GRAPHICAL MODELING OF ASYMMETRIC GAMES AND VALUE OF INFORMATION IN MULTI-AGENT DECISION SYSTEMS

WANG XIAOYING

NATIONAL UNIVERSITY OF SINGAPORE

2007

GRAPHICAL MODELING OF ASYMMETRIC GAMES AND VALUE OF INFORMATION IN MULTI-AGENT DECISION SYSTEMS

WANG XIAOYING

(B.Mgt., Zhejiang University)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF INDUSTRIAL & SYSTEMS ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2007

ACKNOWLEDGEMENTS

I would like to express my thanks and appreciations to Professor Poh Kim Leng and Professor Leong Tze Yun.

My supervisor, Professor Poh Kim Leng, provided much guidance, support, encouragement and invaluable advices during the entire process of my research. He introduced me the interesting research area of decision analysis and discussed with me the ideas in this area. Professor Leong Tze Yun provided insightful suggestions to my topic during our group seminars.

Dr. Zeng Yifeng guided me in the research area of decision analysis and discussed with me on this research topic.

All the past and present students, researchers in the Department of Industrial & Systems Engineering and Bio-medical Decision Engineering (BiDE), serve as a constant source of advices and intellectual support.

To my parents, I owe the greatest debt of gratitude for their constant love, confidence and encouragement on me.

Special thanks to my boyfriend Zhou Mi for his support on me during the entire course of writing this thesis and his help with the revision, my labmate Han Yongbin for helping me with the format and my dearest friend Wang Yuan, Guo Lei for their warm encouragements.

Table of Contents

1 Introduction 1

1.1 Background and Motivation 2

1.2 Multi-agent Decision Problems 4

1.3 Objectives and Methodologies 5

1.4 Contributions 5

1.5 Overview of the Thesis 7

2 Literature Review 9

2.1 Graphical Models of Representing Single Agent Decision Problems 9

2.1.1 Bayesian Networks 9

2.1.2 Influence Diagrams 13

2.1.3 Asymmetric Problems in Single Agent Decision Systems 16

2.2 Multi-agent Decision Systems 19

2.3 Graphical Models of Representing Multi-agent Decision Problems 22

2.3.1 Extensive Form Game Trees 22

2.3.2 Multi-agent Influence Diagrams 23

2.4 Value of Information (VOI) in Decision Systems 28

2.4.1 Value of Information in Single Agent Decision Systems 28

2.4.2 Computation of EVPI 29

3 Asymmetric Multi-agent Influence Diagrams: Model Representation 35

3.1 Introduction 35

3.2 Asymmetric Multi-agent Decision Problems 38

3.2.1 Different Branches of Tree Containing Different Number of Nodes …………………………………………………………………38
3.2.2 Different Branches of Tree Involves Different Agents................ 41
3.2.3 Player's Choices are Different in Different Branches of Tree...... 44
3.2.4 Different Branches of Tree Associated with Different Decision Sequences... 45
3.3 Asymmetric Multi-agent Influence Diagrams 46

4 Asymmetric Multi-agent Influence Diagrams: Model Evaluation... 56

4.1 Introduction... 56
4.2 Relevance Graph and S-Reachability in AMAID................................ 58
4.3 Solution for AMAID... 61
4.3.1 AMAID With Acyclic Relevance Graph..................................... 61
4.3.2 AMAID With Cyclic Relevance Graph....................................... 65
4.4 A Numerical Example... 67
4.5 Discussions 69

5 Value of Information in Multi-agent Decision Systems......... 71

5.1 Incorporating MAID into VOI Computation....................................... 71
5.1.1 N is Observed by Agent A Prior to D_a 74
5.1.2 N is Observed by Agent B Prior to D_b 77
5.1.3 N is Observed by Both Agents A and B 78
5.2 VOI in Multi-agent Systems – Some Discussions and Definitions...... 80
5.3 Numerical Examples... 88
5.4 Value of Information for the Intervened Variables in Multi-agent Decision Systems.. 95
5.4.1 Problem.. 95

5.4.2 Canonical Form of MAIDs 98
5.4.3 Independence Assumption in Canonical Form of MAID 101
6 Qualitative Analysis of VOI in Multi-agent Systems 103
6.1 Introduction 103
6.2 Value of Nature Information in Multi-agent Decision Systems 105
6.3 Value of Moving Information in Multi-agent Decision Systems 114
6.4 Examples 117
7 Conclusion and Future Work 120
7.1 Conclusion 120
7.2 Future Work 123
Reference 124

Summary

Multi-agent decision problem under uncertainty is complicated since it involves a lot of interacting agents. The Pareto optimal set does not remain to be the Nash equilibria in multi-agent decision systems. Many graphical models have been proposed to represent the interactive decisions and actions among agents. Multi-agent Influence Diagrams (MAIDs) are one of them, which explicitly reveal the dependence relationship between chance nodes and decision nodes compared to extensive form trees. However, when representing an asymmetric problem in multi-agent systems, MAIDs do not turn out to be more concise than extensive form trees.

In this work, a new graphical model called Asymmetric Multi-agent Influence Diagrams (AMAIDs) is proposed to represent asymmetric decision problems in multi-agent decision systems. This framework extends MAIDs to represent asymmetric problems more compactly while not losing the advantages of MAIDs. An evaluation algorithm adapted from the algorithm of solving MAIDs is used to solve AMAID model.

Value of information (VOI) analysis has been an important tool for sensitivity analysis in single agent systems. However, little research has been done on VOI in the multi-agent decision systems. Works on games have discussed value of information based on game theory. This thesis opens the discussion of VOI based on the graphical representation of multi-agent decision problems and tries to unravel the properties of VOI from the structure of the graphical models. Results turn out that information value could be less than zero in multi-agent decision systems because of the interactions among agents. Therefore, properties of VOI become much more complex in multi-agent decision systems than in single agent systems. Two types of information value in multi-agent decision systems are discussed, namely *Nature Information* and *Moving Information*. Conditional independencies and s-reachability are utilized to reveal the qualitative relevance of the variables.

VOI analysis can be applied to many practical areas to analyze the agents' behaviors, including when to hide information or release information so as to maximize the agent's own utility. Therefore, discussions in this thesis will turn out to be of interest to both researchers and practitioners.

List of Figures

Figure 2.1 An example of BN ... 10

Figure 2.2 A simple influence diagram ... 15

Figure 2.3 An example of SID ... 17

Figure 2.4 Game tree of a market entry problem ... 23

Figure 2.5 A MAID ... 26

Figure 2.6 A relevance graph of Figure 2.5 ... 27

Figure 3.1 Naive representations of Centipede Game ... 40

Figure 3.2 MAID representation of Killer Game ... 43

Figure 3.3 MAID representation of Take Away Game ... 45

Figure 3.4 MAID representation of the War Game ... 46

Figure 3.5 An AMAID representation of the Centipede Game ... 48

Figure 3.6 The cycle model ... 50

Figure 3.7 Reduced MAID by initiating D_1^1 ... 53

Figure 4.1 De-contextualize contextual utility node ... 60

Figure 4.2 Constructing the relevance graph of the AMAID ... 61

Figure 4.3 Reduced AMAID $M[D_2^{100} \mapsto A]$ of the Centipede Game ... 65

Figure 4.4 The relevance graph of the Killer Game ... 67

Figure 4.5 Numerical example of the Centipede Game ... 68

Figure 4.6 Reduced AMAID $M[D_2^{100} \mapsto A]$ ... 69

Figure 5.1 A MAID without information to any agent ... 74

Figure 5.2 A MAID with agent A knowing the information ... 77

Figure 5.3 A MAID with agent B knowing the information ... 78

Figure 5.4 A MAID with both agents A and B knowing the information ... 80

Figure 5.5 The MAIDs, relevance graphs and game tree of manufacturer example 90
Figure 5.6 An ID of decision-intervened variables in single agent decision systems 97
Figure 5.7 MAID of decision-intervened variables in Multi-agent decision system 98
Figure 5.8 Canonical Form of MAID 100
Figure 5.9 Convert MAID to canonical form. 101
Figure 6.1 An example of VOI properties 117
Figure 6.2 New model $M_{D_a^2|N_3}$, after N_3 is observed by D_a^2 118

List of Tables

Table 5.1 Utility matrices of the two manufacturers 88

Table 5.2 Expected utilities of the four situations 91

Table 5.3 Utility matrices of the two manufacturers-Example 2.......................... 92

Table 5.4 Expected utilities of the four situations-Example 2.............................. 92

[This page intentionally left blank]

1 Introduction

Decision making in our daily life is hard because the decision situations are complex and uncertain. Decision analysis provides decision makers a kind of tools for thinking systematically about hard and complex decision problems to achieve clarity of actions (Clemen 1996). If there is more than one person involved in the decision, the complexity of decision making is raised. Such decision problems are often modeled as multi-agent decision problems in which a number of agents cooperate, coordinate and negotiate with each other to achieve the best outcome in uncertain environments. In multi-agent systems, agents will be representing or acting on behalf of users and owners with very different goals and motivations in most cases. Therefore, the same problems under single agent systems and multi-agent systems would sometimes generate quite different outcomes and properties.

The theories in multi-agent decision systems provide a foundation of this thesis. In this chapter, we will introduce the motivation of writing this thesis and define the basic problem addressed in this thesis. The last section of this chapter gives an overview of the remainder of the thesis.

1.1 Background and Motivation

Making a good decision in a multi-agent system is complicated since both the nature of decision scenarios and the attributes of multiple agents have to be considered. However, such situation is always unavoidable since people are always involved into a large social network. Therefore, analyzing, representing and solving decision problems under such circumstances become meaningful.

Many graphical models in single agent areas have been extended to model and solve decision problems in multi-agent areas, such as Multi-agent Influence Diagrams (MAIDs). MAIDs extend Influence Diagrams (IDs) to model the relevance between chance nodes and decision nodes in multi-agent decision systems. They successfully reveal the dependency relationships among variables, of which extensive game trees lack. However, in representing asymmetric decision problems, the specification load of a MAID is often worse than an extensive game tree. Hence, a new graphical model is needed for representing and solving these asymmetric decision problems. Examples in this thesis will show the practical value of our proposed models.

On the other hand, when agents make decisions in a decision system, information puts a direct influence on the quality of the decisions(Howard 1966). Agents can be better off or worse off by knowing a piece of information and the time to know this information. Information value plays an important role in the decision making process of agents. For example, in Prisoner's Dilemma game, one prisoner can get higher payoff if he/she knows the decision of another prisoner. Since information gathering is usually associated with a cost, computing how much value of this information will add to the total benefit has been a focus for agents.

Until now, researches on value of information (VOI) have been confined in the single agent decision systems. Information value involving multiple agents has been discussed in games. They use mathematical inductions and theorems to discuss the influence of information structure and agents' payoff functions on the sign of information value. Many properties of VOI in multi-agent decision systems have not been revealed yet. Different kinds of information values have not been categorized. Recently, researches in decision analysis have developed graphical probabilistic representation to model decision problems. This work opens the discussion of VOI based on the graphical models.

1.2 Multi-agent Decision Problems

This work is based on multi-agent decision systems, which have different characteristics from single agent decision systems. Firstly, a multi-agent decision problem involves a group of agents, while a single agent decision problem only involves one agent. Secondly, those agents have intervened actions or decisions because their payoff functions are influenced by other agents' actions. Thirdly, each agent's decision may be observed or not observed by other agents, while a decision maker always observes its previous decisions in a single agent decision system. Fourthly, agents can cooperate or compete with each other; Fifthly, agents have their individual objectives although they may seek a cooperative solution. Every agent is selfish and seeks to maximize its own utility, without considering others' utilities. They cooperate with each other by sharing information. Because of these differences, decision problems in multi-agent decision systems and single agent decision systems are quite different. In multi-agent decision models, decision interaction among agents is an interesting and essential problem. The output of a multi-agent decision model may not always be a Pareto optimality set, but the Nash equilibria. However, in single agent systems, the output of the model is always a Pareto optimality set.

1.3 Objectives and Methodologies

The goal of this thesis is to establish a new graphical model for representing and solving asymmetric problems in multi-agent decision systems, as well as discussing value of information in multi-agent decision systems. To achieve this goal, we carry out the stages as follows:

First of all, we build a new flexible framework. The main advantage of this decision-theoretic framework lies in its capability for representing asymmetric decision problems in multi-agent decision systems. It encodes the asymmetries concisely and naturally while maintaining the advantages of MAID. Therefore, it can be utilized to model complex asymmetric problems in multi-agent decision systems.

The evaluation algorithm of MAIDs is then extended to solve this model based on the strategic relevance of agents.

1.4 Contributions

The major contributions of this work are as follows:

Firstly, we propose a new graphical model to represent asymmetric multi-agent decision problems. Four kinds of asymmetric multi-agent decision problems are discussed. This framework is argued for its ability to represent these kinds of asymmetric problems concisely and naturally compared to the existing models. It enriches the graphical languages for modeling multiple agent actions and interactions.

Secondly, the evaluation algorithm is adopted to solve the graphical model. Extending from the algorithm of solving MAIDs, this algorithm is shown to be effective and efficient in solving this model.

Thirdly, we open the door of discussing value of information based on the graphical model in multi-agent decision systems. We define some important and basic concepts of VOI in multi-agent decision systems. Ways of VOI computation using existing MAIDs are studied.

Fourthly, some important qualitative properties of VOI are revealed and verified in multi-agent systems, which also facilitate fast VOI identification in the real world.

Knowledge of VOI of both chance nodes and decision nodes based on a graphical model can guide decision analyst and automated decision systems in gathering information by weighing the importance and information relevance of each node. The methods described in this work will serve this purpose well.

1.5 Overview of the Thesis

This chapter has given some basic ideas in decision analysis, introduced the objective and motivation of this thesis and described the methodologies used and the contributions in a broad way.

The rest of this thesis is organized as follows:

Chapter 2 introduces related work involving graphical models and evaluation methods both in single agent decision system and multi-agent decision system. Most of current work on VOI computation in single agent decision systems is also covered.

Chapter 3 proposes a graphical multi-agent decision model to represent asymmetric multi-agent decision problems. Four main types of asymmetric problems are discussed and the characteristics of this new model are highlighted.

Chapter 4 presents the algorithm for solving this new decision model. The complexity problem is discussed in this section as well.

Chapter 5 defines VOI in multi-agent decision systems illustrated by a basic model of multi-agent decision systems. Different kinds of information value are categorized. A numerical example is used to illustrate some important properties of VOI in multi-agent decision systems.

Chapter 6 verifies some qualitative properties of VOI in multi-agent decision systems based on the graphical model.

Chapter 7 summarizes this thesis by discussing the contributions and limitations of the work. It also suggests some possible directions for future work.

2 Literature Review

This chapter briefly surveys some related work: graphical models for representing single agent decision problems, graphical models for representing multi-agent decision problems, multi-agent decision systems, and value of information in single agent decision systems. This survey provides a background for a more detailed analysis in the following chapters and serves as a basis to the extension of these existing methodologies.

2.1 Graphical Models for Representing Single Agent Decision Problems

2.1.1 Bayesian Networks

Bayesian networks are the fundamental graphical modeling language in probabilistic modeling and reasoning. A Bayesian network (Pearl 1998; Neapolitan 1990; Jensen 1996; Castillo et al. 1997) is a triplet (X, A, P) in which X is the set of nodes in the graph, A is the set of directed arcs between the nodes and P is the joint probability distribution over the set of uncertain variables. Each node $x \in X$ is called a chance node in a BN which has an associated conditional probability distribution $P(x|\pi(x))$ ($\pi(x)$ denotes all x's parents) associated. The arc between nodes indicates the relevance, probabilistic or statistical correlation relationship between the variables. $P = \prod_{x \in X} p(x \mid \pi(x))$ defines a multiplicative factorization function of the conditional probability of individual variables. An example of BN is shown in Figure 2.1.

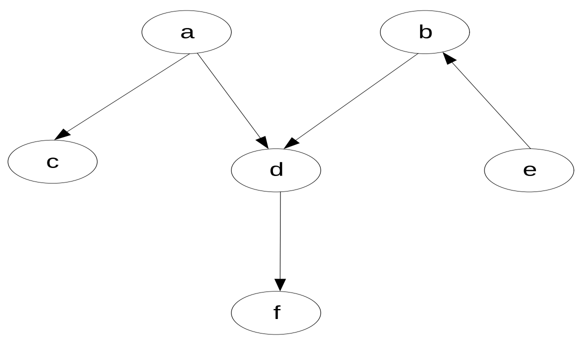


Figure 2.1 An example of BN

This BN contains six nodes $\{ a, b, c, d, e, f \}$. Each node in the BN has one conditional probability given its parents. Take node d for example, $\pi(d)=\{a, b\}$ and the conditional probability associated with it is $p(d \mid (a,b))$. BN is an *acyclic directed graph* (DAG). The joint probability distribution of a BN is defined by its DAG structure and the conditional probabilities associated with each variable. Therefore, in Figure 2.1, the joint probability distribution can be represented as: $P(a,b,c,d,e,f) = p(a)p(e)p(c \mid a)p(f \mid d)p(b \mid e)p(d \mid (a,b))$.

An important property of BNs is d-separation. The notion of d-separation can be used to identify conditional independence of any two distinct nodes in the network given any third node. The definition (Jensen, 1996 & 2001) is given below:

Definition 2.1 Let G be a directed acyclic graph and X, Y, Z are the three disjoint subsets of the nodes in G. Then X and Y are said to be d-separated by Z if for every chain from any node in X to any node in Y, the following conditions are satisfied:

1. If an intermediate node A on the chain is in a converging connection(head-to-head), neither A nor its descendants are in Z;
2. If an intermediate node A on the chain is in a serial (head-to-tail) or diverging (tail-to-tail) connection, and A is in Z.

Each chain satisfying the above conditions is called *blocked*, otherwise it is *active*. In this example, nodes d and e are d-separated given node b.

Probabilistic inference in BNs has been proven to be NP-hard (Cooper 1990). In the last 20 years, various inference algorithms have been developed, including exact and approximate methods. The exact methods include Kim and Pearl's message passing algorithm (Pearl 1988; Neapolitan 1990; Russell & Norvig 2003), junction tree method (Lauritzen & Spiegelhalter 1988; Jensen et al. 1990; Shafer

1996; Madsen & Jensen 1998), cutest conditioning method (Pearl 1988; Suermondt & Cooper 1991), direct factoring method (Li & Ambrosio 1994), variable elimination method (Dechter 1996) etc.

The approximate methods include logic sampling method (Henrion 1988), likelihood weighting (Fung & Chang 1989; Shachter & Peot 1992), Gibbs sampling (Jensen 2001), self-importance sampling and heuristic-importance sampling (Shachter 1989), adaptive importance sampling (Cheng & Druzdzel 2000) and backward sampling (Fung & Favero 1994). A number of other approximate inference methods have also been proposed. Since the exact inference methods usually require a lot of computational costs, approximate algorithms are usually used for large networks. However, Dagum and Luby (1993) showed that the approximate inference methods are also NP-hard within an arbitrary tolerance.

Many extensions have been made to BNs in order to represent and solve some problems under special conditions. For example, the dynamic Bayesian networks (DBNs, Nicholson 1992; Nicholson & Brady 1992; Russell & Norvig 2003), probabilistic temporal networks (Dean & Kanazawa 1989; Dean & Wellman 1991), dynamic causal probabilistic networks (Kjaerulff 1997) and modifiable temporal belief networks (MTBNs, Aliferis et al. 1995, 1997) to model time-dependent problems. All these representations and inferences are in the framework of single agent.

2.1.2 Influence Diagrams

An influence diagram (Howard & Matheson 1984/2005; Shachter 1986) is a graphical probabilistic reasoning model used to represent single-agent decision problems.

Definition 2.2 An influence diagram is a triplet (*N*, *A*, *P*). Its elements can be defined as below:

1. $N = X \cup D \cup U$, where *X* denotes the set of chance nodes, *D* denotes the set of decision nodes and *U* denotes the set of utility nodes. A deterministic node is a special type of chance node.
2. *A* is the set of directed arcs between the nodes which represents the probabilistic relationships between the nodes;
3. *P* is the conditional probability table associated with each node. $P = P(x|\pi(x))$ for each instantiation of $\pi(x)$ where $\pi(x)$ denotes all x 's parents and $x \in N$.

Two conditions must be satisfied in an influence diagram:

1. Single Decision Maker Condition: there is only one sequence of all the decision nodes. In other words, decisions must be made sequentially because of one decision maker.
2. No-forgetting Condition: information available at one decision node is also available at its subsequent decision nodes.

In an influence diagram, rectangles represent the decision nodes, ovals represent the chance nodes and diamonds represent the value or utility. An example of the influence diagram is shown in Figure 2.2. This influence diagram comprises a set of chance nodes $\{a, b, c\}$, a decision node *d* and value node v. The chance nodes a and b are observed before decision *d*, but not chance node c. The arc from one chance node to another chance node is called a relevance arc which means the outcome of the coming chance node is relevant for assessing the incoming chance node. The arc from one chance node to one decision node is called an information arc which means the decision maker knows the outcome of the chance node before making this decision. The corresponding chance nodes are called observed nodes, denoted as the information set *I(D)*. The arc from a decision node to a chance node is called an influence arc which means the decision outcome will influence the probability of the chance node.

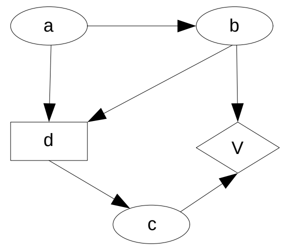


Figure 2.2 A simple influence diagram

The evaluation methods for solving influence diagrams include the reduction algorithm (Shachter 1996, 1988) and strong junction tree (Jensen et al. 1994). The reduction algorithm reduces the influence diagram by methods of node removal and arc reversal. The strong junction tree algorithm first transforms the influence diagram into the moral graph, then triangulates the moral graph following the strong elimination order and finally uses the message passing algorithm to evaluate the strong junction tree constructed from the strong triangulation graph (Nielsen 2001).

Influence diagrams involve one decision maker in a symmetric situation. Some extensions have been proposed to solve other problems under different situations. For example, Dynamic Influence Diagrams (DIDs, Tatman & Shachter 1990), Valuation Bayesian Networks (VBs, Shenoy 1992), Multi-level Influence Diagrams (MLIDs, Wu & Poh 1998), Time-Critical Dynamic Influence Diagrams (TDIDs, Xiang & Poh 1999), Limited Memory Influence Diagrams (LIMIDs, Lauritzen & Vomlelova 2001), Unconstrained Influence Diagrams (UIDs, Jensen

& Vomlelova 2002) and Sequential Influence Diagrams (SIDs, Jensen et al. 2004).

2.1.3 Asymmetric Problems in Single Agent Decision Systems

A decision problem is defined to be *asymmetric* if 1) the number of scenarios is not the same as the elements' number in the Cartesian product of the state spaces of all chance and decision variables in all its decision tree representation; or 2) the sequence of chance and decision variables is not the same in all scenarios in one decision tree representation.

Although IDs are limited in its capability of representing asymmetric decision problems, it provides a basis for extension to solve asymmetric decision problems involving one decision maker, such as Asymmetric Influence Diagrams (AIDs, Smith et al. 1993), Asymmetric Valuation Networks (AVNs, Shenoy 1993b, 1996), Sequential Decision Diagrams (SDDs, Covaliu and Oliver 1995), Unconstrained Influence Diagrams (UIDs, Jensen & Vomlelova 2002), Sequential Influence Diagrams (SIDs, Jensen et al. 2004) and Sequential Valuation Networks (SVNs, Demirer and Shenoy 2006). All these works aim to solve the asymmetric problems under the framework of single agent. None of them is able to represent the asymmetric problems in multi-agent decision systems.

2.1.3.1 Sequential Influence Diagrams

Sequential Influence Diagrams (SIDs, Jensen et al. 2004) are a graphical language for representing asymmetric decision problems involving one decision maker. It inherits the compactness of IDs and extends the expressiveness of IDs in the meantime. There are mainly three types of asymmetries in the single agent decision systems: structural asymmetry, order asymmetry and the asymmetry combined with both structural and order. SIDs are proposed to effectively represent these three asymmetries. The SIDs can be viewed as the combination of the two diagrams. One diagram reveals the information precedence including the asymmetric information. The other diagram represents the functional and probabilistic relations. SIDs are also composed of chance nodes, decision nodes and value nodes. Figure 2.3 shows an example of SID.

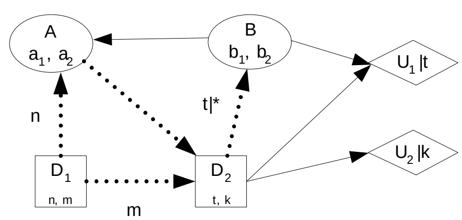


Figure 2.3 An example of SID; The * denotes that the choice D_2=*t* is only allowed when $(D_1 = m) \cup (D_1 = n \cap (A = a_1))$ is satisfied.

The dashed arrow in Figure 2.3 is also called *structural arc* which encodes the information precedence and asymmetric structure of the decision problem. A *guard* may be associated with a structural arc, which is composed of two parts. One part describes the fulfilled context. When the context is fulfilled, the arc is *open*. The other part states the constraints when the context will be fulfilled. For example, in Figure 2.3, the guard n on the dashed arc from node D_1 to A means the next node in all scenarios is A whenever $D_1=n$ and this guard only has one part because the context $D_1=n$ is unconstrained. However, the guard t|* on the dashed arc from node D_2 to B means that the context $D_2=t$ is only allowed when $(D_1 = m) \cup (D_1 = n \cap (A = a_1))$ is satisfied. Therefore, it is composed of two parts. The solid arc serves as the same function as in IDs.

The SIDs are solved by decomposing the asymmetric problem into small symmetric sub-problems which are then organized in a *decomposition graph* (Jensen et al. 2004) and propagating the probability and utility potentials upwards from the root nodes of the decomposition graph.

2.1.3.2 Other Decision Models for Representing Asymmetric Decision Problems

One direct way to represent asymmetric decision problems to use refined decision trees called coalescence (Olmsted 1983) decision tree approach. This method encodes the asymmetries with a natural way which is easy to understand and solve. However, the disadvantage is the decision tree grows exponentially as the decision problem gets larger. The automating coalescence in decision trees is not easy as well since it involves first constructing the uncoalesced tree and then recognizing repeated subtrees. Therefore, it is only limited to small problems. Asymmetric Influence Diagrams (AIDs, Smith et al. 1993) extend IDs using the notion of distribution tree which captures the asymmetric structure of the decision problems. The representation is compact but it has redundant information both in IDs and distribution trees. Asymmetric Valuation Networks (AVNs, Shenoy 1993b, 1996) are based on valuation networks (VNs, Shenoy 1993a) which consist of two types of nodes: variable and valuation. This technique captures asymmetries by using indicator valuations and effective state spaces. Indicator valuation encodes structural asymmetry with no redundancy. However, AVNs are not as intuitive as IDs in modeling of conditionals. Besides, they are unable to model some asymmetries. Sequential Decision Diagrams (SDDs, Covaliu and Oliver 1995) use two directed graphs to model a decision problem. One is an ID to describe the probability model and another sequential decision diagram to capture the asymmetric and information constraints of the problem. This technique can represent asymmetry compactly but there is information duplication in the two graphs. The probability model in this approach cannot be represented consistently.

2.2 Multi-agent Decision Systems

The trend of interconnection and distribution in computer systems have led to the emergence of a new field in computer science: multi-agent systems. An agent is a computer system which is situated in a certain environment and is able to act independently on behalf of its user or owner (Wooldridge & Jennings 1995). Intelligent agents have the following capabilities: 1) Reactivity: they can respond to the changes in the environment in order to satisfy its design objectives; 2) Pro-activeness: they can take the initiative to exhibit goal-directed behavior; 3) Social ability: they can interact with other agents to satisfy their design objectives.

A multi-agent system (Wooldridge 2002) is a system comprising a number of agents interacting with each other by communication. Different agents in the systems may have different "spheres of influence" with a self-organized structure to achieve some goals together (Jennings 2000). There are five types of organizational relationships among these agents (Zambonelli et al. 2001): Control, Peer, Benevolence, Dependency and Ownership. The interactions among different agents include competition and cooperation. Grouped in different organizations, different agents can interact with other agents both inside and outside of the organization to achieve certain objectives in a system, which is called a multi-agent decision system.

Currently, many studies carried out on multi-agent systems are connected with game theory. The tools and techniques discovered in game theory have found many applications in computational multi-agent systems research. Efficient computation of Nash equilibria has been one of the main foci in multi-agent systems. Nash equilibrium is the state when no agent has any incentive to deviate from. Parts of the research focus on the probabilistic graphical models to represent games and compute Nash equilibria. For example, game tree (von Neumann and Morgenstern 1994) represents the agents' actions by nodes and branches. Expected Utility Networks (EUNs, La Mura & Shoham 1999) and Game Networks (G nets, La Mura 2000) incorporate both the probabilistic and utility independence in a multi-agent system. Some algorithms have also been developed for identifying equilibrium in games. TreeNash algorithm (Kearns et al. 2001a, 2001b) treats the global game as being composed of interacting local games and then computes approximate Nash equilibria in one-stage games. Hybrid algorithm (Vickrey & Koller 2002) is based on hill-climbing approach to optimize a global score function, the optima of which are precisely equilibria. Constraint satisfaction algorithm (CSP, Vickrey & Koller 2002) uses a constraint satisfaction approach over a discrete space of agent strategies.

All these research work above adopts a game-theoretic way to represent the interaction between agents and seeks the equilibria among agents. Some related graphical models will be introduced in the next section.

2.3 Graphical Models for Representing Multi-agent Decision Problems

2.3.1 Extensive Form Game Trees

Extensive form tree is developed by von Neumann and Morgenstern when representing n-person games. A completed game tree is composed of chance and decision nodes, branches, possible consequences and information sets. The main difference between decision trees and game trees is the representations of information constraints. In decision trees, the information constraints are represented by the sequence of the chance and decision nodes in each scenario, while in game trees, the information constraints are represented by information sets.

An information set is defined as a set of nodes where a player cannot tell which node in the information set he/she is at. Figure 2.4 shows a game tree for a market entry problem. The nodes connected by one dashed line are in the same information set.

The disadvantage of the game tree is that it obscures the important dependence relationships which are often present in the real world scenarios.

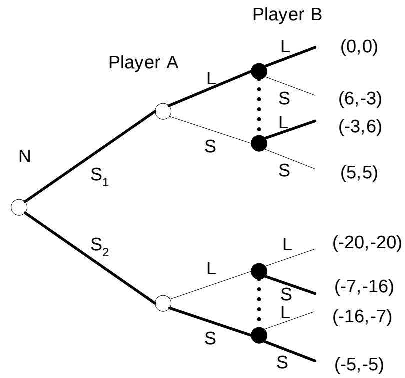


Figure 2.4 Game tree of a market entry problem

2.3.2 Multi-agent Influence Diagrams

In multi-agent decision systems, *multi-agent influence diagram*s (MAIDs, Koller and Milch 2001) are considered as a milestone in representing and solving games. It allows domain experts to compactly and concisely represent the decision problems involving multiple decision-makers. A qualitative notion of strategic relevance is used in MAIDs to decompose a complex game into several interacting simple games, where a global equilibrium of the complex game can be found through the local computation of the relatively simple games. Formally, the definition of a MAID is given as follows (Koller and Milch 2001).

Definition 2.3 A MAID *M* is a triplet (*N, A, P*). *N*= $\chi \cup D \cup U$ is the set of uncertain nodes, where χ is the set of chance nodes which represents the decisions of nature, $D = \bigcup_{a\in \mathrm{A}} D_a$ represents the set of all the agents' decision nodes, $U = \bigcup_{a\in \mathrm{A}} U_a$ represents the set of all the agents' utility nodes. *I* is the set of directed arcs between the nodes in the directed acyclic graph (DAG). Let x be a variable and $\pi(x)$ be the set of x's parents. For each instantiation $\pi(x)$ and x, there is a conditional probability distribution (CPD): $P(x|\pi(x))$ associated. If $x \in D$, then $P(x|\pi(x))$ is called a *decision rule* ($\sigma(x)$) for this decision variable x. A *strategy profile* σ is an assignment of decision rules to all the decisions of all the agents. The joint distribution $P_{M[\sigma]} = \prod_{x \in \chi \cup U} P(x|\pi(x)) \prod_{x \in D} \sigma(x)$.

It can be seen that a MAID involves a set of agents A. Therefore, different decision nodes and utility nodes are associated with different agents. The "no-forgetting condition" is still satisfied in the MAID representation. However, in MAIDs, it means that the information available at the previous decision point is still available at subsequent decision point of the same agent.

Once σ assigns a decision rule to all the decision nodes in a MAID *M*, all the decision nodes are just like chance nodes in BN and the joint distribution $P_{M[\sigma]}$ is the distribution over *N* defined by the BN. The expected utility of each agent a for the strategy profile σ is:

$$\mathrm{EU}_a(\sigma) = \sum_{U \in U_a} \sum_{u \in dom(U)} \mathrm{P}_{\mathrm{M}[\sigma]}(U = u) \cdot u$$

Definition 2.4 Giving decision rules for the decision nodes in the set $\varepsilon \subset D_a$, a strategy σ_{ε}^{*} is optimal for the strategy profile in the MAID $M_{[-\varepsilon]}$, where all the decisions not in ε have been assigned with decision rules, σ_{ε}^{*} has a higher expected utility than any other strategy $\sigma_{\varepsilon}^{'}$ over ε.

This definition illustrates that σ_{ε}^{*} is the local optimal solution of the decisions in $M_{[-\varepsilon]}$.

Definition 2.5 A strategy profile σ is a Nash equilibrium if $\sigma(D_a)$ is optimal for all the agents $a \in A$.

An example of MAID is shown in Figure 2.5. The MAID is a DAG which comprises of two agents' decision nodes and utility nodes. They are represented with different colors. The total utility of each agent a given a specific instantiation of *N* is the sum of the values of all a's utility nodes given this instantiation. In this figure, agent *B*'s total utility is the sum of *B*'s utility 1 and 2 given an instantiation of all the nodes *N*. The dashed line in the graph represents the information precedence when agents make decisions. In this figure, agent *A* knows his first decision and *B*'s decision when *A* makes his/her second decision and *B* observes chance node 1 but not *A*'s first decision when he/she makes decision.

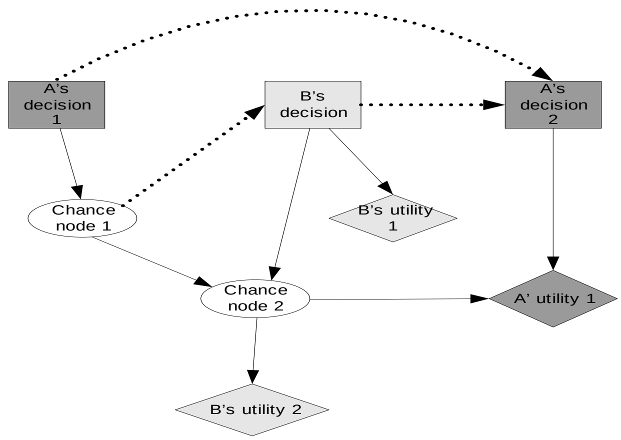


Figure 2.5 A MAID

MAIDs address the issue of non-cooperative agents in a compact model and reveal the probabilistic dependence relationships among variables. Once a MAID is constructed, strategic relevance can be determined solely on the graph structure of the MAID and a strategic relevance graph can be drawn to represent the direct relevance relationships among the decision variables.

We can then draw a *strategic relevance graph* to represent the strategic relationship by adding a directed arc from *D* to *D*' if *D* relies on *D'*. Once the relevance graph is constructed, a divide-and-conquer algorithm (Koller and Milch 2001) can be used to compute the Nash equilibrium of the MAIDs. One example of the relevance graph of Figure 2.5 is shown in Figure 2.6.

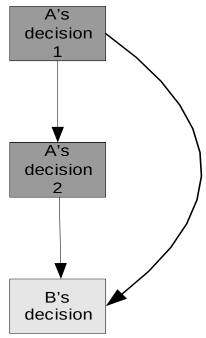


Figure 2.6 A relevance graph of Figure 2.5

With its explicit expression and efficient computing methods, a MAID provides a good solution for representing and solving non-cooperative multi-agent problems. On the other hand, this representation becomes intractably large under asymmetric situations. However, it provides a foundation for us for further development when dealing with asymmetric problems.

Koller and Milch (2001) suggested extending MAIDs to asymmetric situations using context-specificity (Boutilier et al. 1996; Smith et al. 1993). Context can be defined as an assignment of values to a set of variables in the probabilistic sense. This suggestion may be able to integrate the advantages of game tree and MAID representations.

2.4 Value of Information (VOI) in Decision Systems

2.4.1 Value of Information in Single Agent Decision Systems

In single agent systems, VOI analysis has been used as an efficient tool for sensitivity analysis. Calculating VOI can help the decision maker to decide whether it is worthwhile to collect that piece of information and identify which piece of information is the most valuable one to acquire. VOI can be defined as the difference between expected value with information and without information. If the information is complete, then VOI is also called *expected value of perfect information* (EVPI). Otherwise VOI can be called *expected value of imperfect information* (EVIPI). In single-agent decision model, VOI is lower bounded by 0 and upper bounded by EVPI. Therefore, calculating EVPI is important in VOI analysis.

EVPI on an uncertain variable is the difference between expected value with perfect information of that variable and without (Howard, 1996b and 1967). Given a new piece of information X of the uncertain parameters in a decision model I, the EVPI of X is as follows

$$EVPI(X) = E(V_d \mid X, \varepsilon) - E(V_{d_0} \mid \varepsilon) \tag{2.1}$$

In this formula, $d, d_0 \in D$ represents the best decision taken with and without information respectively. E denotes taking expectation and ε denotes the background information. $E(V_d \mid X, \varepsilon)$ is the expected value given information X and background information ε. $E(V_{d_0} \mid \varepsilon)$ is the expected value given background information ε.

From formula (2.1), we can see that EVPI (X, ε) is the average improvement expected to gain resulting from the decision maker's decision choice given the perfect information before making the decision. It represents the maximum amount one should be willing to pay for that piece of perfect information.

2.4.2 Computation of EVPI

Research on computing EVPI can be divided into two groups: qualitative analysis of EVPI and quantitative computation of EVPI. The quantitative computation includes the exact computation and approximate computation.

The traditional economic evaluation of information is introduced by Howard (1966, 1967). In his evaluation, EVPI is calculated by the expected value given the outcomes of the variable minus the expected value without knowing the outcomes of the variable.

Value of evidence (VOE, Ezawa 1994) is a measure of experiment to find out what evidence we would like to observe and what the maximum benefit we can receive from the observation of an evidence. It is defined as:

$$VOE(X_J = x_j) = Max\ EV(X \setminus X_J, X_J = x_j) - Max\ EV(X)$$

For the state space Ω_J of node J. (2.2)

In Formula (2.2), J is the chance node and X_J is the chance variable associated with it. x_j is one instantiation of X_J. $X \setminus X_J$ is the set of chance variables excluding X_J and EV is the expected value. The EVPI given X_J can be defined as:

$$EVPI(X_J) = MaxEV(X \setminus \{D, X_J\}, D \setminus X_J, X_J) - MaxEV(X)$$

For the state space Ω_J of node J. (2.3)

which can then be represented as a function of VOE:

$$EVPI(X_J) = \sum VOE(X_J = x_j) * \Pr\{x_j\}$$

For the state space Ω_J of node J. (2.4)

From formula (2.3), we can see that the EVPI computed from VOE is the EVPI for all the decisions, assuming the evidence is observed before the first decision. Besides, the value of evidence can be negative, but the value of perfect information is always greater than or equal to 0. Note here the value of evidence is different from value of information, since a piece of evidence may have negative impact on the total expected value, but information value can never be negative in single agent decision systems.

Once the evidence x_j is propagated, when the decision maker makes the next decision (remove decision node), this information is already absorbed. Hence by weighing the value of evidence for each x_j with $\Pr\{x_j\}$, we can compute the value of perfect information. The unconditional probability $\Pr\{x_J\}$ can always be obtained by applying arc reversals (Shachter, 1986) between its predecessors, as long as they are not decision nodes.

This method of calculating EVPI is based on VOE, the computation efficiency of which is based on the efficiency of propagation algorithm for influence diagram. In practical usage, when the problem gets large, the computation of EVPI becomes intractable. Under this circumstance, some assumptions have been made to simplify the computation problem. Myopic value of information (Dittmer & Jensen 1997) computation is among one of them. The myopic assumption assumes that the decision maker can only consider whether to observe one more piece of information even when there is an opportunity to make more observations. This method of calculating the expected value of information is based on the strong junction tree framework (Jensen et al. 1994) corresponding to the original influence diagram. The computation procedure for both scenarios, with and without information, can make use of the same junction tree with only a number of tables expanded but not recalculated. Its disadvantage is its limitation in the myopic assumption.

The approximate EVPI computations include the non-myopic approximation method (Heckerman et al. 1991) and Monte Carlo Simulation. The non-myopic approximation method is used as an alternative to the myopic analysis for identifying cost-effective evidence. It assumes linearity in the number of a set of tests which is exponential in the exact computation. The steps of this method are as follows. First, use myopic analysis to calculate the net value of information for each piece of evidence. Second, arrange the evidences in descending order according to their net values of information, and finally compute the net value of information of each m-variable subsequence of the pieces of evidence starting from the first to identify evidence whose observation is cost effective. Because this approach uses the central-limit theorem to compute the value of information, it is limited to the problem with independent or special dependent distribution evidences where the central-limit theorem is valid.

Another traditional approximate method is Monte Carlo Simulation. According to each chance variable's probability distribution, we can generate great amount of random numbers and the expected utility can be determined then (Felli & Hazen, 1998). Although this approach is easy to understand, it is not space and time efficient.

Different from these quantitative methods, Poh and Horvitz (1996) proposed a graph-theoretic way to analyze information value. This approach reveals the dominance relationships of the EVPI on each chance nodes in the graphical decision models based on a consideration of the topology of the models. The EVPIs of chance nodes can then be ordered with non-numerical procedures. An algorithm based on d-separation is proposed to obtain a partial ordering of EVPI of chance nodes in a decision model with single decision node which is represented as an influence diagram expressed in canonical form (Howard, 1990).

Xu (2003) extended this method with u-separation procedure to return a partial EVPI ordering of an influence diagram.

Xu (2003) extended VOI computation to the dynamic decision systems. It is a computation based on dynamic influence diagrams (DIDs, Tatman & Shachter 1990). Different from the VOI computation based on IDs, the discount factors are considered in dynamic decision systems. The steps are as follows: first, decompose DIDs into sub-networks with similar structures. Second, generate sub-junction tree based on the sub-networks. Third, calculate the expected utility from leaf to the root node.

The above-mentioned work involves VOI analysis in single-agent decision systems. Until now, no research work has been done on VOI analysis in multi-agent decision systems. Information value involving multiple agents has been discussed in games using mathematical inductions and theorems to discuss the influence of information structure and the agents' payoff functions on the sign of information value.

3 Asymmetric Multi-agent Influence Diagrams: Model Representation

In IDs and BNs, a naïve representation of asymmetric decision problem will lead to unnecessary blowup. The same problem will be confronted in MAIDs when they are used to represent the asymmetric problems. Therefore, it is important to extend MAIDs when asymmetric situations are confronted.

This chapter discusses four kinds of asymmetric multi-agent decision problems commonly confronted and illustrates the Asymmetric Multi-agent Influence Diagrams (AMAIDs) by modeling these highly asymmetric multi-agent decision problems.

3.1 Introduction

There are mainly two popular classes of graphical languages for representing multi-agent decision problems, namely game trees and Multi-agent Influence Diagrams. Game trees can represent asymmetric problems in a more natural way, but the specification load in a tree (i.e., the size of the graph) increases exponentially as the number of decisions and observations increases. Besides, it is not easy for game tree representation to explicitly reveal the dependence relationships between variables. MAIDs are a modification of influence diagrams for representing decision problems involving multiple non-cooperative agents more concisely and explicitly. A MAID decomposes the real-world situation into chance and decision variables and the dependence relationships among these variables. However, similar blow-up problems are confronted when using MAIDs to represent asymmetric multi-agent decision problem, sometimes even worse than game trees. Take centipede game for example.

[Centipede Game]

Centipede Game was first introduced by Rosenthal (1981) in game theory. In this game, two players take turns to choose either to take a slightly larger share of a slowly increasing pot, or to pass the pot to the other player. The payoffs are arranged so that if one passes the pot to one's opponent and the opponent takes the pot, one receives slightly less than if one had taken the pot. Any game with this structure but a different number of rounds is called a centipede game.

Such a decision problem is called an asymmetric multi-agent decision problem. A special aspect of asymmetric multi-agent decision problems is that the next decision to be made and the information available may depend on the agents' previous decisions or chances moves. For example, in the Centipede game, the next player's move depends on the previous player's choice of whether to take or pass. There are several types of asymmetric multi-agent problems, and we will discuss them in detail in the next section.

The above asymmetric decision scenario could not be solved using traditional methods of influence diagrams and extensions of the representation which have been reviewed in Chapter 2 such as the UIDs, SIDs, AIDs, AVNs and SDDs. The reason is that these formalisms emphasize the single agent based asymmetric decision problem. These graphical models do not take the interaction (or strategic relevance) among multiple agents into consideration.

MAIDs extend the formalisms of BNs and IDs to represent decision problems involving multiple agents. With decision nodes representing the decisions of agents and chance nodes representing the information or observation, MAIDs do not only allow us to capture the important structure of the problem, but make explicit the strategic relevance between decision variables. However, in representing the asymmetric problem, a naïve representation of MAIDs leads to unnecessary blowup.

The representation of an asymmetric multi-agent decision problem requires a new graphical decision model extending from MAIDs. In our work, we integrate game tree and MAIDs together into one language called *asymmetric multi-agent influence diagrams* (AMAIDs).

3.2 Asymmetric Multi-agent Decision Problems

In this section we present four examples to illustrate four types of asymmetries usually confronted in multi-agent decision systems. These examples will also be used in the next section to illustrate our proposed graphical model.

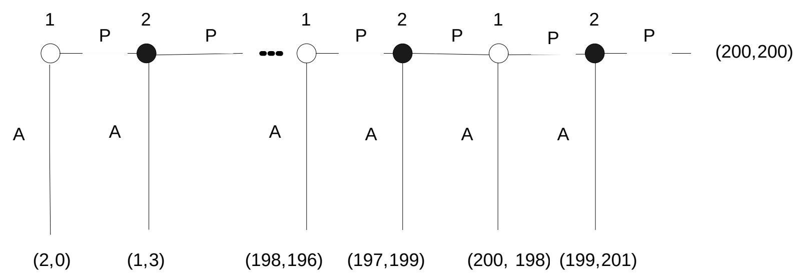
Considering the extensive form trees of the asymmetric problem, we can divide asymmetries in multi-agent decision systems into four types. 1) Different branches of the tree contain different number of nodes; 2) Different branches of the tree involves with different agents; 3) Player's choices are different in different branches of tree; 4) Different decision sequences are associated with different branches of tree.

3.2.1 Different Branches of Tree Containing Different Number of Nodes

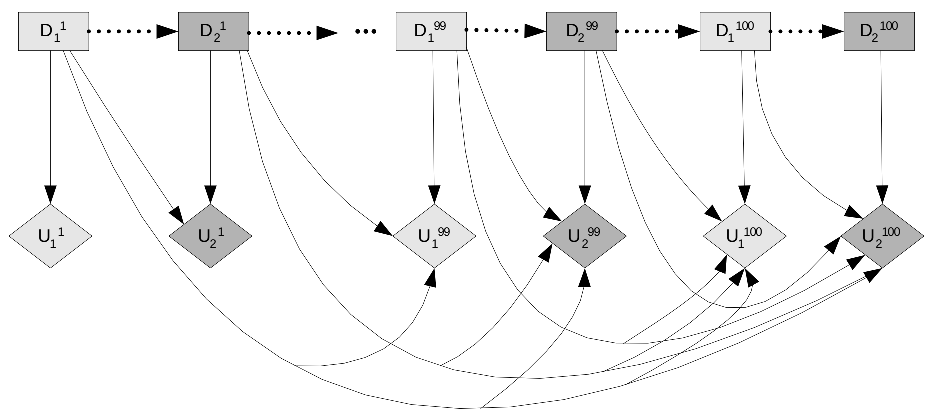
We illustrate this type of asymmetry by Centipede Game mentioned in the above section.

[Centipede Game]

Here we adopt a more detailed version: Consider two players 1 and 2. At the start of the game, Player 1 has two small piles of coins in front of him; very small indeed in fact, as one pile contains only two coins and the other pile has no coins at all. As a first move, Player 1 must make a decision between two choices: he can either take the larger pile of coins (at which point he must also give the smaller pile of coins to the other player) or he can push both piles across the table to Player 2. Each time the piles of coins pass across the table, one coin is added to each pile, such that on his first move, Player 2 can now pocket the larger pile of 3 coins, giving the smaller pile of 1 coin to Player 1 or he can pass the two piles back across the table again to Player 2, increasing the size of the piles to 4 and 2 coins. The game continues for either a fixed period of 100 rounds or until a player decides to end the game by pocketing a pile of coins. If none takes the pile after 100 rounds, then both of them will be given 100 coins.



(a) Game tree representation of the Centipede Game



(b) MAID representation of the Centipede Game

Figure 3.1 Naive representations of Centipede Game

Figure 3.1(a) shows the extensive form tree representation of this problem, with payoffs attached to each end node. In the graph, “A” represents player accepts the larger pile, while “P” represents that player passes to let the next player make a decision. Figure 3.1(b) shows the MAID representation of this problem. Decision node D_i^n represents the decision made by agent i at the n^{th} round. Value node U_i^n represents the utility associated with agent i at the n^{th} round. As we can see, the extensive form does not show the dependence relationships between Player 1 and 2’s decisions explicitly, although it concisely represents the asymmetric decision problem compared to the MAID. The graph size of MAID is prohibitive.

3.2.2 Different Branches of Tree Involves Different Agents

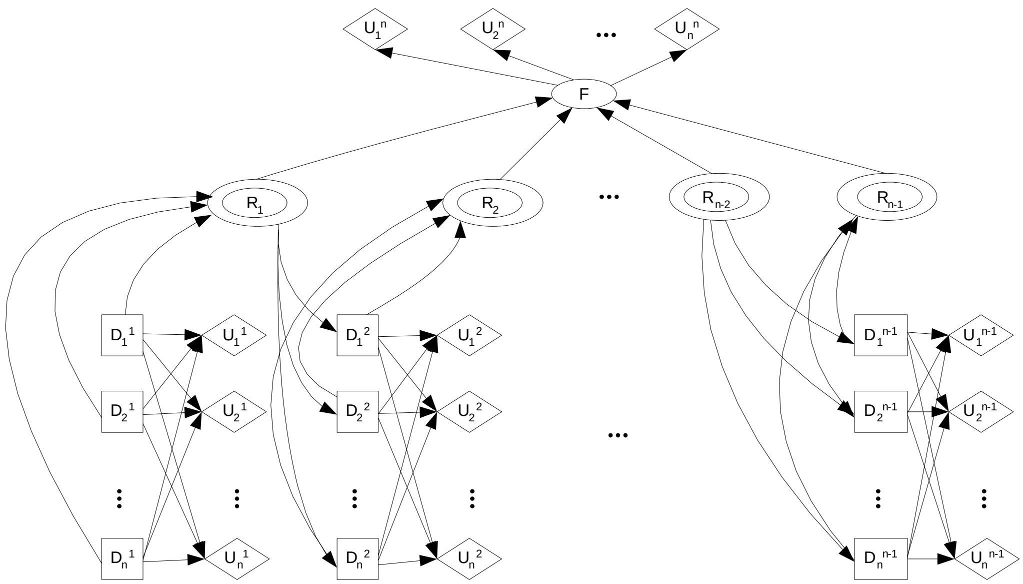
[Killer Game]

There is a popular game called "Killer" among university students. Here we describe the game using a revised version. The game's rule is as follows. Suppose there are *N* players. In each round, they have to vote in order to decide who will be the suspect. The one who gets the highest votes will be "killed" (It means that this person is kicked out of the game and cannot vote again). If there is a tie in a vote, the one with the lowest index amongst those who are tied is "killed". In the final round, a game of chance determines the winner between the remaining two players. To make it simple, we assume everyone is an independent individual. In other words, everyone's decision is not controlled by others. The game ends when *N* rounds of voting have been completed.

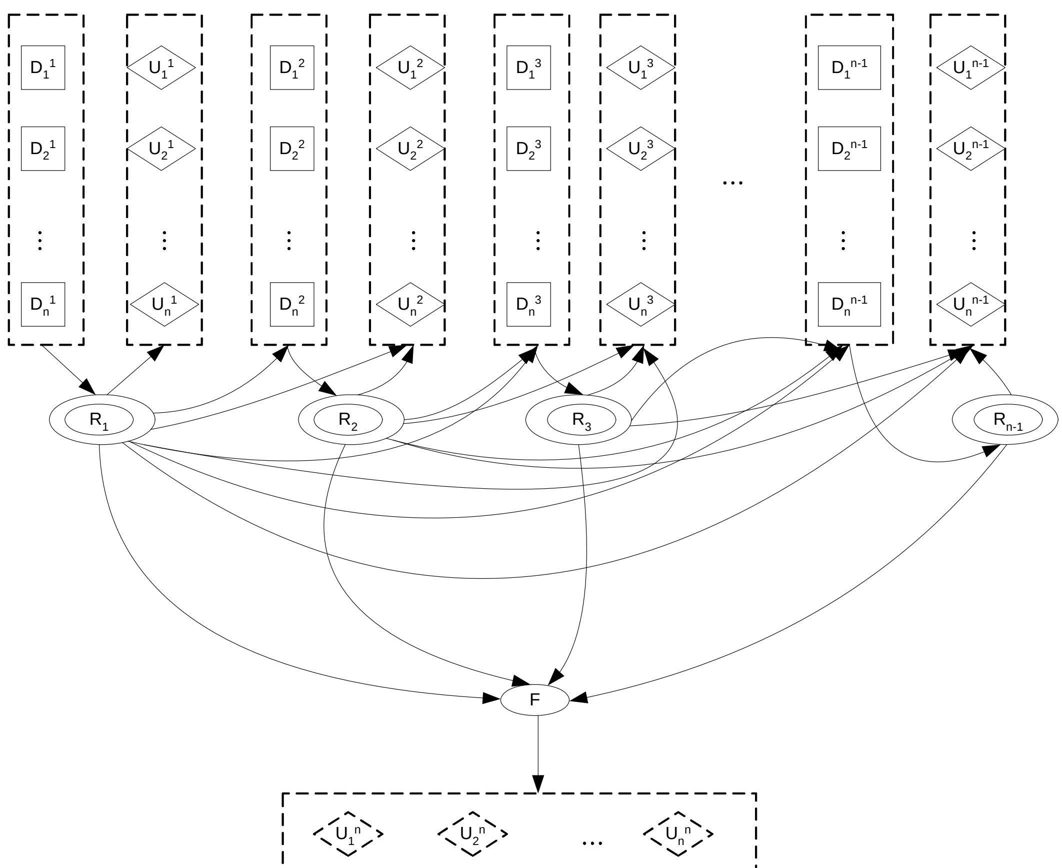
In the first round, there will be $(N\text{-}1)^{N}$ combinations of the possibilities, with *N* outcomes. There will be $(N\text{-}2)^{N-1}$ combinations of the possibilities in the second round, with *(N-1)* outcomes. And third round, $(N\text{-}3)^{N-2}$ combinations , with *(N-2)* outcomes, so on and so forth. The game has to go on with *N* rounds. Using the game tree to represent this game, the game tree would be highly asymmetric. In each round, we represent one outcome with a sub-tree. It means that after everyone has voted in one round, some agent A_i is voted off and he/she is no longer able to vote in the next round. A different sub-tree in the same level may represent the case where a different agent A_j is voted off. Following this rule, the game tree will be very large and the solving time complexity is O(n!).

Figure 3.2 shows the MAID representation of this example, but the specification load of the graph is actually worse than the game tree. For example, the utilities U_1^n, $U_2^n \ldots U_n^n$ in the last round contain all the information from the previous decisions. Even though deterministic nodes R_i are introduced to represent the agents who are voted off during that round *i* and *F* to represent the final result, the CPD table of each utility node still stores the values although a player A_i has already been voted off. This leads to the redundancy of information stored in the nodes.

We do a further refinement of the MAID by introducing *clusters* of nodes in the extended model of Figure 3.2(b), represented by the dashed frames. This refinement makes the original MAID more compactly.



(a) MAID representation of the Killer Game



(b) A further refinement of the MAID model

Figure 3.2 MAID representation of Killer Game

Those nodes in the same dashed frame are in the same cluster, which have the same parents and descendants. A cluster can include a set of decision nodes or utility nodes. If a cluster includes a set of decision nodes, it means that the decisions are made simultaneously. If a cluster includes a set of utility nodes, it simply represents a set of agents' utility nodes under the same condition.

In our extended work, we introduce *clusters* into our AMAID representation to make it more concise.

3.2.3 Player's Choices are Different in Different Branches of Tree

[Take Away Game]

Suppose there is a pile of *N* matches on the tables. Two players take turns to remove the matches from the pile. On the first move a player is allowed to remove any number of objects, but not the whole pile. On any subsequent move, a player is allowed to remove no more than what his or her opponent removed on the previous move. The one who removes the last one match from the table win the game.

This decision problem has two special characteristics: (1) each player's available choices might be changing every step. The scope depends on the choice made by the previous player. (2) The number of the game stages is unknown, depending on the choices made by the players in each step.

The game tree of this decision problem is highly asymmetric. The tree will be very large as it will have O(n!) leaves. However, our MAID representation is worse, not only in the specification load, but also in the expressiveness of MAID. Figure 3.3 shows the MAID representation of this problem. In this representation, it is hard for us to identify when the game will be ended. Besides, in each step, the MAID stores every choice of the players from 1 to *N* even though some of them are impossible. Therefore, redundancy is incurred.

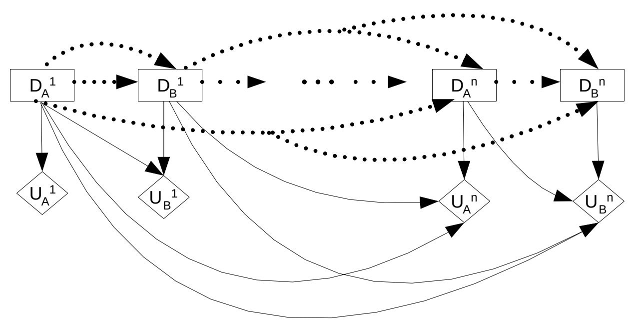


Figure 3.3 MAID representation of Take Away Game

3.2.4 Different Branches of Tree Associated with Different Decision Sequences

[War Game]

Suppose country *A* plans to conquer countries *B* and *C*. *A* should decide whether to fight with *B* first or *C* first. The country which *A* has chosen to fight first should then decide whether to make a coalition with another country or fight by itself. If it decides to make a coalition, the country who is requested should decide whether to help or not.

This problem is asymmetric because the first decision maker *A*'s decision influences the decision sequences of the next two decision makers' decisions. It is quite natural to represent this problem with a game tree. To represent it by a MAID we have to represent the unspecified ordering of the *B* and *C*'s decisions as a linear ordering of decisions. Figure 3.4 depicts an MAID representation of the War Game.

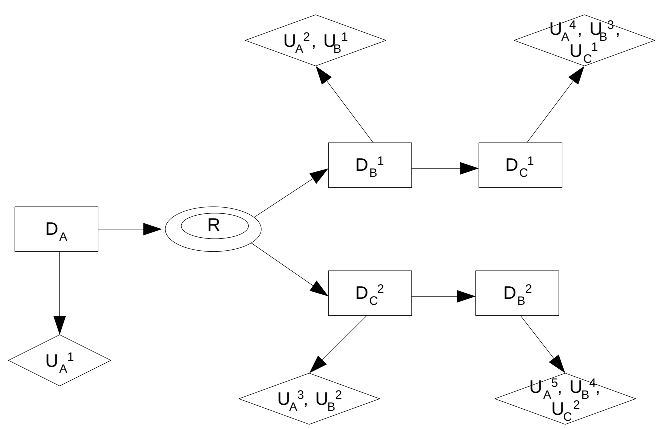


Figure 3.4 MAID representation of the War Game

3.3 Asymmetric Multi-agent Influence Diagrams

In this section, we will describe the main features of asymmetric multi-agent influence diagrams (AMAIDs) by considering the AMAID representation of the

Centipede Game as described in the previous section. This idea is borrowed from the idea of Sequential Influence Diagrams (SIDs) when handling the asymmetric decision problems in single agent decision systems.

Similar to a SID, An AMAID can be viewed as two diagrams superimposed onto each other. One diagram encodes the information precedence as well as asymmetric structure and the other encodes the probabilistic dependence relations for the chance nodes and deterministic functional relations for the utility node.

Assuming a set of agents *I*, an AMAID *M* is a triplet (*N*, *A*, *P*). $N = C \bigcup D \bigcup U$ is the set of uncertain nodes, where C is the set of chance nodes (represented by ellipses) which represents the decisions of nature, $D = \bigcup_{i \in I} D_i$ represents the set of all the agents' decision nodes (represented by rectangles), $U = \bigcup_{i \in I} U_i$ represents the set of all the agents' utility nodes (represented by diamonds). *P* is the joint probability distributions over all the nodes *N*. *A* is the set of directed arcs comprised of dashed arcs and solid arcs between the nodes in the graph. The dashed arc (also called *contextual arc*) encodes the information precedence and asymmetric structure, while the solid arc (also called *probabilistic arc*) encodes the probabilistic dependence and functional relations. In other words, if there is a dashed edge from *X* to *Y*, it means *X* is observed or decided before *Y* is observed or decided. The arc (*X*, *Y*) may be associated with an annotation *g(X, Y)* which describes the *context* under which the next node in the set of scenarios is the node that the arc points to and we call it *contextual condition*. A *context* (Boutilier et al. 1996, Zhang & Poole 1999, Poole & Zhang 2003) refers to an assignment of some actual values to a set of variables. We say the arc is *open* if the context is fulfilled. Otherwise, we say the arc is *closed*.

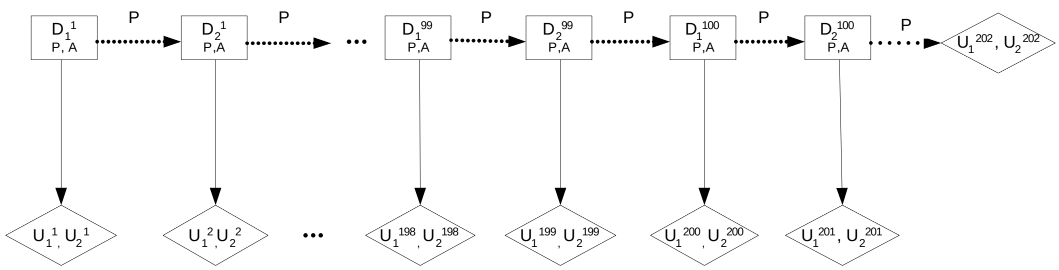


Figure 3.5 An AMAID representation of the Centipede Game

As shown in Figure 3.5, the dashed arc from D_1^1 to D_2^1encodes D_1^1 is decided upon before D_2^1 and asymmetric information is encoded by the contextual condition P associated with the dashed arc. The annotation P on the dashed arc from D_1^1 to D_2^1 means that whenever D_1^1=P, the next node in all scenarios is D_2^1. In other words, D_1^1=P makes the value of D_1^1 irrelevant to the payoff cluster (U_1^1, U_2^1) (in all scenarios, U_1^1=0, U_2^1=0). Whenever D_1^1=P, we say that the dashed arc from D_1^1 to D_2^1 is open. The set of nodes referenced by the contextual condition g is called the domain of g, e.g. dom (g(D_1^1, D_2^1))={D_1^1}. The set of contextual conditions are denoted by g, i.e., if g does not contain an annotation for the dashed arc (X, Y), then we extend g with the annotation $g(X, Y) \equiv 1$.

The decision node in an AMAID is composed of two parts. The part above encodes the name of the decision node, while the part below encodes the available choices of each decision. One utility node may encode the utilities of several agents, we call it a *cluster* of utility nodes and use arrays to describe them. As shown in Figure 3.5, the decision node D_1^1 has two available choices "A" and "P", array (U_1^i, U_2^i) is used to describe every cluster of utility nodes.

A *scenario* in an AMAID can be identified by iteratively following the open arcs from a source node (a node with no incoming dashed arcs) until a node is reached with no open outgoing arcs. In a MAID, a scenario requires one terminal node explicitly. However, this does not hold in AMAID. In the case D_1^1=D, the scenarios end in D_1^1 with a state of D, if D_1^1 =A, the scenarios may end with a state of A at any decision nodes thereafter, except D_1^1.

Unlike MAID, AMAID is not an acyclic graph. It allows the temporal existence of directed cycles. However, the sub-graph representing each scenario must be an acyclic graph. In other words, the cycle should have at least one closed contextual arc in one scenario. For example, A and B are two manufacturing companies in the market. Company A has a new innovation and has to decide (D_A) whether to license it out (L) or release as an open source to the public (R). If it licenses the new technology out, then after a few years, other companies would also know the technology, which means that other companies can produce by mimicking. If it releases the innovation as an open source, then the other companies will know it immediately. *B* is another company who has to decide whether to incorporate *A*'s technology into its own product. If the technology is released to others, there will be a market feedback about the technology (*F*) immediately. Otherwise, there will be a feedback a few years later. The AMAID representation of this scenario is shown in Figure 3.6. As we can see from the figure, the AMAID is in a directed cycle. Only when the decision result of *A* is observed, the directed cycle can be broken. If D_A=*L*, then the arcs 1 and 4 are closed, the cycle is broken. If D_A=*R*, the arcs 2 and 3 are closed.

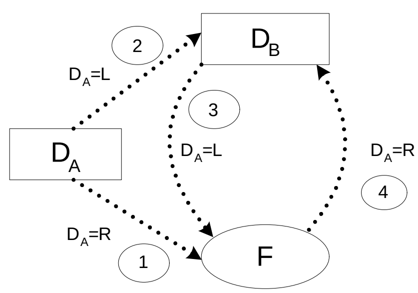


Figure 3.6 The cycle model

A partial temporal order $\prec_M$ can be defined over the chance and decision nodes in an AMAID *M*. If and only if there is a directed path from *X* to *Y* in *M* but not from *Y* to *X* or *Y* is unobserved, we say X$\prec_M$ Y. In Figure 3.6, If D_A=*L*, then $D_B \prec_M F$. If D_A=*R*, then $F \prec_M D_B$.

Apart from the qualitative properties of AMAID, an AMAID also specifies the joint probability distributions over nodes *N*. Let x be a variable and $\pi(x)$ be the set of x's parents (if for any $y \in \pi(x)$, there is a directed arc $y \rightarrow x$ or y --> x). For each instantiation $\pi(x)$ and x, there is a conditional probability distribution (CPD): $P(x|\pi(x))$ associated. If $x \in D$, then $P(x|\pi(x))$ is called a *decision rule* ($\sigma(x)$)for this decision variable x. A *strategy profile* σ is an assignment of decision rules to all the decisions of all the agents. The joint distribution defined over *N* is $P_{M[\sigma]} = \prod_{x \in C \cup U} P(x|\pi(x)) \prod_{x \in D} \sigma(x)$.

Note that if for $y \in \pi(x)$, there is a directed contextual arc y --> x with a contextual condition *g*: $y = y_1$ associated, then the CPD table for x is $P(x \mid \pi(x) \setminus y, y = y_1)$. Otherwise, the CPD table for x is $P(x|\pi(x))$.

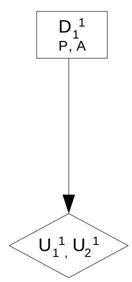
In an AMAID, there may be a series of utility nodes of an agent i, but they are under different contexts. We call it *contextual utility*. For example, the utility $U_1^1|(D_1^1=A)$ means the utility is only available when $D_1^1=A$. We call $D_1^1=A$ is the *context statement* of the contextual utility and those contextual variables involved in the context statement is called the *domain* of the context statement. Apparently, we cannot have a contextual variable with all its available choices in the context statement. Those utility variables specified with different contexts cannot be added together. Only those utility nodes unspecified can be an addend, assuming the additive decomposition of the agent's utility function by breaking an agent's utility function into several variables. Consider the Centipede Game in Figure 3.5, the utility nodes U_1^1, U_1^2, …, U_1^{201} are all the utilities of agent 1, but they cannot be added together since they are contextual utilities. However, if let's say when agent 1 makes the first decision (D_1^1), there is the same amount of cost U_{cost} incurred, no matter what choice he makes. Then U_{cost} should be added to all the other contextual utilities. For all the utility nodes of agent i under the same context, we define a *class*. Those unspecified utility nodes should be added to every class.

With the probability distribution, the utility for each agent can be computed. Suppose $U_i=\{U_1, U_2, ..., U_n\}$. Every element of U_i should be in the same class. The total utility for an agent i if the agents play a given strategy profile σ can be computed with equation below:

$$EU_a(\sigma) = \sum_{(u_1,...,u_n)\in dom(\mathcal{U}_i)} P_{M[\sigma]}(u_1,...u_n)\sum_{k=1}^{n} u_k = \sum_{U\in\mathcal{U}_i}\sum_{u\in dom(U)} P_{M[\sigma]}(U=u)\bullet u$$

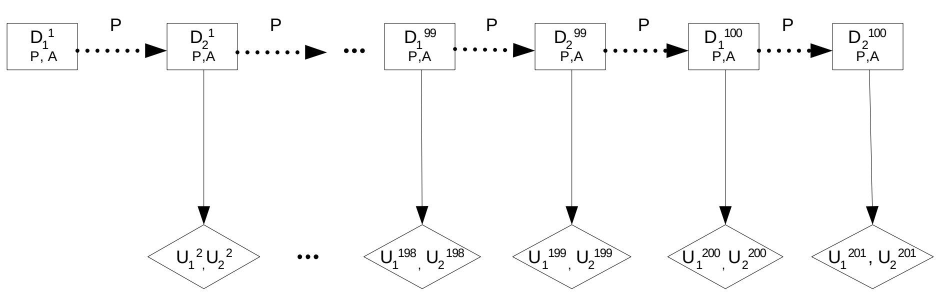
Let M be an AMAID with variables N and contextual condition g. If one variable $X \in N$ appears in the domain of the contextual condition g, we call this X a *split variable*. For a partial temporal order $\prec_M$ in an AMAID M, if there is no other split variables Y before X, i.e., $Y \prec_M X$, then X is called an *initial split variable*.

If a split variable X is initiated (a specific value is assigned to X), then the contextual condition g with X included can be evaluated. If the contextual condition is evaluated to be false, then the associated contextual arc can be removed with all the variables that we can only reach by following that arc. Consider the AMAID representation of the Centipede Problem shown in Figure 3.5. In this representation, D_1^1 is the initial split variable. After initiating D_1^1 by assigning values A and P respectively, we get the following reduced AMAIDs shown in Figure 3.7.



(a) Reduced AMAID M [$D_1^1 \mapsto A$] of the Centipede Game by the instantiation

D_1^1=A



(b) Reduced AMAID M [$D_1^1 \mapsto P$]of the Centipede Game by the instantiation

D_1^1=P

Figure 3.7 Reduced MAID by initiating D_1^1

Below shows the AMAID representation of the other three asymmetric examples mentioned above.

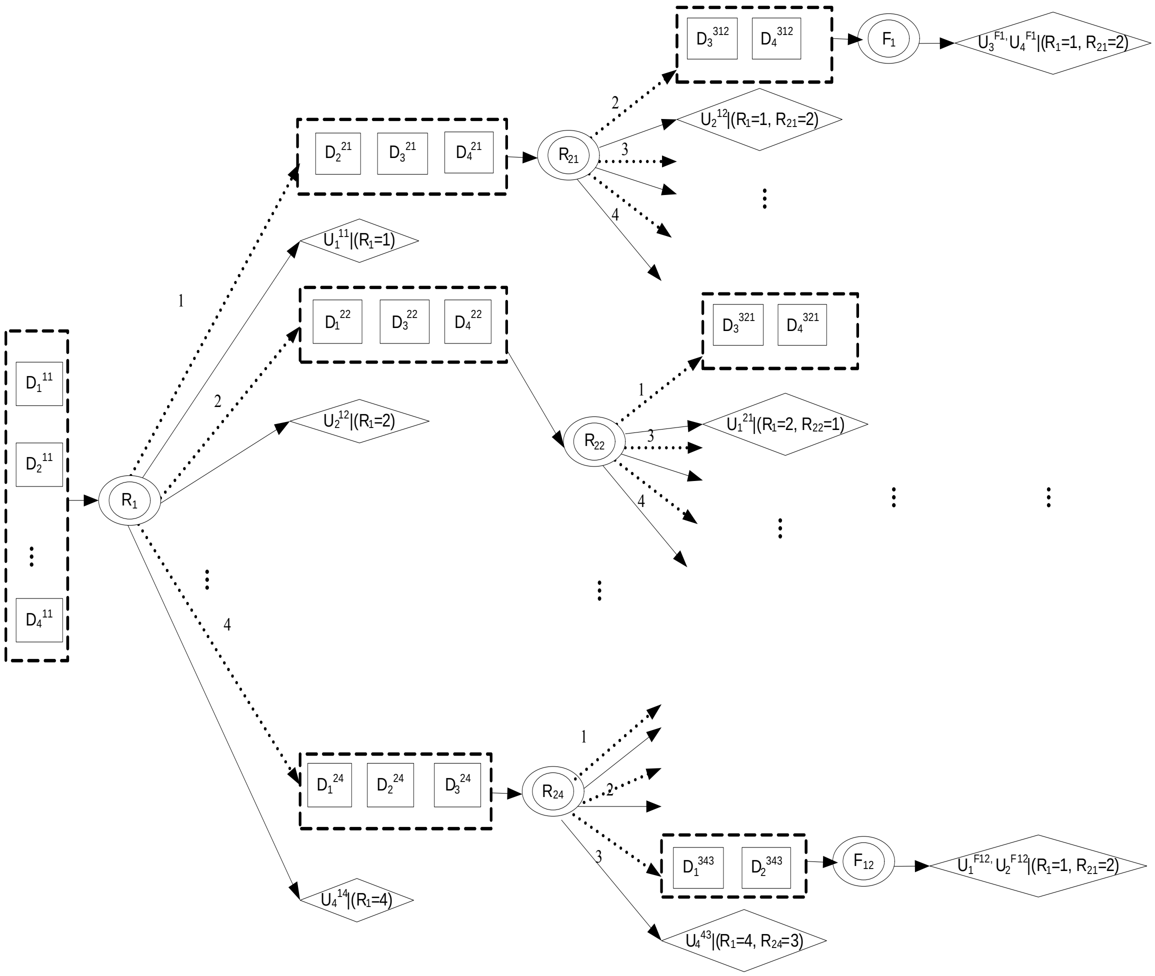


Figure 3.8 AMAID representation of Killer Game (N=4)

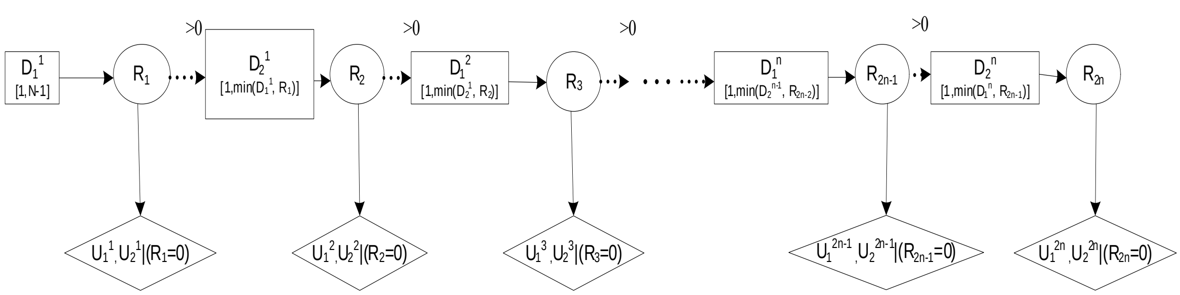


Figure 3.9 AMIAD representation of Take Away Game

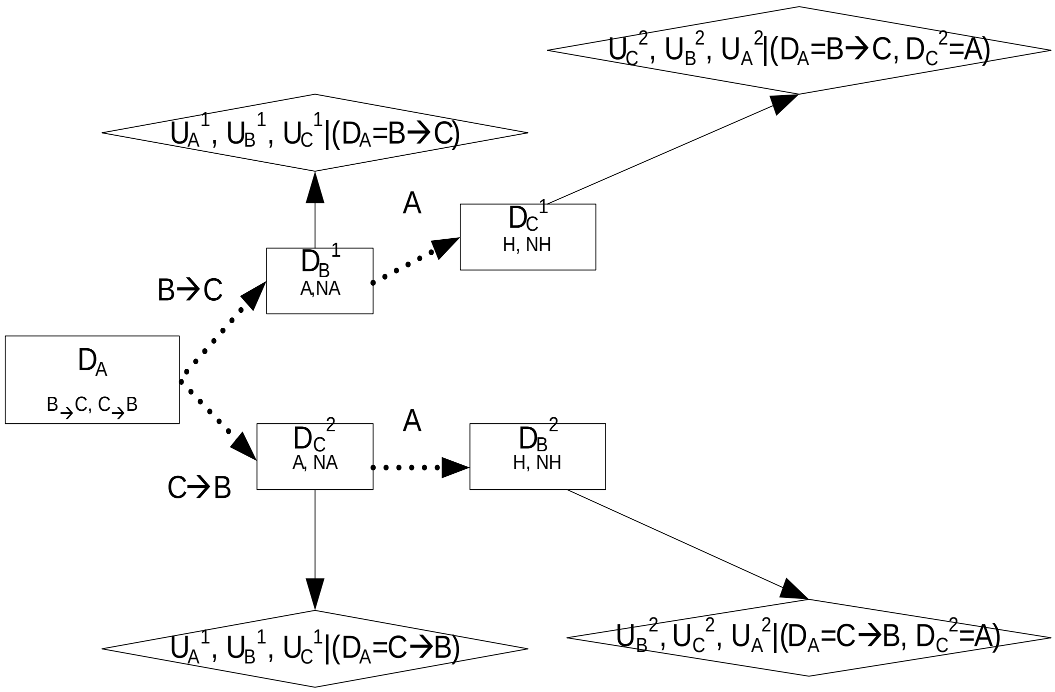


Figure 3.10 AMAID representation of War Game

4 Asymmetric Multi-agent Influence Diagrams: Model Evaluation

In multi-agent systems, the main computational task is to compute the Nash equilibrium. In the previous chapter, the decision models of AMAIDs have been developed to represent asymmetric multi-agent decision problems. This chapter will discuss the evaluation algorithms to solve proposed decision models.

4.1 Introduction

The multi-agent decision problems involve multiple interacting agents in an uncertain environment. One agent's decision will influence another agent's decisions which may in turn affect other agents' decisions. The aim of a specific agent is to seek the optimal decision rule, given decision rules of other agents. Because of the intricate interactions among these agents, finding Nash equilibrium becomes extremely difficult. A straightforward and easy approach is to convert the AMAID into game tree and then use backward induction to solve the game tree. Unfortunately, this straightforward approach described above does not provide any computational efficiency since it will create some unnecessary blowups.

Koller and Milch (2001) proposed the definition of strategic relevance to break the complex game into a series of relatively simple games, taking advantage of the independence structure in a MAID which reduced the task of finding a global equilibrium to several relatively local computations. We will adopt this concept in our algorithm for evaluating AMAIDs.

We begin the discussion by introducing some definitions related to strategic relevance (Koller & Milch 2001).

Definition 4.1 S-Reachability

A node *D’* is s-reachable from a node *D* in a MAID *M* if there is some utility node $U \in U_D$ such that if a new parent $\hat{D}'$ were added to *D’*, there would be an active path in *M* from $\hat{D}'$ to *U* given $Pa(D) \cup \{D\}$, where a path is active in a MAID if it is active in the same graph, viewed as a BN.

Definition 4.2 Relevance Graph

The relevance graph for a MAID *M* is a directed graph, the nodes of which are the decision nodes of *M*. There is a directed arc between *D’* to $D, D' \to D$, if and only if *D’* is s-reachable from *D*.

Definition 4.3 *Nash Equilibrium* (Nash 1950)

A Nash equilibrium is a state that no agent has the incentive to deviate from its decision rule specified by the strategy profile, given no other agents deviate.

4.2 Relevance Graph and S-Reachability in AMAID

In order to apply the definition of relevance graph and s-reachability in AMAIDs, we would first extend an AMAID to de-contextualize AMAID.

Definition 4.4 *De-contextualize AMAID*

An AMAID containing no contextual utility node is called a *de-contextualize AMAID*.

We can change an AMAID to a de-contextualize AMAID whenever a contextual utility node is met, and add a directed arc from the split variable X which is in the domain of contextual utility to the utility node. If an arc already exists, do nothing. For example, if the contextual utility node is represented as $U_1^1|(D_1^1=A)$, we add an arc from the split variable D_1^1 to the contextual utility U_1^1 and change contextual utility node to a normal form utility node by deleting the context conditions contained in the contextual utility node. The contextual utility node after removing the context statement is called *de-contextualized utility node*.

After changing AMAID to *De-contextualize AMAID*, we can check the s-reachability of all the decision nodes in *De-contextualize AMAID*.

Therefore, the steps of constructing the relevance graph of an AMAID are as follows:

1. For a decision node *D’* and *D* in an AMAID *M* and there is some contextual utility node $U \in U_D$, De-contextualize the utility node *U* by using the method listed above.
2. Add a new parent $\hat{D}'$ to D’ ;
3. If there is an active path from $\hat{D}'$ to the de-contextualized *U* given $Pa(D) \cup \{D\}$, the node *D’* is said to be s-reachable from a node *D* in the AMAID *M*.
4. Check the s-reachability between every two decision nodes in the AMAID;
5. Construct a new graph only which contains every decision node in *M*. If *D’* is s-reachable from *D*, add a directed arc between *D’* to *D*, $D' \rightarrow D$.

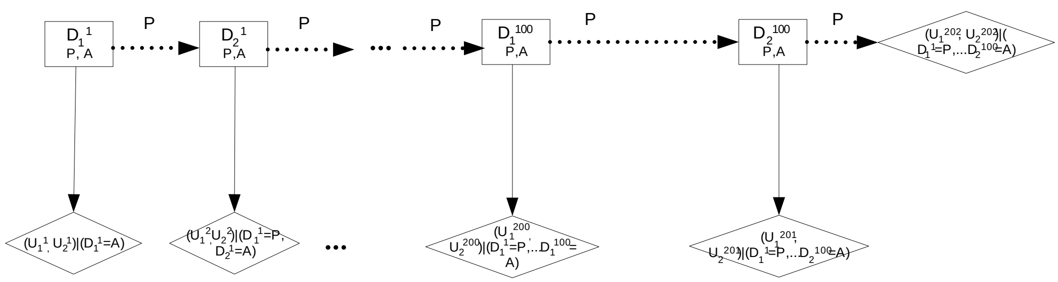
A path is said to be active if along this chain (the directed path), every intermediate node *A* satisfies:

a) If *A* is a head-to-head node in the chain, *A* or its descendents are in $Pa(D) \cup \{D\}$;

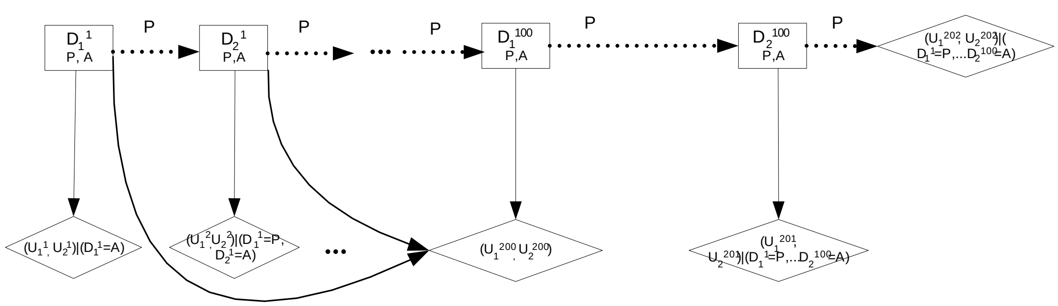
b) If *A* is not a head-to-head node in the chain, *A* is not in $Pa(D) \cup \{D\}$.

If there is dashed arc along the path, make sure the arc is open.

We take the AMAID of Centipede Game for example. Figure 4.1(a) shows the AMAID representation of Centipede Game by showing the contextual condition in contextual utilities explicitly. Take the contextual utility node (U_1^{200}, U_2^{200})|(D_1^{1}=P,...D_1^{100}=A) for example, since D_1^{1}, …D_1^{100} are split variables contained in the domain of the contextual condition, we should add a directed arc from D_1^{1},…D_1^{100} to the contextual utility node (U_1^{200}, U_2^{200})|(D_1^{1}=P,...D_1^{100}=A) to de-contextualize it. Since the arc from D_1^{100} to contextual utility node (U_1^{200}, U_2^{200})|(D_1^{1}=P,...D_1^{100}=A) already exists, there is no need to add another one. We get Figure 4.1(b) showing the graph after contextual utility node (U_1^{200}, U_2^{200})|(D_1^{1}=P,...D_1^{100}=A) is de-contextualized.



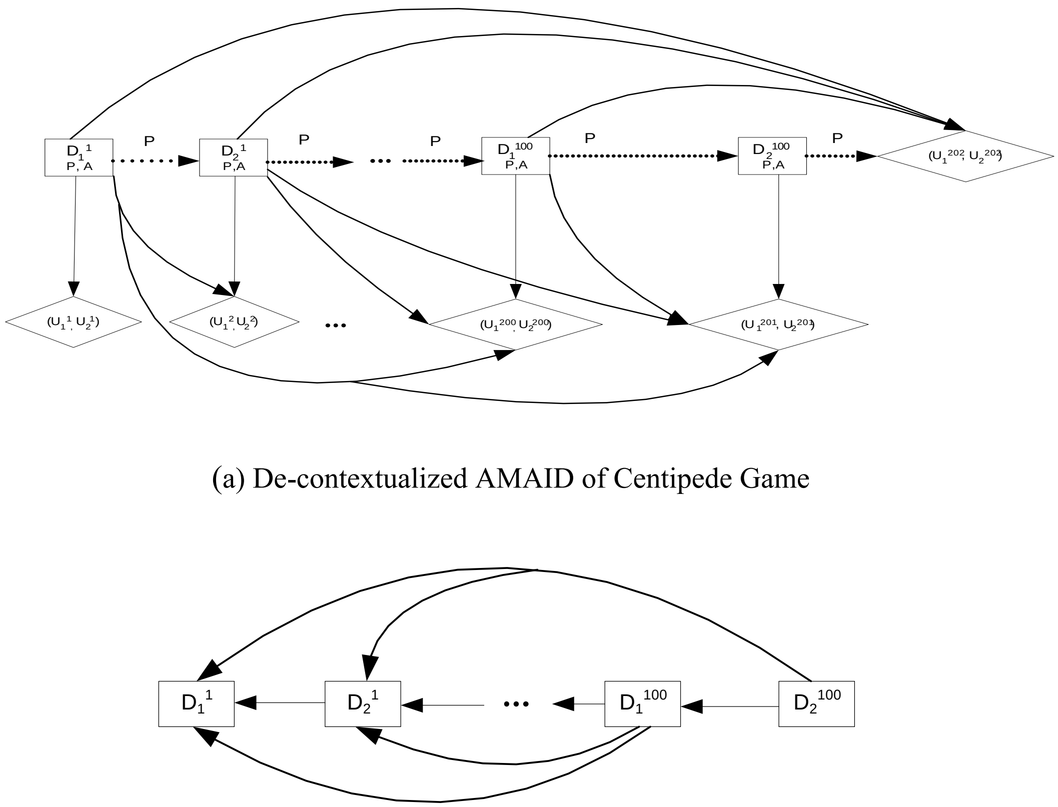
(a) AMAID of Centipede Game



(b) AMAID after contextual utility node is de-contextualized

Figure 4.1 De-contextualize contextual utility node

Figure 4.2(a) shows the De-contextualized AMAID of Centipede Game. Figure 4.2(b) shows the relevance graph of Centipede Game according to the De-contextualized AMAID.



(a) De-contextualized AMAID of Centipede Game

(b) Relevance graph of AMAID of the Centipede Game

Figure 4.2 Constructing the relevance graph of the AMAID

4.3 Solution for AMAID

4.3.1 AMAID With Acyclic Relevance Graph

The goal of evaluating the AMAIDs is to find an optimal decision rule δ_i for each decision node D_i and to maximize each agent's expected utility given other agents' chosen decision rule. The computation is based on the following expression:

$$\delta_{D_i}{}^{*} = \arg\max_{\delta_{D_i}{}^{*}} \sum_{d \in dom(D_a)} \delta_{D_i}{}^{*}(d_i \,|\, \mathrm{pa}_{D_i}) \times \sum_{U \in U_{D_i}} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U = u \mid d_i, \mathrm{pa}_{D_i}) \cdot u$$

Where u is the utility function specified by each utility node U , σ is the strategy profile specified by the AMAID.

In multi-agent decision problems, the agents' decisions are always related. In order to optimize the decision rule of one decision node, the decision rule for those decisions that are relevant for it should be clarified first. Therefore, we can construct a topological ordering of the decision nodes in AMAID according to the constructed relevance graph. The topological ordering is an ordering D_1, …D_n such that if D_i is s-reachable from D_j, then $i<j$ (Koller & Milch 2001). Each decision node relies only on the decision nodes that precede it. Therefore, we can compute the optimal decision rule of each decision node in this topological order so that when we are computing the optimal decision rule of one decision node, the decision rule of the decision nodes that it relies on has already been clarified. The algorithm for solving AMAID can be stated as follows:

Algorithm 4.1

Given an AMAID with an acyclic relevance graph

1. Identify a topological ordering D_1,…D_n of the relevance graph for M;

2. Let σ^0 be an arbitrary fully mixed strategy profile for *M*;
3. For *i*=1 to *n*:

   Let δ be a decision rule for D_i that is optimal for σ^{i-1}.

   Let $\sigma^i = (\sigma^{i-1}_{-D_i}, \delta)$;
4. Output σ^n as an equilibrium of *M*.

All the above mentioned steps are the same as the steps of solving MAIDs except for the step 3. In the step 3, we want to find a δ such that for every instantiation pa_{D_i} of $Pa(D_i)$ where $P_{M[\sigma^{i-1}]}(\mathrm{pa}_{D_i}) \succ 0$, the probability distribution $\delta(D_i \mid \mathrm{Pa}_{D_i})$ is a solution of:

$$\arg\max_{P^*} \sum_{d \in dom(D_i)} P^*(d) \times \sum_{U \in U_{D_i}} \sum_{u \in dom(U)} P_{M[(\sigma^{i-1})]}(U = u \mid d_i, \mathrm{pa}_{D_i}) \cdot u$$

Here, *U* can be divided into two types: 1) the utility nodes which are the descendents of the decision D_i, we use $U^0_{D_i}$ to denote it; 2) the contextual utility nodes whose domain contains the decision node D_i if D_i is a split variable, we use $U^C_{D_i=d_i}$ to denote it. Therefore, in step 3, in order to find a value $d^* \in dom(D_i)$ that maximizes:

$$\sum_{U \in U^0_{D_i}} \sum_{u \in dom(U)} P_{M[(\sigma^{i-1})]}(U = u \mid d_i, \mathrm{pa}_{D_i}) \cdot u + \sum_{U \in U^C_{D_i=d_i}} \sum_{u \in dom(U)} P_{M[(\sigma^{i-1})]}(U = u \mid D_i = d_i, \mathrm{pa}_{D_i=d_i}) \cdot u$$

We let P^* assign probability 1 to d^* and 0 to the other possible values of D_i. The resulting σ^n is always a pure strategy profile.

When we are performing this computation, all other decision nodes are changed into chance node with probability assigned by σ^{i-1} . Then we can use BN inference to obtain the optimal decision rule of D_i which maximizes:

$$\sum_{U \in U^0_{D_i}} \sum_{u \in dom(U)} P_{M[(\sigma^{i-1})]}(U = u \mid d_i, \mathrm{pa}_{D_i}) \cdot u + \sum_{U \in U^C_{D_i = d_i}} \sum_{u \in dom(U)} P_{M[(\sigma^{i-1})]}(U = u \mid D_i = d_i, \mathrm{pa}_{D_i = d_i}) \cdot u$$

Take the AMAID of Centipede Game for example,

According to the relevance graph in Figure 4.2(b), the topological ordering should be $D_2^{100}, \ldots D_1^{1}$.

Consider D_2^{100} firstly, it has two contextual utility nodes associated, namely $U^C_{D_2^{100}=A}$ and $U^C_{D_2^{100}=P}$. Therefore, in step 3, in order to find a value $d_2^{100*} \in \{A, P\}$ that maximizes:

$$\sum_{U \in U^C_{D_2^{100}=d_2^{100}}} \sum_{u \in dom(U)} P_{M[(\sigma^{i-1})]}(U = u \mid d_2^{100}, d_1^{100}) \cdot u$$

We let P^* assign probability 1 to d_2^{100*} and 0 to the other possible values of D_2^{100} .

After the decision rule for D_2^{100} is decided, the next decision node in the topological order is D_1^{100}. At that time, the optimal decision rule for D_2^{100} has already been decided. Assume d_2^{100*}=A, then the split variable of D_2^{100} is initiated. The AMAID can therefore being reduced to AMAID $M[D_2^{100} \mapsto A]$, which is shown in Figure 4.3.

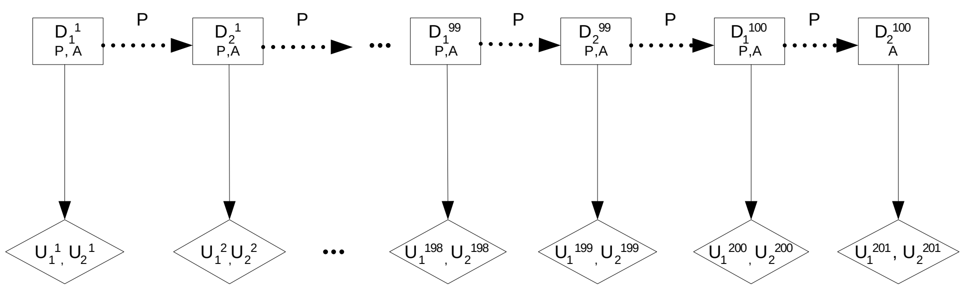


Figure 4.3 Reduced AMAID $M[D_2^{100} \mapsto A]$ of the Centipede Game

4.3.2 AMAID With Cyclic Relevance Graph

The algorithm of solving AMAID with cyclic relevance graph is similar to the algorithm of solving MAID. The only difference is when converting the sub-AMAID to a game tree, we use the asymmetric game tree.

Definition 4.5 Strongly Connected Component (Koller & Milch 2001)

A set S of nodes in a directed graph is a strongly connected component (SCC) if for every pair of nodes $D \neq D' \in S$, there exists a directed path from D to D'.

A maximal SCC is an SCC that is not a strict subset of any other SCC.

The maximal SCCs of a relevance graph can be constructed in linear time. We can construct a component graph, and the nodes of which are the maximal SCCs of the relevance graph. We can find a topological ordering $C_1, \ldots C_m$ of the component graph, such that if some element of C_i is s-reachable from some element of C_j, then $i<j$.

Algorithm 4.2 Given an AMAID *M*

1. Construct a component graph of the relevance graph for *M*;
2. Identify a topological ordering $C_1,\ldots C_m$ of the component graph;
3. Let σ^0 be an arbitrary fully mixed strategy profile for *M*;
4. For i=1 to *n*:

   Let τ be a partial strategy profile for C_i that is a Nash equilibrium in M[$\sigma^{i-1}_{-C_i}$];
5. Let $\sigma^i = (\sigma^{i-1}_{-C_i}, \tau)$;
6. Output σ^n as an equilibrium of *M*.

In computing every partial strategy profile for C_i, we convert the sub-AMAID into an asymmetric game tree and then use backward induction to find the partial strategy τ .

Take the AMAID of the Killer Game with n=4 for example. The relevance graph of it is shown as in Figure 4.4. In this example, every component forms a sub-AMAID. Then we can convert the sub-AMAID into three asymmetric game trees and use backward induction to find the partial strategy τ .

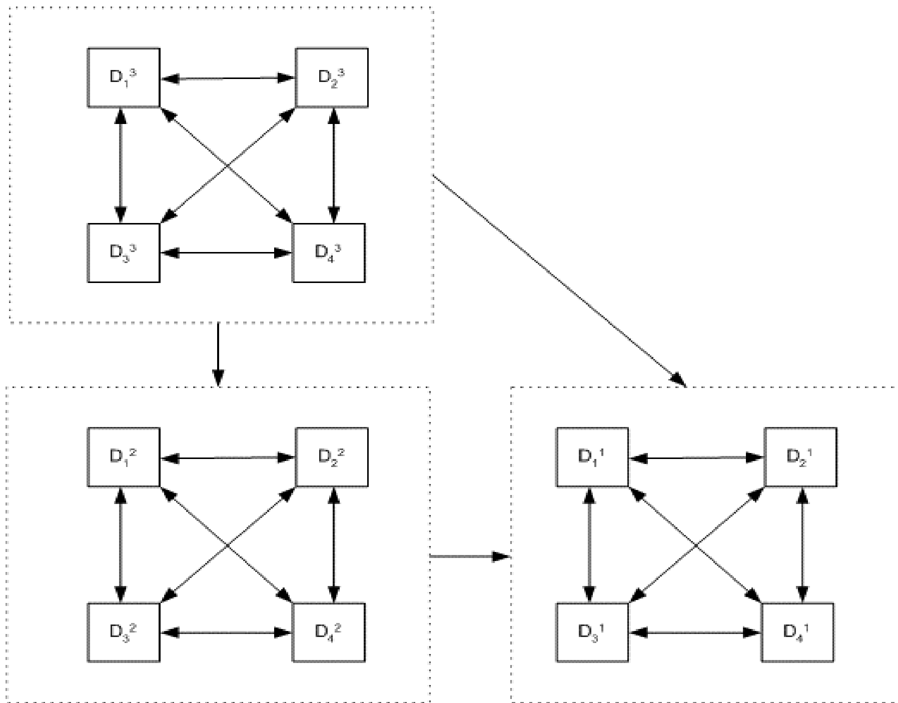
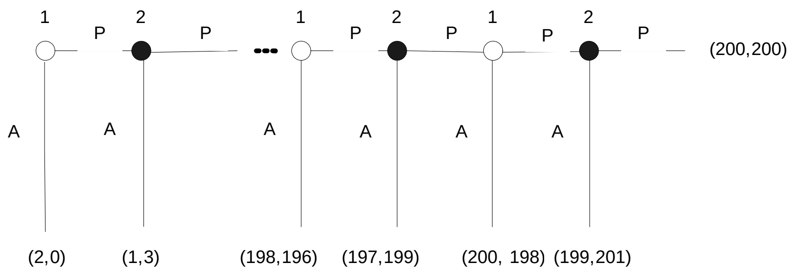


Figure 4.4 The relevance graph of the Killer Game

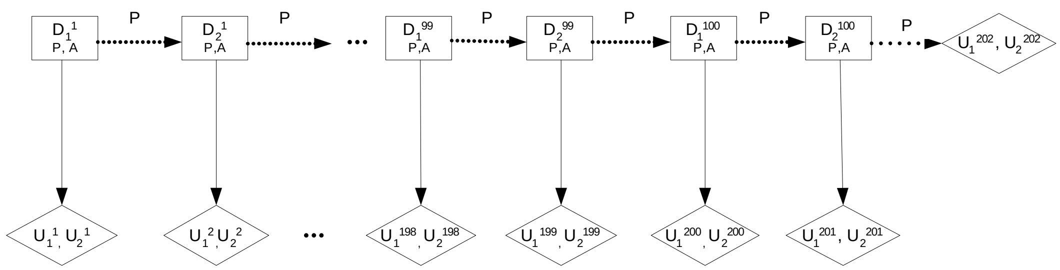
4.4 A Numerical Example

In this section, we take some numerical examples of the above Centipede Game.

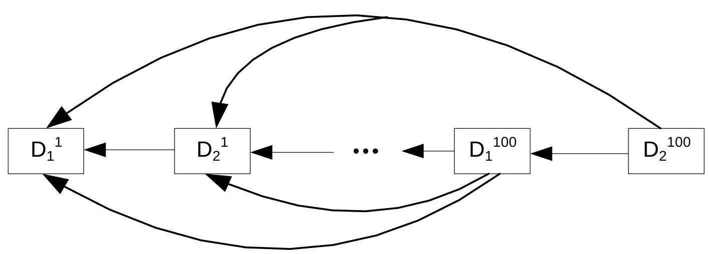
Figure 4.5 shows the numerical example of the Centipede Game. According to the relevance graph in Figure 4.5(c), the topological ordering should be $D_2^{100}, \ldots D_1^1$.



(a) Game tree of the numerical example



(b) AMAID of the Centipede Game



(c) Relevance graph

Figure 4.5 Numerical example of the Centipede Game

Consider D_2^{100} firstly, it has two contextual utility nodes associated, namely $U^C_{D_2^{100}=A}$ and $U^C_{D_2^{100}=P}$. Therefore, in step 3, in order to find a value $d_2^{100*} \in \{A, P\}$ that maximizes:

$$\sum_{U \in U^C_{D_2^{100}=d_2^{100}}} \sum_{u \in dom(U)} P_{M[(\sigma^{i-1})]}(U = u \mid d_2^{100}, d_1^{100}) \cdot u$$

Therefore,

$$P^* = \arg\max_{P^*}[P(d_2^{100} = P) \cdot P_{M[(\sigma^{i-1})]}(U = 200 \mid d_2^{100} = P, d_1^{100}) \cdot 200$$
$$+P(d_2^{100} = A) \cdot P_{M[(\sigma^{i-1})]}(U = 201 \mid d_2^{100} = A, d_1^{100}) \cdot 201]$$

After assigning an arbitrary fully mixed strategy profile for *M*, we can find $d_2^{100*} = A$.

We let P^* assign probability 1 to d_2^{100*}=*A* and 0 to d_2^{100*}=*P*.

After the decision rule for D_2^{100} is decided, the next decision node in the topological order is D_1^{100}. At that time, The AMAID can be reduced to AMAID $M[D_2^{100} \mapsto A]$, which is shown in Figure 4.6.

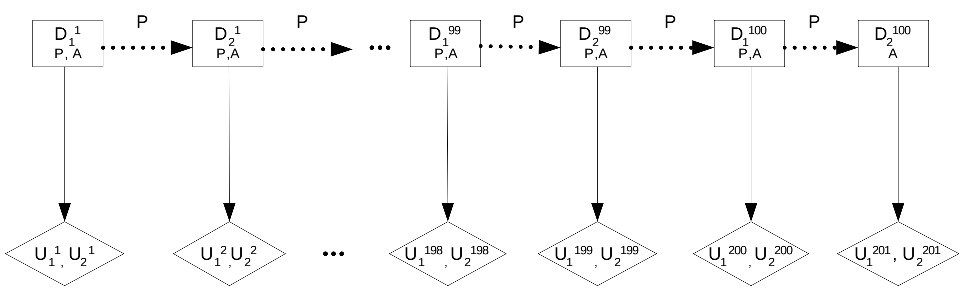


Figure 4.6 Reduced AMAID $M[D_2^{100} \mapsto A]$

Using the same method, we can get the optimal decision rule of D_1^{100} is to assign probability 1 to $d_1^{100*} = A$ and 0 to $d_1^{100*} = P$.

Finally, we get the Nash equilibrium is ($d_1^{1*} = A, \ldots d_2^{100*} = A$).

4.5 Discussions

To demonstrate the potential savings resulting from our decision model, we take the Centipede Game and Killer Game for example. In the Centipede Game, whose relevance graph is acyclic, we can reduce the AMAID step by step after deciding the optimal decision rule of one decision node. This is a computational saving compared to MAID.

Besides, in the algorithm of deciding the optimal decision rule of each decision, the maximization function is reduced to:

$$P^* = \arg\max_{P^*}[P(d_2^{100} = P) \cdot P_{M[(\sigma^{i-1})]}(U = 200 \mid d_2^{100} = P, d_1^{100}) \cdot 200$$
$$+P(d_2^{100} = A) \cdot P_{M[(\sigma^{i-1})]}(U = 201 \mid d_2^{100} = A, d_1^{100}) \cdot 201]$$

While in MAID, the maximization function would be very large, since it contains a lot of impossible happenings. For example, it will contain items $P_{M[(\sigma^{i-1})]}(U = 200 \mid d_2^{100} = A, d_1^{100}) \cdot 200$, $P_{M[(\sigma^{i-1})]}(U = 200 \mid d_2^{100} = P, d_1^{100} = A) \cdot 200$, which in fact have probability 0. The algorithm for solving AMAID eliminates these impossible happenings automatically with contextual utility nodes.

In the Killer Game, the algorithm ends up generating a sequence of n games, each with n decisions. Therefore, the computation complexity still remains to be O(n!). But the computation complexity is indeed lower than MAID as the contextual utilities in our AMAID eliminates many events with probability 0 and will not be worse than extensive form tree.

5 Value of Information in Multi-agent Decision Systems

Value of information in multi-agent decision systems is quite important but much more complex than in single agent decision systems (Howard 1996, 1997). Agents can buy information to increase their expected utilities, but it is hard for them to decide the most valuable piece of information and the best time to buy that piece of information given their available resources. Agents can also release the information they have in hand to the other agents to influence the outcome of the system. But deciding what to share and when to share are the difficulties agents confronted. Other agents can also choose to believe the information or disbelieve it. We can also adopt Bayes' Theorem to update agents' believes, while this is not covered in our topic of discussion. This chapter defines value of information in multi-agent systems, followed by a numerical example.

5.1 Incorporating MAID into VOI Computation

Multi-agent influence diagram (Koller & Milch 2001) is a graphical representing model used to compute the Nash Equilibrium in games. The structure of MAIDs can be used to provide efficient algorithms for finding equilibria in non-cooperative games by dividing large games into smaller ones, which are proven to be more efficient than other algorithms based on game trees for certain type of games. Therefore, it can be used as a graphical representation to compute VOI in multi-agent systems.

Let us start with a simple multi-agent influence diagram *M* shown in Figure 5.1. This model involves two agents *A* and *B* and considers the relevance of one chance variable *N*. The expected utility of agent *A* if the agents play a given strategy profile σ is:

$$EU_a(\sigma) = \sum_{u_a \in dom(U_a)} P_{M[\sigma]}(U = u_a)u_a \tag{5.1}$$

The expected utility for agent *B* given the agents play strategy profile σ is:

$$EU_b(\sigma) = \sum_{u_b \in dom(U_b)} P_{M[\sigma]}(U = u_b)u_b$$

Assume δ_a and δ_b are the decision rules for the decision variables D_a and D_b respectively. Then the optimal $\delta_a{}^*$ for σ is:

$$\delta_a{}^* = \arg\max_{\delta_a{}^*} EU_a(\sigma_{D_b}, \delta_a{}^*)$$

By (5.1), this is equivalent to:

$$\delta_a{}^* = \arg\max_{\delta_a{}^*} \sum_{u_a \in dom(U_a)} P_{M[\sigma_{D_b}, \delta_a{}^*]}(U = u_a)u_a$$

Since $P_{M[\sigma_{D_b}, \delta_a{}^*]}(U = u_a) = \sum_{d_a \in dom(D_a)} P_{M[(\sigma_{D_b}, \delta_a *)]}(d_a) P_{M[(\sigma_{D_b}, \delta_a *)]}(U = u_a \mid d_a)$

$$= \sum_{d_a \in dom(D_a)} \delta^*(d_a) P_{M[(\sigma)]}(U = u_a \mid d_a),$$

Thus, the optimal $\delta_a{}^*$ for σ is

$$\delta_a{}^* = \arg\max_{\delta_a^*} \sum_{u_a \in dom(U_a)} \sum_{d_a \in dom(D_a)} \delta^*(d_a) P_{M[(\sigma)]}(U = u_a \mid d_a) \cdot u_a$$
$$= \arg\max_{\delta_a^*} \sum_{d_a \in dom(D_a)} \delta^*(d_a) \times \sum_{u_a \in dom(U_a)} P_{M[(\sigma)]}(U = u_a \mid d_a) \cdot u_a$$

Similar to δ_b, the optimal $\delta_b{}^*$ for σ is:

$$\delta_b{}^* = \arg\max_{\delta_b^*} EU_b(\sigma_{D_a}, \delta_b{}^*)$$

Which is equivalent to:

$$\delta_b{}^* = \arg\max_{\delta_b^*} \sum_{u_b \in dom(U_b)} P_{M[\sigma_{D_a}, \delta_b{}^*]}(U = u_b) u_b$$
$$\delta_b{}^* = \arg\max_{\delta_b^*} \sum_{d_b \in dom(D_b)} \delta^*(d_b) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma)]}(U = u_b \mid d_b) \cdot u_b$$

Therefore, the Nash equilibrium σ^* for this MAID M is ($\delta_a{}^*$, $\delta_b{}^*$), and the expected utilities of agent A and agent B are:

$$EU_a(\sigma^*) = \sum_{d_a \in dom(D_a)} \delta^*(d_a) \times \sum_{u_a \in dom(U_a)} P_{M[(\sigma^*)]}(U = u_a \mid d_a) \cdot u_a$$
$$EU_b(\sigma^*) = \sum_{d_b \in dom(D_b)} \delta^*(d_b) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma^*)]}(U = u_b \mid d_b) \cdot u_b$$

Note the difference between the expected value of multi-agent systems and single agent systems. When we compute VOI in single agent decision systems, we use the maximum expected utility and the optimal action of decision maker. However, in multi-agent systems, we use the maximum expected utility of each agent and the optimal decision rule of each agent given the decision rule of other agents. This implies that the action chosen by each agent may not be a Pareto optimality; but the given actions of the other agents, which means the action of each agent may not be an optimal one, but the optimal one given actions of the other agents. This is also the property of Nash equilibrium that no agent is willing to deviate from the strategy specified by the strategy profile if the other agents do not deviate. Therefore, in multi-agent systems, the maximum expected utility of each agent is the expected value when all agents choose their strategies specified by Nash equilibrium for that decision model.

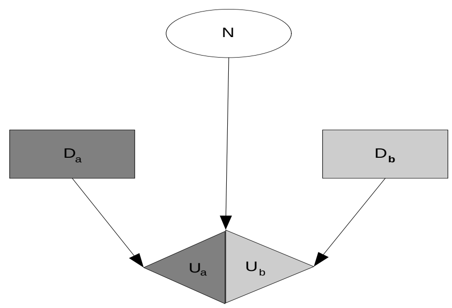


Figure 5.1 A MAID without information to any agent

Here we discuss the value of *N* in the multi-agent decision model (shown in Figure 5.1) by considering three situations: 1) the value of *N* is observed by agent *A* prior to making D_a, while agent *B* does not observe *N* but he/she is aware that *A* knows *N;* 2) the value of *N* is observed by agent *B* prior to making D_b, while agent *A* does not observe *N* but he/she is aware that *B* knows *N*; 3) the value of *N* is observed by both agents *A* and *B* prior to D_a and D_b, but they all know the other agent also observes *N*.

5.1.1 *N* is Observed By Agent *A* Prior to D_a

Figure 5.2 shows the MAID under the situation that the value of *N* is observed by agent *A* prior to D_a, while agent *B* does not observe *N* but he/she is aware that *A* knows N , A knows B knows A knows N so on and so forth (Assumption of *common knowledge*[1]).

The expected utility of agent A given a strategy profile σ if N is observed by agent A prior to D_a is

$$EU_a(\sigma) = \sum_{u_a \in dom(U_a)} P_{M[\sigma]}(U = u_a)u_a \tag{5.2}$$

Assume δ_a and δ_b are the decision rules for the decision variables D_a and D_b respectively. Then the optimal $\delta_a{}^*$ for σ is:

$$\delta_a{}^* = \arg\max_{\delta_a{}^*} EU_a(\sigma_{D_b}, \delta_a{}^*)$$

By (5.2), this is equivalent to:

$$\delta_a{}^* = \arg\max_{\delta_a{}^*} \sum_{u_a \in dom(U_a)} P_{M[\sigma_{D_b}, \delta_a{}^*]}(U = u_a)u_a$$

Breaking up the joint probability expression, we get

$$\begin{aligned} &P_{M[\sigma_{D_b}, \delta_a{}^*]}(U = u_a) \\ &= \sum_{\mathrm{Pa}_{D_a} \in dom(\mathrm{Pa}(D_a))} P_{M[(\sigma_{D_b}, \delta_a *)]}(\mathrm{Pa}_{D_a}) \\ &\qquad \times \sum_{d_a \in dom(D_a)} P_{M[(\sigma_{D_b}, \delta_a *)]}(d_a \mid \mathrm{Pa}_{D_a}) P_{M[(\sigma_b, \delta_a *)]}(U = u_a \mid d_a, \mathrm{Pa}_{D_a}) \end{aligned}$$

The CPD table for a node in a BN does not influence the prior distribution of its parents, so $\sum_{\mathrm{Pa}_{D_a} \in dom(\mathrm{Pa}(D_a))} P_{M[(\sigma_{D_b}, \delta_a *)]}(\mathrm{Pa}_{D_a}) = \sum_{\mathrm{Pa}_{D_a} \in dom(\mathrm{Pa}(D_a))} P_{M[(\sigma)]}(\mathrm{Pa}_{D_a})$. Given values for D and its parents, a distribution does not depend on the CPD of D, so $P_{M[(\sigma_b, \delta_a *)]}(U = u_a \mid d_a, \mathrm{Pa}_{D_a}) = P_{M[(\sigma)]}(U = u_a \mid d_a, \mathrm{Pa}_{D_a})$.

[1] Common knowledge is the basic assumption in the games which is a special kind of knowledge for a group of agents. There is common knowledge P in a group of agents G if all agents in G know P, they all know that they know P, they all know that they all know that they know P, and so on and infinitum.

Therefore,

$$P_{M[\sigma_{D_b},\delta_a^*]}(U=u_a)$$
$$=\sum_{\mathrm{Pa}_{D_a}\in dom(\mathrm{Pa}(D_a))} P_{M[(\sigma)]}(\mathrm{Pa}_{D_a})\times\sum_{d_a\in dom(D_a)}\delta^*(d_a\mid \mathrm{Pa}_{D_a})P_{M[(\sigma)]}(U=u_a\mid d_a,\mathrm{Pa}_{D_a})$$

Thus, the optimal $\delta_a{}^*$ for σ is:

$$\delta_a{}^*=\arg\max_{\delta_a{}^*}\sum_{u_a\in dom(U_a)}\sum_{\mathrm{Pa}_{D_a}\in dom(\mathrm{Pa}(D_a))}P_{M[(\sigma)]}(\mathrm{Pa}_{D_a})$$
$$\times\sum_{d_a\in dom(D_a)}\delta^*(d_a\mid \mathrm{Pa}_{D_a})P_{M[(\sigma)]}(U=u_a\mid d_a,\mathrm{Pa}_{D_a})\cdot u_a$$

Rearranging the summations, we get:

$$\delta_a{}^*=\arg\max_{\delta_a{}^*}\sum_{\mathrm{Pa}_{D_a}\in dom(\mathrm{Pa}(D_a))}P_{M[(\sigma)]}(\mathrm{Pa}_{D_a})\sum_{d_a\in dom(D_a)}\delta^*(d_a\mid \mathrm{Pa}_{D_a})$$
$$\times\sum_{u_a\in dom(U_a)}P_{M[(\sigma)]}(U=u_a\mid d_a,\mathrm{Pa}_{D_a})\cdot u_a$$

In this decision model, Pa_{D_a}=n and $n\in dom(N)$, thus $\delta_a{}^*$ is optimal for σ if and only if $\delta_a{}^*$ is the solution of the following maximization problem:

$$\delta_a{}^*=\arg\max_{\delta_a{}^*}\sum_{n\in dom(N)}P_{M[(\sigma)]}(n)\sum_{d_a\in dom(D_a)}\delta^*(d_a\mid n)$$
$$\times\sum_{u_a\in dom(U_a)}P_{M[(\sigma)]}(U=u_a\mid d_a,n)\cdot u_a$$

Similarly for agent *B*, δ_b is optimal for σ if and only if δ_b is the solution of the following maximization problem:

$$\delta_b{}^*=\arg\max_{\delta_b{}^*}\sum_{d_b\in dom(D_b)}\delta^*(d_b)\times\sum_{u_b\in dom(U_b)}P_{M[(\sigma)]}(U=u_b\mid d_b)\cdot u_b$$

Therefore, the Nash equilibrium σ^* for this MAID *M* is ($\delta_a{}^*$, $\delta_b{}^*$), and the maximum expected utilities of agent *A* and agent *B* are:

$$EU_a(\sigma^*)=\sum_{n\in dom(N)}P_{M[(\sigma^*)]}(n)\sum_{d_a\in dom(D_a)}\delta^*(d_a\mid n)\times\sum_{u_a\in dom(U_a)}P_{M[(\sigma^*)]}(U=u_a\mid d_a,n)\cdot u_a$$
$$EU_b(\sigma^*)=\sum_{d_b\in dom(D_b)}\delta^*(d_b)\times\sum_{u_b\in dom(U_b)}P_{M[(\sigma^*)]}(U=u_b\mid d_b)\cdot u_b$$

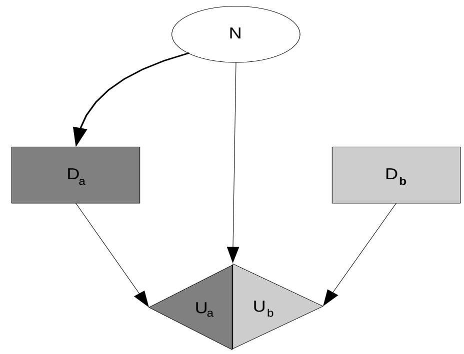


Figure 5.2 A MAID with agent *A* knowing the information

5.1.2 *N* is Observed By Agent *B* Prior to D_b

Figure 5.3 shows the MAID under the situation that the value of *N* is observed by agent *B* prior to D_b, while agent *A* does not observe *N* but he/she knows that *B* knows *N*.

The expected utility of agent *A* given a strategy profile σ if *N* is not observed by agent *A* prior to D_a is:

$$EU_a(\sigma) = \sum_{u_a \in dom(U_a)} P_{M[\sigma]}(U = u_a)u_a$$

The expected utility for agent *B* given the agents play strategy profile σ if *N* is observed by agent *B* prior to D_b is:

$$EU_b(\sigma) = \sum_{u_b \in dom(U_b)} P_{M[\sigma]}(U = u_b)u_b$$

The optimal δ_a^* and δ_b^* for σ are:

$$\delta_a^* = \arg\max_{\delta_a^*} \sum_{d_a \in dom(D_a)} \delta^*(d_a) \times \sum_{u_a \in dom(U_a)} P_{M[(\sigma)]}(U = u_a \mid d_a) \cdot u_a$$

$$\delta_b^* = \arg\max_{\delta_b^*} \sum_{\mathrm{Pa}_{D_b} \in dom(\mathrm{Pa}(D_b))} P_{M[(\sigma)]}(\mathrm{Pa}_{D_b}) \sum_{d_b \in dom(D_b)} \delta^*(d_b \mid \mathrm{Pa}_{D_b})$$

$$\times \sum_{u_b \in dom(U_b)} P_{M[(\sigma)]}(U = u_b \mid d_b, \mathrm{Pa}_{D_b}) \cdot u_b$$

$$= \arg\max_{\delta_b^*} \sum_{n \in dom(N)} P_{M[(\sigma)]}(n) \sum_{d_b \in dom(D_b)} \delta^*(d_b \mid n)$$

$$\times \sum_{u_b \in dom(U_b)} P_{M[(\sigma)]}(U = u_b \mid d_b, n) \cdot u_b$$

Therefore, the Nash equilibrium σ^* for this MAID M is (δ_a^* , δ_b^*), and the maximum expected utilities of agent *A* and agent *B* are:

$$EU_a(\sigma^*) = \sum_{d_a \in dom(D_a)} \delta^*(d_a) \times \sum_{u_a \in dom(U_a)} P_{M[(\sigma^*)]}(U = u_a \mid d_a) \cdot u_a$$

$$EU_b(\sigma^*) = \sum_{n \in dom(N)} P_{M[(\sigma^*)]}(n) \sum_{d_b \in dom(D_b)} \delta^*(d_b \mid n) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma^*)]}(U = u_b \mid d_b, n) \cdot u_b$$

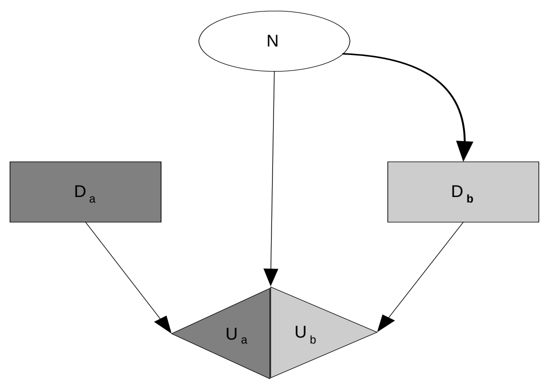


Figure 5.3 A **MAID with agent *B* knowing the information**

5.1.3 *N* is Observed By Both Agents *A* and *B*

Figure 5.4 shows the MAID under the situation that the value of *N* is observed by both agents *A* and *B* prior to D_a and D_b and it is the common knowledge that the value of *N* is revealed to both of them.

The expected utility of agent *A* given a strategy profile σ if N is observed by agent *A* prior to D_a is:

$$EU_a(\sigma) = \sum_{u_a \in dom(U_a)} P_{M[\sigma]}(U = u_a) u_a$$

The optimal $\delta_a{}^*$ for σ is:

$$\delta_a{}^* = \arg\max_{\delta_a{}^*} \sum_{\mathrm{Pa}_{D_a} \in dom(\mathrm{Pa}(D_a))} P_{M[(\sigma)]}(\mathrm{Pa}_{D_a}) \sum_{d_a \in dom(D_a)} \delta^*(d_a \mid \mathrm{Pa}_{D_a})$$
$$\times \sum_{u_a \in dom(U_a)} P_{M[(\sigma)]}(U = u_a \mid d_a, \mathrm{Pa}_{D_a}) \cdot u_a$$

Similarly for $\delta_b{}^*$,

$$\delta_b{}^* = \arg\max_{\delta_b{}^*} \sum_{\mathrm{Pa}_{D_b} \in dom(\mathrm{Pa}(D_b))} P_{M[(\sigma)]}(\mathrm{Pa}_{D_b}) \sum_{d_b \in dom(D_b)} \delta^*(d_b \mid \mathrm{Pa}_{D_b})$$
$$\times \sum_{u_b \in dom(U_b)} P_{M[(\sigma)]}(U = u_b \mid d_b, \mathrm{Pa}_{D_b}) \cdot u_b$$

In this decision model, Pa_{D_a}=n, $\mathrm{Pa}_{D_b} = n$ and $n \in dom(N)$, thus $\delta_a{}^*$ and $\delta_b{}^*$ are optimal for σ if and only if $\delta_a{}^*$ and $\delta_b{}^*$ are the solutions of the following maximization problem:

$$\delta_a{}^* = \arg\max_{\delta_a{}^*} \sum_{n \in dom(N)} P_{M[(\sigma)]}(n) \sum_{d_a \in dom(D_a)} \delta^*(d_a \mid n)$$
$$\times \sum_{u_a \in dom(U_a)} P_{M[(\sigma)]}(U = u_a \mid d_a, n) \cdot u_a$$
$$\delta_b{}^* = \arg\max_{\delta_b{}^*} \sum_{n \in dom(N)} P_{M[(\sigma)]}(n) \sum_{d_b \in dom(D_b)} \delta^*(d_b \mid n)$$
$$\times \sum_{u_b \in dom(U_b)} P_{M[(\sigma)]}(U = u_b \mid d_b, n) \cdot u_b$$

Therefore, the Nash equilibrium σ^* for this MAID M is ($\delta_a{}^*$, $\delta_b{}^*$), and the maximum expected utilities of agent A and agent B are:

$$EU_a(\sigma^*) = \sum_{n \in dom(N)} P_{M[(\sigma^*)]}(n) \sum_{d_a \in dom(D_a)} \delta^*(d_a \mid n) \times \sum_{u_a \in dom(U_a)} P_{M[(\sigma^*)]}(U = u_a \mid d_a, n) \cdot u_a$$
$$EU_b(\sigma^*) = \sum_{n \in dom(N)} P_{M[(\sigma^*)]}(n) \sum_{d_b \in dom(D_b)} \delta^*(d_b \mid n) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma^*)]}(U = u_b \mid d_b, n) \cdot u_b$$

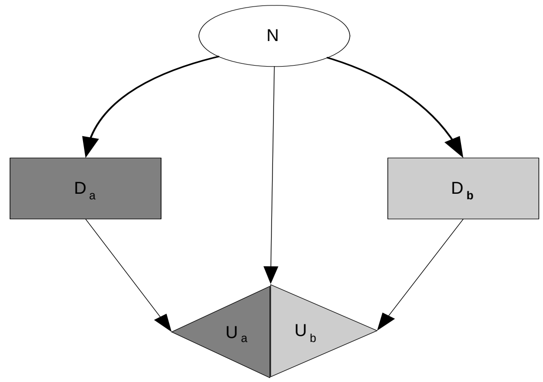


Figure 5.4 A MAID with both agents *A* and *B* knowing the information

5.2 VOI in Multi-agent Systems – Some Discussions and Definitions

In the previous section, we have discussed a simple multi-agent decision model with extension to three situations: all agents without information; part of the agents having the perfect information and all agents with perfect information. Having computed the expected utilities of each agent in each decision model, we can now define the VOI in multi-agent decision systems.

Definition 5.1 In Multi-agent decision systems with a set of agents *I*, s*ymmetric information* is the information which is revealed to all the agents in the decision system. *Asymmetric information* is the information that is revealed to a confined subset of agents in the decision system.

Take the multi-agent decision model represented by Figure 5.2 for example, the information for observing the value of N before D_a is the *asymmetric information* of agent A, since it is only revealed to agent A. In the decision model represented by Figure 5.4, the information for observing the value of N before D_a and D_b is the symmetric information, since the information is revealed to all the agents A and B in the system.

In the single agent decision systems, VOI is calculated decision by decision. In other words, the same variable may have different values to different decisions. According to the "no-forgetting" rules that the information observed at one decision point is always available at subsequent decision points, it is not difficult to imagine that the information value viewed at an earlier decision point is no less than the value of same information viewed at the subsequent decision points. In multi-agent systems, this fact may not hold anymore.

We first begin our discussion by defining VOI in multi-agent systems firstly. In all the definitions and calculations below, utility is the value measure.

Definition 5.2 An agent i has *perfect recall* with respect to a total order $D_{1,}...,D_n$ ($D_{1,}...,D_n \in D_i$) if $D_k \in Pa(D_l)$ and $Pa(D_k) \subset Pa(D_l)$ for all $D_k, D_l \in D_i$, $k \prec l$, where D_i is a decision node of agent *i*.

This definition implies the "no-forgetting" condition in multi-agent decision systems, which guarantees that any information that is observed before one decision point of a specific agent i, is also observed at subsequent decision points of agent *i*.

Definition 5.3 In Multi-agent decision systems with a set of agents *I*, the *expected value of information* on an uncertain variable *N* before decision D of agent i (where $D \in D_i$ and D_i is a decision node of agent $i \in I$) is the difference between the expected value of agent i given the state of *N* is known to agent *i* before decision D in the system and the expected value of agent i without knowing the state of *N*. We use $EVPI_i$ (*D*|*N*) to denote it.

This definition is based on the EVPI definition in single agent systems. The analysis of EVPI is helpful for a specific agent to decide which piece of information is more valuable to him/her or when to buy that piece of information will maximize its expected utility on the condition that the other agents do not know the information. However, in multi-agent systems, things are no longer that simple. Agents are not only allowed to buy the information, but they could also release the information to other agents to influence their expected utilities.

Therefore, it is important for agents to evaluate the value of information given a subset of agents also knows that information. Specially, what we evaluate most in games are the changes of expected value of each agent when the information is released to all the agents at the beginning of the game. We define *expected value of symmetric information* and *expected value of perfect asymmetric information* as follows.

Definition 5.4 In Multi-agent decision systems with a set of agents *I*, the *expected value of symmetric information* on an uncertain variable *N* to an agent $i \in I$, denoted by $\text{EVPSI}_i(N)$, is the difference between the expected value of agent *i* if the state of *N* is known before all agents' decisions in the system and the expected value of agent i if the state of *N* remains unknown to any agent in the system. The *expected value of perfect asymmetric information* (EVPAI) of each agent $i \in I$, denoted by $\text{EVPAI}_i(D|N)$, is the difference between the expected value of agent i with perfect asymmetric information that is revealed to a subset of agents K ($K \subset I$) and without it.

Express it with the formula,

$EVPSI_i$=EV_i(with perfect symmetric information)-EV_i(with no information);

$EVPAI_i$= EV_i(with perfect asymmetric information)-EV_i(with no information).

The models described in Section 5.1 will be used to illustrate this definition.

1) Model represented by Figure 5.1: Expected Utility without Information

Assuming Nash equilibrium σ^* in the model without information represented by Figure 5.1 is (δ_a^*, δ_b^*),

$$\delta_a^* = \arg\max_{\delta_a^*} \sum_{d_a \in dom(D_a)} \delta^*(d_a) \times \sum_{u_a \in dom(U_a)} P_{M[(\sigma)]}(U = u_a \mid d_a) \cdot u_a$$

$$\delta_b^* = \arg\max_{\delta_b^*} \sum_{d_b \in dom(D_b)} \delta^*(d_b) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma)]}(U = u_b \mid d_b) \cdot u_b$$

Thus the expected utilities of agents A and B are :

$$EU_a(\sigma^*) = \sum_{d_a \in dom(D_a)} \delta^*(d_a) \times \sum_{u_a \in dom(U_a)} P_{M[(\sigma^*)]}(U = u_a \mid d_a) \cdot u_a$$

$$EU_b(\sigma^*) = \sum_{d_b \in dom(D_b)} \delta^*(d_b) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma^*)]}(U = u_b \mid d_b) \cdot u_b$$

2) Model represented by Figure 5.2: Expected Utility with Asymmetric Information

If agent A knows the information before making decision D_a, Nash equilibrium $\sigma^*{}_{N \to D_a}$ in the model with asymmetric information N to A is $(\delta_a^*{}_{N \to D_a}, \delta_b^*{}_{N \to D_a})$,

$$\delta_a^*{}_{N \to D_a} = \arg\max_{\delta_a^*{}_{N \to D_a}} \sum_{n \in dom(N)} P_{M[(\sigma_{N \to D_a})]}(n) \sum_{d_a \in dom(D_a)} \delta^*{}_{N \to D_a}(d_a \mid n)$$
$$\times \sum_{u_a \in dom(U_a)} P_{M[(\sigma_{N \to D_a})]}(U = u_a \mid d_a, n) \cdot u_a$$

$$\delta_b{}^*{}_{N\to D_a} = \arg\max_{\delta_b{}^*{}_{N\to D_a}} \sum_{d_b \in dom(D_b)} \delta^*{}_{N\to D_a}(d_b) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma_{N\to D_a})]}(U = u_b \mid d_b) \cdot u_b$$

The expected utilities of agents *A* and *B* are:

$$EU_a(\sigma^*{}_{N\to D_a}) = \sum_{n\in dom(N)} P_{M[(\sigma^*{}_{N\to D_a})]}(\mathrm{Pa}_{D_a}) \sum_{d_a \in dom(D_a)} \delta^*{}_{N\to D_a}(d_a \mid \mathrm{Pa}_{D_a})$$
$$\times \sum_{u_a \in dom(U_a)} P_{M[(\sigma^*{}_{N\to D_a})]}(U = u_a \mid d_a, \mathrm{Pa}_{D_a}) \cdot u_a$$

$$EU_b(\sigma^*{}_{N\to D_a}) = \sum_{d_b \in dom(D_b)} \delta^*{}_{N\to D_a}(d_b) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma^*{}_{N\to D_a})]}(U = u_b \mid d_b) \cdot u_b$$

Therefore, the EVPAI of agent *A* is as follows:

EVPAI$_a$(D_a |*N*)= $EU_a(\sigma^*{}_{N\to D_a}) - EU_a(\sigma^*)$

$$= \sum_{n\in dom(N)} P_{M[(\sigma^*{}_{N\to D_a})]}(n) \sum_{d_a \in dom(D_a)} \delta^*{}_{N\to D_a}(d_a \mid n) \times \sum_{u_a \in dom(U_a)} P_{M[(\sigma^*{}_{N\to D_a})]}(U = u_a \mid d_a, n) \cdot u_a$$
$$- \sum_{d_a \in dom(D_a)} \delta^*(d_a) \times \sum_{u_a \in dom(U_a)} P_{M[(\sigma^*)]}(U = u_a \mid d_a) \cdot u_a$$

With agent *A* knowing the information, the expected value of agent *B* also will have a change. The difference can be expressed by the following formula:

$$EU_b(\sigma^*{}_{N\to D_a}) - EU_b(\sigma^*)$$
$$= \sum_{d_b \in dom(D_b)} \delta^*{}_{N\to D_a}(d_b) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma^*{}_{N\to D_a})]}(U = u_b \mid d_b) \cdot u_b$$
$$- \sum_{d_b \in dom(D_b)} \delta^*(d_b) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma^*)]}(U = u_b \mid d_b) \cdot u_b$$

If D_b is not s-reachable from D_a, which means that $\delta_b{}^*$ is independent of the decision rule that σ^* assigns to D_a, then

$$\sum_{d_b \in dom(D_b)} \delta^*{}_{N\to D_a}(d_b) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma^*{}_{N\to D_a})]}(U = u_b \mid d_b) \cdot u_b$$
$$= \sum_{d_b \in dom(D_b)} \delta^*(d_b) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma^*)]}(U = u_b \mid d_b) \cdot u_b \ ,$$

thus $EU_b(\sigma^*_{N\to D_a})$ - $EU_b(\sigma^*)$=0, which means that the expected value of agent B will not change with agent A knowing the information if D_b is not s-reachable from D_a (refer to Chapter 6 for detailed proof).

3) Model represented by Figure 5.4: Expected Utility with Full Information

If both agent A and B know the information before making decision D_a, Nash equilibrium $\sigma^*_{N\to D_{ab}}$ in the model with symmetric information N to A is ($\delta_a{}^*{}_{N\to D_{ab}}$, $\delta_b{}^*{}_{N\to D_{ab}}$).

$$\delta_a{}^*{}_{N\to D_{ab}} = \arg\max_{\delta_a{}^*{}_{N\to D_{ab}}} \sum_{n\in dom(N)} P_{M[(\sigma_{N\to D_{ab}})]}(n) \sum_{d_a\in dom(D_a)} \delta^*{}_{N\to D_{ab}}(d_a \mid n)$$
$$\times \sum_{u_a\in dom(U_a)} P_{M[(\sigma_{N\to D_{ab}})]}(U=u_a \mid d_a, n)\cdot u_a$$
$$\delta_b{}^*{}_{N\to D_{ab}} = \arg\max_{\delta_b{}^*{}_{N\to D_{ab}}} \sum_{n\in dom(N)} P_{M[(\sigma_{N\to D_{ab}})]}(n) \sum_{d_b\in dom(D_b)} \delta^*{}_{N\to D_{ab}}(d_b \mid n)$$
$$\times \sum_{u_b\in dom(U_b)} P_{M[(\sigma_{N\to D_{ab}})]}(U=u_b \mid d_b, n)\cdot u_b$$

The expected utilities of agent A and B are:

$$EU_a(\sigma^*{}_{N\to D_{ab}}) = \sum_{n\in dom(N)} P_{M[(\sigma^*{}_{N\to D_{ab}})]}(n) \sum_{d_a\in dom(D_a)} \delta^*{}_{N\to D_{ab}}(d_a \mid n)$$
$$\times \sum_{u_a\in dom(U_a)} P_{M[(\sigma^*{}_{N\to D_{ab}})]}(U=u_a \mid d_a, n)\cdot u_a$$
$$EU_b(\sigma^*{}_{N\to D_{ab}}) = \sum_{n\in dom(N)} P_{M[(\sigma^*{}_{N\to D_{ab}})]}(n) \sum_{d_b\in dom(D_b)} \delta^*{}_{N\to D_{ab}}(d_b \mid n)$$
$$\times \sum_{u_b\in dom(U_b)} P_{M[(\sigma^*{}_{N\to D_{ab}})]}(U=u_b \mid d_b, n)\cdot u_b$$

Therefore,

EVPSI$_a$(N)= $EU_a(\sigma^*{}_{N\to D_{ab}})$ - $EU_a(\sigma^*)$

$$= \sum_{n \in dom(N)} P_{M[(\sigma^*_{N \to D_{ab}})]}(n) \sum_{d_a \in dom(D_a)} \delta^*_{N \to D_{ab}}(d_a \mid n)$$
$$\times \sum_{u_a \in dom(U_a)} P_{M[(\sigma^*_{N \to D_{ab}})]}(U = u_a \mid d_a, n) \cdot u_a$$
$$- \sum_{d_a \in dom(D_a)} \delta^*(d_a) \times \sum_{u_a \in dom(U_a)} P_{M[(\sigma^*)]}(U = u_a \mid d_a) \cdot u_a$$

EVPSI$_b$(N)= $EU_b(\sigma^*_{N \to D_{ab}}) - EU_b(\sigma^*)$

$$= \sum_{n \in dom(N)} P_{M[(\sigma^*_{N \to D_{ab}})]}(n) \sum_{d_b \in dom(D_b)} \delta^*_{N \to D_{ab}}(d_b \mid n)$$
$$\times \sum_{u_b \in dom(U_b)} P_{M[(\sigma^*_{N \to D_{ab}})]}(U = u_b \mid d_b, n) \cdot u_b$$
$$- \sum_{d_b \in dom(D_b)} \delta^*(d_b) \times \sum_{u_b \in dom(U_b)} P_{M[(\sigma^*)]}(U = u_b \mid d_b) \cdot u_b$$

We can tell the difference between EVPSI, EVPAI and EVPI. EVPI computes the information decision by decision. Alternatively, it computes the information value difference among decisions given the same information. This is sufficient in single agent decision systems. However, in multi-agent decision systems, agents are interactive. Even if they do not know the information, the fact that other agents know the information may also influence their expected utility. Besides, we sometimes need to compare the value difference among agents given they have the same information. When taking the joint value of information into consideration, the problem gets more complex in multi-agent decision systems, and the joint value of information can involve different agents. Here we would confine our discussions in the computation of EVPI, EVPSI and EVPAI.

5.3 Numerical Examples

We use a numerical example to illustrate our above-mentioned example of VOI computation in MAS further. This numerical example is adopted from Bruno Bassan et al. (2003).

A and B are two manufacturers in the market. The next year's market competition will have two possible states: intense (S_2) and not intense (S_1), each with probability 1/2. Assume A and B both have two choices for their production quantities: large quantity and small quantity. For the different states and different choices, there are different payoffs associated with them. Below shows the two different payoff matrices S_1 and S_2 associated with the two different market states.

$S_1=$	L	S
L	0,0	6,-3
S	-3,6	5,5

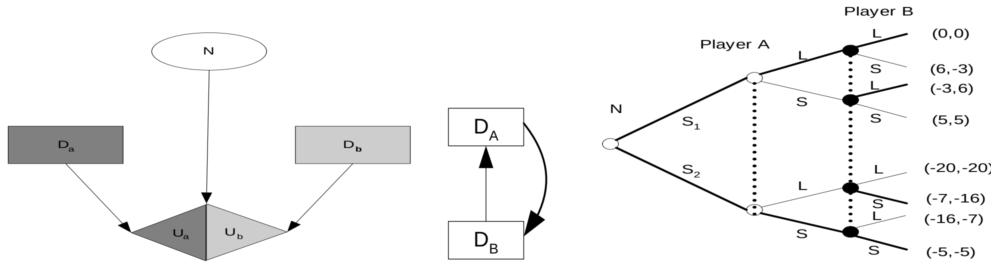
$S_2=$	L	S
L	-20,-20	-7,-16
S	-16,-7	-5,-5

Table 5.1 Utility matrices of the two manufacturers

Consider these four situations: 1) the value of N is not revealed to both of the manufacturers; 2) the state value of N is revealed to manufacturer A before A makes its decision, while manufacturer B does not observe N and it is common knowledge that A knows N ; 3) the value of N is revealed to manufacturer B

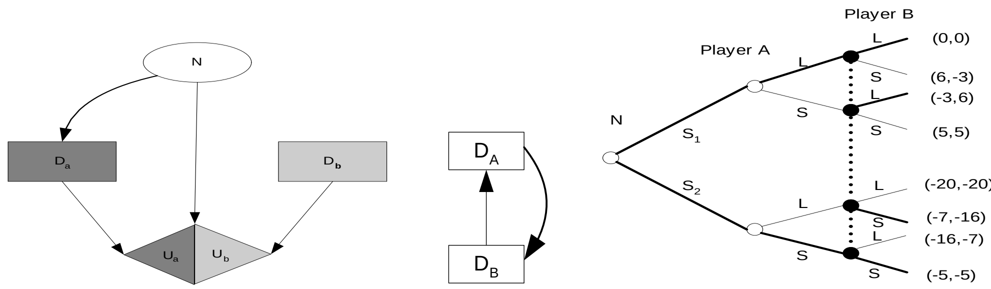
before B makes its decision, while manufacturer A does not observe N and B knows N is a common knowledge;4) the value of N is observed by both agents A and B prior to their decisions, and it is common knowledge that they all observes N.

The MAIDs, the corresponding relevance graphs and game trees under the four situations are shown in Figure 5.5.



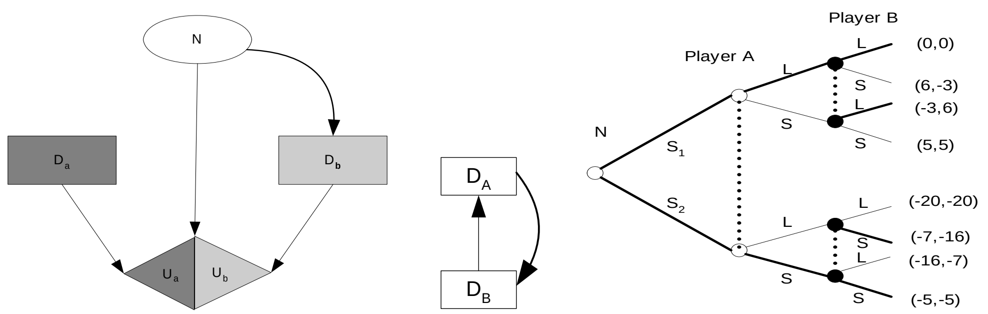
(a-1) (a-2) (a-3)

(a) Graph representation under situation 1



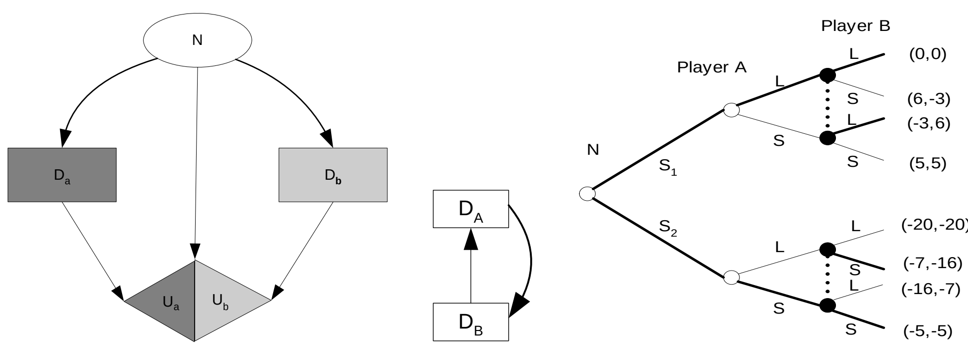
(b-1) (b-2) (b-3)

(b) Graph representation under situation 2



(c-1) (c-2) (c-3)

(c) Graph representation under situation 3



(d-1) (d-2) (d-3)

(d) Graph representation under situation 4

Figure 5.5 The MAIDs, relevance graphs and game tree of manufacturer example

Solving the MAIDs by using the divide and conquer algorithm in cyclic relevance graphs of each situation, we get the following result of expected utilities.

	B-Informed	B-Uninformed
A-Informed	(-2.5, -2.5)	(-8, -3.5)
A-Uninformed	(-3.5, -8)	(0,0)

Table 5.2 Expected utilities of the four situations

In this numerical example, $EVPSI_i$=EV_i(With perfect symmetric information)-EV_i(With no information)=-2.5-0=-2.5. $EVPI_i$=$EVPAI_i$=EV_i(With perfect asymmetric information to agent *i*)-EV_i(With no information)=-8-0=-8.

For each agent, the values of perfect symmetric information and perfect asymmetric information are both negative. This is unlike the situation in single agent decision systems where the decision maker is better off knowing the information. Therefore, in multi-agent decision systems, EVPI is not bounded by 0, which means that knowing the information may be even worse than not knowing the information. This makes the VOI problem in multi-agent decision systems much more complex than in single agent systems. Information may be of no value in multi-agent decision systems. Because of this, agents can increase their own utilities by releasing the information to the other agents. It is also possible that the agents have to pay in order to choose not to know the information.

If the utility matrices of agent A and B are changed to the following matrices without changing the game structure:

$S_1=$	L	S
L	2,2	-1,-6
S	-6,-1	-2,-2

$S_2=$	L	S
L	-20,-20	-5,-5
S	-5,-5	2,2

Table 5.3 Utility matrices of the two manufacturers-Example 2

The resulting solutions are:

	B-Informed	B-Uninformed
A-Informed	(2,2)	(-1.5,-1.5)
A-Uninformed	(-1.5, -1.5)	(0,0)

Table 5.4 Expected utilities of the four situations-Example 2

In this example, the EVPSI of each agent is 2, which is positive. EVPAI still remains to be negative.

Although the value of information in multi-agent systems can be negative, whether it is negative or not is influenced by the utility function, which is beyond our scope here. In this thesis, we would confine studies on the influence of graph structure to the value of information.

From this example, we can get our justification below.

Justification 5.1: In single agent decision systems, VOI is bounded by 0 and EVPI; While in multi-agent decision systems, VOI can be less than 0.

Definition 5.5 The absolute value of VOI in MAS is how much the agents are willing to pay on that set of information. If VOI≤0, it is called the *Escaping Cost*, which is the cost the agents are willing to pay for escaping to know the information. If VOI ≥ 0, it is called *Revealing Cost*, which is the cost the agents are willing to pay for knowing the information.

In the first numerical example 1, the *escaping cost* of agent *A* is –3.5 given *B* does not know the information and the *revealing cost* of agent *A* is 1 given B knows the information.

In single agent decision systems, the information involved only has one type, the state of chance variables. Due to the "no-forgetting" condition in single agent decision systems, the decision maker will never forget the state of its previous decision variables. However, decisions made by one agent are not necessarily observed by other agents in multi-agent decision systems, as in the simultaneous games. Therefore, it is meaningful to discuss another type of information- the state of decision variables. By calculating the information of other agents' decision variables, agents can decide whether to spy on other agents' decision or whether agents can cooperate together if knowing the other agents' decision will increase their total utilities.

Definition 5.6 In multi-agent systems, the information coming from the chance node is called *Nature Information*; while the information coming from other agents' decision node is called *Moving Information*.

Note that this definition is based on the assumption of common knowledge, i.e. agent A knows agent B knows agent A know… and so on. Although in real life, if agent A spies on agent B's action, agent B may not know agent A knows his/her action already, but this situation is not in our scope of discussion.

In the commercial market, the competitor's production type is a kind of nature information (For example, whether his production function is high cost or low cost); and what the competitor is going to produce is the moving information.

5.4 Value of Information for the Intervened Variables in Multi-agent Decision Systems

5.4.1 Problem

In single agent decision systems, some IDs have chance nodes with decision nodes as their parents. These chance variables are called decision-intervened variables. In traditional EVPI computation in single agent decision systems, the decision-intervened variables are not taken into consideration since adding an arc from the decision-intervened variable to its parent (the decision node) will form a directed cycle, which is contradictory to the DAG foundation in graphical decision models. Not only the quantitative computation of VOI for intervened variables becomes a problem, but also the actual meanings of VOI for intervened variables.

Assume a clairvoyant is able to provide us the perfect information about the future. Consider the following situation: one decision maker needs information about an uncertain variable *X* before his/her decision *D*, but *X* is influenced by his decision *D*. If the clairvoyant tells him the prefect information of *X*'s state *x*, the decision maker can adjust his/her action *D* to achieve the maximum utility. However, X is influenced by the decision *D*. By changing *D*, X may no longer be in the state *x*, which is contradictory to the perfect information assumption.

Otherwise, if the clairvoyant wants to provide perfect information, he/she must know the decision maker's decision before he/she provides the perfect information of X. This indicates the decision maker's action is predetermined, which is also contradictory to the freewill assumption of decision maker's decision.

However, knowing VOI of a decision-intervened variable can still facilitate decision maker's decision in the practical use. In single agent decision systems, knowing VOI of decision-intervened variables provides the decision-maker an upper limit of the benefit that the current decision can achieve. This can be explained by a simple example shown in Figure 5.6. A farmer has to decide how many orange trees to grow in the coming year in order to achieve the maximum profit. However, the profit is also influenced by the price of the orange in the market. Assume this is a free market and the price of the product in the market is completely determined by the demand and supply. The farmer would like to pay for the information about the price of the product. In the meantime, the price of the product in the market next year is also influenced by the quantity the farmer is going to grow next year. In this example, the farmer may still be willing to know the exact price of the orange on the market of the next year before he/she grows any orange trees this year. Because knowing the price of the product helps the farmer to decide how many orange trees to grow or whether to grow any other plants. If the price of the orange is below the cost, then the farmer can grow other plants in his/her field to avoid this loss.

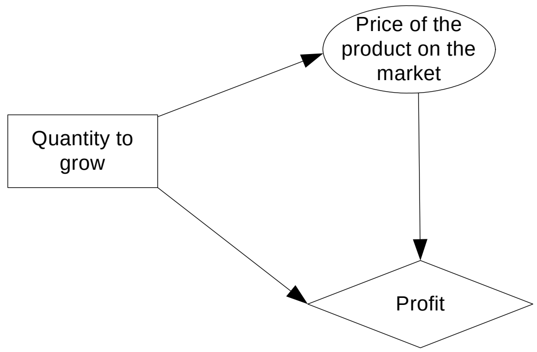


Figure 5.6 An ID of decision-intervened variables in single agent decision systems

In multi-agent decision systems, discussing the VOI of the decision-intervened variables may be much more meaningful. We extend the above example to multi-agent decision systems for further illustration. Figure 5.7 shows the MAID representation of the example. Assume there are two monopolists A and B on the market. The price of that product on the market is completely determined by the demand and the total quantities they produce. In the MAID representation of this example, the "price of the product on the market" chance node is influenced by two decisions nodes Q_A and Q_B, which means the price can only be known after the two decision nodes have been decided. If monopolist A can get the perfect information of the "Price" chance node, it can adjust its own decision to increase its profit, and so does monopolist B. If the information of the "Price" chance node is revealed to both of them, the situation becomes much more complex. We introduce the canonical form of MAID firstly.

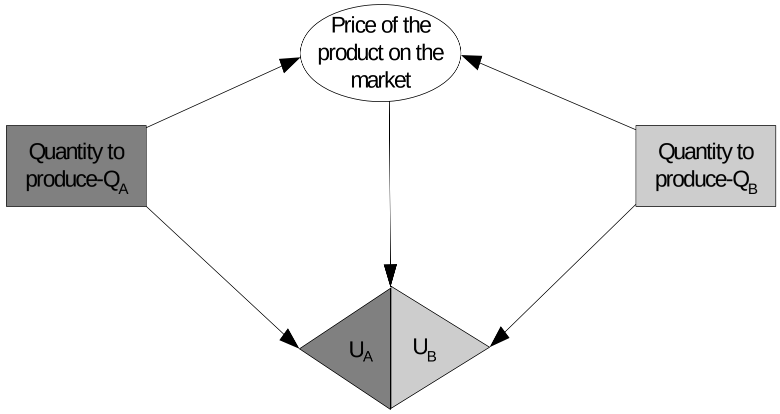


Figure 5.7 MAID of decision-intervened variables in Multi-agent decision system

5.4.2 Canonical Form of MAIDs

When computing VOI of decision-intervened variables in single agent decision models, we cannot simply add an arc from the decision node to the chance node, since this will incur a cycle in the influence diagram. The requirement of the chance nodes not being the descendants of decision nodes when computing VOI in the influence diagrams can be satisfied by reformulating the influence diagrams into canonical form (Howard 1990, Heckerman 1995). An influence diagram is said to be in canonical form if there are no chance nodes that are descendants of decision nodes in IDs. We extend this definition to MAIDs by defining that a MAID is in canonical form if there are no chance nodes that are descendants of decision nodes in MAIDs. Then we can reformulate MAIDs with decision-intervened chance nodes into canonical form following the Howard procedure of converting descendant chance nodes into deterministic nodes and introducing mapping variables which are not influenced by the decision nodes.

The algorithm developed by Heckerman and Shachter (1995) for constructing canonical form of generic influence diagram is as follows.

Given a decision problem with respect to the set of decision nodes D and set of chance nodes C:

1. Add a node to the diagram corresponding to each variable in $C \bigcup D$
2. Order the variables X_1, …X_n in C in the order that the variables unresponsive[2] to D comes first
3. For each variable $X_i \in C$ that is responsive to D,

a) Add a causal-mapping-variable chance node Xi(Vi) to the diagram, where $V_i \subseteq D \bigcup \{X_1,...,X_{i-1}\}$

b) Make X_i a deterministic node with parents V_i and $X_i(V_i)$

4. Assess independencies among the variables that are unresponsive to D

We now extend this algorithm to the MAIDs to construct canonical form MAIDs.

Definition 5.7 A MAID is said to be in *canonical form* if (1) all chance nodes that

[2] In Heckerman and Shachter (1995), a chance node *X* is unresponsive to decision *D* means *X* has the same outcome no matter what D is taken. In other word, *X* is probabilistically independent of *D* if *X* is unresponsive to *D*, but not vice versa.

are responsive to D are descendants of one or more decision nodes and (2) all chance nodes that are descendants of one or more decision nodes are deterministic nodes. Thus, there are no chance nodes that are descendants of decision nodes in the MAIDs.

We can now construct the canonical form of the MAID shown in Figure 5.8 as follows.

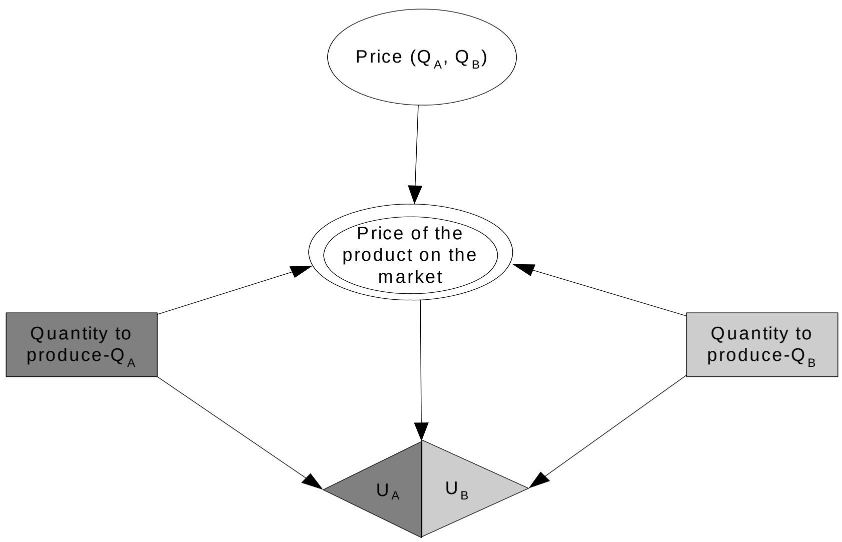


Figure 5.8 Canonical Form of MAID

5.4.3 Independence Assumption in Canonical Form of MAID

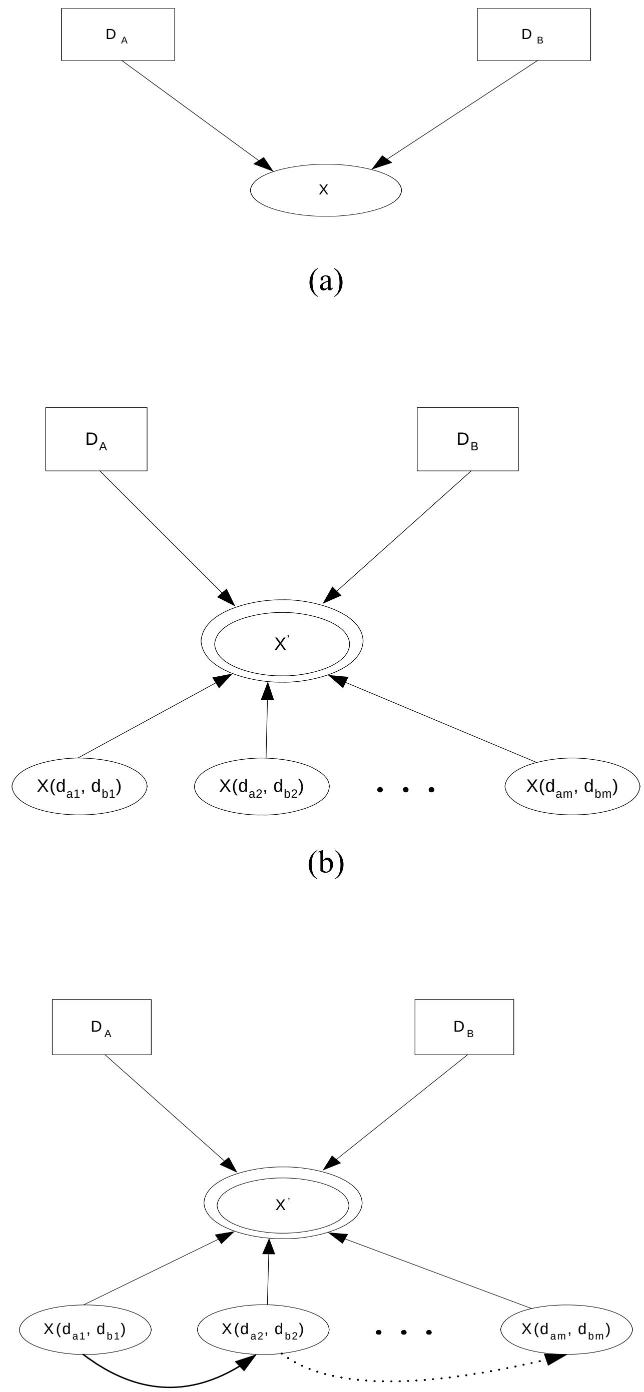


Figure 5.9 Convert MAID to canonical form; (a) is the original MAID, (b) is the canonical form assuming independence and (c) is the canonical form under general dependence relationship.

When changing MAIDs to canonical form, if the decisions D_A and D_B have k and l instances respectively, the chance variable X has n states, then $X(D_{ai},D_{bj})$ has n^{kl} instances, while the deterministic variable X' still has n states. In the original MAID the random variable X has only $k \cdot l \cdot n$ instances, so it is necessary to assign more probabilities for the equivalent conversion. The independence assumption between variables $X(D_{ai},D_{bj})$ represented in figure5.9(b) simplifies this conversion. In figure5.9(b), we have $P((X_p,D_{ai},D_{bj}),(X_q,D_{ar},D_{bt}))=P(X_p,D_{ai},D_{bj}) \cdot P(X_q,D_{ar},D_{bt})$, and the number of probabilities that need assigning is reduced from n^{kl} to $k \cdot l \cdot n$. Other probabilities can be derived from these $k \cdot l \cdot n$ outcomes.

The dependencies among the mapping variables may lead to more complex computation of VOI. However, when there is no specified information about the dependencies, we can assume independence to simplify our computation.

6 Qualitative Analysis of VOI in Multi-agent Systems

Under many complex situations, computing the exact value of information is very time-consuming and resource-consuming. Sometimes some quick justification about value of information is needed than the one that is more accurate but time-consuming. In this chapter, we will analyze VOI in multi-agent systems by reviewing some results obtained in single agent systems and present extensions in multi-agent systems. Our study is based on the multi-agent models in canonical form, which is already introduced in the previous chapter.

6.1 Introduction

As mentioned earlier, value of information has already been discussed quantitatively and qualitatively in single agent systems based on the graphical models. Most of the work on value of information in games is based on the extensive game tree and the discussion topics are also limited, for example to the influence of utility functions on information value. No research has been done on value of information based on the graphical model in multi-agent systems. Many important properties of information value in multi-agent systems have been omitted so far.

The complexity of exact computation of EVPI in an arbitrary single agent decision model with arbitrary utility function is known to be intractable. Even with the simplified assumptions that a decision maker is risk neutral or has a constant degree of risk aversion, the problem still remains intractable. Things become more complex in multi-agent systems.

Such situations motivated researchers to discuss value of information in a qualitative way by computing the information value indirectly. An earlier work (Poh & Horvitz, 1996) discussed the qualitative relationship about the information relevance of chance variables in graphical models based on a consideration of the topology of the models, most of which are based on the notion of d-separation.

A similar concept has also existed in multi-agent systems, which is called s-reachability. Without s-reachability, one decision cannot be relevant to another. Unlike d-separation in *BN*s, s-reachability is not necessarily a symmetric relation.

As we have defined in chapter 5, the information value in multi-agent systems has two types, one is nature information coming from the chance node, another is moving information from the decision node. We will discuss them respectively.

6.2 Value of Nature Information in Multi-agent Decision Systems

Theorem 6.1 Given a general MAID *M* with a set of agents *A*, *C* is the set of chance nodes, D_a is the set of decision nodes of agent a and U_a is the set of value nodes of each agent a, where $a \in A$. Let $D \in D_a$ be a decision node of agent a, $N \in C$ be an uncertain chance variable. If $N \perp U_a | D$, then $EVPI_a(D|N) = 0$ and $EVPAI_{b \in \{A/a\}}(D|N) = 0$.

Proof: We use *M* to denote the model *N* is not observed by *D*, and use $M_{D|N}$ to denote the model after *N* is observed by *D*.

According to the definition of Nash equilibrium that no agent has the incentive to deviate from its decision rule specified by the strategy profile, given no other agents deviate.

Firstly, we are going to verify that the optimal decision rule of *D* will not change if the decision rules of other decisions do not change.

a) In the general MAID model *M*:

Let δ_D be a decision rule for the decision variable $D \in D_a$ and σ be a strategy profile for *M*. Then δ_D is optimal for σ if and only if the probability distribution satisfies the following maximization problem:

$$\delta_D{}^* = \arg\max_{\delta_D{}^*} \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma_{-D}, \delta_{D^*})]}(U = u) \cdot u ,$$

The events ($U = u, D = d, Pa(D) = \mathrm{pa}_D$) for $d \in dom(D)$, $pa_D \in dom(Pa(D))$ form a partition of event $U = u$, so we can express $P_{M[(\sigma_{-D}, \delta_{D^*})]}(U = u)$ as the sum of the probabilities of these more specific events. By breaking up the joint probability we get:

$$\begin{aligned} & P_{M[(\sigma_{-D}, \delta_{D^*})]}(U = u) \\ & = \sum_{\mathrm{pa}_D \in dom(Pa(D))} P_{M[(\sigma_{-D}, \delta_{D^*})]}(\mathrm{pa}_D) \\ & \quad \times \sum_{d \in dom(D)} P_{M[(\sigma_{-D}, \delta_{D^*})]}(d \mid \mathrm{pa}_D) P_{M[(\sigma_{-D}, \delta_{D^*})]}(U = u \mid d, \mathrm{pa}_D) \end{aligned}$$

The CPD for a node in a Bayesian network has no prior distribution over its parents, so $P_{M[(\sigma_{-D}, \delta_{D^*})]}(\mathrm{pa}_D) = P_{M[(\sigma)]}(\mathrm{pa}_D)$ and $P_{M[(\sigma_{-D}, \delta_{D^*})]}(d \mid \mathrm{pa}_D) = \delta_D{}^*(d \mid \mathrm{pa}_D)$. $P_{M[(\sigma_{-D}, \delta_{D^*})]}(U = u \mid d, \mathrm{pa}_D) = P_{M[(\sigma)]}(U = u \mid d, \mathrm{pa}_D)$, because a distribution does not depend on the CPD for *D* given values of *D* and its parents. Therefore,

$$\begin{aligned} & P_{M[(\sigma_{-D}, \delta_{D^*})]}(U = u) \\ & = \sum_{\mathrm{pa}_D \in dom(Pa(D))} P_{M[(\sigma)]}(\mathrm{pa}_D) \\ & \quad \times \sum_{d \in dom(D)} \delta_D{}^*(d \mid \mathrm{pa}_D) P_{M[(\sigma)]}(U = u \mid d, \mathrm{pa}_D) \end{aligned}$$

Thus, δ_D is optimal for σ if and only if it satisfies the following maximization problem:

$$\delta_D^* = \arg\max_{\delta_D^*} \sum_{\mathrm{pa}_D \in dom(Pa(D))} P_{M[(\sigma)]}(\mathrm{pa}_D) \sum_{d \in dom(D)} \delta_D^*(d \,|\, \mathrm{pa}_D)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U = u \mid d, \mathrm{pa}_D) \cdot u$$

Therefore,

$$EU_a(M) = \max_{\delta_D^*} \sum_{\mathrm{pa}_D \in dom(Pa(D))} P_{M[(\sigma)]}(\mathrm{pa}_D) \sum_{d \in dom(D)} \delta_D^*(d \,|\, \mathrm{pa}_D)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U = u \mid d, \mathrm{pa}_D) \cdot u$$

b) If *N* is observed before *D*, in model $M_{D|N}$

We are going to verify that $\delta_D^*(d \,|\, \mathrm{pa}_D)$ will not change if the decision rules of other decision nodes do not change.

Assume σ' is a strategy profile for $M_{(D|N)}$ that only differs from σ only at *D* and a decision rule δ_D'. Let Pa_D' denote the new parents of *D* in model $M_{D|N}$.

$Pa_D' = \{Pa_D, N\}$, then δ_D' is optimal for σ' if and only if the probability distribution satisfies the following maximization problem:

$$\delta_D'^* = \arg\max_{\delta_D'^*} \sum_{\mathrm{pa}_D \in dom(Pa(D))} P_{M[(\sigma')]}(\mathrm{pa}_D, n) \sum_{d \in dom(D)} \delta_D'^*(d \,|\, \mathrm{pa}_D, n)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma')]}(U = u \mid d, \mathrm{pa}_D, n) \cdot u$$

Therefore,

$$EU_a(M_{D|N}) = \max_{\delta_D'^*} \sum_{n \in dom(N)} \sum_{\mathrm{pa}_D \in dom(Pa(D))} P_{M[(\sigma')]}(\mathrm{pa}_D, n) \sum_{d \in dom(D)} \delta_D'^*(d \,|\, \mathrm{pa}_D, n)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma')]}(U = u \mid d, \mathrm{pa}_D, n) \cdot u$$

Since $N \perp U_a \,|\, D_a$, thus,

$$P_{M[(\sigma')]}(U=u \mid d, \mathrm{pa}_D, n) \cdot u = P_{M[(\sigma')]}(U=u \mid d, \mathrm{pa}_D) \cdot u$$

So we get:

$$EU_a(M_{D|N}) = \max_{\delta_D{}'^*} \sum_{n \in dom(N)} \sum_{\mathrm{pa}_D \in dom(Pa(D))} P_{M[(\sigma')]}(\mathrm{pa}_D, n) \sum_{d \in dom(D)} \delta_D{}'^*(d \,|\, \mathrm{pa}_D, n)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U=u \mid d, \mathrm{pa}_D) \cdot u$$

Because $\sigma' = \sigma_{-D,\delta_D{}'^*}$, by rearranging the summations, we get:

$$EU_a(M_{D|N})$$
$$= \max_{\delta_D{}'^*} \sum_{\mathrm{pa}_D \in dom(Pa(D))} \sum_{d \in dom(D)} \sum_{n \in dom(N)} P_{M[(\sigma_{-D,\delta_D{}'^*})]}(d \mid \mathrm{pa}_D, n) P_{M[(\sigma_{-D,\delta_D{}'^*})]}(\mathrm{pa}_D, n)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma_{-D,\delta_D{}'^*})]}(U=u \mid d, \mathrm{pa}_D) \cdot u$$

Using the chain rule,

$$EU_a(M_{D|N}) = \max_{\delta_D{}'^*} \sum_{\mathrm{pa}_D \in dom(Pa(D))} \sum_{d \in dom(D)} \sum_{n \in dom(N)} P_{M[(\sigma_{-D,\delta_D{}'^*})]}(d, \mathrm{pa}_D, n)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma_{-D,\delta_D{}'^*})]}(U=u \mid d, \mathrm{pa}_D) \cdot u$$

Marginalizing *N*, we get:

$$EU_a(M_{D|N}) = \max_{\delta_D{}'^*} \sum_{\mathrm{pa}_D \in dom(Pa(D))} \sum_{d \in dom(D)} P_{M[(\sigma_{-D,\delta_D{}'^*})]}(d, \mathrm{pa}_D)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma_{-D,\delta_D{}'^*})]}(U=u \mid d, \mathrm{pa}_D) \cdot u$$
$$= \max_{\delta_D{}'^*} \sum_{\mathrm{pa}_D \in dom(Pa(D))} \sum_{d \in dom(D)} P_{M[(\sigma_{-D,\delta_D{}'^*})]}(\mathrm{pa}_D) P_{M[(\sigma_{-D,\delta_D{}'^*})]}(d \mid \mathrm{pa}_D)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma_{-D,\delta_D{}'^*})]}(U=u \mid d, \mathrm{pa}_D) \cdot u$$

Since the CPD for a node in BN has no effect on the prior distribution over its parents, so $P_{M[(\sigma_{-D,\delta_D{}'^*})]}(\mathrm{pa}_D) = P_{M[(\sigma')]}(\mathrm{pa}_D) = P_{M[(\sigma)]}(\mathrm{pa}_D)$. And

$$P_{M[(\sigma_{-D,\delta_D{}'^*})]}(U=u \mid d, \mathrm{pa}_D) \cdot u = P_{M[(\sigma')]}(U=u \mid d, \mathrm{pa}_D) \cdot u = P_{M[(\sigma)]}(U=u \mid d, \mathrm{pa}_D) \cdot u$$

because a distribution does not depend on the CPD for D given values for *D* and its parents.

We get:

$$EU_a(M_{D|N}) = \max_{\delta_D{}'^*} \sum_{\mathrm{pa}_D \in dom(Pa(D))} P_{M[(\sigma)]}(\mathrm{pa}_D) \sum_{d \in dom(D)} P_{M[(\sigma_{-D}, \delta_D{}'^*)]}(d \mid \mathrm{pa}_D)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U = u \mid d, \mathrm{pa}_D) \cdot u$$

We use $P'^*(d \mid \mathrm{pa}_D)$ to denote $P_{M[(\sigma_{-D}, \delta_D{}'^*)]}(d \mid \mathrm{pa}_D)$

$$EU_a(M_{D|N}) = \max_{P'^*} \sum_{\mathrm{pa}_D \in dom(Pa(D))} P_{M[(\sigma)]}(\mathrm{pa}_D) \sum_{d \in dom(D)} P'^*(d \mid \mathrm{pa}_D)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U = u \mid d, \mathrm{pa}_D) \cdot u$$

Since

$$EU_a(M) = \max_{\delta_D{}^*} \sum_{\mathrm{pa}_D \in dom(Pa(D))} P_{M[(\sigma)]}(\mathrm{pa}_D) \sum_{d \in dom(D)} P_{M[(\sigma_{-D}, \delta_D{}^*)]}(d \mid \mathrm{pa}_D)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U = u \mid d, \mathrm{pa}_D) \cdot u$$

We use $P^*(d \mid \mathrm{pa}_D)$ to denote $P_{M[(\sigma_{-D}, \delta_D{}^*)]}(d \mid \mathrm{pa}_D)$,

$$EU_a(M) = \max_{P^*} \sum_{\mathrm{pa}_D \in dom(Pa(D))} P_{M[(\sigma)]}(\mathrm{pa}_D) \sum_{d \in dom(D)} P^*(d \mid \mathrm{pa}_D)$$
$$\times \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U = u \mid d, \mathrm{pa}_D) \cdot u$$

Compare the expression $EU_a(M)$ with $EU_a(M_{D|N})$, we can find that for any $d \in dom(D), U \in U_D, u \in dom(U)$, the two maximization problems are the same, i.e. any $P'^*(d \mid \mathrm{pa}_D)$ will be a solution of $EU_a(M)$ and any $P^*(d \mid \mathrm{pa}_D)$ is also a solution of $EU_a(M_{D|N})$. Therefore, $EU_a(M) = EU_a(M_{D|N})$.

That is to say, agent a will not deviate from his/her action at *D* given the value of the original parents of *D*, if other agents do not deviate from the strategy specified by σ_{-D}.

From the discussion above, we can also see that

$$P'^{*}(d\,|\,\mathrm{pa}_D) = \arg\max_{P'^{*}} \sum_{d \in dom(D)} P'^{*}(d\,|\,\mathrm{pa}_D) \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U = u \mid d, \mathrm{pa}_D) \cdot u$$

and

$$P''^{*}(d\,|\,\mathrm{pa}_D{}')$$
$$= \arg\max_{P''^{*}} \sum_{d \in dom(D)} P''^{*}(d\,|\,\mathrm{pa}_D{}') \sum_{U \in U_a} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U = u \mid d, \mathrm{pa}_D) \cdot u$$

Therefore, we can get

$$P''^{*}(d\,|\,\mathrm{pa}_D{}') = P'^{*}(d\,|\,\mathrm{pa}_D) = P^{*}(d\,|\,\mathrm{pa}_D)\,,$$

which is

$$P''^{*}(d\,|\,\mathrm{pa}_D, n) = P^{*}(d\,|\,\mathrm{pa}_D)$$

This tells us that given the value of *D*'s original parents, if $N \perp U_a \,|\, D$, then the value of *N* will not change the optimal probability distribution of *D*. While the CPD table of *D* becomes large, since *D* has a new parent *N* in the new model $M_{D|N}$. If the decision node *D* has m parents in model *M,* each with t values, and *D* has n choices associated with it, *N* has p values. The size of the original CPD table of *D* is $n \times t^m$, but is $n \times t^m \times p$ after *N* is observed by *D*.

This also implies that if any of $Pa(D)$ is independent of U_a, then the CPD table of it will not have influence on the optimal choice of *D*.

Secondly, we are going to verify that any agent will not deviate from his/her strategy at any of their decision node *D'* which is different from *D*, given other agents do not deviate from their strategies at other decision nodes.

a) In a general MAID model *M*:

Let δ be a decision rule for any decision variable *D'* in model *M* except *D*, and σ be a strategy profile for *M*. Assume *D'* belongs to any agent $i, i \in I$. δ is optimal for σ if and only if the probability distribution satisfies the following maximization problem:

$$\delta^* = \arg\max_{\delta^*} \sum_{\mathrm{pa}_{D'} \in dom(Pa(D'))} P_{M[(\sigma)]}(\mathrm{pa}_{D'}) \sum_{d' \in dom(D')} \delta^*(d' | \mathrm{pa}_{D'})$$
$$\times \sum_{U \in U_i} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U = u \mid d', \mathrm{pa}_{D'}) \cdot u$$

If $\sum_{\mathrm{pa}_{D'} \in dom(Pa(D'))} P_{M[(\sigma)]}(\mathrm{pa}_{D'})$ =0, the maximization problem is trivial, since all distributions yield a value of zero. Thus, it is necessary and sufficient that for all $\mathrm{pa}_{D'}$ such that $\sum_{\mathrm{pa}_{D'} \in dom(Pa(D'))} P_{M[(\sigma)]}(\mathrm{pa}_{D'})$ >0, the distribution $\delta(D' | \mathrm{pa}_{D'})$ be a solution of the following maximization problem:

$$\delta^* = \arg\max_{\delta^*} \sum_{d' \in dom(D')} \delta^*(d' | \mathrm{pa}_{D'}) \times \sum_{U \in U_i} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U = u \mid d', \mathrm{pa}_{D'}) \cdot u \quad (6.1)$$

$$EU_i(M) = \max_{\delta^*} \sum_{d' \in dom(D')} \delta^*(d' | \mathrm{pa}_{D'}) \times \sum_{U \in U_i} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U = u \mid d', \mathrm{pa}_{D'}) \cdot u$$

b) If *N* is observed before *D*, in model $M_{D|N}$

Let δ' be a decision rule for the same decision variable *D'* in model $M_{D|N}$ as in *M*, and σ' be a strategy profile for $M_{D|N}$ which differs from σ only at *D'*. Then δ' is optimal for σ' if and only if the probability distribution satisfies the following maximization problem:

$$\delta'^* = \arg\max_{\delta^*} \sum_{d' \in dom(D')} \delta'^*(d' | \mathrm{pa}_{D'}) \times \sum_{U \in U_i} \sum_{u \in dom(U)} P_{M[(\sigma')]}(U = u \mid d', \mathrm{pa}_{D'}) \cdot u$$

Since σ' is the strategy profile which differs from σ only at *D'* and a distribution does not depend on the CPD for *D'* given values for *D'* and its parents.

Therefore, $P_{M[(\sigma')]}(U=u \mid d', \mathrm{pa}_{D'}) \cdot u = P_{M[(\sigma)]}(U=u \mid d', \mathrm{pa}_{D'}) \cdot u$

By substituting, we get

$$\delta'^{*} = \arg\max_{\delta'^{*}} \sum_{d' \in dom(D')} \delta'^{*}(d' \big| \mathrm{pa}_{D'}) \times \sum_{U \in U_i} \sum_{u \in dom(U)} P_{M[(\sigma)]}(U=u \mid d', \mathrm{pa}_{D'}) \cdot u \quad (6.2)$$

The two maximization problems of (1) and (2) are the same, we get $\delta'^{*} = \delta^{*}$.

That is to say, no matter *N* is observed before *D* or not, the optimal decision rule of any decision node except *D* will not change, if N is independent of U_a given *D*. Thus, the expected utilities of other agents will not change after *N* is observed by *D*.

Therefore, $EVPI_a(D \big| N) = 0$ and $EVPAI_{b \in \{A/a\}}(D \big| N) = 0$. □

Theorem 6.2: Given a general MAID *M* in canonical form with a set of agents *A*, *C* is the set of chance nodes, D_i is the set of decision nodes of agent a and U_i is the set of value nodes of each agent a, where $a \in A$. Let $D \in D_a$ be a decision node of agent a, $N \in C$ be an uncertain chance variable. In both decision models *M* and $M_{D|N}$, if for any decision node *D'* of agent b, *D* is not s-reachable from *D'*, and *D'* is not a descendant of *D*. If for $D' \subset \{\bigcup_{i \in \{A \setminus a\}} D_i\}$, then $EVPAI_{i \in \{A/a\}}(D \big| N) = 0$, where $b \in A$, $D' \subset \{\bigcup_{a \in A} D_a \setminus D\}$.

Proof: Let δ be a decision rule for a decision variable *D* in the MAID *M*, δ' be a decision rule for *D'*, $\delta_{D|N}$ be a decision rule for the decision variable *D* in the MAID $M_{D|N}$ and $\delta'_{D|N}$ be a decision rule for the decision variable *D'* in the MAID $M_{D|N}$. σ be a strategy profile for *M* and σ' be a strategy profile for $M_{D|N}$.

Since any *D'* is not s-reachable from *D*, then changing σ at *D* will not affect the optimality of δ' (According to algorithm of solving MAID). Therefore, except δ, any optimal decision rule of any decision variable will not change, i.e. $\delta_{D|N}{}'^{*} = \delta'^{*}$.

The distribution of δ' should be a solution of the following maximization problem:

$$\delta'^{*} = \arg\max_{\delta'^{*}} \sum_{d'\in dom(D')} \delta'^{*}(d'|\mathrm{pa}_{D'}) \times \sum_{U\in U_i} \sum_{u\in dom(U)} P_{M[(\sigma)]}(U=u \mid d', \mathrm{pa}_{D'})\cdot u$$

After *N* is observed by *D*, the distribution of $\delta'_{D|N}{}^{*}$ should be a solution of the following maximization problem:

$$\delta'_{D|N}{}^{*} = \arg\max_{\delta'_{D|N}{}^{*}} \sum_{d'\in dom(D')} \delta'_{D|N}{}^{*}(d'|\mathrm{pa}_{D'}) \times \sum_{U\in U_i} \sum_{u\in dom(U)} P_{M[(\sigma')]}(U=u \mid d', \mathrm{pa}_{D'})\cdot u$$

Since $\delta_{D|N}{}'^{*} = \delta'^{*}$, the two maximization problems should be the same. And the induced *M*, $M_{D|N}$ are two identical BNs which only differs at the CPD table they assign to *D*, then

$$P_{M[(\sigma)]}(U=u \mid d', \mathrm{pa}_{D'})\cdot u = P_{M[(\sigma')]}(U=u \mid d', \mathrm{pa}_{D'})\cdot u$$

$$\delta'^{*} = \arg\max_{\delta'^{*}} \sum_{\mathrm{pa}_{D'}\in dom(Pa(D'))} P_{M[(\sigma')]}(\mathrm{pa}_{D'}) \sum_{d\in dom(D')} \delta'^{*}(d'|\mathrm{pa}_{D'}) \times \sum_{U\in U_b} \sum_{u\in dom(U)} P_{M[(\sigma')]}(U=u \mid d', \mathrm{pa}_{D'})\cdot u$$

$$EU_b(M_{D|N}) = \max_{\delta'^{*}} \sum_{\mathrm{pa}_{D'}\in dom(Pa(D'))} P_{M[(\sigma')]}(\mathrm{pa}_{D'}) \sum_{d\in dom(D')} \delta'^{*}(d|\mathrm{pa}_{D'}) \times \sum_{U\in U_b} \sum_{u\in dom(U)} P_{M[(\sigma')]}(U=u \mid d', \mathrm{pa}_{D'})\cdot u$$

For those $D' \subset \{\bigcup_{i\in\{A\backslash a\}} D_a\}$, *D'* is not a descendant of *D*, then

$EU_b(M_{D|N})$ is equal to:

$$EU_b(M) = \max_{\delta^*} \sum_{\mathrm{pa}_{D'} \in dom(Pa(D'))} P_{M[(\sigma)]}(\mathrm{pa}_{D'}) \sum_{d \in dom(D')} \delta^*(d \,|\, \mathrm{pa}_{D'})$$
$$\times \sum_{U \in U_b} \sum_{u \in dom(U)} P_{M[(\sigma')]}(U = u \mid d', \mathrm{pa}_{D'}) \cdot u$$

Therefore, $EVPAI_{i \in \{I/a\}}(D \,|\, N) = 0$. □

6.3 Value of Moving Information in Multi-agent Decision Systems

Theorem 6.3: Given a general MAID *M* in canonical form with a set of agents *A*, *C* is the set of chance nodes, D_a is the set of decision nodes of agent a and U_a is the set of value nodes of each agent a, where $a \in A$. Let $D \in D_a$ be a decision node of agent a, $N \in C$ be an uncertain chance variable. If for any decision node *D'* of agent b, *D* is not s-reachable from *D'*, where $b \in A, D' \subset \{\bigcup_{a \in A} D_a \setminus D\}$, then the moving information of *D* to *D'* is equal to 0, i.e. $EVPI_{i \in A}(D' \mid D)$=0.

Proof: Since *D* is not s-reachable from *D'*, therefore, *D'* does not rely on *D*. Changing the decision rule of *D* will not affect the optimal decision rule of *D'*.

Let δ be a decision rule for a decision variable *D* in the MAID *M*, δ' be a decision rule for *D'*, σ be a strategy profile for *M* and σ' be a strategy profile for $M_{D|D'}$.

In order to verify that the optimal σ^* in *M* will still remain to be the Nash equilibrium in model $M_{D|N}$, we verify below that no agent will deviate from its decision rule given other agents do not deviate.

When *D* is observed before *D'*, the value of *D*=*d* is revealed to *D'*. To *D'*, it is like agent *a* takes *D*=*d* with probability 1. Since we know no matter how the decision rule *D* takes, it will not change the decision rule of *D'*. Therefore, by observing *D*, the optimal decision rule of *D'* will not be changed. Therefore, δ' will remain to be the same, i.e, $\delta'(D' \mid Pa_{D'}) = \delta'(D' \mid Pa_{D'}, D)$.

Since decision node *D* is not s-reachable from any *D'*, we can know that the optimal decision rule of any decision node *D'* will not change.

If *D'* is not s-reachable from *D*, changing the decision rule of *D'* will not change the optimal decision rule of *D*; the optimal decision rule δ of *D* will remain the same. If *D'* is s-reachable from *D*, *D* relies on *D'*, since δ' will remain the same, the optimal decision rule of *D* will not change.

Therefore, the optimal decision rule of every decision node will not change. The Nash equilibrium remains to be the same. As the CPD table in *M* and $M_{D'|D}$ are all the same, the utility of each agent will not change. Therefore, the moving information of *D* to *D'* is equal to 0, i.e. $EVPI_{i \in A}(D' \mid D)$=0. □

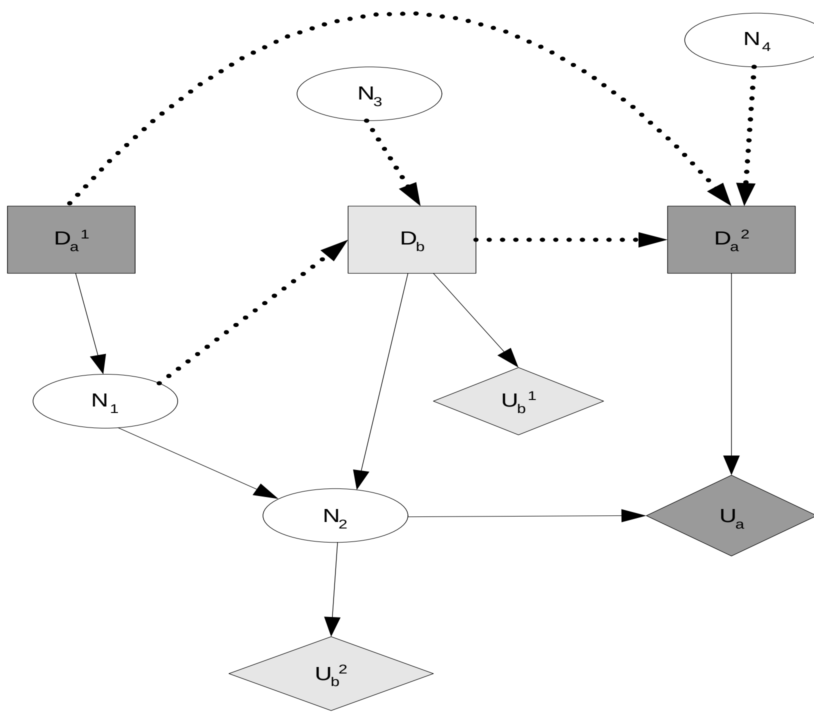
Theorem 6.4: Given a general MAID *M* with a set of agents *A*, *C* is the set of chance nodes, D_a is the set of decision nodes of agent a and U_a is the set of value nodes of each agent a, where $a \in A$. Let $D \in D_a$ be a decision node of agent a, $N \in C$ be an uncertain chance variable. If for any decision node *D'* of agent i, $i \in \{A \setminus a\}$, then the moving information of *D* to *D'* is also larger or equal to 0, i.e. $EVPI_{i \in \{A/a\}}(D' \mid D) \geq 0$.

Proof: If *D* is not s-reachable from *D'*, from Theorem 6.3, $EVPI_{i \in \{A/a\}}(D' \mid D)$=0.

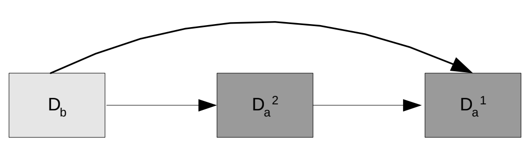
If *D* is s-reachable from *D'*, by observing *D*, the decision rule of *D'* might change, in the condition that $EU_i(\sigma') \succ EU_i(\sigma)$ where σ is a strategy profile for *M* and σ' is a strategy profile for $M_{D|D'}$. Otherwise, given the decision rule of other decision nodes does not change, the agent has no incentive to change its optimal decision rule of *D'* and *D* will therefore remain to be the same. In other words, the Nash equilibrium will change only when the expected utility of agent *i* increases by changing its decision rule. Otherwise, he/she can take his/her original decision rule. Therefore, $EVPI_{i \in \{A/a\}}(D' \mid D) \geq 0$. □

6.4 Examples

Let us consider an example of the application of these properties. Consider the decision model shown in Figure 6.1; Using Theorem 6.1, since $N_4 \perp U_a \mid D_a^{\ 2}$, we can know that $EVPI_a(D_a^{\ 2} \mid N_4) = 0$ and $EVPAI_b(D_a^{\ 2} \mid N_4) = 0$. This means that after N_4 is observed by agent a before making decision $D_a^{\ 2}$, the expected utility of both agents a and b will not change.



(a) MAID *M* of the simple example

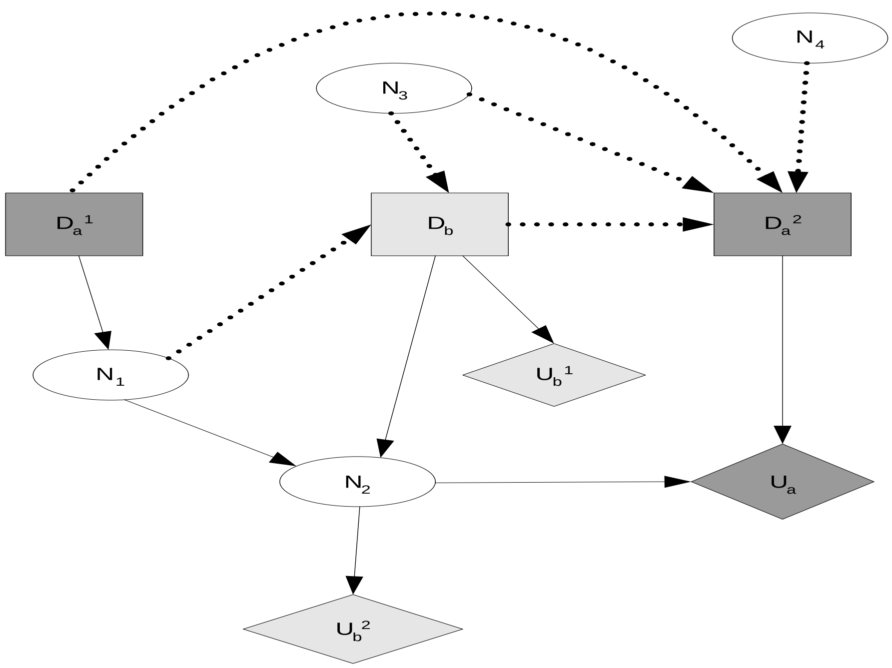


(b) Relevance graph of the MAID *M* shown in (a)

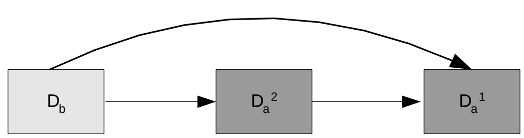
Figure 6.1 An example of VOI properties

Similarly, since $N_3 \perp \{U_b{}^1, U_b{}^2\} \mid D_b$, using Theorem 6.1, we can infer $EVPI_b(D_b \mid N_3) = 0$ and $EVPAI_a(D_b \mid N_3) = 0$.

Figure 6.2 shows the new MAID model $M_{D_a{}^2|N_3}$ after N_3 is observed by $D_a{}^2$ and its corresponding relevance graph. Since $D_a{}^2$ is not s-reachable from D_b in both model M and new model $M_{D_a{}^2|N_3}$, and D_b is not a descendant of $D_a{}^2$. Hence, using Theorem 6.2, we can infer that $EVPAI_b(D_a{}^2 \mid N_3) = 0$.



(a) MAID $M_{D_a{}^2|N_3}$



(b) Relevance graph of the MAID $M_{D_a{}^2|N_3}$

Figure 6.2 New model $M_{D_a{}^2|N_3}$, after N_3 is observed by $D_a{}^2$

Since both D_a^1 and D_a^2 are not s-reachable from D_b, the moving information from D_a^1, D_a^2 to D_b will equal to 0, i.e. $EVPI_a(D_b \mid D_a^1) = 0$, $EVPI_b(D_b \mid D_a^1) = 0$, $EVPI_a(D_b \mid D_a^2) = 0$ and $EVPI_b(D_b \mid D_a^2) = 0$, according to Theorem 6.3.

The moving information from decision node D_b to decision nodes D_a^1 and D_a^2 will always be larger or equal to 0, i.e. $EVPI_a(D_a^2 \mid D_b) \geq 0$ and $EVPI_a(D_a^1 \mid D_b) \geq 0$, according to Theorem 6.4.

7 Conclusion and Future Work

In this chapter, a summary of the merits and the limitations of the work conducted is presented and possible extensions are suggested to conclude this thesis.

7.1 Conclusion

In this study, we developed a graphical model for representing and solving asymmetric multi-agent decision problems in an uncertain environment and extended the computation of information value to multi-agent decision systems.

Firstly, we developed a new framework to represent asymmetric multi-agent decision problems, called Asymmetric Multi-agent Influence Diagrams (AMAIDs). Four main types of asymmetric problems are discussed by giving out illustrative examples. Our model is compared with the existing models in representing them. An AMAID is a probabilistic model that is composed of two parts; One part encodes the information precedence as well as asymmetric structure and the other encodes the probabilistic dependence relations for the chance nodes and deterministic functional relations for the utility node. The asymmetric structures and information precedence are represented by contextual arcs. An AMAID does not only represent the asymmetric problems more concisely, but also solves some problems such as order asymmetry which cannot be represented by normal MAID.

Secondly, we proposed the evaluation algorithm for solving the decision model of AMAIDs. This evaluation algorithm is extended from the evaluation algorithm for solving MAIDs. S-reachability is identified by converting an AMAID to a de-contextualized AMAID.

Thirdly, value of information is discussed and defined in multi-agent decision systems. We incorporate MAIDs into our computation of VOI by illustrating three different information conditions in a basic MAID model: with all agents knowing the information, with part of the agents knowing the information and with no information. According to the number of agents knowing the information in the system, we divide the information in multi-agent decision systems into two types: symmetric information and asymmetric information. According to the information source, we divide the information into another two types: nature information and moving information. Three concepts: expected value of perfect information, expected value of perfect asymmetric information and expected value of perfect symmetric information are defined and compared in the multi-agent systems.

Numerical examples are given to illustrate the computation of VOI based on the decision model of MAIDs. The property that VOI can be less than 0 in multi-agent systems is shown by the example. We define another two concepts of escaping cost and revealing cost with respect to the sign of VOI.

Fourthly, value of information for the intervened variables in multi-agent decision systems is discussed. The algorithm developed by Heckerman and Shachter (1995) for constructing the canonical form of influence diagram is extended to construct the canonical form of multi-agent influence diagram.

Finally, we revealed and verified some properties of nature information value and moving information value in multi-agent decision systems based on the graphical model of MAIDs. These properties discussed reveal the qualitative relationship about the information relevance of chance variables and decision variables based on the topology of the models, mostly on the notion of conditional independence and s-reachability in MAID. These properties can be used to guide researchers and practitioners in some fast identification.

7.2 Future Work

This work proposes a new model to solve asymmetric decision problems and opens the discussion of information value in multi-agent decision systems. Therefore, the first possible extension is to see whether AMAID model can also be used for computing VOI in asymmetric decision problems.

Secondly, some interesting properties of VOI in multi-agent decision systems still remain unrevealed. Research still needs to be conducted on VOI in multi-agent decision systems.

Thirdly, The VOI computation in multi-agent decision systems mainly depends on the computation method of Nash equilibrium of the decision model. Therefore, finding new efficient inference methods will facilitate the VOI computation.

Fourthly, the properties of VOI discussed in this work can be applied to solve various real-world problems and analyze the actions of the agents.

Last but not least, information problems involving trust, imperfect information and signaling are interesting problems that remain to be discussed, which can also be analyzed based on the graphical models.

Reference

Becker, A, Geiger, D. (1996), A sufficiently fast algorithm for finding close to optimal junction trees, In *Proceedings of the Twelfth Uncertainty in Artificial Intelligence*, Morgan Kaufmann.

Bellman, R. (1957), *Dynamic Programming*. Princeton University Press, Princeton, NJ.

Boutilier, C., Friedman, N., Goldszmidt, M. and Koller, D. (1996), Context-specific independence in Bayesian networks. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pp. 115–123.

Blum, B., C. R. Shelton and Koller, D. (2003), A Continuation Method for Nash Equilibria in Structured Games, In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pp. 757-765.

Castillo, E., Jose Manuel Gutierrez and Ali S. Hadi (1997), *Expert Systems and Probabilistic Network Models*, Springer.

Castelo, R. and Kocka, T. (2003), On Inclusion-Driven Learning of Bayesian.

Clemen, R., T. (1996), Making Hard Decisions. *Second edition*. Duxbury.

Cooper, G. F. (1987), *Probabilistic Inference Using Belief Networks is NP-hard*, Knowledge Systems Laboratory, Stanford University, Memo KSL-87-27.

Cooper, G. F. (1988), A Method for Using Belief Networks as Influence Diagrams, In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pp. 211-219.

Cooper, G. F. (1990), The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks, *Artificial Intelligence* **42**, pp. 393-405.

Cooper, G. F. and Herskovits, E. (1992), A Bayesian Method for the Induction of Probabilistic Networks from Data, *Machine Learning* **9**, pp. 309-347.

Cover, T. M., Thomas, J. A. (1991), *Elements of Information Theory*, John Wiley & Sons.

Dagum, P. and Luby, M. (1993), Approximating Probabilistic Inference in Belief Networks in NP-hard, *Artificial Intelligence 60(1),* pp.41-48.

Dean, T. and Kanazawa, K. (1989), A Model for Reasoning about Persistence and Causation, *Computational Intelligence* **5**, pp. 142-150.

Dechter, R. (1996), Bucket Elimination: A Unifying Framework for Probabilistic Inference, In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pp. 211-219.

Dittmer, S., L. and Jensen, F., V. (1997), Myopic Value of Information in Influence Diagrams. In D. Geiger and P. P. Shenoy (editors), *Proceedings* *of the Thirteenth Conference on Uncertainty in Artificial Intelligence,* pp.142-149. Morgan Kaufmann.

Edwards, W. (1998), Hailfinder: Tools for and Experiences with Bayesian Normative Modeling, *American Psychologist 53*, pp. 416-428.

Ezawa, K., J. (1994), Value of Evidence on Influence Diagrams. In R. L. de Mantaras and D. Poole (editors), *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 212-220, Morgan Kaufmann.

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

Felli, J. C. and Hazen, G. B. (1998), Sensitivity Analysis and the Expected Value of Perfect Information, *Medical Decision Making*, Vol.18, No.1, pp.95-109

Geiger, D., Verma, T. and Pearl, J. (1989), D-separation: From Theorems to Algorithms, In *Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence*, pp. 118-125.

Heckerman, D., Meek, C. and Koller, D. (2004), Probabilistic Models for Relational Data, *Technical Report*, Microsoft Research.

Heckerman, D., Horvitz E., and Middleton B. (1991), An Approximate Non-myopic Computation for Value of Information, In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 135-141, Morgan Kauffman.

Heckerman, D. E., Horvitz, E. J. and Nathwani, B. N. (1992), Toward Normative Expert Systems: Part I The Pathfinder Project. *Methods of Information in Medicine*, 31, pp. 90-105.

Heckerman, D., and Shachter, R. (1995), Decision-Theoretic Foundations for Causal Reasoning. *Journal of Artificial Intelligence Research*, 3, pp. 405-430.

Henrion, M. (1988), Propagating Uncertainty in Bayesian Networks by Probabilistic Logic Sampling, In *Uncertainty in Artificial Intelligence* ***2*** (Eds.: Lemmer, J.F. and L. N. Kanal), pp. 149-163.

Howard, R.A. (1960), Dynamic Programming and Markov Processes, the MIT press.

Howard, R.A. (1966), Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, SSC-2(1), pp. 22-26.

Howard, R.A. (1967), Value of information lotteries. *IEEE Transactions on Systems Science and Cybernetics*, SSC-3(1), pp. 54-60.

Howard, R. A. and Matheson, J. E. (1984), Influence Diagrams. In *Readings on the Principles and Applications of Decision Analysis* ***2*** (Eds.: Howard, R. A. and J. E. Matheson), pp. 719-726.

Jensen, F. V., Lauritzen, K. G. and Olesen, K. G. (1990), Bayesian Updating in Causal Probabilistic Networks by Local Computations, *Computational Statistical Quarter 4*, pp. 269-282.

Jensen, F., Jensen, F. V, and Dittmer, S. L (1994), From Influence Diagrams to Junction Trees, In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 367-374.

Jensen, F. V. (1996), *An Introduction to Bayesian Networks*, Springer, New York.

Jensen, F. V. (2001), *Bayesian Networks and Decision Graphs*, Springer, New York.

Jensen, F. V. and Marta Vomlelova (2002), Unconstrained Influence Diagrams, In *Proceedings of the Eighteenth Conference of Uncertainty in Artificial Intelligence*, pp. 234-241.

Jensen, F. V., Nielsen, T. D. and P. P. Shenoy (2004), Sequential Influence Diagrams: A Unified Asymmetric Framework, In *Proceedings of the Second Workshop on Probabilistic Graphical Models*, pp. 121-128.

Kearns, M., M. L. Littman and Singh, S. (2001a), Graphical Models for Game Theory, In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pp. 253-260.

Kearns, M., M. L. Littman and Singh, S. (2001b), An Efficient Exact Algorithm for Singly Connected Graphical Games, In *Proceedings of the Fourteenth Conference on Neural Information Processing Systems*, pp. 817-823

Kjærulff, U. (1994), Reduction of Computation Complexity in Bayesian Networks Through Removal of Weak Dependencies, In *Proceedings of Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 374-382.

Kjærulff, U. (1997), A Computational Scheme for Reasoning in Dynamic Probabilistic Networks, In *Proceedings of Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 121-129.

Koller, D. and B. Milch (2001), Multi-Agent Influence Diagrams for Representing and Solving Games, In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, Washington, pp. 1027-1034.

La Mura, P. (2000), Game Networks, In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 335-342.

La Mura, P. and Shoham, Y. (1999), Expected Utility Networks, In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 366-373.

Lauritzen, S. L. and D. Nilsson (2001), Representing and Solving Decision Problems with Limited Information, *Management Science* 47, pp. 1238-1251.

Lauritzen, S.L., and D.J. Spiegelhalter (1988), Local computations with probabilities on graphical structures and their application to expert systems (with discussion), *Journal of the Royal Statistics Society*, B50, pp. 157-224.

Leong, T. Y. (1994), *An Integrated Approach to Dynamic Decision Making under Uncertainty*. PhD thesis, MIT, Cambridge, MA.

Madsen, A. L. and Jensen F. V. (1998), Lazy Propagation in Junction Trees, In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp. 211-219.

Mckelvey, R. and McLennan, A. (1996), Computation of Equilibria in Finite Games. *Handbook of Computational Economics*, pp. 87-142.

Nash, J. (1950), Equilibrium Points in N-Person Games, PNAS 36, pp. 48-49.

Neapolitan, R. E. (1990), Probabilistic Reasoning in Expert Systems, Wiley and Sons, New York.

Neapolitan, R. E. (2004), Learning Bayesian Networks, Prentice Hall.

Nielsen, T. and Jensen, F. V. (1999), Well-defined Decision Scenarios, In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp 502-511.

Olmsted, S. M. (1983), On Representing and Solving Decision Problems, PhD Thesis, Department of Engineering-Economic Systems, Stanford University.

Ortiz L. E., and Kaelbling L. P. (2000), Sampling Methods for Action Selection in Influence Diagrams. *AAAI* 2000.

Pearl, J. (1988), Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann.

Pearl, J., Geiger, D. and Verma, T. (1990), The logic of influence diagram. In Oliver, R.M. & Smith, J.Q. (editors), *Influence diagrams, belief nets and decision analysis*, pp 67-87, John Wiley.

Pearl, J. (2000), *Causality, Models, Reasoning and Inference*. Cambridge University Press.

Rege, A. and Agogino, A. M. (1988), Topological Framework for Representing and Solving Probabilistic Inference Problems in Expert Systems, *IEEE Transactions on Systems, Man and Cybernetics* 18 (3), pp. 402-414.

Robert, T. C. and Terry, R. (2001), Making Hard Decisions with Decision Tools, Duxbury/Thomson Learning.

Russell, S. and Norvig, P. (2003), Artificial Intelligence: A Modern Approach, Prentice Hall, Englewood Cliffs.

Ryan, P., Eugene, N. and Shoham, Y. (2004), Simple Search Methods for Finding a Nash Equilibrium, In Proceedings of American Association for Artificial Intelligence (AAAI), pp. 664-669.

Poh, K.L. and Horvitz, E. (1993), Reasoning about the Value of Decision Model Refinement: Methods and Application. In *Proceedings of Ninth Conference on Uncertainty in Artificial Intelligence*, pp.174-182, Morgan Kaufmann.

Poh, K.L., and Horvitz, E. (1996), A Graph-Theoretic Analysis of Information Value. In E. Horvitz and F. Jensen (editors), *Proceedings of the Twelfth*

Conference on Uncertainty in Artificial Intelligence, pp. 427-435, Morgan Kaufmann.

Shachter, R. D. (1986), Evaluating Influence Diagrams, *Operations Research* 34 (6), pp. 871-882.

Shachter, R. D. (1988), Probabilistic Inference and Influence Diagrams, *Operations Research* 36, pp. 589-605.

Shachter, R.D., and Peot, M.A. (1992), Decision Making Using Probabilistic Inference Methods, in *Proceedings of 8th Conference on Uncertainty in Artificial Intelligence*, pp. 276-283.

Shachter, R. D. (1999), Efficient Value of Information Computation, In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 594-601.

Shacthter, R.D., B. D'Ambrosio, B. and B. A. Del Favero (1990), Symbolic Probabilistic Inference in Belief Networks, In *Proceedings of the Eighth National Conference on Artificial Intelligence* I, pp. 126-131.

Shachter, R. and M. A. Peot. (1992), Decision Making Using Probabilistic Inference Methods, In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 276-283.

Shachter, R. D. (1990), Evidence Absorption and Propagation through Evidence Reversals. In *Uncertainty in Artificial Intelligence* 5, pp. 173-190. North-Holland.

Shachter, R. D. (1998), Bayes-Ball: The Rational Pastime (for Determining Irrelevance and Requisite Information in Belief Networks and Influence Diagrams), In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence,* pp.480-487, San Francisco, CA, Morgan Kaufmann.

Shafer, G. (1996), *Probabilistic Expert Systems*, Society for Industrial and Applied Mathematics, Philadelphia.

Shenoy, P. (1992), Valuation-Based Systems for Bayesian Decision Analysis, *Operations Research* 40 (3), pp. 463-484.

Suermondt, J., and Cooper, G. (1991), Initialization for the Method of Conditioning in Bayesian Belief Networks, *Artificial Intelligence* **50**, pp. 83-94.

Tatman, J. A. and Shachter, R.D. (1990), Dynamic Programming and Influence Diagrams, *IEEE Transactions on Systems, Man and Cybernetics* 20 (2), pp.365-379.

Vickrey, D. and Koller, D. (2002), Multi-Agent Algorithms for Solving Graphical Games, In *Proceedings of American Association for Artificial Intelligence* (AAAI), pp. 345-351.

Von Stengel, B. (2002), Computing Equilibria for Two-Person Games, *Handbook of Game Theory*, pp. 1723-1759.

Von Neumann, J. and Morgenstern, O. (1947), *The Theory of Games and Economic Behavior*, Princeton: Princeton University Press, 2nd Edition.

Wooldridge, M. (2002), *An Introduction to Multiagent Systems*. John Wiley and Sons Ltd.

Wu, X. and Poh, K. L. (1998), Decision Model Construction with Multilevel Influence Diagrams, In *Proceedings of AAAI 1998 Spring Symposium on Interactive and Mixed-Initiative Decision Theoretic Systems*, pp. 142-147.

Xiang, Y., Poole, D. and Beddoes, M. P. (1993) Multiply Sectioned Bayesian Networks and Junction Forests for Large Knowledge Based Systems, *Computational Intelligence* 9(2), pp.171-220.

Xiang, Y., Pant, B., Eisen, A., Beddoes, M. P. and Poole, D. (1993), Multiply Sectioned Bayesian Networks for Neuromuscular Diagnosis, *Artificial Intelligence in Medicine* 5, pp. 293-314.

Xiang, Y. (1994), *Distributed Multi-agent Probabilistic Reasoning with Bayesian Networks, Methodologies for Intelligent Systems* (Eds.: Z. W. Ras and M. Zemankova), pp. 285-294.

Xiang, Y. P. and Poh, K. L. (1999), Time-Critical Dynamic Decision Making, In *Proceedings of Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 688-695.

Xu, S., Poh, K.L., and Horvitz, E.,J. (2001) Graphical Analysis of Value of Information in Decision Models, in *Proceedings of Fourteenth*

*International Florida Artificial Intelligence Research Society Conference,*155-159.

Xu, S. and Poh, K.L., (2002) Value of Information Analysis in Dynamic Influence Diagram. A. Abraham, M. Koeppen (Eds.), *Hybrid Information Systems*, pp.597-610, Physica Verlag, Heidelberg, 2002

Xu, S. and Poh, K.L., (2002), Value of Information Analysis in Dynamic Decision Models. In *Proceedings of the 2002 AAAI Spring Symposium on Information Refinement and Revision for Decision Making: Modeling for Diagnostics, Prognostics, and Prediction*, pp 109-114, AAAI Press, Menlo Park, California, 2002. Available as AAAI Technical Report SS-02-03, 2002

Xu, S. and Poh, K.L., (2003), *Value of Information in Decision Systems*, PHD Thesis, Department of Industrial & Systems Engineering, National University of Singapore, Singapore

Zambonelli, F., Jennings N. R. and Wooldridge, M. (2001), Organizational Abstractions for the Analysis and Design of Multiagent Systems, In *Agent-Oriented Software Engineering* (Eds.: Ciancarini, P. and M. Wooldridge), Springer-Verlag Lecture Notes in Artificial Intelligence.

Zeng, Y and Poh, K.L., (2005), *Probabilistic Modeling and Reasoning in Multiagent Decision Systems*, PHD Thesis, Department of Industrial & Systems Engineering, National University of Singapore, Singapore.

Zhang, N. L. (1998), Probabilistic Inference in Influence Diagrams, *Journal of Computational Intelligence* 4, pp. 476-497.