodes not only within the same time slice but nodes across time slices as well.

The nodes in a DID, like the one in Fig. 2.2, correspond to the elements of a POMDP. That is, the values of the decision node A^t, correspond to the set of actions, A, in a POMDP. The values of the chance nodes, S^t and O^t, correspond to the sets of states and observations, respectively, in a POMDP. The conditional probability distribution (CPD), Pr($S^{t+1}|S^t$, A^t), of the chance node, S^{t+1}, is analogous to the transition function, *T* in a POMDP. The CPD, Pr($O^{t+1}|S^{t+1}$, A^t), of the chance node, O^{t+1}, is analogous to the observation function, O, and the utility table of the utility node, *U*, is analogous to the reward function, *R*, in a POMDP. The links in DIDs also known as influence links connect nodes not only within the same time slice but also across different time slices as well indicating causal relationships not only within the same time slice but also between time slices.

DIDs perform planning using a forward exploration technique. This technique explores the possible states of belief an agent may have in the future, the likelihood of reaching each state of belief, and the expected utility of each belief state. The agent then adopts the plan which maximizes

the expected utility. DIDs provide exact solutions for finite horizon POMDP problems, and finite look-ahead approximations for POMDPs of infinite horizon.

2.4 INTERACTIVE INFLUENCE DIAGRAMS (I-IDS)

Interactive Influence Diagrams (I-IDs) [13] generalize IDs [44] to make them applicable to settings shared with other agents, who may act, observe and update their beliefs. In this section, we describe I-IDs for modeling specifically two-agent interactions. I-IDs are graphical representations of decision making in uncertain multi-agent environments. In this framework, agents are represented using chance nodes and their actions are controlled using a static probability distribution. Any real world scenario in which the agents are interacting may be decomposed into chance and decision variables, and the dependencies between the variables. I-IDs ascribe procedural models to other agents: these may be IDs, Bayesian networks (BNs), or I-IDs themselves leading to recursive modeling. As agents act and make observations, beliefs over others models are updated. With the implicit assumption that the true model of other is contained in the model space, I-IDs use Bayesian learning to update beliefs, which gradually converge.

2.4.1 SYNTAX

In addition to the usual chance, decision, and utility nodes, I-IDs include a new type of node called the *model node*. We show a general level *l* I-ID in Fig. 2.3(a), where the model node $(M_{j,l-1})$ is denoted using a hexagon. We note that the probability distribution over the chance node, S, and the model node together represents agent i's belief over its *interactive state space*. In addition to the model node, I-IDs differ from IDs by having a chance node, A_j, that represents the distribution over the other agent's actions, and a dashed link, called a *policy link* between the model node and the chance node, A_j. In the absence of other agents, the model node and the chance node, A_j , vanish and I-IDs collapse into traditional IDs.

The model node consists of the decisions made by the different models ascribed by i to the other agent. Each model in the model node may itself be an I-ID or an ID giving rise to recursive

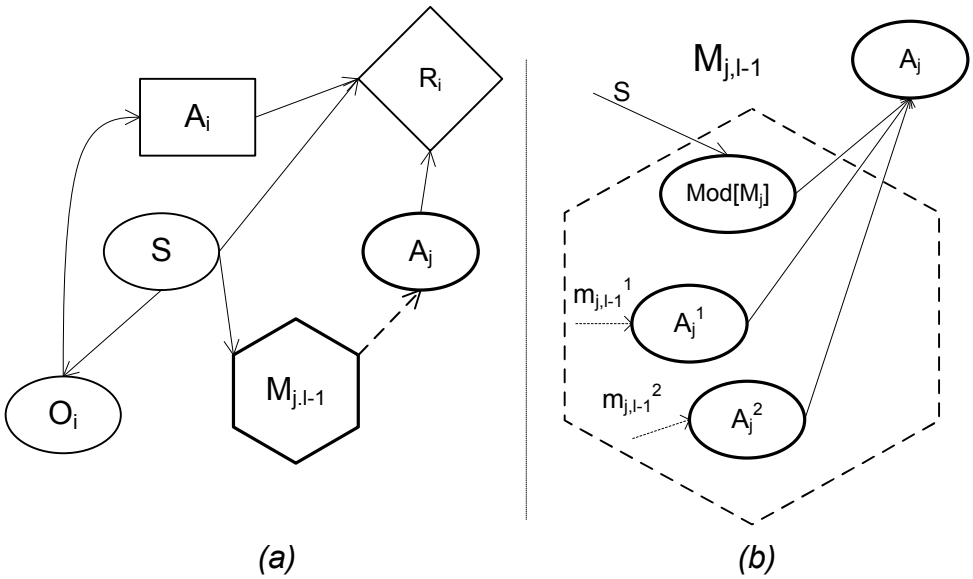


Figure 2.3: (a) Level $l > 0$ I-ID for agent i sharing the environment with one other agent j. The hexagon is the model node ($M_{j,l-1}$) and the dashed arrow is the policy link. (b) Representing the model node and policy link using chance nodes and causal relationships. The decision nodes of the lower-level I-IDs or IDs ($m^1_{j,l-1}$, $m^2_{j,l-1}$) are mapped to the corresponding chance nodes (A^1_j, A^2_j), which is indicated by the dotted arrows. Depending on the value of node, $Mod[M_j]$, distribution of each of the chance nodes is assigned to node A_j with some probability.

modeling. This recursion ends when a model is an ID. Formally, we denote a model of j as, $m_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_j \rangle$, where $b_{j,l-1}$ is the level $l-1$ belief, and $\hat{\theta}_j$ is the agent's *frame* consisting of action, observation and utility nodes. Because the model node contains the alternative models of the other agent as its values, its representation is not simple. In particular, some of the models within the node are I-IDs that when solved generate the agents optimal policy in their decision nodes. Each decision node is mapped to the corresponding chance node, say A^1_j , in the following way: if OPT is the set of optimal actions obtained by solving the I-ID (or ID), then $Pr(a_j \in A^1_j) = \frac{1}{|OPT|}$ if $a_j \in OPT$, 0 otherwise.

The dashed policy link between the model node and the chance node A_j can be represented as shown in Fig. 2.3(b). The decision node of each level $l-1$ I-ID is transformed into a chance node as we mentioned previously, so that the actions with the largest value in the decision node

are assigned uniform probability in the chance node while the rest are assigned zero probability. Each of the alternate models of the other agent can be represented as chance nodes A_j^1, A_j^2, one for each model. The chance node labeled $Mod[M_j]$ forms the parents of the chance node A_j. Thus, there are as many action nodes (A_j^1, A_j^2) in $M_{j,l-1}$ as the number of alternative models of the other agent. Each of these models is denoted by the states of the $Mod[M_j]$ node. The distribution over $Mod[M_j]$ is *i*'s belief over *j*'s candidate models (model weights) given the physical state *S*. The conditional probability table (CPT) of the chance node, A_j, is a ***multiplexer***, that assumes the distribution of each of the action nodes (A_j^1, A_j^2) depending on the value of $Mod[M_j]$. In other words, when $Mod[M_j]$ has the value $m_{j,l-1}^1$, the chance node A_j assumes the distribution of the node A_j^1, and A_j assumes the distribution of A_j^2 when $Mod[M_j]$ has the value $m_{j,l-1}^2$. Note that in Fig. 2.3(b), the dashed policy link can be replaced using traditional dependency links.

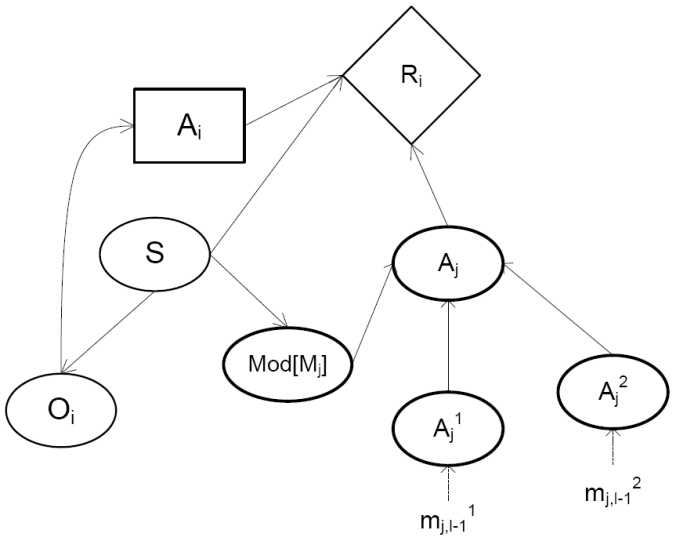


Figure 2.4: The transformed I-ID with the model node replaced by the chance nodes and the relationships between them.

In Fig. 2.4, we show the transformed I-ID when the model node is replaced by the chance nodes and relationships between them. In contrast to the representation in Fig. 2.3(a) , there are no special-purpose policy links, rather the I-ID is composed of only those types of nodes that are found in traditional IDs and dependency relationships between the nodes.

2.4.2 SOLUTION

Solution of an I-ID proceeds in a bottom-up manner, and is implemented recursively.

1. Solve the lower level models, which are traditional IDs or BNs. Their solutions provide probability distributions over the other agents actions, which are entered in the corresponding chance nodes found in the model node of the I-ID.

2. The mapping from the level 0 models decision nodes to the chance nodes is carried out so that actions with the largest value in the decision node are assigned uniform probabilities in the chance node while the rest are assigned zero probability.

3. Given the distributions over the actions within the different chance nodes (one for each model of the other agent), the I-ID is transformed into a traditional ID.

4. During the transformation, the CPT of the node, A_j , is populated such that the node assumes the distribution of each of the chance nodes depending on the state of the node, $Mod[M_j]$.

5. The transformed I-ID is a traditional ID that may be solved using the standard expected utility maximization method [12].

6. This procedure is carried out up to the level l I-ID whose solution gives the non-empty set of optimal actions that the agent should perform given its belief. Notice that analogous to IDs, I-IDs are suitable for online decision-making when the agents current belief is known.

2.5 INTERACTIVE DYNAMIC INFLUENCE DIAGRAMS (I-DIDS)

In this section, we describe the interactive dynamic influence diagrams (I-DIDs) for two-agent interactions which are the extensions of interactive influence diagrams to dynamic settings (multiple time slices).

I-DIDs extend I-IDs to allow sequential decision making over multiple time slices (see Fig. 2.5). Just as DIDs are the structured graphical representations of POMDPs, I-DIDs are the graphical representations for I-POMDPs.

2.5.1 SYNTAX

Fig. 2.5 shows a general two time slice I-DID. Here, in addition to the model nodes and the dashed policy link, what differentiates an I-DID from a DID is the *model update link* shown as a dotted arrow in Fig. 2.5. We explain the semantics of the model update next.

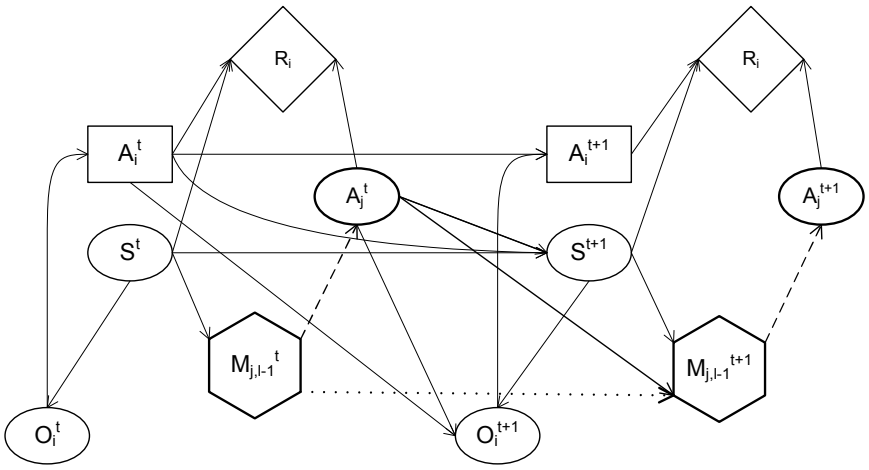
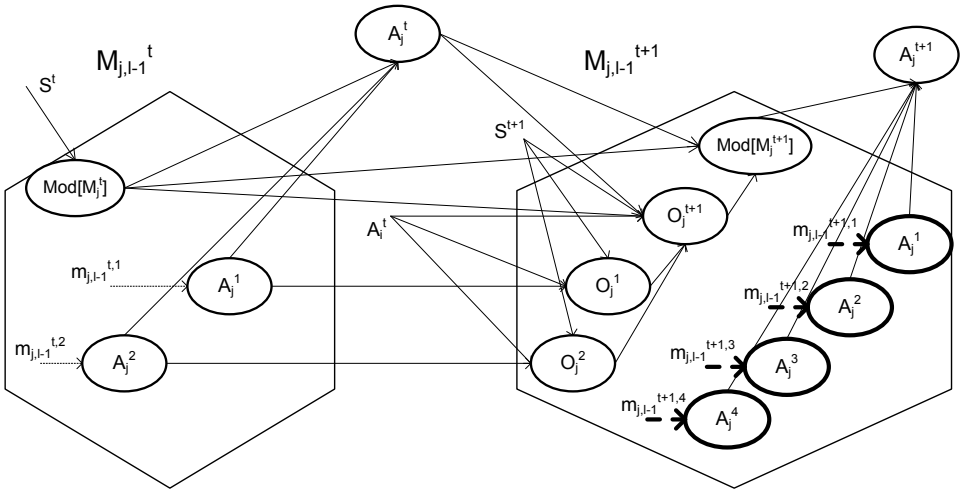


Figure 2.5: A generic two time-slice level l I-DID for agent i.

The model update link symbolically represents the update of the model node. There are two steps in the update process. First, the models need to be updated to reflect the change in beliefs that occur because the agents interact with the environment and with each other by acting on recieved observations. It can be observed that the number of models in the model node increase exponentially upon update. Since the set of optimal actions for a model could include all the actions and the agent may recieve any one of $|\Omega_j|$ possible observations, the updated set at time step $t+1$ will have up to $|\mathcal{M}^t_{j,l-1}||A_j||\Omega_j|$ models where $|\mathcal{M}^t_{j,l-1}|$ is the number of models at time step t, $|A_j|$ and $|\Omega_j|$ are the largest spaces of actions and observations respectively, among all the models. The CPT of $Mod[M^{t+1}_{j,l-1}]$ encodes the function, $\tau(b^t_{j,l-1}, a^t_j, o^{t+1}_j, b^{t+1}_{j,l-1})$ which is 1 if the belief $b^t_{j,l-1}$ in the model $m^t_{j,l-1}$ using the action a^t_j and observation o^{t+1}_j updates to $b^{t+1}_{j,l-1}$ in a model $m^{t+1}_{j,l-1}$; otherwise it is 0.

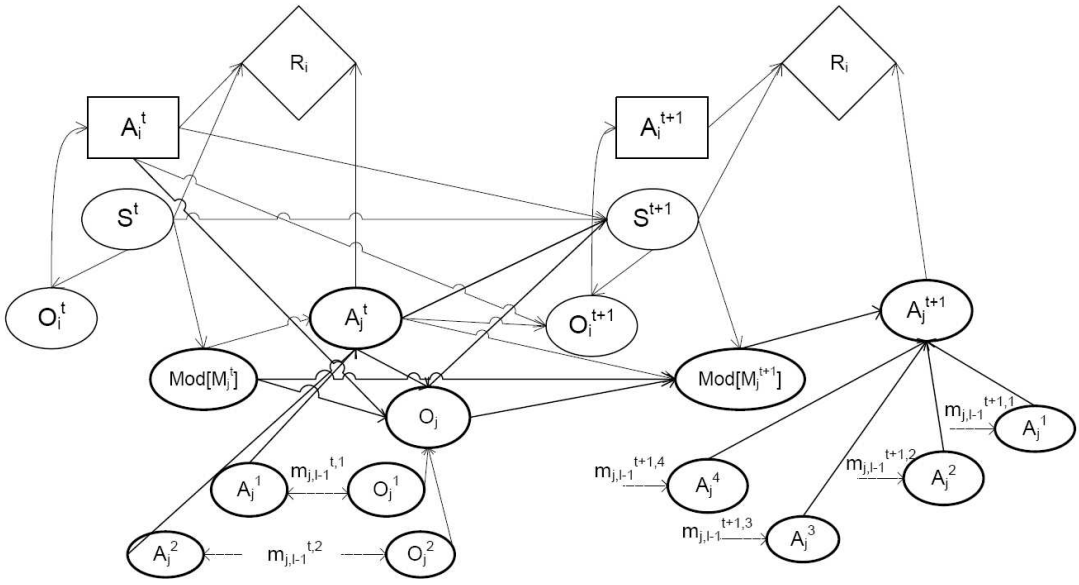
Second, the new distribution over the updated models needs to be computed, given the original distribution and the probability of the agent performing the action and receiving the observation



.

Figure 2.6: The semantics of the model update link. Notice the growth in the number of models at $t + 1$ shown in bold.

that led to the updated model. The dotted model update link in the I-DID may be replaced using standard dependency links and chance nodes, as shown in Fig. 2.6 transforming it into a flat DID.



.

Figure 2.7: Transformed I-DID with the model nodes and model update link replaced with the chance nodes and the relationships (in bold).

In order to clearly understand the model update preocess, we will use an example to show how the dotted model update link is implemented in the I-DID as in Fig. 2.6. First, let us assume two level $l-1$ models are ascribed to agent j at time step t. Suppose, they result in one action and each agent j can make one of two possible observations, then the number of models in the updated set will be four. Hence, at time step $t+1$, the model node will contain four updated models, say, ($m_{j,l-1}^{t+1,1}$, $m_{j,l-1}^{t+1,2}$, $m_{j,l-1}^{t+1,3}$, and $m_{j,l-1}^{t+1,4}$). Each of these models will have different initial beliefs because of agent j updating its beliefs due to its action and one of two possible observations. The next step is to compute the distribution over the updated set of models. In other words, the distribution over the chance node $Mod[M_j^{t+1}]$ (in $M_{j,l-1}^{t+1}$) is to be computed. The probability that j's updated model is, say $m_{j,l-1}^{t+1,1}$, depends on the probability of *j* performing the action and receiving the observation that led to this model, and the prior distribution over the models at time step *t*. Because the chance node A_j^t assumes the distribution of each of the action nodes based on the value of $Mod[M_j^t]$, the probability of the action is given by this chance node. In order to obtain the probability of j's possible observation, we introduce the chance node O_j which depending on the value of $Mod[M_j^t]$ assumes the distribution of the observation node in the lower level model denoted by $Mod[M_j^t]$. Because the probability of *j*'s observations depends on the physical state and the joint actions of both agents, the node O_j is linked with S^{t+1}, A_i^t, and A_j^t. Analogous to A_j^t, the conditional probability table of O_j is also a multiplexer modulated by $Mod[M_j^t]$. Finally, the distribution over the prior models at time *t* is obtained from the chance node, $Mod[M_j^t]$ in $Mod[M_{j,l-1}^t]$. Consequently, the chance nodes, $Mod[M_j^t]$, A_j^t, and O_j , form the parents of $Mod[M_j^{t+1}]$ in $M_{j,l-1}^{t+1}$. Notice that the model update link may be replaced by the dependency links between the chance nodes that constitute the model nodes in the two time slices. In Fig. 2.7 we show the two time-slice I-DID with the model nodes replaced by the chance nodes and the relationships between them. Chance nodes and dependency links that not in bold are standard, usually found in DIDs. Expansion of the I-DID over more time steps requires the repetition of the two steps of updating the set of models that form the values of the model node and adding the relationships between the chance nodes, as many times as there are model update links. We note that the possible set of models of the other

agent *j* grows exponentially with the number of time steps. For example, after *T* steps, there may be at most $|\mathcal{M}_{j,l-1}^{t=1}|(|A_j||\Omega_j|)^{T-1}$ candidate models residing in the model node.

SOLUTION

Analogous to I-IDs, the solution to a level *l* I-DID for agent *i* expanded over *T* time steps may be carried out recursively. For the purpose of illustration, let *l* = 1 and *T* = 2. The solution method uses the standard look-ahead technique, projecting the agents action and observation sequences forward from the current belief state [38], and finding the possible beliefs that *i* could have in the next time step. Because agent *i* has a belief over j's models as well, the lookahead includes finding out the possible models that *j* could have in the future. Consequently, each of j's level 0 models (represented using a standard DID) in the first time step must be solved to obtain its optimal set of actions. These actions are combined with the set of possible observations that *j* could make in that model, resulting in an updated set of candidate models (that include the updated beliefs) that could describe the behavior of *j* in the second time step. Beliefs over this updated set of candidate models are calculated using the standard inference methods using the dependency relationships between the model nodes as shown in Fig. 2.6. We note the recursive nature of this solution: in solving agent i's level 1 I-DID, j's level 0 DIDs must be solved. If the nesting of models is deeper, all models at all levels starting from 0 are solved in a bottom-up manner.

We briefly outline the recursive algorithm for solving agent *i*'s level *l* I-DID expanded over *T* time steps with one other agent *j* in Fig. 2.8. A two-phase approach is adopted: Given an I-ID of level l (described previously in Section 2.4) with all lower level models also represented as I-IDs or IDs (if level 0), the first step is to expand the level *l* I-ID over *T* time steps adding the dependency links and the conditional probability tables for each node. The focus is particularly on establishing and populating the model nodes (lines 3-11). In the second phase, a standard look-ahead technique is used projecting the action and observation sequences over *T* time steps in the future, and backing up the utility values of the reachable beliefs. Similar to I-IDs, the I-DIDs reduce to DIDs in the absence of other agents. As we mentioned previously, the 0-th level models are the traditional

I-DID EXACT(level $l \geq 1$ I-ID or level 0 ID, T)
Expansion Phase
1. **For** t **from** 1 **to** $T-1$ **do**
2. **If** $l \geq 1$ **then**
Populate $M^{t+1}_{j,l-1}$
3. **For each** m^t_j **in** $\mathcal{M}^t_{j,l-1}$ **do**
4. Recursively call algorithm with the $l-1$ I-ID(or ID) that represents m^t_j and the horizon, $T-t+1$
5. Map the decision node of the solved I-ID (or ID), $OPT(m^t_j)$, to the chance node A^t_j
6. **For each** a_j **in** $OPT(m^t_j)$ **do**
7. **For each** o_j **in** O_j (part of m^t_j) **do**
8. Update j's belief, $b^{t+1}_j \leftarrow SE(b^t_j, a_j, o_j)$
9. $m^{t+1}_j \leftarrow$ New I-ID (or ID) with b^{t+1}_j as belief
10. $\mathcal{M}^{t+1}_{j,l-1} \overset{\cup}{\leftarrow} \{m^{t+1}_j\}$
11. Add the model node, $M^{t+1}_{j,l-1}$, and the model update link between $M^t_{j,l-1}$ and $M^{t+1}_{j,l-1}$
12. Add the chance, decision and utility nodes for t+1 time slice and the dependency links between them
13. Establish the CPTs for each chance node and utility node

Look-Ahead Phase
14. Apply the standard look-ahead and backup method to solve the expanded I-DID (other solution approaches may also be used)

Figure 2.8: Algorithm for exactly solving a level $l \geq 1$ I-DID or level 0 DID expanded over T time steps.

DIDs. Their solutions provided probability distributions over actions of the agent modeled at that level to I-DIDs at level 1. Given probability distributions over other agents actions the level 1 I-DIDs can themselves be solved as DIDs, and provide probability distributions to yet higher level models. It is assumed that the number of models considered at each level is bound by *M*. Solving an I-DID of level *l* is then equivalent to solving $O(M^l)$ DIDs.

CHAPTER 3

RELATED WORK

Suryadi and Gmytrasiewicz [42] proposed modeling other agents by modifying IDs to better reflect the observed behavior. Unlike I-DIDs, other agents did not model the original agent and the distribution over the models was not updated over time based on the actions and observations.

Recent advancements in I-DIDs contribute to the increasing popularity of multi-agent graphical models such as Multi-agent Influence Diagrams (MAIDs) [26] and Networks of Influence Diagrams (NIDs) [18, 19] that seek to model the embedded structure in many real-world decision making problems. This is done by encoding the structure as chance and decision variables, and the dependencies between the variables. Unlike extensive forms of games, MAID games are compact and readable. They graphically represent games of imperfect information with decision nodes for each agent's actions and chance nodes for the agent's private information. Their objectivity in analysing games and efficiency in computing Nash equilibrium is aided by exploiting the conditional independence structure. NIDs extend MAIDs to include agents' uncertainty over the game being played and over models of the other agents. Both MAIDs and NIDs provide an analysis of the game from an external viewpoint, and adopt Nash equilibrium as the solution concept. MAIDs do not allow us to define a distribution over non-equilibrium behaviors of other agents. MAIDs are applicable only for single play games and in static environments. But I-DIDs address this gap by extending DIDs to multi-agent settings and therefore allowing its application in repeated games and in dynamic environments. They represent the other agents′ models as states in their model node. Other agents' models and the original agent's beliefs over these models are then updated over time. I-DIDs provide a way to exploit predicted non-equilibrium behavior.

In this chapter, we will discuss the following exact and approximation techniques in some level of detail and refer the readers to their respective papers for more information:

1. Exact algorithm to solve I-DIDs

   Using Behavioral Equivalence (BE).

2. Approximate algorithms to solve I-DIDs

   Using Model Clustering (MC).

   Using Discriminative Model Updates (DMU).

3.1 Exactly Solving I-DIDs using Behavioral Equivalence

Since the BE approach lays the foundation for our new approximation technique (ϵ-subjective equivalence), we will discuss this approach in greater detail. However, an overview of the approximation algorithms will also be presented to enable the readers to understand the need for an improved approximation method.

3.1.1 Behavioral Equivalence (BE)

In order to reduce the dimensionality of the interactive state space, it is required to reduce the number of models being solved at every time step. At the same time, doing so will reduce the optimality of the solution if the actual models of the other agents were pruned before they were solved. Hence, it is important to carefully prune models from the infinitely large model space. Some methods limit the maximum number of models they solve at each time step as a way to mitigate the impact of the history that afflicts the other modeled agent. Although the space of possible models is very large, not all models need to be considered. This is because some models in the model node of the I-DID have behavioral predictions for the other agent that are identical. These models are classified as *behaviorally equivalent* [36, 37]. Thus, all such models could be pruned and a single representative model could be considered. This is because the solution of the subject agent's I-DID

is affected by the predicted behavior of the other agent only; thus we need not distinguish between behaviorally equivalent models.

The main idea of the exact algorithm to solve I-DIDs using behavioral equivalence is to aggregate the behaviorally equivalent models into a finite number of equivalence classes and instead of reasoning over the infinite set of interactive states, we operate over the finite set of equivalence classes each having one representative model.

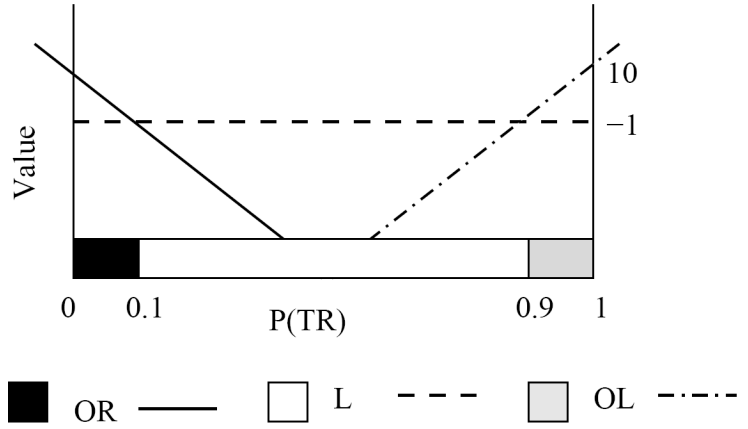


Figure 3.1: Horizon-1 value function in the tiger game and the belief ranges corresponding to different optimal actions.

In order to clearly understand the construction of behavioral equivalence classes, let us consider a simple example - the classical tiger problem introduced in [25]. According to the problem, there is an agent waiting to open one of two doors. Behind one of the doors, there is a tiger that would eat the agent that opens that door and behind the other is a pot of gold. There is a reward of +10 to get the gold and -100 when the agent is eaten by the tiger. There are two states signifying the location of the tiger - TL, when the tiger is behind the left door and TR, when the tiger is behind the right door. The agent can choose to perform one of three actions - opening the left door (OL), opening the right door (OR), and listen (L). The agent can receive two observations when it chooses to listen that will guide it to making the right decision - GL, it hears a Tiger's growl from behind the left door, and GR, it hears a growl from behind the right door each with 85% certainty. The value function gives the value of performing the optimal plan given the belief. In Fig. 3.1, we show the value function in the tiger game and the belief ranges corresponding to different optimal actions.

We note that the agent opens the right door if it believes the probability that the tiger is behind the right door is less than 0.1. It will listen if the probability is between 0.1 and 0.9 and open the left door if the probability is greater than 0.9. We observe that each optimal action spans over multiple belief points. For example, opening the right door is the optimal action for all beliefs in the set [0-0.1). Thus, the beliefs in the set [0-0.1) are equivalent in that it induces the same optimal behavior. Such beliefs are *behaviorally equivalent*. The collection of the equivalence classes forms a partition of the belief space. For finite horizons, and a finite number of actions and observations, the number of distinct optimal actions and therefore the number of equivalence classes is also finite.

Using this insight, behavioral equivalence is used to solve I-DIDs exactly by pruning the models that induced the same optimal behavior and replacing all the models in a behavioral equivalence class with one representative model. Thus, at every time step, the number of models that have to be solved is reduced to only the number of these equivalence classes. Let **BehavioralEq**($\mathcal{M}_{j,l-1}$) be the procedure that prunes the behaviorally equivalent models from $\mathcal{M}_{j,l-1}$ returning the set of representative models. The algorithm for exactly solving I-DIDs using behavioral equivalence is given below. The algorithm for solving the I-DID is the same as before, except that the updated set of models is minimized by excluding the behaviorally equivalent models (line 17).

3.2 Approximately solving I-DIDs using Model Clustering

This approach was introduced by Zeng et al. [46]. They presented a method to reduce the dimensionality of the interactive state space and mitigate the impact of the curse of history. This is done by limiting and holding a constant number of models, $0 < K << M$, where M is the possibly large number of candidate models of the other agent included in the state space. Using the insight that beliefs that are spatially close are likely to be behaviorally equivalent [37], Zeng et al. cluster the models of the other agents and select representative models from each cluster. They utilize the popular k-means clustering method, which gives an iterative way to generate the clusters. Intuitively, the clusters contain models that are likely to be behaviorally equivalent and hence may be replaced

I-DID EXACT(level $l \geq 1$ I-DID or level 0 DID, T)
Expansion Phase
1. **For** t **from** 1 **to** $T-1$ **do**
2. **If** $l \geq 1$ **then**
Populate $M^{t+1}_{j,l-1}$
3. **For each** m^t_j **in** $\mathcal{M}^t_{j,l-1}$ **do**
4. Recursively call algorithm with the $l-1$ I-DID(or DID) that represents m^t_j and the horizon, $T-t$
5. Map the decision node of the solved I-DID (or DID), $OPT(m^t_j)$, to the chance node A^t_j
6. **For each** a_j **in** $OPT(m^t_j)$ **do**
7. **For each** o_j **in** O_j (part of m^t_j) **do**
8. Update j's belief, $b^{t+1}_j \leftarrow SE(b^t_j, a_j, o_j)$
9. $m^{t+1}_j \leftarrow$ New I-DID (or DID) with b^{t+1}_j as belief
10. $\mathcal{M}^{t+1}_{j,l-1} \overset{\cup}{\leftarrow} \{m^{t+1}_j\}$
11. Add the model node, $M^{t+1}_{j,l-1}$, and the model update link between $M^t_{j,l-1}$ and $M^{t+1}_{j,l-1}$
12. Add the chance, decision and utility nodes for t+1 time slice and the dependency links between them
13. Establish the CPTs for each chance node and utility node

Solution Phase
14. **If** $l \geq 1$ **then**
15. Represent the model nodes and the model update link as in Fig. 2.6 to obtain the DID
Minimize model spaces
16. **For** t **from** 1 **to** T **do**
17. $\mathcal{M}^t_{j,l-1} \leftarrow$ **BehavioralEq**($\mathcal{M}^t_{j,l-1}$)
18. Apply the standard look-ahead and backup method to solve the expanded DID (other solution approaches may also be used)

Figure 3.2: Algorithm for exactly solving a level $l \geq 1$ I-DID or level 0 DID expanded over T time steps.

by a subset of representative models without a significant loss in the optimality of the decision maker. *K* representative models from the clusters are selected and updated over time.

3.2.1 MODEL CLUSTERING APPROACH

The approximation technique is based on clustering the agent models and selecting K, where $0 < K << M$, representative models from the clusters. In order to initiate clustering, the initial means was identified around which the models would be clustered. The selection of the initial means is crucial as we hope to select them minimally and avoid discarding models that are behaviorally distinct from the representative ones. The initial means were selected as those that lie on the intersections of the behaviorally equivalent regions (see previous section for an illustration to help understand these regions). This allows models that are likely to be behaviorally equivalent to be grouped on each side of the mean. These intersection points are called sensitivity points (SPs). In order to compute the SPs, we observe that they are the beliefs at the non-dominated intersection points (or lines) between the value functions of pairs of policy trees. A linear program (LP) shown in [46] provides a straightforward way of computing the SPs. If the intersections were lines, then the LP returned a point on this line. The initial clusters group together models of the other agent possibly belonging to multiple behaviorally equivalent regions. Additionally, some of the SPs may not be candidate models of the other agent j as believed by the subject agent i . In order to promote clusters of behaviorally equivalent models and segregate the non-behaviorally equivalent ones, the means are updated using an iterative method often utilized by the *k-means* clustering approach. This iterative technique converges because over increasing iterations less new models will be added to a cluster, thereby making the means gradually invariant. Given the stable clusters, a total of K representative models are selected from them. Depending on its population, each cluster contributes a proportion k of models to the set. The k models whose beliefs are the closest to the mean of the cluster are selected for inclusion in the set of models that are retained. Remaining models in the cluster are discarded. The selected models provide representative behaviors for the original set of models included in the cluster. The algorithm for approximately solving I-DIDs using model clustering is a slight variation of the one in Fig. 2.7 that solves I-DIDs exactly. In particular, on generating the candidate models in the model node during the expansion phase, K models are selected after clustering using the procedure *KModelSelection* explained in [46]. It can be noted that models at

all levels will be clustered and pruned. Also, this approach is more suited to situations where agent *i* has some prior knowledge about the possible models of others, thereby facilitating the clustering and selection. We refer the readers to [46] for more details on this approach.

As mentioned earlier, the insight for this approach comes from the fact that behaviorally equivalent models are spatially closer to each other than the behaviorally distinct ones. However, this approach first generates all possible models before reducing the space at each time step, and utilizes an iterative and often time-consuming k-means clustering method. Despite its favorable results when compared to the exact approaches, it can be noted that there is no way to show the degree to which models are behaviorally equivalent. Our approximation technique (ϵ-subjective equivalence), provides a definition for subjective equivalence in terms of the distribution over the future action-observation paths, that allows a way to measure the degree to which the models are subjectively equivalent. Apart from this, the *Chapter 8* contains more information that will highlight the advantages of our approach over the model clustering approach.

3.3 Approximately solving I-DIDs using Discriminative Model Updates

This approximation method was introduced by Doshi and Zeng [12]. This work is also motivated by the fact that the complexity of I-DIDs increased predominantly due to the exponential growth of candidate models, over time. Hence, they formalized a *minimal set* of models of other agents, a concept that was previously discussed in [36]. Their new approach for approximating I-DIDs significantly reduced the space of possible models of other agents that needed to be considered by discriminating between model updates. In other words, the models were discriminatively updated only if the resulting models were not behaviorally equivalent to the previously updated ones. Furthermore, in this technique, solving all the initial models was avoided. The outline of the algorithm is given below. The algorithm takes the I-DID of level l, the horizon T, and the number K of random models to be solved initially, and the threshold for euclidean distance between belief points, as input. First, K models are randomly selected from the candidate model space and solved. For each of the remaining models, if the belief of that model is close to that of one of the solved models by

atleast a threshold (supplied as input), then that model assumes the solution of the solved model. Otherwise, the model is solved. At each time step, only those models are selected for updating which will result in predictive behaviors that are distinct from others in the updated model space. In other words, models that on update resulted in predictions that are identical to those that existed were not selected for updating. For these models, their revised probability masses were transfered to the existing behaviorally equivalent models. The solutions of the solved models are then merged bottom up to obtain the policy graph. This approach improves on the previous one that uses model clustering (discussed earlier) because it does not generate all possible models prior to selection at each time step; rather it results in a minimal set of models.

We empirically compare this approach with our approximation method in terms of the average rewards obtained and results are shown in *Chapter 7*. For more details on this approach, we refer the readers to [12].

CHAPTER 4

SUBJECTIVE EQUIVALENCE

In this chapter, we provide a definition for subjective equivalence in terms of the distribution of the future action-observation paths, that allows a way to measure the degree to which the models are subjectively equivalent. We first assume that the models of the other agent j have identical frames and differ only in their beliefs. Because our technique is closely related to a previous concept - behavioral equivalence (BE), we will first define BE. We will then introduce subjective equivalence (SE) [1] and finally relate the two definitions.

As we mentioned previously, two models of the other agent are BE if they produce identical behaviors for the other agent. Formally, models $m_{j,l-1}, \hat{m}_{j,l-1} \in \mathcal{M}_{j,l-1}$ are BE if and only if $OPT(m_{j,l-1}) = OPT(\hat{m}_{j,l-1})$, where $OPT(\cdot)$ denotes the solution of the model that forms the argument. If the model is a DID or an I-DID, its solution is a policy tree. Our initial aim was to identify models that are *approximately* behaviorally equivalent. But due to the nature of the definition of BE, direct comparisons of disparate policy trees are not possible. A pair of policy trees may only be checked for equality. Thus, making it difficult to define a measure of approximate BE, motivating further investigations.

Analogous to BE, it can be noted that some subsets of models may impact the decision-making of the modeling agent similarly, thereby motivating interest in grouping such models together. We use this insight and introduce a new concept called subjective equivalence.

[1] We will use BE, SE as acronyms for *behaviorally* and *subjectively equivalent* in their adjective forms and *behavioral* and *subjective equivalence* in their noun forms, respectively. Appropriate usage will be self-evident.

4.1 DEFINITION

Let $h = \{a_i^t, o_i^{t+1}\}_{t=1}^{T}$ be the action-observation path for the modeling agent i, where o_i^{T+1} is null for a T horizon problem. If $a_i^t \in A_i$ and $o_i^{t+1} \in \Omega_i$, where A_i and Ω_i are i's action and observation sets respectively, then the set of all paths is, $H = \Pi_1^T(A_i \times \Omega_i)$, and the set of action-observation histories up to time t is $H^t = \Pi_1^{t-1}(A_i \times \Omega_i)$. The set of future action-observation paths is, $H_{T-t} = \Pi_t^T\ (A_i \times \Omega_i)$, where t is the current time step.

We show an example of future action-observation paths of agent i in a 2-horizon multi-agent tiger problem in Fig. 4.1. Agent i's actions are represented by nodes, and i's possible perceived observations are represented by the edges. In this example, agent i starts with listening and then it may receive one of six possible observations dependent on j's action. We use the action-observation paths of just agent i since our focus is on the decision making of i. Each of i's future paths have a probability associated with it. This probability is the chance with which that particular path is chosen by the subject agent i. The sum of each of these future action-observation path probabilities is 1. Also note that as the number of time steps increases, the number of action-observation paths and hence the size of the distribution table containing individual path probabilities, increases exponentially. As we discuss later, this is one of the main reasons for memory issues when the algorithm is executed. Also, the size of the distribution is directly proportional to the the number of actions and observations for agent i.

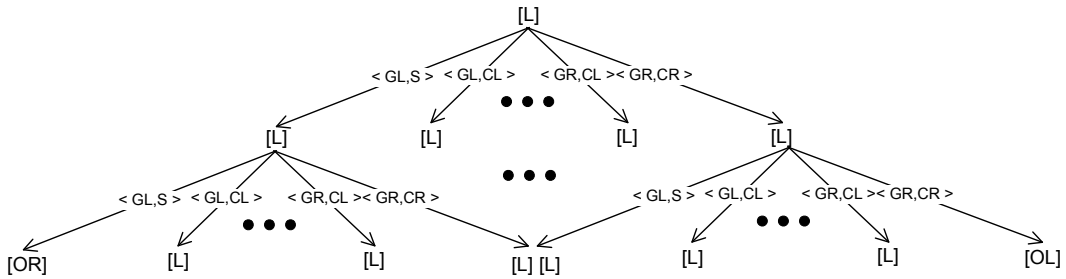


Figure 4.1: Future action-observation paths of agent i in a 2-horizon multiagent tiger problem. The nodes represent i's action, while the edges are labeled with the possible observations. This example starts with i listening. Agent i may receive one of six observations conditional on j's action, and performs an action that optimizes its resulting belief.

The distribution over i's future action-observation paths such as the one shown in Fig. 4.1 is induced by agent j's model and agent i's perfect knowledge of its own model and its action-observation history. This distribution plays a critical role in our approach and we denote it as, $Pr(H_{T-t}|h^t, m_{i,l}, m^t_{j,l-1})$, where $h^t \in H^t$, $m_{i,l}$ is i's level l I-DID and $m^t_{j,l-1}$ is the level $l-1$ model of j in the model node at time t. For the sake of brevity, we rewrite the distribution term as, $Pr(H_{T-t}|m^t_{i,l}, m^t_{j,l-1})$, where $m^t_{i,l}$ is i's horizon $T-t$ I-DID with its initial belief updated given the actions and observations in h^t. We will present a way to compute this distribution in the next section. We define SE below:

Definition 2 (Subjective Equivalence)**.** *Two models of agent j, $m^t_{j,l-1}$ and $\hat{m}^t_{j,l-1}$, are subjectively equivalent if and only if $Pr(H_{T-t}|m^t_{i,l}, m^t_{j,l-1}) = Pr(H_{T-t}|\, m^t_{i,l}, \hat{m}^t_{j,l-1})$, where H_{T-t} and $m^t_{i,l}$ are as defined previously.*

In other words, SE models are those that induce an identical distribution over agent i's future action-observation history. This reflects the fact that such models impact agent i's behavior similarly. We note that BE models, by definition, would induce a similar distribution over the future action-observation paths. However, models that induce similar distribution over agent i's future paths are not necessarily behaviorally equivalent. There could be models which induce a similar distribution and still differ in their behavior. The behavioral difference is not observed since the difference would become explicit over paths that are never followed (those which receive probability 0). This is why we call models that induce similar distributions as subjectively equivalent since these models are equivalent from the perspective of the subject agent.

4.2 Computing the Distribution over Future Paths

As mentioned earlier, each of the future action-observation paths has a probability associated with it. This probability is the chance with which that particular path is chosen by the subject agent i. The probabilities of all the paths put together constitute the distribution over the action-observation paths of agent i. Let h_{T-t} be some future action-observation path of agent i, $h_{T-t} \in H_{T-t}$. In

Proposition 1, we provide a recursive way to arrive at the probability, $Pr(h_{T-t}|m^t_{i,l}, m^t_{j,l-1})$. Of course, the probabilities over all possible paths sum to 1.

Proposition 1. $Pr(h_{T-t}|m^t_{i,l}, m^t_{j,l-1})$

$=Pr(a^t_i, o^{t+1}_i|m^t_{i,l}, m^t_{j,l-1}) \sum_{a^t_j, o^{t+1}_j} Pr(h_{T-t-1}|a^t_i, o^{t+1}_i, m^t_{i,l}, a^t_j, o^{t+1}_j, m^t_{j,l-1})$

$Pr(a^t_j, o^{t+1}_j|a^t_i, m^t_{i,l}, m^t_{j,l-1})$

$= Pr(a^t_i, o^t_i|m^t_{i,l}, m^t_{j,l-1}) \sum_{a^t_j, o^{t+1}_j} Pr(h_{T-t-1}|m^{t+1}_{i,l}, m^{t+1}_{j,l-1})\ Pr(a^t_j, o^{t+1}_j|a^t_i, m^t_{i,l}, m^t_{j,l-1})$

where

$$Pr(a^t_i, o^{t+1}_i|m^t_{i,l}, m^t_{j,l-1}) = Pr(a^t_i|OPT(m^t_{i,l}))\ \sum_{a^t_j} Pr(a^t_j|OPT(m^t_{j,l-1}))$$
$$\sum_{s^{t+1}} O_i(s^{t+1}, a^t_i, a^t_j, o^{t+1}_i) \sum_{s,m_j} T_i(s, a^t_i, a^t_j, s^{t+1})\ b^t_{i,l}(s, m_j) \tag{4.1}$$

and

$$Pr(a^t_j, o^{t+1}_j|a^t_i, m^t_{i,l}, m^t_{j,l-1}) = Pr(a^t_j|OPT(m^t_{j,l-1}))\ \sum_{s^{t+1}} O_j(s^{t+1}, a^t_j, a^t_i, o^{t+1}_j)$$
$$\sum_{s,m_j} T_i(s, a^t_i, a^t_j, s^{t+1}) b^t_{i,l}(s, m_j) \tag{4.2}$$

In Eq. 4.1, $O_i(s^{t+1}, a^t_i, a^t_j, o^{t+1}_i)$ is i's observation function contained in the CPT of the chance node, O^{t+1}_i, in the I-DID, $T_i(s, a^t_i,\ a^t_j, s^{t+1})$ is i's transition function contained in the CPT of the chance node, S^{t+1}, $Pr(a^t_i|OPT(m^t_{i,l}))$ is obtained by solving agent i's I-DID, $Pr(a^t_j|OPT(m^t_{j,l-1}))$ is obtained by solving j's model and appears in the CPT of node, A^t_j. In Eq. 4.2, $O_j(s^{t+1}, a^t_j,\ a^t_i, o^{t+1}_j)$ is j's observation function contained in the CPT of the chance node, O^{t+1}_j, given j's model is $m^t_{j,l-1}$. We give the proof of Proposition 1 below.

Proof of Proposition 1. $Pr(h_{T-t}|m^t_{i,l}, m^t_{j,l-1})$

$= Pr(h_{T-t-1}, a^t_i, o^{t+1}_i|m^t_{i,l}, m^t_{j,l-1})$

$= Pr(h_{T-t-1}|a^t_i, o^{t+1}_i, m^t_{i,l}, m^t_{j,l-1})\ Pr(a^t_i, o^{t+1}_i|m^t_{i,l}, m^t_{j,l-1})$ (using Bayes rule)

We focus on the first term next:

$Pr(h_{T-t-1}|a^t_i, o^{t+1}_i, m^t_{i,l}, m^t_{j,l-1})$

$= \sum_{a^t_j, o^{t+1}_j} Pr(h_{T-t-1}|\ a^t_i, o^{t+1}_i, m^t_{i,l}, a^t_j, o^{t+1}_j, m^t_{j,l-1})\ Pr(a^t_j, o^{t+1}_j|a^t_i, m^t_{i,l}, m^t_{j,l-1})$

$= Pr(h_{T-t-1}|m^{t+1}_{i,l}, m^{t+1}_{j,l-1})\ Pr(a^t_j, o^{t+1}_j|a^t_i, m^t_{i,l}, m^t_{j,l-1})$

In the above equation, the first term results due to an update of the models at time step t with

actions and observations. This term is computed recursively. For the second term, j's level $l-1$ actions and observations are independent of i's observations.

We now focus on the term, $Pr(a_i^t, o_i^{t+1}|m_{i,l}^t, m_{j,l-1}^t)$:

$Pr(a_i^t, o_i^{t+1}|m_{i,l}^t, m_{j,l-1}^t) = Pr(o_i^{t+1}|a_i^t, m_{i,l}^t, m_{j,l-1}^t)\ Pr(a_i^t|OPT(m_{i,l}^t))$

(i's action is conditionally independent of j given its model)

$= Pr(a_i^t|OPT(m_{i,l}^t))\ \sum_{a_j^t}\ Pr(o_i^{t+1}|a_i^t, a_j^t, m_{i,l}^t, m_{j,l-1}^t)\ Pr(a_j^t|OPT(m_{j,l-1}^t))$

$= Pr(a_i^t|OPT(m_{i,l}^t))\ \sum_{a_j^t}\ Pr(o_i^{t+1}|a_i^t, a_j^t, m_{i,l}^t)\ Pr(a_j^t|OPT(m_{j,l-1}^t))$

(i's observation is conditionally independent of j's model)

$= Pr(a_i^t|OPT(m_{i,l}^t))\ \sum_{a_j^t}\ Pr(a_j^t|OPT(m_{j,l-1}^t))\ Pr(o_i^{t+1}|a_i^t, a_j^t, b_{i,l}^t)$ ($b_{i,l}^t$ is i's belief in $m_{i,l}^t$)

$= Pr(a_i^t|OPT(m_{i,l}^t))\ \sum_{a_j^t}\ Pr(a_j^t|OPT(m_{j,l-1}^t))\ \sum_{s^{t+1}}\ Pr(o_i^{t+1}|s^{t+1}, a_i^t, a_j^t)\ Pr(s^{t+1}|a_i^t, a_j^t, b_{i,l}^t)$

$= Pr(a_i^t|OPT(m_{i,l}^t))\ \sum_{a_j^t}\ Pr(a_j^t|OPT(m_{j,l-1}^t))\ \sum_{s^{t+1}}\ O_i(s^{t+1}, a_i^t, a_j^t, o_i^{t+1})$

$\sum_{s,m_j}\ T_i(s, a_i^t, a_j^t, s^{t+1})\ b_{i,l}^t(s, m_j)$

where O_i and T_i are i's observation and transition functions respectively, in the I-DID denoted by model, $m_{i,l}^t$. This proves Eq. 4.1 in Proposition 1.

Finally, we move to the term, $Pr(a_j^t, o_j^{t+1}|a_i^t, m_{i,l}^t, m_{j,l-1}^t)$, to obtain Eq. 4.2:

$Pr(a_j^t, o_j^{t+1}|a_i^t, m_{i,l}^t, m_{j,l-1}^t) = Pr(o_j^{t+1}|a_j^t, a_i^t, m_{i,l}^t, m_{j,l-1}^t)\ Pr(a_j^t|a_i^t, m_{i,l}^t, m_{j,l-1}^t)$

$= Pr(o_j^{t+1}|a_j^t, a_i^t, m_{i,l}^t, m_{j,l-1}^t)\ Pr(a_j^t|OPT(m_{j,l-1}^t))$

(j's action is conditionally independent of i given its model)

$= Pr(a_j^t|OPT(m_{j,l-1}^t))\ \sum_{s^{t+1}}\ Pr(o_j^{t+1}|a_j^t, a_i^t, s^{t+1})\ Pr(s^{t+1}|a_j^t, a_i^t, m_{i,l}^t, m_{j,l-1}^t)$

$= Pr(a_j^t|OPT(m_{j,l-1}^t))\ \sum_{s^{t+1}}\ O_j(s^{t+1}, a_j^t, a_i^t, o_j^{t+1})\ \sum_{s,m_j}\ Pr(s^{t+1}|a_j^t, a_i^t, s)\ b_{i,l}^t(s, m_j)$

($b_{i,l}^t$ is i's belief in $m_{i,l}^t$)

$= Pr(a_j^t|OPT(m_{j,l-1}^t))\ \sum_{s^{t+1}}\ O_j(s^{t+1}, a_j^t, a_i^t, o_j^{t+1})\ \sum_{s,m_j}\ T_i(s, a_i^t, a_j^t, s^{t+1})\ b_{i,l}^t(s, m_j)$

(agent i's I-DID is used)

where O_j is j's observation function in model $m_{j,l-1}^t$, which is a part of i's I-DID. ∎

Now that we have a way of computing the distribution over the future paths, we may relate Definition 2 to our previous understanding of behaviorally equivalent models :

Proposition 2. *If* $OPT(m^t_{j,l-1}) = OPT(\hat{m}^t_{j,l-1})$*, then* $Pr(H_{T-t}|m^t_{i,l}, m^t_{j,l-1}) = Pr(H_{T-t}|m^t_{i,l}, \hat{m}^t_{j,l-1})$ *where* $m^t_{j,l-1}$ *and* $\hat{m}^t_{j,l-1}$ *are* j*'s models.*

Proof. The proof is reducible to showing the above for some individual path, $h_{T-t} \in H_{T-t}$. Given $OPT(m^t_{j,l-1}) = OPT(\hat{m}^t_{j,l-1})$, we may write, $Pr(a^t_j|OPT(m^t_{j,l-1})) = Pr(a^t_j|OPT(\hat{m}^t_{j,l-1}))$ for all a^t_j. Because all other terms in Eqs. 4.1 and 4.2 are identical, it follows that $Pr(h_{T-t}|m^t_{i,l}, m^t_{j,l-1})$ must be the same as $Pr(h_{T-t}|m^t_{i,l}, \hat{m}^t_{j,l-1})$.

∎

Consequently, the set of subjectively equivalent models includes those that are behaviorally equivalent. It further includes models that induce identical distributions over agent i's action−observation paths, but these models could be behaviorally distinct over those paths that have a zero probability. Thus, these latter models may not be behaviorally equivalent. Doshi and Gmytrasiewicz [11] call these models as (strictly) observationally equivalent. Therefore, the converse of the above proposition is not true.

We use a simple method to compute the distribution over the paths given the models of i and j by transforming the I-DID into a Dynamic Bayesian Network (DBN). We do this by replacing agent i's decision nodes in the I-DID with chance nodes so that $Pr(a_i \in A^t_i) = \frac{1}{|OPT(m^t_{i,l})|}$ and removing the utility nodes. The desired distribution is then computed by finding the marginal over the chance nodes that represent i's actions and observations with j's model entered as evidence in the Mod node at t.

In the next chapter, we will introduce the notion of ϵ-subjective equivalence that uses our definition of SE to approximately solve I-DIDs. We also describe the algorithm used.

CHAPTER 5

ϵ-SUBJECTIVE EQUIVALENCE

The definition of SE described in the previous section has the advantage of being rigorous in addition to the merit of permitting us to measure the degree to which models are SE, thereby allowing us to introduce *approximate SE*.

5.1 DEFINITION

We introduce the notion of ϵ-subjective equivalence (ϵ-SE) and define it as follows:

Definition 3 (ϵ-SE)**.** *Given $\epsilon \geq 0$, two models, $m^t_{j,l-1}$ and $\hat{m}^t_{j,l-1}$, are ϵ-SE if the divergence between the distributions $Pr(H_{T-t}|m^t_{i,l}, m^t_{j,l-1})$ and $Pr(H_{T-t}|m^t_{i,l}, \hat{m}^t_{j,l-1})$ is no more than ϵ.*

Here, the distributions over i's future paths are computed as shown in Proposition 1. There exists multiple ways to measure the divergence between distributions. Kullback-Leibler (KL) divergence [27] is one of the most well known information-theoretic measures of divergence of probability distributions, in part because their mathematical properties are well studied. There is a strong precedent of using KL divergence successfully in agent research to measure distance between distributions. As KL divergence is not symmetric, we use a symmetric version in this work, thereby providing added ease of use. Consequently, the models are ϵ-SE if,

$$D_{KL}(Pr(H_{T-t}|m^t_{i,l}, m^t_{j,l-1})||Pr(H_{T-t}|m^t_{i,l}, \hat{m}^t_{j,l-1})) \leq \epsilon$$

where $D_{KL}(p||p')$ denotes the symmetric KL divergence between distributions, p and p', and is calculated as:

$$D_{KL}(p||p') = \frac{1}{2}\sum_k \left(p(k) log \frac{p(k)}{p'(k)} + p'(k) log \frac{p'(k)}{p(k)} \right)$$

If $\epsilon = 0$, ϵ-SE collapses into exact SE. Sets of models exhibiting ϵ-SE for some non-zero but small ϵ do not differ significantly in how they impact agent i's decision making. As we mention in the next section, these models could be candidates for pruning.

5.2 APPROACH

We first compute the distributions over the future observation paths for all the initial models in the candidate model space. We then pick a model of j at random, say, $m_{j,l-1}^{t=1}$, from the model node and call it the representative model. The divergences in the distributions of each of the remaining models is computed with respect to that of the representative. All other models in the model node whose divergence values are less than or equal to ϵ are classified as ϵ-SE with $m_{j,l-1}^{t=1}$ and are grouped together with it. Of the remaining models, another representative is picked at random and the previous procedure is repeated. The procedure terminates when no more models remain to be grouped. It can be seen that this iteration converges quickly because there are only a finite number of behavioral equivalence classes. Recall that we had assumed a finite horizon problem with finite number of actions and observations. This process is illustrated in Fig. 5.1. In general, when $\epsilon > 0$, more models will likely be grouped together than if we considered exact SE. This will result in a fewer number of classes in the partition and at most as many representatives as there are behaviorally distinct models at each time step, after pruning.

The above procedure result in partitioning the model space into ϵ-SE classes and the representatives of each class are ϵ-subjectively distinct. This is because as we pick each representative model, we make sure that we group all the models in the model space that are equivalent with it before proceeding to pick another. However, this set is not unique and the partition could change with different representatives. Only the representative model from each class is retained and all other models are pruned. The representatives are distinguished in that all models in its group are ϵ-SE with it. Unlike exact SE, ϵ-SE relation is not necessarily transitive. Hence, it would be wrong to select any arbitrary model in the class to be the representative since others may not be ϵ-SE with

it. Let $\hat{\mathcal{M}}_j$ be the largest set of behaviorally distinct models, also called the *minimal set* [12]. Then, the following proposition holds:

Proposition 3 (Cardinality)**.** *The ϵ-SE approach results in at most $|\hat{\mathcal{M}}_j|$ models after pruning.*

Intuitively, the proposition follows from the fact that in the worst case, $\epsilon = 0$, resulting in subjectively distinct models. This set is no larger than the set of behaviorally distinct models.

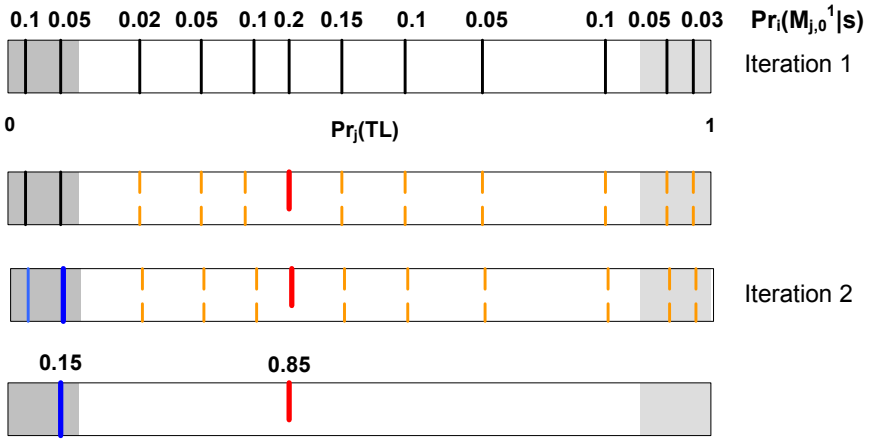


Figure 5.1: Illustration of the iterative ϵ-SE model grouping using the tiger problem. Black vertical lines denote the beliefs contained in different models of agent j included in the initial model node, $M^1_{j,0}$. Decimals on top indicate i's probability distribution over j's models. We begin by picking a representative model (red line) and grouping models that are ϵ-SE with it. Unlike exact SE, models in a different behavioral (shaded) region get grouped as well. Of the remaining models, another is selected as representative. Agent i's distribution over the representative models is obtained by summing the probability mass assigned to the individual models in each class.

5.2.1 Transfer of Probability Mass

A transfer of probability mass needs to happen in any approach which prunes some models of agent *j*, so that the mass assigned to those models is not lost. Hence, it is also done in an exact approach when models that are exactly SE are pruned. Agent i's belief assigns some probability mass to each model in the model node. Pruning some of the models would result in the loss of the mass assigned to those models. This loss would induce an error in the optimality of the solution and this error is avoided by transferring the probability mass over the pruned models in each class to the ϵ-SE representative that is retained in the model node (see Fig. 5.1).

5.2.2 SAMPLING ACTIONS AND OBSERVATIONS

For a time−extended I−DID, since the clustering process is done while solving the I-DID at every subsequent time step at which the the actual history of i's observations are not known, we obtain a likely history h^t by sampling i's actions and observations for subsequent time steps in the I-DID. This is because the predictive distribution over i's future action-observation paths, $Pr(H_{T-t}|h^t, m_{i,l}, m^t_{j,l-1})$, is conditioned on the history, as well. The sampling procedure is given below.

Initially, since the probability of occurrence of all of agent i's actions is assumed to be equal, we pick an action a^t_i at random. Using the sampled action and the belief, $o^{t+1}_i \sim Pr(\Omega_i|a^t_i, b^t_{i,l})$ (where $b^t_{i,l}$ is the prior belief) as the likelihood, we sample an observation. This sampled action-observation pair is used as the history, $h^t \overset{\cup}{\leftarrow} \langle a^t_i, o^{t+1}_i \rangle$. The above procedure is implemented by entering randomly, one of agent i's actions, as evidence in the chance node, A^t_i, of the DBN (mentioned in section 4) and sampling from the inferred distribution over the chance node, O^{t+1}_i.

In order to compute the distribution over the paths, we note that the agent i's I-DID's solution is needed as well ($Pr(a^t_i|OPT(m^t_{i,l}))$ term in Eq. 4.1). We avoid this complication by assuming a uniform distribution over i's actions, $Pr(a^t_i|OPT(m^t_{i,l})) = \frac{1}{|A_i|}$. However, even though the set of ϵ-SE models may change, this does not affect the set of behaviorally equivalent models. Thus, a different set of models of j may now be observationally equivalent. Nevertheless, a uniform distribution minimizes the change as models that are now observationally equivalent would continue to remain so for any other distribution over i's actions. This is because given a model of j, a uniform distribution for i induces a distribution that includes the largest set of paths in its support.

5.3 APPROXIMATION ALGORITHM

In this section, we present our algorithm for approximately solving I-DIDs using the previously described concept of ϵ-SE. The algorithm follows a similar approach as the exact solution using BE, except the procedure, **ϵ-SubjectiveEquivalence** replaces the procedure, **BehaviorEq**, in the algorithm in Fig. 3.2. The procedure, **ϵ-SubjectiveEquivalence** differs from the procedure, **BehaviorEq**, in the way the models are partitioned in the model node of the I-DID at each time step. This

is shown in Fig. 5.2. The procedure takes as input, the set of j's models, $\mathcal{M}_j$, agent i's DID, m_i, current time step and horizon, and the approximation parameter, ϵ. The algorithm begins by computing the distribution over the future paths of i for each model of j. If the time step is not the initial one, the prior action-observation history is first sampled. We may compute the distribution by transforming the I-DID into a DBN as mentioned in *Chapter 4* and entering the model of j as evidence – this implements Eqs. 4.1 and 4.2.

Then a representative model is picked at random and all the models of the other agent in the subject agent's model node, that have a distribution whose divergence from the distribution of the representative model is within ϵ, are grouped together. For this, we utilize the previously cached distributions of all the candidate models. This process is repeated until all the remaining ungrouped models are grouped. Each iteration results in a new unique class of ϵ-SE models including their respective representatives. In the final selection phase, only the representative model for each class is retained and the remaining models in the class are pruned after their belief masses are transferred to the representative. The set of representative models, which are ϵ-subjectively distinct, are returned.

ϵ-**SUBJECTIVEEQUIVALENCE** (Model set $\mathcal{M}_j$, DID m_i, current time step tt, horizon T, ϵ) **returns** $\mathcal{M}'_j$

1. Transform DID m_i into DBN by replacing i's decision nodes with chance nodes having uniform distribution
2. **For** t **from** 1 **to** tt **do**
3. Sample, $a_i^t \sim Pr(A_i^t)$
4. Enter a_i^t as evidence into chance node, A_i^t, of DBN
5. Sample, $o_i^{t+1} \sim Pr(O_i^{t+1})$
6. $h^t \overset{\cup}{\leftarrow} \langle a_i^t, o_i^{t+1} \rangle$
7. **For each** m_j^k **in** $\mathcal{M}_j$ **do**
8. Compute the distribution, $P[k] \leftarrow Pr(H_{T-t}|h^t, m_i, m_j^k)$, obtained from the DBN by entering m_j^k as evidence (Proposition 1)

Clustering Phase

9. **While** $\mathcal{M}_j$ not empty
10. Select a model, $m_j^{\hat{k}} \in \mathcal{M}_j$, at random as representative
11. Initialize, $\mathcal{M}_j^{\hat{k}} \leftarrow \{m_j^{\hat{k}}\}$
12. **For each** m_j^k **in** $\mathcal{M}_j$ **do**
13. **If** $D_{KL}(P[\hat{k}]||P[k]) \leq \epsilon$
14. $\mathcal{M}_j^{\hat{k}} \overset{\cup}{\leftarrow} m_j^k$, $\mathcal{M}_j \overset{-}{\leftarrow} m_j^k$

Selection Phase

15. **For each** $\mathcal{M}_j^{\hat{k}}$ **do**
16. Retain the representative model, $\mathcal{M}'_j \overset{\cup}{\leftarrow} m_j^{\hat{k}}$
17. **Return** $\mathcal{M}'_j$

Figure 5.2: Algorithm for partitioning j's model space using ϵ-SE. This function replaces **BehaviorEq()** in Fig. 3.2.

CHAPTER 6

TEST PROBLEM DOMAINS

In order to illustrate the usefulness of I-DIDs, we apply them to two illustrative problems. We describe, in particular, the formulation of the I-DIDs for these examples.

6.1 MULTI-AGENT TIGER PROBLEM

We begin our illustrations of using I-IDs and I-DIDs with a slightly modified version of the multiagent tiger problem [20]. It differs from other multi-agent versions of the same problem [30] by assuming that the agents not only hear growls to know about the location of the tiger, but also hear creaks that may tell if the other agent has opened a door. The problem has two agents, each of which can open the right door (OR), the left door (OL) or listen (L). In addition to hearing growls (from the left (GL) or from the right (GR)) when they listen, the agents also hear creaks (from the left (CL), from the right (CR), or no creaks (S)), which noisily indicate the other agents opening one of the doors or listening. When any door is opened, the tiger persists in its original location with a probability of 95%. Agent *i* hears growls with a reliability of 65% and creaks with a reliability of 95%. Agent *j*, on the other hand, hears growls with a reliability of 95%. Thus, the setting is such that agent *i* hears agent *j* opening doors more reliably than the tiger's growls. This suggests that *i* could use *j*'s actions as an indication of the location of the tiger. Each agents preferences are as in the single agent game discussed in the original version [25].

Let us consider a particular setting of the tiger problem in which agent *i* considers two distinct level 0 models of *j*. This is represented in the level 1 I-ID shown in Fig. 6.1. The two IDs could differ, for example, in the probability that *j* assigns to the tiger being behind the left door as modeled by the node *TigerLocation*. Given the level 1 I-ID, we may expand it into the I-DID shown in

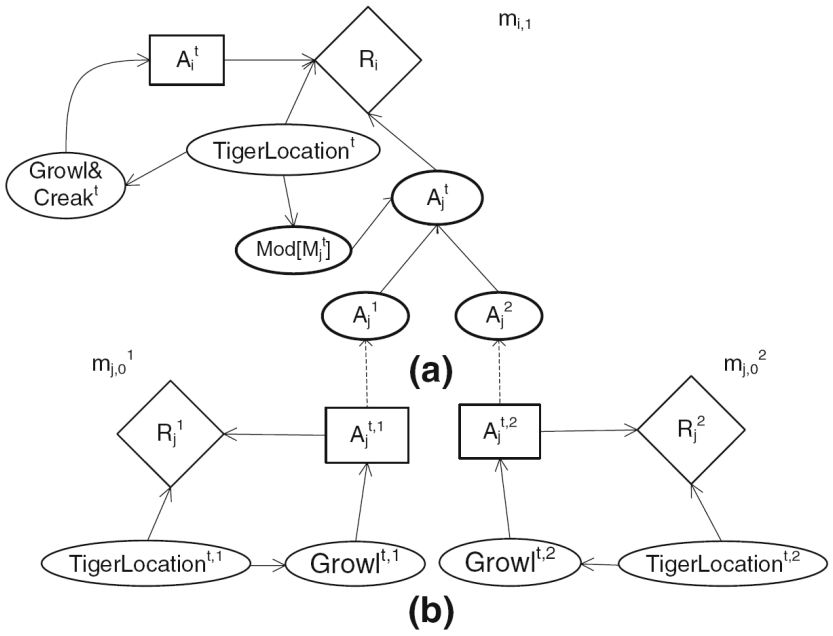


Figure 6.1: (a) Level 1 I-ID of agent *i*, (b) two level 0 IDs of agent *j* whose decision nodes are mapped to the chance nodes, $A1_j$ and $A2_j$, in (a), indicated by the dotted arrows. The two IDs differ in the distribution over the chance node, TigerLocation [14].

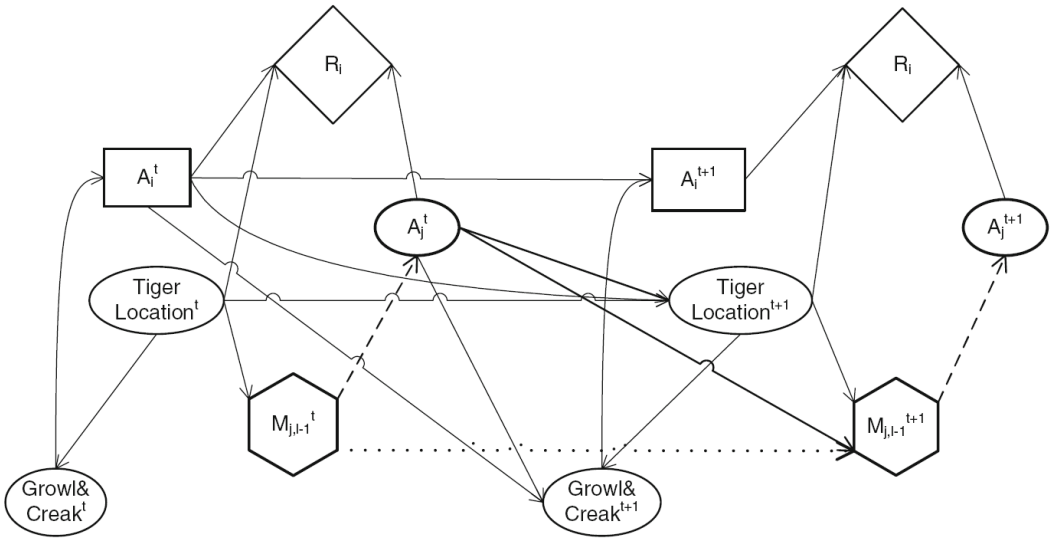


Figure 6.2: Level 1 I-DID of agent *i* for the multiagent tiger problem. The model node contains *M* level 0 DIDs of agent *j* . At horizon 1, the models of *j* are IDs [14].

Fig. 6.2. The model node, $M^t_{j,0}$ contains the different DIDs that are expanded from the level 0 IDs in Fig. 6.1(b). The DIDs may have different probabilities about the tiger location at time step *t*. We get the probability distribution of *j*'s actions in chance node A^t_j by solving the level 0 DIDs of *j*. On performing the optimal action(s) at time step *t*, *j* may receive observations of the tiger's growls. This is reflected in new beliefs on the tiger's position within *j*'s DIDs at time step *t + 1*. Consequently, the model node, $M^{t+1}_{j,0}$, contains more models of *j* and *i*'s updated belief on *j*'s possible DIDs.

$\langle a^t_i, a^t_j \rangle$	TigerLocation^t_i	TL	TR
(a)			
$\langle OL, * \rangle$	*TL*	0.95	0.05
$\langle OL, * \rangle$	*TR*	0.05	0.95
$\langle OR, * \rangle$	*TL*	0.95	0.05
$\langle OR, * \rangle$	*TR*	0.05	0.95
$\langle *, OL \rangle$	*TL*	0.95	0.05
$\langle *, OL \rangle$	*TR*	0.05	0.95
$\langle *, OR \rangle$	*TL*	0.95	0.05
$\langle *, OR \rangle$	*TR*	0.05	0.95
$\langle L, L \rangle$	*TL*	1.0	0
$\langle L, L \rangle$	*TR*	0	1.0
(b)			
$\langle OL, * \rangle$	*	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

Figure 6.3: CPD of the chance node $TigerLocation^{t+1}_i$ in the I-DID of Fig. 6.2 when the tiger (a) likely persists in its original location on opening doors, and (b) randomly appears behind any door on opening one.

$\langle a^t_i, a^t_j \rangle$	TgrLoc^{t+1}_i	⟨GL, CL⟩	⟨GL, CR⟩	⟨GL, S⟩	⟨GR, CL⟩	⟨GR, CR⟩	⟨GR, S⟩
$\langle L, L \rangle$	*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

Figure 6.4: The CPD of the chance node $Growl\&Creak^{t+1}_i$ in the level 1 I-DID.

We showed the nested I-DID unrolled over two time steps for the multiagent tiger problem in Fig. 6.2. Agent *i* at level 1 considers *M* models of agent *j* of level 0 which, for example, differ in the distributions over the chance node *TigerLocation*. In agent *i*'s I-DID, we assign the marginal distribution over the tigers location to the CPD of the chance node $TigerLocation_i^t$. In the next time step, the CPD of the chance node $TigerLocation_i^{t+1}$ conditioned on $TigerLocation_i^t$, A_i^t, and A_j^t is the transition function, shown in Fig. 6.3. We show the CPD of the observation node, $Growl\&Creak_i^{t+1}$, in Fig. 6.4. The CPDs of the observation nodes in level 0 DIDs are identical to the observation function in the single agent tiger problem.

$\langle a_i^t, a_j^t \rangle$	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

Figure 6.5: Reward function of agent *i* for the multi-agent tiger problem.

The decision node A_i^t includes possible actions of agent *i* in the scenario such as listening (L), opening the left door (OL), and opening the right door (OR). The utility node R_i in the level 1 I-DID relies on both agents actions, A_i^t and A_j^t, and the physical states, $TigerLocation_i^t$. We show the utility table in Fig. 6.5. The utility tables for level 0 models are identical to the reward function in the single agent tiger problem which assigns a reward of 10 if the correct door is opened, a penalty of 100 if the opened door is the one behind which is a tiger, and a penalty of 1 for listening.

6.2 Multi-agent Machine Maintenance Problem

The *multiagent machine maintenance problem* (MM) [20] is a multi-agent variation of the original machine maintenance problem presented in [41]. In this version, we have two agents that cooperate. The non-determinism of the original problem is increased to make it more realistic, allowing

for more interesting policy structures when solved. The original MM problem involved a machine containing two internal components operated by a single agent. Either one or both components of the machine may fail spontaneously after each production cycle. The machine that is under maintenance can have three possible states: *0-fail* implying that none of the internal components in that machine failed; *1-fail* implying that one of the internal components in that machine failed; and *2-fail* implying that two of the internal components in that machine failed. If an internal component has failed, then there is some chance that when operating upon the product, it will cause the product to be defective. An agent may choose to manufacture the product (*M*) without examining it, examine the product (*E*), inspect the machine (*I*), or repair it (*R*) before the next production cycle. On an examination of the product, the subject may find it to be defective. Of course, if more components have failed, then the probability that the product is defective is greater.

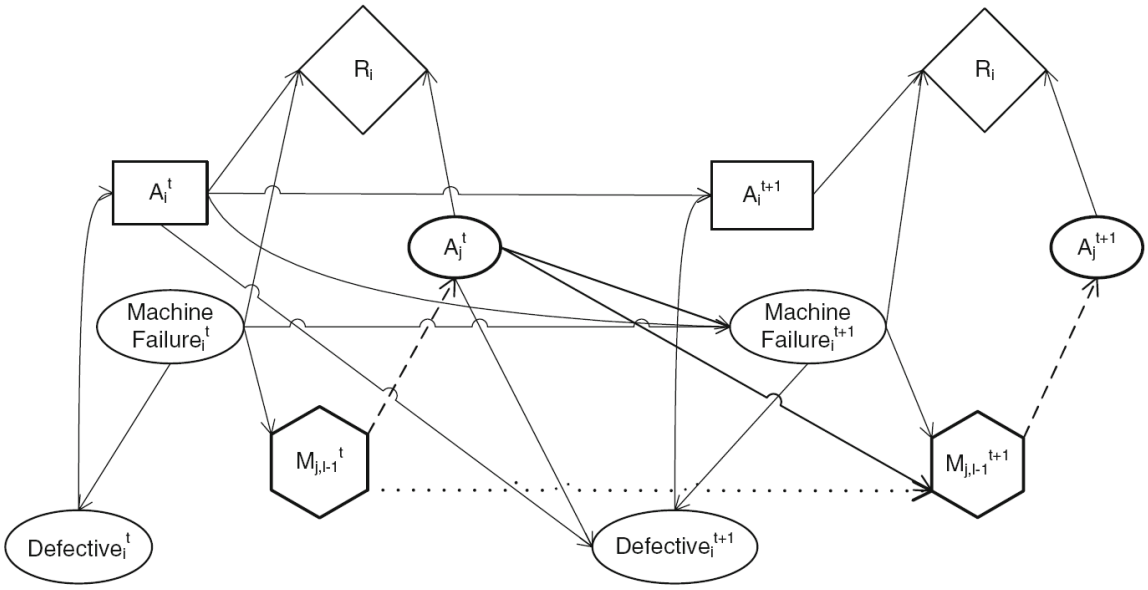


Figure 6.6: Level 1 I-DID of agent *i* for the multiagent MM problem. The hexagonal model node contains *M* level 0 DIDs of agent *j* . At horizon 1, the models of *j* are IDs [14].

We show the design of a level 1 I-DID for the multiagent MM problem in Fig. 6.6. We consider *M* models of agent *j* at level 0 which differ in the probability that *j* assigns to the chance node $MachineFailure_j$. In the I-DID, the chance node, $MachineFailure_i^{t+1}$, has incident arcs from the nodes $MachineFailure_i^t$, A_i^t, and A_j^t . The CPD of the chance node is shown in Fig. 6.7.

$\langle a_i^t, a_j^t \rangle$	Mch Fail$_i^{t+1}$	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

Figure 6.7: CPD of the chance node $MachineFailure_i^{t+1}$ in the level 1 I-DID of Fig. 6.6.

$\langle a_i^t, a_j^t \rangle$	Mch fail$_i^{t+1}$	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

Figure 6.8: The CPD of the chance node $Defective_i^{t+1}$ in the level 1 I-DID.

For the observation chance node, $Defective_i^{t+1}$, we associate the CPD shown in Fig. 6.8. Note that arcs from $MachineFailure_i^{t+1}$ and the nodes, A_i^t, and A_j^t, in the previous time step are incident to this node. The observation nodes in the level 0 DIDs have CPDs that are identical to the observation function in the original MM problem.

The decision node, A_i , consists of agent *i*'s actions including manufacture (*M*), examine (*E*), inspect (*I*), and repair (*R*). It has one information arc from the observation node $Defective_i^t$ indicating that *i* knows the examination results before making the choice. The utility node R_i is associated with the utility table in Fig. 6.9. The utility function of the agent j which is a level 0 agent is

$\langle a_i^t, a_j^t \rangle$	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

Figure 6.9: Reward function of agent *i*. For the level 0 agent *j*, the reward function is identical to the one in the classical MM problem with some modifications shown in Fig. 6.10.

$\langle a_j^t \rangle$	0-fail	1-fail	2-fail
⟨M⟩	0.9025	0.475	0.0
⟨E⟩	0.6525	0.225	0.0
⟨I⟩	−0.5	−1.5	−2.5
⟨R⟩	−2.0	−2.0	−2.0

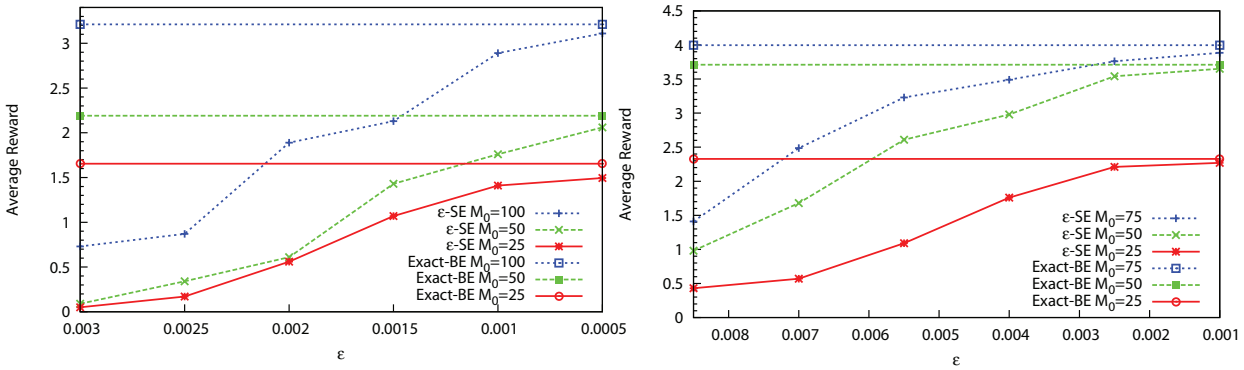
Figure 6.10: Reward function of agent *j*. Agent *j* is a level 0 agent whose reward function is identical to the one in the classical MM problem with some modifications.

shown in Fig. 6.10. The CPD of the chance node, $Mod[M_j^{t+1}]$, in the model node, $M_{j,l-1}^{t+1}$, reflects which prior model, action and observation of *j* results in a model contained in the model node.

CHAPTER 7

EXPERIMENTAL EVALUATION

We implemented the algorithms in Figs. 3.2 and 5.2 utilizing the HUGIN Java API for DIDs. HUGIN is a commercial software used for solving graphical models such as Bayesian networks and influence diagrams [1]. HUGIN not only has a GUI, but also APIs in several languages such as JAVA, C++ e. t. c., where these graphical models can be implemented and used in other applications. We show results for the well-known problems in the literature: the two-agent *tiger problem* ($|S|$=2, $|A_i|$=$|A_j|$=3, $|\Omega_i|$=6, $|\Omega_j|$=3) [20] and the multiagent version of the machine maintenance (MM) problem ($|S|$=3, $|A_i|$=$|A_j|$=4, $|\Omega_i|$=2, $|\Omega_j|$=2) [41] described in the previous chapter. These problems are popular but relatively small, having a physical state space size of 2 and 3 respectively. But note that in an interactive state space, we must consider all possible models of other agents, thus making the interactive state space (IS) considerably larger. We formulate level 1 I-DIDs of increasing time horizons for the problems, and solve it approximately for varying ϵ. We show that, (i) the quality of the solution generated using our approach (ϵ-SE) improves as we reduce ϵ for given numbers of initial models of the other agent, M_0, and converges toward that of the exact solution. This is indicative of the flexibility of the approach; (ii) in comparison to the approach of updating models discriminatively (DMU) [12], which is the current efficient technique, ϵ-SE is able to obtain larger rewards for an identical number of initial models. This indicates a more informed clustering and pruning using ϵ-SE in comparison to DMU, although it is less efficient in doing so.



(a) (b)

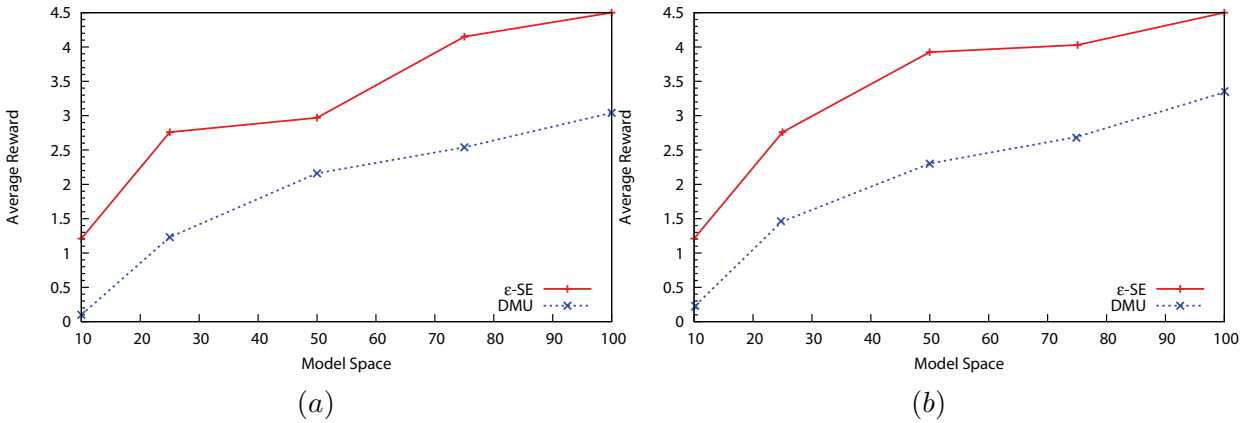
Figure 7.1: Performance profile obtained by solving a level 1 I-DID for the multiagent tiger problem using the ϵ-SE approach for (a) 3 horizons and (b) 4 horizons. As ϵ reduces, quality of the solution improves and approaches that of the exact.

7.1 Multi-agent Tiger Problem

In Fig. 7.1(a, b), we show the average rewards gathered by executing the policies obtained from solving level 1 I-DIDs approximately within a simulation of the problem domain. Each data point is the average of 300 runs where the true model of j is picked randomly according to i's belief. The exact solutions are represented by the flat lines. As ϵ decreases and approaches zero, the policies tend to converge to the exact solution. As the number of candidate models of the other agent considered by the agent i increases, its chances of modeling the other agent correctly also increases. Note that the error bound in *Chapter 8* does not apply here because we prune models in subsequent time steps as well.

Next, we compare the performance of this approach with that of DMU. While both approaches cluster and prune models, DMU does so only in the initial model node, thereafter updating only those models which on update will be behaviorally distinct. Thus, we compare the average rewards obtained by the two approaches when an identical number of models remain in the initial model node (a) before and (b) after clustering and selection as shown in Fig. 7.2(a) and (b) respectively. In the comparison involving the initial models that remain in the model node before clustering, it

is possible that the DMU approach might prune more models than ϵ-SE. This could be responsible, in part, for its poor performance compared to e-SE. Hence, this might not be the best indicator for correctly comparing the effectiveness of the two pruning strategies. However, the latter comparison is done by varying ϵ in both approaches until the desired number of models are retained. This enables us to compare the quality of the solution for the same number of models retained and in turn allowing us to compare the effectiveness of the clustering and selection techniques of the two approaches. The DMU data for case (b) were provided by Dr. Yifeng Zeng, Aalborg University, Denmark.



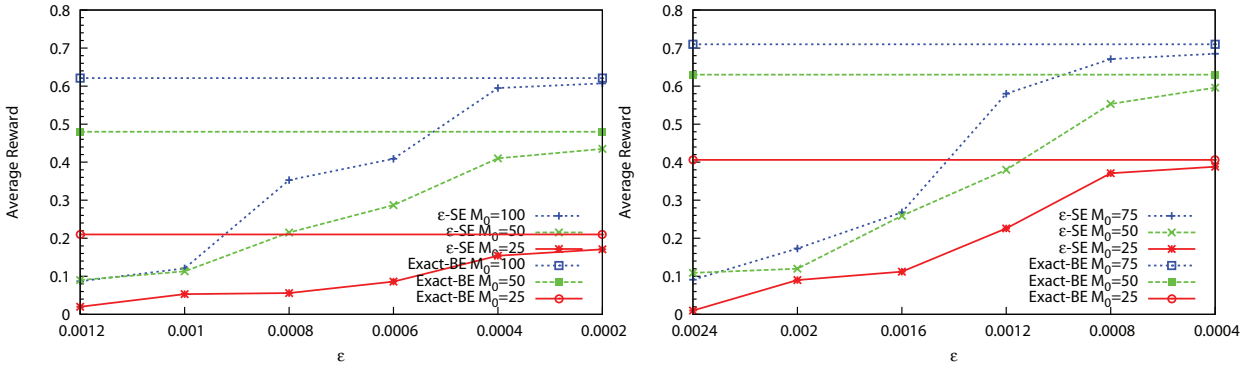
(a) (b)

Figure 7.2: Comparison of ϵ-SE and DMU for the multi-agent tiger problem in terms of the rewards obtained given identical numbers of models in the initial model node (a) before clustering and pruning and (b) after clustering and pruning.

From Fig. 7.2(b), we observe that ϵ-SE results in better quality policies that obtain significantly higher average reward. This indicates that the models pruned by DMU were more valuable than those pruned by ϵ-SE, thereby indicating a more informed way in which clustering and selection were done in our approach. DMU's approach of measuring simply the closeness of beliefs in models for clustering resulted in significant models being pruned. However, the trade off is the increased computational cost in calculating the distributions over the future paths. To illustrate, ϵ-SE consumed an average of 34.4 secs in solving a 4 horizon I-DID with $25-100$ initial models and differing ϵ, on an Intel Pentium Dual CPU 1.87GHz, 3GB RAM machine which represents approximately a three-fold increase compared to DMU.

7.2 MULTI-AGENT MACHINE MAINTENANCE PROBLEM

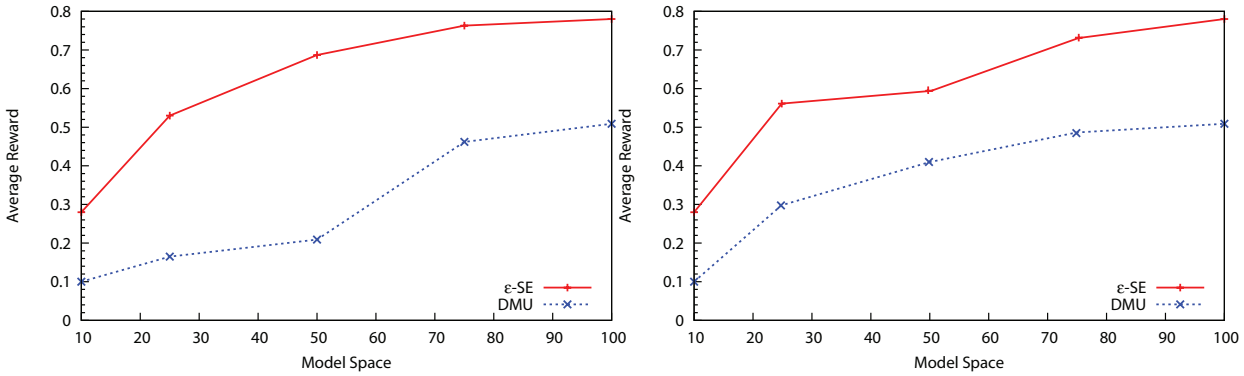
We show a similar set of results for the MM problem in Fig. 7.3. The MM problem differs in having one more physical state and action in comparison to the tiger problem, and less observations. We observe a similar convergence toward the performance of the exact solution as we gradually reduce ϵ. This affirms the flexibility in selecting ϵ provided by the approach.



(a) (b)

Figure 7.3: Performance profile for the multiagent MM problem obtained by solving level 1 I-DIDs approximately using ϵ-SE for (a) 3 horizon and (b) 4 horizon. Reducing ϵ results in better quality solutions.

Furthermore, in Fig. 7.4, we again note the significant increase in average reward exhibited by ϵ-SE compared to DMU given an identical number of retained models.



(a) (b)

Figure 7.4: Significant increase in rewards obtained for ϵ-SE compared to DMU, given identical numbers of retained models in the initial model node (a) before clustering and pruning and (b) after clustering and pruning for the MM problem.

This clearly illustrates the improvement in clustering models that are truly approximately similar, in comparison to using heuristics such as closeness of beliefs. As mentioned earlier, even though the results presented in Fig. 7.4(a) may not be a reliable indicator for comparing the effectiveness of the two clustering strategies, the results shown in Fig. 7.4(b) further reinforce the appeal of ϵ-SE. This provides empirical evidence that our approach performed a more informed clustering and that the models retained are significantly more valuable than those retained by DMU translating into greater reward, albeit at the cost of efficiency. The approach incurred on average 54.5 secs exhibiting a four-fold increase in time taken compared to DMU in order to solve a horizon 4 I-DID with 25-100 initial models. On the other hand, while ϵ-SE continues to solve I-DIDs of 5 horizons, the exact approach runs out of memory.

In summary, experiments on two multiagent problem domains indicate that the ϵ-SE approach models subjective similarity between models of the other agent more accurately resulting in favorable performance in terms of quality of the solutions, but at the expense of computational efficiency. As a part of the evaluation, we also theoretically analyze the performance of our approximation technique and compare it with that of the model clustering approach (described previously in *Chapter* 3) in the next chapter.

CHAPTER 8

THEORETICAL ANALYSIS

Our main motivation toward the proposed approximation technique is to mitigate the curse of history and dimensionality by considerably reducing the size of the state space and at the same time preserving the quality of the solution. In this chapter, we will focus on specifying how exactly we achieved computational savings and also on bounding the error due to the approximation. We will also theoretically analyze our savings with respect to exact SE algorithm and the Model Clustering approach.

8.1 COMPUTATIONAL SAVINGS

The computational complexity of solving I-DIDs is primarily due to the large number of models that must be solved over *T* time steps. Let M_j^0 be the number of candidate models of the other agent, A_j be the number of actions the agent can perform, and Ω_j be the number of possible observations. Hence at time step t, there could be $|\mathcal{M}_j^0|(|A_j||\Omega_j|)^t$ many models of the other agent j. As mentioned earlier, nested modeling further contributes to the complexity of the problem because it requires solving of lower level models recursively upto level 0. In an N+1 agent setting, if the number of models considered at each level for an agent is bound by $|\mathcal{M}|$, then solving an I-DID at level l requires the solutions of $\mathcal{O}((N|\mathcal{M}|)^l)$ many models. As we mentioned in Proposition 3, the ϵ-SE approximation reduces the number of agent models at each level to at most the size of the minimal set, $|\hat{\mathcal{M}}^t|$. Thus, $|\mathcal{M}_j^0|$ many models are solved initially and the complexity is incurred due to the distribution computations while performing the inference in a DBN. This complexity is less than that of solving DIDs. Hence, we need to solve atmost $\mathcal{O}((N|\hat{\mathcal{M}}^*|)^l)$ number of models at

each non-initial time step, typically less, where $\hat{\mathcal{M}}^*$ is the largest of the minimal sets, in comparison to $\mathcal{O}((N|\mathcal{M}|)^l)$. Here $\mathcal{M}$ grows exponentially over time. In general, $|\hat{\mathcal{M}}| \ll |\mathcal{M}|$, resulting in a substantial reduction in the computation. Additionally, a reduction in the number of models in the model node also reduces the size of the state space, which makes solving the upper-level I-DID more efficient.

We will now compare our approach with that of the model clustering (MC) approach [46].

1. In the MC approach, constant number (*K*) of models are solved at every time step where as in our ϵ-SE approach, all initial models are solved in order to compute the distribution over the future action-observation paths. However, from the next step onwards, only a maximum of as many models as there are behaviorally distinct ones have to be solved.

2. In MC, in order to partition the model space, it is required to find the sensitivity points which involves complex linear programming whereas the process of partitioning SE regions in our approach is simple. We simply pick a model randomly and cluster all ϵ-SE models with it. Hence, when another model is picked randomly from those that remain after the grouping, it is assured that it is ϵ-subjectively distinct from the previous representative. However, computing the distributions for all the candidate models, which is required for the clustering process, is time consuming.

3. In MC, the *k*-means clustering process is known to take some time to converge where as in ϵ-SE the clustering methodology is simple and the clustering is quick due to the presence of only finite number of SE classes.

4. In MC, when *K* models are selected we may end up having more than one model from the same subjectively equivalent region. This results in redundancy (because two SE models are effectively identical as they affect the subject agent similarly) and unnecessary computations. Instead, if these models were from different SE regions, the solution quality could be improved. However, in the ϵ-SE approach, such redundancies are avoided.

It can be shown theoretically that the ϵ-subjective equivalence approach always performs better or equal to, but never worse, than the model clustering approach in terms of the number of candidate models ascribed to the other agents. This claim follows from the analysis that we conduct as shown below.

For the purpose of this analysis, let us consider *R* to be the number of behaviorally equivalent classes at any particular time step *t* and *K* to be the number of models picked in the MC approach. We present results for three exhaustive cases as follows:

1. $R < K$: In this case, the ϵ-SE approach ends up solving at most *R* models. Hence, even the worst case of this approach is better in terms of the number of candidate models solved with respect to the model clustering approach. In terms of quality, in the worst case of the ϵ-SE approach where $\epsilon = 0$, since no redundancy occurs in the models picked, it results in an exact solution but the MC approach is unable to guarantee this. Thus, better solution quality is more probable with the former.

2. $R = K$: In this case, the MC approach and the worst case of the ϵ-SE approach (when $\epsilon = 0$), end up solving the same number of models. In terms of quality, the worst case of the SE approach guarantees at least one representative per subjectively equivalent region thus producing an exact solution but the MC approach does not, as there may be redundant models.

3. $R > K$: In this case, the worst case of the ϵ-SE approach ends up solving greater number of models. But quality-wise, the ϵ-SE approach is more likely to perform better than the MC approach because a greater number of ϵ-subjectively distinct models are solved in the former and a there exists atleast *R-K* regions without a representative model in the latter.

8.2 Error Bound

In the ϵ-SE approach, we may partially bound the error that arises due to the approximation. We assume that the lower-level models of the other agent are solved exactly and also assume that

we limit the pruning of ϵ-SE models to the initial model node. Doshi and Zeng [12] show that, in general, it is difficult to usefully bound the error if lower-level models are themselves solved approximately. Trivially, when $\epsilon = 0$ there is no optimality error in the solution. The error is due to transferring the probability mass of the pruned model to the representative, effectively replacing the pruned model with the representative. In other words, error arises when ϵ is such that models from some subjectively equivalent regions get clustered with a representative model from another region.

For example, say there are *R* behaviorally equivalent regions and *k* representative models remain after the clustering process, at a particular time step, from *M* candidate models of agent *j* that were initially considered. Note that the value of *k* is dynamic; it changes at every time step. We can bound the error for excluding all but *k* models. This presents us with two situations where approximation errors can occur:

1. When $\boldsymbol{k = R}$: In this case, there is a model representing each ϵ-subjectively equivalent region *R* and the number of ϵ-*subjectively equivalent regions* equal the number of *behaviorally subjectively regions*. Hence, there will be no optimality error.

2. When $\boldsymbol{k < R}$: In the trivial case where ϵ = 0, approximation error arises because there will be *R-k* regions without representatives. In the case where $\epsilon > 0$, approximation error arises because there may be more than or equal to *R-k* regions without representatives.

Note that our approach can never result in a situation where $\boldsymbol{k > R}$ (see *Proposition 3*).

Our definition of SE provides us with a unique opportunity to bound the error for i. We observe that the expected value of the I-DID could be obtained as the expected reward of following each path weighted by the probability of that path. Let $\rho_{b_{i,l}}(H_T)$ be the vector of expected rewards for agent i given it's belief when each path in H_T is followed. Here, T is the horizon of the I-DID. The expected value for i is:

$$EV_i = Pr(H_T|m_{i,l}, m_{j,l-1}) \cdot \rho_{b_{i,l}}(H_T)$$

where $m_{j,l-1}$ is the model of j.

If the above model of j is pruned in the Mod node, let model $\hat{m}_{j,l-1}$ be the representative that replaces it. Then $\hat{b}_{i,l}$ is i's belief in which model $m_{j,l-1}$ is replaced with the representative. Expected value for i, $\hat{EV}_i$, is:

$$\hat{EV}_i = Pr(H_T|m_{i,l}, m_{j,l-1}) \cdot \rho_{\hat{b}_{i,l}}(H_T)$$

Then, the effective error bound is:

$$\begin{aligned}
&\Delta = ||\hat{EV}_i - EV_i||_\infty \\
&= ||Pr(H_T|m_{i,l}, m_{j,l-1}) \cdot \rho_{\hat{b}_{i,l}}(H_T) - Pr(H_T|m_{i,l}, m_{j,l-1}) \cdot \rho_{b_{i,l}}(H_T)||_\infty \\
&= ||Pr(H_T|m_{i,l}, m_{j,l-1}) \cdot \rho_{\hat{b}_{i,l}}(H_T) - Pr(H_T|m_{i,l}, \hat{m}_{j,l-1}) \cdot \rho_{b_{i,l}}(H_T) \\
&\quad + Pr(H_T|m_{i,l}, \hat{m}_{j,l-1}) \cdot \rho_{b_{i,l}}(H_T) - Pr(H_T|m_{i,l}, m_{j,l-1}) \cdot \rho_{b_{i,l}}(H_T)||_\infty \quad \text{(add zero)} \\
&\leq ||Pr(H_T|m_{i,l}, m_{j,l-1}) \cdot \rho_{\hat{b}_{i,l}}(H_T) - Pr(H_T|m_{i,l}, \hat{m}_{j,l-1}) \cdot \rho_{\hat{b}_{i,l}}(H_T) \\
&\quad + Pr(H_T|m_{i,l}, \hat{m}_{j,l-1}) \cdot \rho_{b_{i,l}}(H_T) - Pr(H_T|m_{i,l}, m_{j,l-1}) \cdot \rho_{b_{i,l}}(H_T)||_\infty \quad (|\rho_{\hat{b}_{i,l}}| \leq |\rho_{b_{i,l}}|) \\
&\leq ||\rho_{\hat{b}_{i,l}}(H_T) - \rho_{b_{i,l}}(H_T)||_\infty \cdot ||Pr(H_T|m_{i,l}, m_{j,l-1}) - Pr(H_T|m_{i,l}, \hat{m}_{j,l-1})||_1 \quad \text{(Hölder's inequality)} \\
&\leq (R_i^{max} - R_i^{min})T \times 2\epsilon \qquad\qquad \text{(Pinsker's inequality)}
\end{aligned}$$

Matters become more complex when we additionally prune models in the subsequent model nodes as well. This is because rather than comparing over distributions given each history of i, we sample i's action-observation history. Consequently, additional error incurs due to the sampling, which is difficult to bound. As mentioned earlier, it is difficult to usefully bound the error if lower-level models are themselves solved approximately. This limitation is significant because approximately solving lower level models could bring considerable computational savings.

In summary, error in i's behavior due to pruning ϵ-SE models in the initial model node may be bounded, but we continue to investigate how to usefully bound the error due to multiple additional approximations.

CHAPTER 9

CONCLUSION

Interactive dynamic influence diagrams (I-DIDs) provide a graphical formalism for modeling the sequential decision making of an agent in an uncertain multi-agent setting. In this thesis, we present a new approximation method, called ϵ *Subjective Equivalence* (ϵ-SE), to solve interactive dynamic influence diagrams (I-DIDs). This is an approximation technique that allows an agent to plan sequentially in multi-agent scenarios, which could be cooperative, competitive or even neutral. The main motivation behind the development of this method is that the curses of dimensionality and history that impact I-DIDs, limited existing algorithms from scaling to larger multi-agent problem domains. These curses manifests in the exponentially growing space of candidate models ascribed to other agents over time. Hence, our goal was to come up with an approximation technique that could mitigate these curses better than those that already existed.

Existing approximation techniques used clustering and pruning of behaviorally equivalent models as the way to identify the *minimal* model set. Our approximation technique reduces the complexity by additionally pruning models that are *approximately* subjectively equivalent. Toward this objective, we defined subjective equivalence in terms of the distribution over the subject agent's future action-observation paths that allowed a way to measure the degree to which the models are subjectively equivalent, which helped formulate our approximation technique. Defining SE by explicitly focusing on the impact that the other agents' models have on the subject agent in the interaction allowed us to better identify subjective similarity. This translated into solutions of better quality given a limit on the number of models that could be held in memory. Consequently, other approaches may need more models to achieve comparable quality, which could translate into better efficiencies for our approach.

We showed the performance of our approach for two test problems: the multi-agent tiger problem, and the multi-agent machine maintenance problem and compared the results of our approach with the existing best technique for solving I-DIDs (DMU) and also the exact SE method. Highlights of the results obtained are presented below:

1. The quality of the solution generated using our approach improves as we reduce ϵ for given numbers of initial models of the other agent, and approaches that of the exact solution. This is indicative of the flexibility of the approach.

2. In comparison to the approach of updating models discriminatively (DMU), which is the current efficient technique, ϵ-SE is able to obtain larger rewards for an identical number of initial models. This indicates a more informed clustering and pruning using ϵ-SE in comparison to DMU. The trade off was the increased computational cost due to calculating the distributions over future paths. ϵ-SE consumed three times the average time consumed by DMU in solving a 4 horizon I-DID with 25-100 initial models and differing ϵ for the multi-agent tiger problem and a four-fold increase in the time consumed with the same setting for the multi-agent machine maintenance problem.

We also theoretically analyzed the savings from our approach and compared it with that of the model clustering approach. Our analysis revealed that ϵ-SE either ascribes less models to other agent or is likely to perform qualitatively better in comparison to the model clustering approach.

9.1 Limitations and Future Work

Scalability to higher horizons using our approximation technique is limited mainly by the curse of history due to the exponential increase in the number of future paths over increasing number of horizons. We are investigating ways to mitigate the impact of this curse. We are also investigating ways of reducing the computational cost, for example, by directly computing the distributions instead of using the DBN and preemptively discriminating between model updates. The new definition showed potential when bounding the final error due to replacing some candidate models with

an approximate representative. However, this error bound only applies when lower level models are solved exactly. This is a problem as it is the lower levels which offer the greatest potential for savings. We are also currently working on ways to usefully bound the error when these lower level models are solved approximately. We are optimistic that all of this can be done in a systematic way, and this will facilitate application to larger multi-agent problem domains.

BIBLIOGRAPHY

[1] Hugin expert: The leading decision support tool. www.hugin.com.

[2] D. Aberdeen. A survey of approximate methods for solving partially observable markov decision processes. In *Technical report*, National ICT Australia.

[3] R. J. Aumann. Interactive epistemology i: Knowledge. In *International Journal of Game Theory*, pages 263–300, 1999.

[4] D. E. Bell, H. Raiffa, and A. Tversky. Decision making: Descriptive, normative, and prescriptive interactions. Cambridge University Press, 1988.

[5] C. Boutilier. Sequential optimality and coordination in multiagent systems. In *International Joint Conference on Artificial Intelligence*, pages 478–485, 1999.

[6] C. Boutilier and D. Poole. Computing optimal policies for partially observable decision processes using compact representations. In *Association for the Advancement of Artificial Intelligence*, pages 1168–1175, 1996.

[7] C. Camerer. Behavioral game theory: Experiments in strategic interaction. Princeton University Press, 2003.

[8] A. R. Cassandra, M. L. Littman, and N. L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In *Uncertainty in Artificial Intelligence*, 1997.

[9] J. M. Charnes and P. Shenoy. Multistage monte carlo methods for solving influence diagrams using local computation. In *Management Science*, pages 405–418, 2004.

[10] P. Doshi. *Optimal sequential planning in partially observable multiagent settings*. PhD thesis, University of Illinois, 2005.

[11] P. Doshi and P. Gmytrasiewicz. On the difficulty of achieving equilibrium in interactive pomdps. In *International Symposium of AI and Math*, 2006.

[12] P. Doshi and Y. Zeng. Improved approximation of interactive dynamic influence diagrams using discriminative model updates. In *Autonomous Agents and Multiagent Systems*, pages 907–914, 2009.

[13] P. Doshi, Y. Zeng, and Q. Chen. Graphical models for online solutions to interactive pomdps. In *Autonomous Agents and Multiagent Systems*, pages 809–816, 2007.

[14] P. Doshi, Y. Zeng, and Q. Chen. Graphical models for interactive pomdps: Representations and solutions. In *Journal of Autonomous Agents and Multiagent Systems*, pages 376–416, 2009.

[15] G. E.Monahan. A survey of partially observable markov decision processes: Theory, models, and algorithms. In *Management Science*, pages 1–16, 1982.

[16] D. Fudenberg and D. K. Levine. *The theory of learning in games*. MIT Press, 1998.

[17] D. Fudenberg and J. Tirole. *Game theory*. MIT Press, 1991.

[18] K. Gal and A. Pfeffer. Networks of influence diagrams: A formalism for representing agents beliefs and decision-making processes. In *Journal of Artificial Intelligence Research*, pages 109–147, 2008.

[19] Y. Gal and A. Pfeffer. A language for modeling agents decision-making processes in games. In *Autonomous Agents and Multiagent Systems*, pages 265–272, 2003.

[20] P. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multiagent settings. In *Journal of Artificial Intelligence Research*, pages 49–79, 2005.

[21] E. Hansen, D. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *Association for the Advancement of Artificial Intelligence*, 2004.

[22] J. C. Harsanyi. Games with incomplete information played by bayesian players. In *Management Science*, pages 159–182, 1967.

[23] M. Hauskrecht. Value-function approximations for partially observable markov decision process. In *Journal of Artificial Intelligence*, 2000.

[24] R. A. Howard and J. E. Matheson. Influence diagrams. In *Readings on the Principles and Applications of Decision Analysis*, pages 721–762, 1984.

[25] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. In *Artificial Intelligence*, pages 99–134, 1998.

[26] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. In *International Joint Conferences on Artificial Intelligence*, pages 1027–1034, 2001.

[27] S. Kullback and R. Leibler. On information and sufficiency. In *Annals of Mathematical Statistics*, pages 79–86, 1951.

[28] W. S. Lovejoy. Computationally feasible bounds for partially observed markov decision processes. In *Operations Research*, pages 162–175, 1991.

[29] J. F. Mertens and S. Zamir. Formulation of bayesian analysis for games with incomplete information. In *International Journal of Game Theory*, pages 1–29, 1985.

[30] R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella. Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In *International Joint Conferences on Artificial Intelligence*, 2005.

[31] D. Nilsson and S. Lauritzen. Evaluating influence diagrams using limids. In *Uncertainty in Artificial Intelligence*, pages 436–445, 2000.

[32] J. Pearl. Probabilistic reasoning in intelligent systems: Networks of plausible inference. In *Morgan-Kaufmann: Los Altos, California*, 1988.

[33] J. Pineau, G. Gordon, and S. Thrun. Anytime point-based value iteration for large pomdps. In *Journal of Artificial Intelligence Research*, pages 335–380, 2006.

[34] K. Polich and P. Gmytrasiewicz. Interactive dynamic influence diagrams. In *Autonomous Agents and Multiagent Systems*, 2006.

[35] M. L. Puterman. Markov decision processes:discrete stochastic dynamic programming. Wiley series in probability and mathematical statistics. Wiley-Interscience, 1994.

[36] D. Pynadath and S. Marsella. Minimal mental models. In *Association for the Advancement of Artificial Intelligence*, pages 1038–1044, 2007.

[37] B. Rathnasabapathy, P. Doshi, and P. J. Gmytrasiewicz. Exact solutions to interactive pomdps using behavioral equivalence. In *Autonomous Agents and Multiagent Systems*, pages 1025–1032, 2006.

[38] S. Russell and P. Norvig. *Artificial Intelligence, a modern approach*. Prentice Hall, 2003.

[39] S. Seuken and S. Zilberstein. Memory bounded dynamic programming for dec-pomdps. In *International Joint Conferences on Artificial Intelligence*, 2007.

[40] R. D. Shachter. Evaluating influence diagrams. In *Operations Research*, pages 871–882, 1986.

[41] R. Smallwood and E. Sondik. The optimal control of partially observable markov decision processes over a finite horizon. In *Operations Research*, pages 1071–1088, 1973.

[42] D. Suryadi and P. Gmytrasiewicz. Learning models of other agents using influence diagrams. In *User Modeling*, pages 223–232, 1999.

[43] D. Szer and F. Charpillet. Point based dynamic programming for dec-pomdps. In *Association for the Advancement of Artificial Intelligence*, 2006.

[44] J. A. Tatman and R. D. Shachter. Dynamic programming and influence diagrams. In *IEEE Transactions on Systems, Man, and Cybernetics*, pages 365–379, 1990.

[45] A. Turing. Computing machinery and intelligence. In *Mind LIX*, pages 433–460, 1950.

[46] Y. Zeng, P. Doshi, and Q. Chen. Approximate solutions of interactive dynamic influence diagrams using model clustering. In *Association for the Advancement of Artificial Intelligence*, pages 782–787, 2007.