



VALUE OF INFORMATION IN DECISION SYSTEMS

BY

XU SONGSONG

(B. Eng. XI'AN JIAO TONG UNIVERSITY)

(M. Eng. SHANGHAI JIAO TONG UNIVERSITY)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2003

Acknowledgements

I would like to express my thanks and appreciation in heart to:

Professor Poh Kim Leng, my supervisor, for his encouragement, guidance, support and valuable advice during the whole course of my research. He introduced me to the interesting research area of normative decision analysis and discussed with me the exciting explorations in this area. Without his kind help, I am not able to finish this project.

Professor Leong Tze Yun, for her warm encouragement, guidance and support. She granted me access to some of the data used in this dissertation.

Dr. Eric J. Horvitz, for his helpful advice regarding the topics in this thesis.

Dr. Xiang Yanping, Mr. Qi Xinzhì and other CDS (now BiDE) group members for their advice and suggestions; Mr. Huang Yuchi, for providing part of the data; and all my other friends, for their assistance, encouragement and friendship.

I would also like to thank my family, for their hearty support, confidence and constant love on me.

Abstract

The value of information (VOI) on an uncertain variable is the economic value to the decision maker of making an observation about the outcome of the variable before taking an action. VOI is an important concept in decision-analytic consultation as well as in normative systems. Unfortunately, exact computation of VOIs in a general decision model is an intractable task. The task is not made any easier when the model falls in the class of dynamic decision model (DDM) where the effect of time is explicitly considered.

This dissertation first examines the properties and boundaries of VOI in DDMs under various dynamic decision environments. It then proposes an efficient method for the exact computation of VOI in DDMs. The method first identifies some structure in the graphical representation of Dynamic Influence Diagrams (DID) which could be decomposed to temporal invariant sub-DIDs. The model is then transformed into reusable sub-junction trees to reduce the effort in inference, and hence to improve the efficiency in the computation of both the total expected value and the VOI. Furthermore, this method is also tailored to cover a wider range of issues, for example, computing VOIs for uncertainty variables intervened by decisions, the discounting of optimizing metric over time and elapsing time being stochastic. A case study example is used to illustrate the computational procedure and to demonstrate the results.

The dissertation also considers computation of VOI in hard Partially Observable Markov Decision Processes (POMDPs) problems. Various kinds of approximations for the belief update and value function construction of POMDPs which take advantages of divide-and-conquer or compression techniques are considered and the recommendations are given based on studies of the accuracy-efficiency tradeoffs.

In general decision models, conditional independencies reveal the qualitative relevance of the uncertainties. Hence by exploiting these qualitative graphical relationships in a graphical representation, an efficient non-numerical search algorithm is developed for identifying partial orderings over chance variables in terms of their informational relevance.

Finally, in summery of all the above achievements, a concluding guideline for VOI computation is composed to provide decision makers with approaches suitable for their objectives.

Keywords

Decision analysis, Value of information, dynamic decision model, graphical decision model.

Table of Contents

1	Introduction.....	1
1.1	The Problem.....	1
1.2	Related topics.....	4
1.3	Methodologies	5
1.3.1	Junction Trees	5
1.3.2	Approximation Methods	6
1.3.3	Graphical Analysis.....	7
1.4	Contributions	8
1.5	Organization of the Dissertation	9
2	Literature Review	11
2.1	Value of Information Computation in Influence Diagrams.....	11
2.1.1	Quantitative Methods for Computing EVPI.....	12
2.1.1.1	Exact Computation of EVPI	13
2.1.1.2	Approximate EVPI Computation.....	21
2.1.2	Qualitative Method for Ordering EVPI	22
2.2	Dynamic Decision Problems	23
2.2.1	Dynamic Influence Diagrams	23
2.2.2	Temporal Influence Diagrams	24
2.2.3	Markov Decision Processes	25
2.2.3.1	Solution methods for MDPs	26
2.2.3.2	Solution methods in POMDPs.....	29
2.3	Summary	32
3	Value of Information in Dynamic Systems	33
3.1	Properties of VOI in Dynamic Decision Models.....	34

3.1.1	A Simple Example	34
3.1.2	Order the Information Values	39
3.1.3	EVPI in Partially Observable Models.....	41
3.1.4	Bounds of EVPI in Partially Observable Models	44
3.2	Value of clairvoyance for the intervened variables	48
3.3	Summary	56
4	Exact VOI Computation in Dynamic Systems.....	57
4.1	Temporally Invariant Junction Tree for DIDs	57
4.2	The Problem.....	62
4.3	Adding Mapping Variables to the Junction Tree.....	68
4.4	Cost of gathering information.....	70
4.4.1	Discounting the cost.....	71
4.4.2	Discounting the benefits	71
4.4.3	Semi-Markov Processes.....	73
4.5	Calculating VOI in Dynamic Influence Diagrams.....	73
4.6	Implementation	75
4.6.1	The follow-up of colorectal cancer	75
4.6.2	The model	76
4.6.3	Methods	80
4.6.4	Results and Discussion	83
4.7	Conclusions.....	85
5	Quantitative Approximations in Partially Observable Models	87
5.1	Structural Approximation	88
5.1.1	Finite History Approximations	88
5.1.2	Structural Value Approximations	91

5.1.3	Factorize the Network.....	94
5.2	Parametric Approximation.....	97
5.3	Comments on the approximations	98
6	Qualitative Analysis in General Decision Models	100
6.1	Introduction.....	100
6.2	Value of Information and Conditional Independence.....	102
6.2.1	Basic Information Relevance Ordering Relations	102
6.2.2	Examples.....	104
6.2.3	Computational Issues	108
6.3	Efficient Identification of EVPI Orderings.....	109
6.3.1	Treatment of Barren Nodes.....	109
6.3.2	Neighborhood Closure Property of u -separation with the Value Node	
	112	
6.3.3	An Algorithm for Identifying EVPI Orderings.....	114
6.4	Computational Evaluation of the Algorithm	118
6.4.1	Applications of the Algorithm to Sample Problems.....	118
6.4.2	Combination of Qualitative and Quantitative Methods.....	119
6.4.3	Application in Dynamic Decision Models.....	121
6.5	Summary and Conclusion.....	121
7	Conclusions and Future Work	124
7.1	Summary.....	124
7.1.1	VOI in Dynamic Models.....	124
7.1.2	Qualitative VOI in General IDs.....	127
7.1.3	Guideline for VOI computation in Decision Models.....	128
7.2	Future Work.....	132

Reference	134
Appendix A: Concepts and Definitions	150
Appendix B VOI Given Dependencies Among Mapping Variables.....	157

List of Figures

Figure 1-1: A simple influence diagram	3
Figure 2-1: An example of influence diagram	19
Figure 2-2: Moral graph and triangulated graph for Figure 2-1 (b).....	20
Figure 2-3: Junction trees derived from influence diagrams in Figure 2-1	21
Figure 2-4: An example of temporal influence diagram.....	24
Figure 2-5: Piece-wise linear value function of POMDP	30
Figure 3-1: Toy maker example without information on market	35
Figure 3-2: Toy maker example with full information	36
Figure 3-3: Toy maker example with information of history	36
Figure 3-4: Condensed form of the three scenarios	38
Figure 3-5: Two-stage DID for a typical partially observable problem	42
Figure 3-6: Decision model with S_i observed before A	43
Figure 3-7: Value function and the EVPI over a binary state b	47
Figure 3-8: DID for calculating VOI of intervened nodes.....	50
Figure 3-9: More complex example.....	53
Figure 3-10: Convert ID (a) to canonical form (b), (c).....	54
Figure 4-1: Partition of a DBN	58
Figure 4-2: Partition of Influence Diagram	58
Figure 4-3: An example of DID.....	63
Figure 4-4: Resulting DBN for the example above	63
Figure 4-5: ID without or with mapping variable added	68
Figure 4-6: Sequentially add mapping variables and cliques	69
Figure 4-7: A part of properly constructed junction tree	72
Figure 4-8: The follow-up problem	80

Figure 4-9: Subnet for the follow-up problem.....	81
Figure 4-10: A sub-junction tree (for 2 stages).....	81
Figure 4-11: Condensed canonical form for VOI of S_i before D_i	82
Figure 4-12: Follow-up without diagnostic tests	85
Figure 5-1: LIMID version of Figure 4-3 (after converting decisions to chance nodes)	90
Figure 5-2: Graphical description of MDP	91
Figure 5-3: DID for Q-function MDP approximation	92
Figure 5-4: DID for fast informed bound approximation	92
Figure 5-5: Even-Odd POMDP (2-stage MDP)	94
Figure 6-1: Influence diagram for Example 1.	105
Figure 6-2: Partial ordering of EVPI for Example 1.	105
Figure 6-3: Influence diagram for Example 2.	107
Figure 6-4: The partial ordering of EVPI for Example 2.....	108
Figure 6-5: EVPI of barren nodes are always bounded by those of their parents	110
Figure 6-6: Nodes with the same EVPI	111
Figure 6-7: Extension of u -separation from value node to a direct neighbor.	112
Figure 6-8: U-separation of Y from V by X can be extended to the maximal connected sub-graph containing Y	113
Figure 6-9: Propagation of EVPI from Y to its neighborhood.	115
Figure 6-10: Part of the ordering obtained in example.....	119

List of Tables

Table 4-1: Cost for alternatives in follow-up case.....	78
Table 4-2: Value functions for the follow-up case	79
Table 4-3 Comparison of Computation Time.....	84
Table 6-1: Comparison of Running Time Using Practical Examples	118
Table 7-1: Guideline for VOI computation	131

Nomenclature

γ vectors	30
Barren nodes	106
belief state	29
Canonical form	51
Chord	16
Clique.....	16
clique-width	66
Complete Graph.....	16
decision-intervened.....	48
Delta-property.....	70
DID	
Dynamic Influence Diagram.....	23
discounting factor	27
D-separation.....	60
grid-based MDP	98
Irrelevant Nodes.....	59
K-L divergence	95
Mapping variables.....	50
Marginalizing.....	15
MDP	
Markov Decision Process	25
no-forgetting arc.....	89
NP-complete	17
POMDP	
Partially Observable Markov	
Decision Process.....	25
projection scheme	95
PSPACE-hard	87
Requisite Observation Nodes.....	59
Requisite Probability Nodes	59
risk-neutral	14
Spreading variables.....	64
strong junction tree	16
Triangulated Graph	16
unresponsive	51
u -separation.....	104
value of evidence	13
value of the information.....	2
zero-congruent	95

1 Introduction

Everyone makes decisions in everyday life. Frequently, people make these decisions just out of common sense or instinct, even though the situations are complex and uncertain. Such decisions are not always rational under close examination. Decision analysis provides a rational way for achieving clarity of action under complex and uncertain decision situations. Decision analysis has grown over the last two decades from a mathematical theory to a powerful professional discipline used in many industries and professions. Managers, engineers, medical doctors, military commanders, management consultants and other professionals are now implementing decision analytic tools to direct their actions under uncertain, complex and even rapidly changing situations.

The theories in normative decision analysis provide a foundation of this dissertation. Hence in this chapter, we shall define the basic problem addressed by this dissertation and provide some general review of related modeling and solution approaches. The last section of this chapter provides a brief summary of the remainder of the dissertation.

1.1 The Problem

Accurate, crucial and prompt information usually will improve the quality of decisions, though an undesirable cost might accompany the activity of gathering such

information. For example, various kinds of medical tests help doctors diagnose a patient more accurately, and introduce more efficient therapies to cure the patient quickly. However the tests may cost the patient some fortune, hence he/she faces the problem of determining whether the test is worthy of the benefits it brings, i.e., how much value will this information add to the total benefits and is it cost-effective.

For decision problems, the computation of information value is regarded as an important tool in sensitivity analysis. By obtaining information for previously uncertain variables, there may be a change in the economic value of the decision under consideration; this is the *value of the information* (VOI). Knowing this VOI is quite useful for the decision maker, since it will help him/her decide which variable is more important, and should be clarified first; or whether the uncertain factor should be clarified at all considering to the cost spent on gathering the information.

However, it is hard to obtain perfect information (or clairvoyance) because the future is full of uncertainty. This uncertainty can be ‘screened out’ by using probability theory, which calculates the expected value as one criterion for random variables. So traditionally the Expected Value of Perfect Information (EVPI) is used to analyze the sensitivity of the effects of gathering information on the final decision.

Recently researchers in decision analysis have adopted graphical probabilistic representations to model decision problems. These representations include Bayesian belief networks and influence diagrams, which are both illustrative and able to deal with the uncertainty in real world problems (Russell and Norvig, 1995).

A Bayesian network is a triplet $\{X, A, T\}$ in which X is the set of uncertain nodes, A is the set of directed arcs between the nodes and T is the set of probability tables associated with the nodes.

An influence diagram includes a set of decision nodes and utilities other than the triplet in a Bayesian network. In influence diagrams, rectangles represent decisions or actions, ovals represent chance events or uncertain events, and diamonds represent the value that can be obtained through the decision process. The directed arcs in the diagram indicate the possible relationship between the variables linked with the arcs. It is quite convenient to build decision models using influence diagrams. Figure 1-1 shows an example of an influence diagram with one decision variable D, one observed variable A, one chance variable B not observed before any decisions, and one value node V.

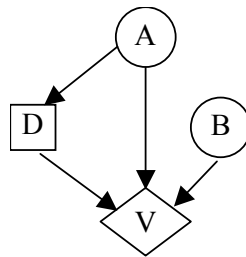


Figure 1-1: A simple influence diagram

The EVPI of an uncertain variable or a set of uncertain variables is the difference between the expected value of the value node with the states of these variables known and unknown. In a decision model, the expected value of any bit of information must be zero or greater, and the upper bound of this value is the EVPI for this piece of information.

Other terms for Expected Value of Perfect of Information include value of clairvoyance, and value of information of observing the evidence.

1.2 Related topics

A great deal of effort has been spent on evaluating the EVPIs of uncertain variables in a decision model, including quantitative and qualitative methods, exact and approximate computations.

The traditional economic evaluation of information in decision making was first introduced by Howard (1966, 1967). Raiffa's (1968) classical textbook described an exact method for computing EVPI. Statistical methods were adopted in these papers to calculate the difference in values between knowing the information and not. Ezawa (1994) used evidence propagation operations in influence diagrams to calculate the value of information out of value of evidence.

Unfortunately, the computational complexity of such exact computation of EVPI in a general decision model with any general utility function is known to be intractable. (Heckerman, Horvitz and Middleton, 1991; Poh and Horvitz, 1996) Even with the simplifying assumption that a decision maker is risk neutral or has a constant degree of risk aversion, the problem remains intractable.

The intractability of EVPI computation has motivated researchers to explore a variety of quantitative approximations, including myopic, iterative one-step look-ahead procedures (Gorry, 1973; Heckerman, Horvitz & Nathwani, 1992; Dittmer and Jensen, 1997; Shachter, 1999) and non-myopic procedures based on employing arguments hinging on the law of large numbers, e.g., central-limit theorem. (Heckerman, Horvitz & Middleton, 1991). Poh & Horvitz (1996) have found that the EVPIs of chance nodes in a decision model can be arrayed if conditional independence statements (CISs) hold among the chance nodes and the value node. In this way, an ordering of EVPIs of chance nodes can be obtained without conducting expensive quantitative computation.

With the knowledge of this EVPI ordering, a decision maker is able to allocate the resource for information gathering.

When the time that decisions are taken also influences the total benefits of the decision maker, we address such problems as dynamic decision models with the explicit reference to time. Besides the traditional ways to address dynamic decision problems, e.g., dynamic programming and Markov decision processes (MDPs) (Bellman, 1957; Howard, 1960), there are dynamic influence diagrams (DIDs) (Tatman & Shachter, 1990), Markov cycle trees (Beck & Pauker, 1983), stochastic trees (Hazen, 1992), and temporal influence diagrams (Provan, 1993). On the other hand, researchers have explored the temporal invariant features of dynamic systems in stochastic models without decisions such as dynamic Bayesian networks (Kjærulff, 1995; Xiang, 1999).

1.3 Methodologies

Since the value of information is typically a sensitivity analysis tool applied before the actual decision is made to guide the collection of important information, timely results of information value are preferred. Thus a major concern in choosing methodologies could be efficiency.

1.3.1 Junction Trees

A relatively fast algorithm for probability propagation in trees of cliques was first developed in Lauritzen & Spiegelhalter (1988). Shafer and Shenoy (1990) introduced junction trees (“Markov tree”) and Jensen & Dittmer (1994) improved the method and applied it in influence diagrams.

As described by Aji & McEliece (2000), the junction tree method is a kind of “General Distributed Law” which distributes the probability marginalization problem into

several local structures called *cliques* and thus saves efforts in the computation of the probability product function (joint probability). The method first renders the DAG of a Bayesian network into an undirected graph by adding arcs between parents of every node, which are called moral arcs, and then adds necessary arcs to make it triangulated, out of which a sequence of cliques can be generated. Calculations upon such cliques were proved to be quite efficient. (Lauritzen & Spiegelhalter, 1988; Aji & McEliece, 2000).

In an influence diagram, the operations we adopt are: First take the expectations over the unknown variables, then maximize over the actions alternately, and finally take expectations over the variables known by the time we choose actions. A general marginalization operation for both maximization and summation was introduced in Jensen & Dittmer (1994), which introduced the junction tree method to decision problems. Kjærulff (1995) and Xiang (1999) applied junction tree propagation in dynamic Bayesian networks, making use of the time-invariant features.

We identify the decomposability of time-invariant dynamic influence diagrams (DIDs), and make use of the repeated features in such DIDs by constructing sub-junction trees on the identified parts. This method is applied in a dynamic case in the medical domain to illustrate the computation for the total expected value and the value of information.

1.3.2 Approximation Methods

The exact solution of general graphical and partially observable decision problems is hard (Cooper, 1990; Papadimitriou & Tsitsiklis, 1987). When it comes to the computation of EVPI, the complexity can be twice that of an exact solution. Even calculating a bound for EVPI will be intractable.

On the other hand, the purpose for VOI computation is to guide the information gathering process and ultimately improve the decision quality. Therefore, in many occasions it is necessary to consider some approximation methods with higher efficiency, but with some tradeoff in accuracy.

The approximate VOI computations considered in this thesis are mainly based on graphical models that consist of a graph topology and a set of parameters associated with it. Hence the original decision model can be approximated by revising either the structures or the parameters, or both, to reduce the total complexity.

1.3.3 Graphical Analysis

Graphs are among the basic tools for establishing probabilistic or other models, especially those in Artificial Intelligence (AI). Many theoretical and practical conclusions of graph theory facilitate researchers in AI to analyze and solve problems explicitly.

A directed acyclic graph (DAG) is defined as a directed graph that contains no directed cycles (Castillo et al, 1997). Basically, Bayesian belief networks and influence diagrams are all DAGs with probability table and conditional independent statements (CIS) embedded in them. The CISs can be checked for validity by implementing some graph separation criterion, namely directed separation or d -separation in DAGs. A formal definition will be introduced in Chapter4 Section 1.

We have sought to find more methods for computation of VOI by leveraging the priorities of the chance nodes in an influence diagram with regards to their VOI, based on graph separation relationships which imply CISs. We have explored the properties in undirected graphs to accelerate the procedure of finding such qualitative

relationships about the information relevance of chance variables in graphical decision models.

In order to show the effectiveness of our methodology we have presented the results obtained on several networks structurally based on real-world models.

1.4 Contributions

This section briefly summarizes the major contributions of the work described in this dissertation.

The problem of value of information is discussed in dynamic decision models, mainly dynamic influence diagrams. Temporal VOI priority is revealed in a dynamic environment. Ways of VOI computation using existing Partially Observable Markov Decision Processes (POMDP) solution methods are studied and boundaries for maximum EVPI of chance nodes are given as well. Moreover, the VOI for decision-intervened chance variables is also investigated for dynamic models.

In order to facilitate fast computation for VOI, we have identified a group of DIDs which can be decomposed into sub-networks with similar structures, and hence a sub-junction tree can be generated based on such sub-networks as the computing template.

This method of time-invariant reusable junction tree is realized and applied to a practical medical case. Experimental results show the method is quite efficient.

To handle the intractability of general VOI computation, quantitative and qualitative approximate approaches are suggested to present timely results.

For hard POMDPs, structural and parametric model reductions are surveyed and analyzed to provide the decision maker guidance in selecting an approximation scheme that best suits the need.

Furthermore, we have worked on a qualitative algorithm for the identification of partial orderings of EVPI for chance nodes in graphical decision models. It considers all the chance nodes in the diagram simultaneously. The algorithm is based on non-numerical graphical analysis on the basis of the idea of undirected graph separation.

The algorithm is tested on a large network based structurally on real-world models. Dramatic savings in time have been observed compared to numerical approaches. Hence we proposed a heuristic combining both the qualitative and quantitative methods together to obtain efficiency and accuracy.

Knowledge of EVPI orderings of the chance nodes in a graphical decision network can help decision analysts and automated decision systems weigh the importance or information relevance of each node and direct information-gathering efforts to variables with the highest expected payoffs. We believe that the methods described in this dissertation can serve the purpose well.

1.5 Organization of the Dissertation

This chapter has given a brief introduction to some basic ideas in decision analysis, reviewed some major work related to the topics addressed in this dissertation, and described the methodologies used and the contributions roughly.

The rest of this thesis is arranged as the following:

As the basis of all further discussions, Chapter 2 introduces related work involving different representations, computation methods used in these representations and various other problems addressed.

Chapter 3 mainly discusses opinions on VOI computation in dynamic environments, both general dynamic influence diagrams and partially observable Markov processes.

Chapter 4 presents an algorithm for identifying time-decomposable DIDs and the VOI computation after the decomposition of the DIDs. The complexity problem is also addressed in this section. Implementation of the method and experimental results on a dynamic medical problem are reported at the end of the section.

Chapter 5 compares various kinds of approximation schemes for POMDP. Issues on the approximation quality and computational complexity are addressed.

Chapter 6 describes methods for identifying the qualitative ordering of VOI of chance nodes in influence diagrams. An algorithm is proposed and shown to be computationally efficient both by theoretical analysis and experiments.

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

2 Literature Review

This chapter briefly surveys some related work: Value of information computation in influence diagrams, dynamic systems including dynamic influence diagram, dynamic Bayesian networks, Markov decision processes and partially observable Markov decision processes.

2.1 Value of Information Computation in Influence Diagrams

Value of information analysis is an effective and important tool for sensitivity analysis in decision theoretical models. It can be used to determine whether to gather information for unknown factors or which information source to consult before taking costly actions. In a decision model, the expected value of any bit of evidence must be zero or greater (Jensen, 1996), and the upper bound of this value is the expected value of perfect information (EVPI) for this piece of evidence. Hence the computation of EVPI is one of the important foci in decision analysis.

EVPI is the difference between a decision maker's expected value calculated with and without the information. When a decision maker considers only money, the expected value can be simply substituted by Expected Monetary Value (EMV).

In a simple decision model \mathbf{M} , \mathbf{X} denotes the set of uncertain parameters; V is the value for the decision maker, \mathbf{D} is the decision set, then

$$EVPI(X) = E_M [\max_d E(V_d | X) - E(V_{d_0} | X)] \quad (2-1)$$

Here $d_0 \in \mathbf{D}$ is the best strategy taken without information, for each instantiation of \mathbf{X} , the best strategy is the same. E stands for taking expectation, and E_M denotes the expected value of model M , which is equivalent to taking expectation over uncertainty X here. For a binary X with probability distribution $p(x_0)$ and $p(x_1)$, a binary D with d_0 and d_1 , $EVPI(X) = p(x_0)[\max_d V(d, x_0) - V(d_0, x_0)] + p(x_1) [\max_d V(d, x_1) - V(d_0, x_1)]$.

As shown by formula (2-1), $EVPI(X)$ denotes the average improvement the decision maker could expect to gain over the payoff resulting from his selection of alternative d_0 given perfect information on the parameter X prior to the time of making the decision.

Other terms for the Expected Value of Perfect of Information include value of clairvoyance, and the value of information of observing the evidence, etc.

2.1.1 Quantitative Methods for Computing EVPI

There are several ways to evaluate the EVPI in a decision model. They can be divided into two main groups: quantitative computation that returns a certain number and qualitative evaluations that returns an ordering of EVPIs of the uncertain variables. The EVPIs can be exactly calculated, or approximated under some assumptions.

The earliest computation of EVPI started from Howard (1966, 1967). The expected profit given clairvoyance about an uncertain variable is calculated by evaluating the expected profit given that the state of the variable is known and then summing up the expectation with respect to the probability assignment on it. Deducted by the expected

value without knowing the outcomes of the variable, we get the EVPI of the specific uncertain variable.

2.1.1.1 Exact Computation of EVPI

The value of evidences (VOE) is calculated in the process of updating evidence (observations) through the influence diagram. VOE can be used to find out what evidence we would like to observe to increase the benefit, and the maximum benefit can be received by removing the uncertainties, i.e., the EVPI. VOE is defined below (Ezawa, 1994):

$$VOE(X_J = x_j) = EV(X \setminus X_J, X_J = x_j) - EV(X) \quad (2-2)$$

where X_J is the chance variable associated with node J that an observation can be made, x_j is one of the instances for X_J , and EV is the expected value. $X \setminus X_J$ is the chance node set excluding X_J , with X_J taking the value x_j . EVPI of X_J can be represented by a function of VOE:

$$EVPI(X_J) = \sum VOE(X_J = x_j) * P(x_j)$$

$$\text{For the state space } \Omega_J \text{ of node } J. \quad (2-3)$$

In other words, 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 VOE for each x_j with $P(x_j)$, EVPI can be computed. The unconditional probability $P(X_J)$ can always be obtained by applying arc reversals (Shachter, 1986) between its predecessors as long as they are not decision nodes.

The value of evidence could be negative, but the value of perfect information is always greater than or equal to zero.

Note that the EVPI computed from VOE is the EVPI for overall decisions, assuming the observing of the evidence before the first decision if a sequence of decisions are involved in the influence diagram.

This method of calculating VOI by evidence propagation solely depends on the computational efficiency of general propagation algorithms in the influence diagrams. It just performs the operation of evidence propagation j times, where j is the number of outcomes of the uncertain node J under concern.

In practical use, the problem may grow very large and be complicated, thus the exact computation of EVPI becomes intractable (Cooper, 1990). In order to avoid the intractability, some assumptions were proposed to simplify the computation in practical use, e.g., myopic assumption. This is a situation that the decision maker considers whether to observe one more piece of evidence before acting given he has zero or more pieces of evidence in the influence diagram. For each piece of evidence, the decision maker will act after observing only that piece of evidence. This assumption is very often used in sequential decision-making, e.g., the pathfinder project (Heckerman et. al., 1992). Another frequently used simplification is assuming the decision maker is risk-neutral so that the value can be replaced by utility. The decision maker's risk profile, i.e., risk-neutral, risk-seeking or risk-averting makes him value differently of certain amount of money. Risk-neutral is the only linear mapping from money value to utility, while the other two are nonlinear.

Lauritzen & Spiegelhalter (1988) developed a relatively fast algorithm for probability propagation in trees of cliques in belief networks. Shafer and Shenoy (1990) introduced the concept of junction trees ("Markov tree") and Jensen & Dittmer (1994) improved the method by extending the marginalization of probability nodes to decision nodes and thus applied it in influence diagrams.

Such an inference method could be adopted in the computation of EVPI as well. Dittmer & Jensen (1997) pointed out that constructing strong junction trees corresponding to the original influence diagram facilitates the computation of the EVPI for different information scenarios. The computation procedure for both scenarios, with and without information, can make use of the same junction tree.

Let's denote the chance node set in an influence diagram as W , decision node set as D , the value node as V . For all the chance nodes and decision nodes, we can partition them into a collection of disjoint sets $W_0, \dots, W_k, \dots, W_n$; for $0 < k < n$, W_k is the set of chance nodes that will be observed between decision D_k and D_{k+1} ; W_0 is the initial evidence variables, W_n is the set of variables that will be observed only after the last decision. This induces an order \prec in W :

$$W_0 \prec D_1 \prec W_1 \prec D_2 \prec \dots \prec D_k \prec W_k \prec D_{k+1} \prec \dots \prec D_n \prec W_n \quad (2-4)$$

In graphical representation, this means that W_k is the parent set of D_{k+1} , and W_n thus includes all the other chance nodes that cannot be observed before the last decision.

For this partitioned influence diagram, Jensen et al. (1994) have shown that the maximum expected utility U_k for decision D_k is:

$$U_k = \max_{D_k} \sum_{W_k} \dots \max_{D_n} \sum_{W_n} P(W_k, \dots, W_n | W_0, \dots, W_{k-1}, D_1, \dots, D_n) * U \quad (2-5)$$

Here, U is the utility function. This equation means the maximum expected utility for a decision problem could be calculated by performing a series of marginalizations of summation and maximization alternatively.

Marginalizing a chance variable A out of the joint probability distribution we get the joint probability of all the remaining variables: $P(X_1, \dots, X_n) = \sum_A P(A, X_1, \dots, X_n)$.

Marginalization can be conducted in a graph which consists of a vertex set and an edge set. Hence the following gives out definitions for some basic concepts in Graph Theory which are related to marginalization.

Def. 2.1 *Chord* (West 2002)

A chord of a cycle C in a graph is an edge not in the edge set of C whose endpoints lie in the vertex set of C .

Def. 2.2 *Complete Graph* (West 2002)

A graph in which each pair of graph vertices is connected by a graph edge.

Def. 2.3 *Clique* (West 2002)

A clique of a graph is its maximal complete subgraph.

Def. 2.4 *Triangulated Graph* (Castillo, 1997)

Triangulated graph refers to the undirected graph that every loop of length four or more has at least one chord

The marginalization corresponds to the following operations on the undirected graph: complete the set of neighbors of A in the graph, and then remove A . All variables can be eliminated in this manner without adding edges if and only if the graph is triangulated (Jensen, 1996). The operation of *triangulation* is making a graph into a triangulated one by adding chords to break the loops. The procedure of adding such chords is called *fill-in*. The fill-in that gives the smallest state space for a triangulation is an optimal fill-in.

For a triangulated undirected graph, the cliques in this graph can be organized into a *strong junction tree* with the following definition:

A tree of cliques is called a junction tree if for each pair (C_1, C_2) of cliques; $C_1 \cap C_2$ belongs to every clique on the path connecting C_1 and C_2 . For two adjacent cliques C_1 and C_2 , the intersection $C_1 \cap C_2$ is called a *separator*. If a junction tree has at least one distinguished clique R , called a strong root, such that for each pair (C_1, C_2) of adjacent cliques in the tree, with C_1 closer to R than C_2 , there exists an ordering of C_2 that respects the order \prec and with the vertices of the separator $C_1 \cap C_2$ preceding the vertices of $C_2 \setminus C_1$, then the junction tree is a strong one.

Finding an optimal junction tree is NP-complete (Arnborg, Corneil, & Proskurowski, 1987), which means the problem is both NP (verifiable in nondeterministic polynomial time¹) and NP-hard (any other NP-problem can be translated into this problem). The simple greedy algorithms by Rose (1974) will often give smaller state space than the fill-ins generated by the vertex ordering of the algorithm Maximum Cardinality Search of Tarjan and Yannakakis (1984), but a mistake in the first step will lead to a junction tree far from optimal. Kjærulff (1990) discussed algorithms for finding a fill-in given a small state space based on simulated annealing. They are better in performance but take longer time to run. Jensen & Jensen (1994) proposed an approach to construct optimal junction trees from triangulated graphs and Becker and Geiger (1996) developed some sufficiently fast algorithm to find close-to-optimal junction trees.

In the junction tree, two functions a probability potential ϕ_C and a utility potential ψ_C are associated to each clique C . The joint potential ϕ and ψ of a junction tree J are defined as $\phi = \prod_{C \in J} \phi_C$, $\psi = \sum_{C \in J} \psi_C$. For a chance variable X , the marginalization operation is $M_X \phi = \sum_X \phi_C$; and for a decision variable D , $M_D \phi = \max_D \phi$. If J is a

¹ A problem is assigned to the NP (nondeterministic polynomial time) class if it is verifiable in polynomial time by a nondeterministic Turing machine. A nondeterministic Turing machine is a “parallel” Turing machine which can take many computational paths simultaneously, with the restriction that the parallel Turing machines cannot communicate.

junction tree, C_1 and C_2 are adjacent cliques with separator $S \subset J$, and if $C_1 \prec C_2$ which indicates C_1 is closer to the root than C_2 is, then C_1 updates its potential functions by absorbing from C_2 :

$$\phi'_{C_1} = \phi_{C_1} * \phi_S; \quad \psi'_{C_1} = \psi_{C_1} + \psi_S / \phi_S \quad (2-6)$$

here,

$$\phi_S = \prod_{C_2 \setminus S} \phi_{C_2}; \quad \psi_S = \prod_{C_2 \setminus S} \phi_{C_2} * \psi_{C_2}$$

$C_2 \setminus S$ refer to nodes in C_2 excluding those also in separator set S .

By successively absorbing leaves into the strong root in the junction tree constructed, it is easy to obtain the overall probability and utility potentials.

Dittmer and Jensen (1997) proposed a method to calculate VOI based on only one junction tree, i.e., reusing the original junction tree for calculating the expected utility (or value) with information obtained. The method can be more clearly described after we introduce the following definitions (Shachter, 1999):

“Clique C is inward of another clique C’ if C is either the strong root clique or between the root clique and C’. And C’ is said to be outward of C. If all cliques containing a variable A are outward of some cliques containing variable B, then A is said to be strictly outward of B and B strictly inward of A. If all clusters containing A either contain B or are outward of a cluster containing B, then A is weakly outward of B and B is weakly inward of A.”

The case of observing a variable A before D can be calculated by adding A to all the cliques between A and D 's inward-most cliques.

We illustrate the propagation and VOI computation through junction tree by an example from Dittmer and Jensen (1997). Scenario (a) in Figure 2-1 is an influence

diagram with three decision nodes, four chance nodes and one utility node. Scenario (b) indicates the situation of observing chance node B before the decision D_1 . In order to turn this directed graph into an undirected graph for further operations, we need to add arcs between each pair of parents for every node, and drop the direction of arcs. This procedure is called moralization, and the arcs added between parents are moral arcs. The calculation of the expected utility for both scenarios can be based on the same strong junction tree; hence Figure 2-2 only shows the moralization and triangulation for scenario (b) in Figure 2-1, which is more complicated.

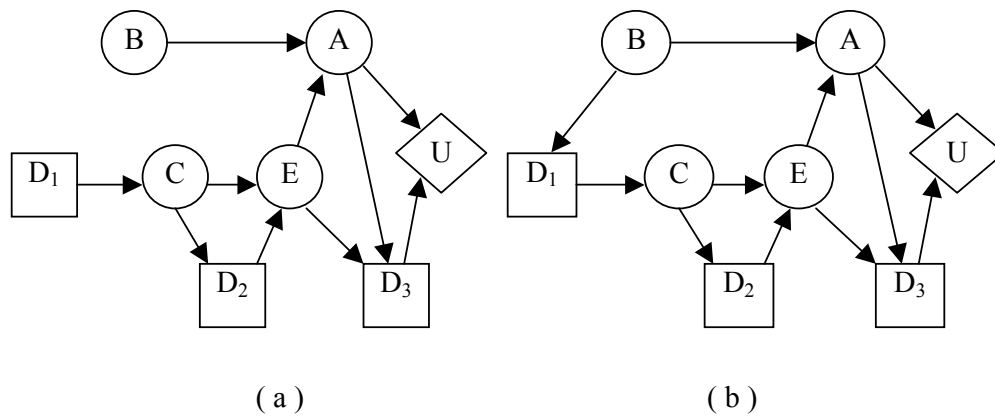
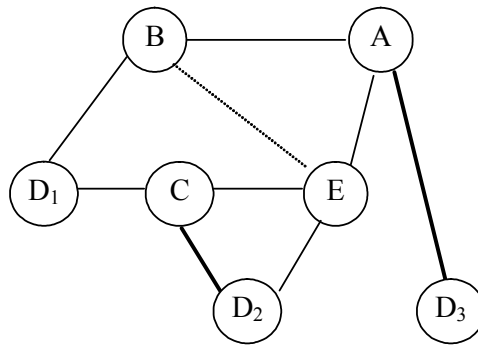
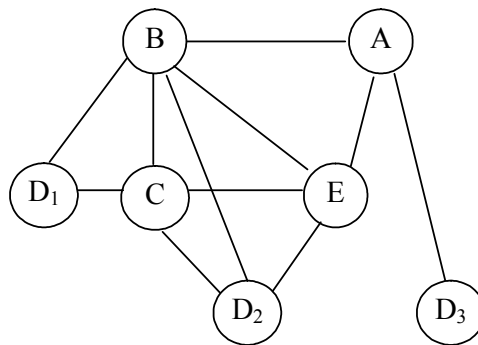


Figure 2-1: An example of influence diagram

(a) with no information on B before D_1 and (b) with information prior to D_1



(a) Moral graph



(b) Triangulated graph

Figure 2-2: Moral graph and triangulated graph for Figure 2-1 (b)

In Figure 2-2 (a), dotted line from B to E is a moral arc to ‘marry’ A’s parents B and E. The solid lines (C, D_2) and (A, D_3) C and A are requisite observations before D_2 and D_3 respectively. The concept of requisite observation will be introduced in detail in Section 4.1. E is not a requisite observation of D_3 , hence not appeared in Figure 2-2 (a).

Figure 2-3 shows the junction trees for both scenarios. Here D_1C and BD_1C are the root cliques respectively. Using junction tree for scenario (b), BD_1C is inward of BCD_2E , node A is strictly outward of C, but weakly outward of E. The difference of the two junction trees only lies in the cliques that are from inward-most clique of the decision D_1 to inward-most clique of B.

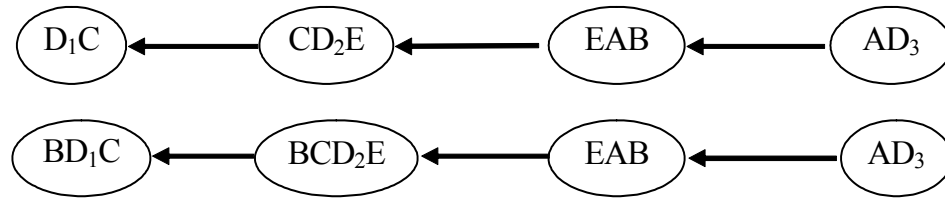


Figure 2-3: Junction trees derived from influence diagrams in Figure 2-1

Above, scenario (a); below, scenario (b)

In (Dittmer and Jensen 1997), decision nodes were treated as chance nodes graphically in triangulation and junction tree construction; the difference only lies on the marginalization operations. In Jensen (1996) the computation in influence diagrams was analogous to that in Bayesian networks after Cooper's transformation (Cooper, 1988), which turns the decision and value nodes into chance nodes. Shachter (1999) used the Bayes-Ball algorithm (Shachter, 1998) to find requisite observations for decisions, which may lead to a simpler (unfortunately, sometimes more complex) diagram. Decision nodes are treated as deterministic nodes afterwards.

2.1.1.2 Approximate EVPI Computation

Heckerman et al (1991) proposed a non-myopic approximation for identifying cost-effective evidence. First, calculate the net value of information for each piece of evidence using the exact method under the myopic assumption. Second, arrange the evidence in descending order according to their net value of information, and finally compute the net value of information of each m -variable subsequence ($1 \leq m \leq$ number of all the chance nodes).

For a diagnosis problem with evidences independent to each other given the hypothesis, the weight of evidence could be added up based on the central-limit theorem for large m . This approximated method can be extended to the special classes of dependent distributions where the central-limit theorem is valid for these dependent distributions as well.

A more traditional approximate method is Monte Carlo Simulation. Supposing the probability distributions of each chance variable is known, it is easy to generate great amount of random numbers for these variables. The best strategy and the corresponding expected utility can be determined thereafter (Felli & Hazen, 1998).

This approach is simple and easy to understand and execute. However, it consumes a great deal of time and space in order to generate enough examples to obtain statistical significance. When the number of random variables gets large, which is quite common in practice, the simulation becomes hard to apply.

2.1.2 Qualitative Method for Ordering EVPI

Besides all the quantitative methods in calculating the EVPI in a decision model, Poh & Horvitz (1996) proposed a way to reveal the qualitative relationships about the informational relevance of variables in graphical decision models based on conditional independencies through graphical separations of uncertain nodes from utility nodes, thus to obtain a partial ordering of EVPI without considering the numerical value of nodes.

The details of this method will be left for further discussion in later chapters.

2.2 Dynamic Decision Problems

A decision problem may have a sequence of decisions taken at different time stages. When time is explicitly considered, such problems are called *dynamic decision problems*.

Researchers have addressed dynamic decision problems with various kinds of dynamic decision models. They usually depict several essential parameters for decision analysis, e.g., the states of the system that vary with time, a set of control laws that can influence the future states of the system, some criteria for the selection of the control laws, (maximize values, utilities, probabilities or minimize costs), and an underlying stochastic process that governs the evolution of the above elements in time. Some of these dynamic decision models will be introduced in the following sections.

2.2.1 Dynamic Influence Diagrams

Tatman and Shachter (1990) extended the general influence diagrams into dynamic influence diagrams (DIDs) by allowing time-separable value functions, one function for each time unit or decision stage. These time-separable value nodes can be summed up or multiplied into a super value node. The operations of chance node removal and decision node removal in general influence diagrams can also be performed over an addend (if sum Σ) or factor (if product Π) in the value function instead of over the entire value function. And non-super value nodes can be reduced into the super value node that is the direct or indirect successor of them.

Dynamic influence diagrams are typically used to address finite stage problems with partially observable state variables. DIDs allow a compact specification of the relationships between observable and non-observable variables, decisions and values

received in every stage. Furthermore, this representation gives direct information about the topology of the model.

However, when the system evolves for more time stages, the graphical representation grows unnecessarily large. Xiang & Poh (1999) mentioned a condensed form for dynamic influence diagrams which represent the repeating features of an N -stage DID into one snap-shot stage.

As a non-decision counterpart of DID, the dynamic Bayesian networks (DBNs) capture the dynamic process by representing multiple copies of the state variables, one for each time step (Dean and Kanazawa, 1989).

Some other temporal models, such as hidden Markov Models (HMM) and Kalman filters can be considered as special cases of DBN, where the former are DBNs with a single discrete state variable and the latter are DBNs with continuous state/evidence variables and linear Gaussian transition/observation probabilities.

2.2.2 Temporal Influence Diagrams

Provan (1993) used temporal influence diagrams (TIDs) to represent a sequence of influence diagrams which evolve with time. Like Figure 2-4, each influence diagram ID_0 to ID_n models an interval of the system, assuming static states in these time intervals. Temporal arcs between the time-indexed influence diagrams depict the dependencies of a future stage on a past k th stage, $1 < k < N$ (N is the total time horizon).

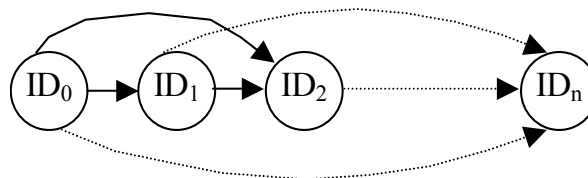


Figure 2-4: An example of temporal influence diagram

Since the more temporal arcs added, the harder the inference in the temporal influence diagrams, Provan (1993) proposed two ways to restrict the network size to ensure the computational feasibility. One way is to construct the IDs in each time interval only with a particular set of observations; the other is assigning temporal arcs for only a subset of variables instead of all the variables.

Later, modifiable temporal belief networks (MTBNs) were developed by Aliferis (1996) as a temporal extension of general Bayesian networks (BNs) to facilitate modeling in dynamic environments. There are three types of variables in MTBN, ordinary observable variables, arc variables and time-lag variables. These variables correspond to chance nodes, the dependency arcs and the temporal arcs in temporal BNs and IDs, respectively. The author used a condensed form of MTBN to facilitate model definition, and a deployed form with variables replicated for each time interval for inference.

2.2.3 Markov Decision Processes

Markov decision processes (MDPs) are mathematical models for sequential optimization problems with stochastic formulation and state structure (Howard, 1960). A Markov decision process consists of five elements: decision epochs T , states S , actions A , transition probabilities P and rewards r . Semi-Markov decision processes (SMDPs) are MDPs with stochastic time-intervals between transitions.

A partially observable Markov Decision Process (POMDP) is a generalization of a Markov Decision Process, which allows for incomplete information regarding the state of the system. At each decision epoch, the decision maker must select an action based only on the incomplete information at hand.

In a POMDP, $S = \{S_1, S_2, \dots, S_t, S_{t+1}, \dots, S_n\}$ is the set of system states. At any discrete time stage $t \in T$, the system is in state S_t . The decision maker then performs an action $a_t \in A$, makes the system change into S_{t+1} and receives an observation (evidence) O_t afterwards. The process is characterized by a value function $V(R_t | S_t, A_t)$, a transition probability distribution $P(S_{t+1} | S_t, A_t)$ and an observation probability distribution $P(O_t | S_t, A_t)$ ². Let $H_t = \{a_1, o_1, a_2, o_2, \dots, a_{t-1}, o_{t-1}\}$ denote the history of actions and messages received up to time t . If based on this information, the decision maker chooses action a_t , a real value $V(s_t, a_t)$ is received when the state of the system is s_t . Time increments by one, $H_{t+1} = H_t \cup \{a_t, o_t\}$, the decision maker choose action a_{t+1} , and the process repeats.

The information in H_t can be encapsulated in the vector S_t (Aoki, 1965; Bertsekas, 1976), and partially observed process can be remodeled as an equivalent fully observed MDP with continuous state space.

2.2.3.1 Solution methods for MDPs

Let $V_t(s)$ be the optimal total expected revenue, given the system starts in state s , taking action a , and results in state s' with transition probability $p(s'|s, a)$, the backward recursive equation for MDP is:

$$V_t(s) = \max_a [r(s, a) + \sum_{s' \in S} p(s'|s, a)V_{t+1}(s')], \quad t \in 1 \cdots N \quad (2-7)$$

Here $r(s, a)$ is the reward received every stage and N is the total number of stages.

Start from $V_{N+1}(s')$, the additional salvage value received at the beginning of time $N+1$ given $S_{N+1} = s'$, a direct approach to calculate the total expected value is to determine

²In the literature, e.g., (Cassandra et al, 1997), this distribution is expressed as $P(O_t | S_t, S_{t+1}, A_t)$.

the values of $V_t(s)$ iteratively for N times, which is known as the *value iteration* approach.

Originated from Howard (1960) and Bellman (1961), the value iteration has a lot of variants that aim to improve the efficiency. The solution is an optimal policy $\pi = \{\delta_1, \delta_2, \dots\}$ where δ_t is a function ($S \rightarrow A$), $t \in N$, and the maximum cumulative expected value $V(s)$ for each s in the state space.

Generally, equation (2-7) can be modified to make the total revenue convergent when $N \rightarrow \infty$ by adding an economic discounting factor β (e.g., interest rate) greater than 0 and less than 1. Hence the MDP can be extended to infinite-stage problems by performing enough iterations until a certain small tolerance is reached. In the discounted case, $V(s)^* = \lim_{t \rightarrow \infty} V_t(s)$, proven by White (1978). Thus we can search for a stationary policy that satisfies $|V_{t+1}(s) - V_t(s)|$ less than an arbitrary small ε . However, it is not efficient to iterate the computation until N is sufficiently large.

In order to deal with infinite-stage problems, Howard (1960) proposed *policy iteration*. As a simplest implementation, the policy iteration can be initiated with any policy, and then determine the optimal policy through the iteration over all the possible policies. A more efficient way (Bellman, 1957; Howard, 1960) is to find a sequence of policies of increasing quality, hence avoid considering many sub-optimal policies.

The sequence of the policies generated by policy iteration is monotonically increasing in value. The algorithm will converge on the optimal solution within finite number of steps as there are a finite number of policies.

Infinite-stage problems can be formulated as linear programs (Derman, 1970; Kushner and Kleinman, 1971). It solves the optimization problem as the following:

$$\text{Maximize } E = \sum_{i \in S} \sum_{a \in A} \rho_i^\pi r_i^\pi$$

S.T.

$$\boldsymbol{\rho}^\pi P^\pi = \boldsymbol{\rho}^\pi,$$

$$\rho_1^\pi + \rho_2^\pi + \dots + \rho_{|S|}^\pi = 1,$$

$$\rho_i^\pi \geq 0, \quad i \in S \quad (2-8)$$

ρ_i^π is the long run stationary probabilities of the transition probabilities matrix P^π corresponds to policy π . $\boldsymbol{\rho}^\pi$ is the vector of ρ_i^π .

Various techniques have been developed for solving large linear programming problems, e.g., the simplex method, and Karmarkar interior-point algorithm.

Among all the solution methods introduced, linear programming supports better sensitivity analysis. Furthermore, we can add more constraints to (2-8) to solve a wider class of problems. The disadvantage is it prohibits the analysis for any specific time stage.

The policy evaluation routine of policy iteration method needs to solve a set of linear formulas, which requires $O(|S|^3)$ computation time if using Gaussian elimination approach. When the state space grows large, the computational cost of obtaining the exact solution will become quite expensive.

One alternative is to solve such set of linear formulas by approximation. This forms an approximate value in the policy evaluation step. Hence when the number of controls is large, such approximation is much less expensive.

Another way is to form super-states by lumping together the states of the original system, and then solve a system with smaller state space. This is the adaptive state aggregation method (Bertsekas, 1987), which is effective when the number of states is very large.

2.2.3.2 Solution methods in POMDPs

It has been pointed out in the literature (Cassandra, 1998) that the policy for a POMDP is not necessarily Markovian. In general, finding an optimal policy requires the decision maker remember the entire history H_t of past observations and actions. Instead of remembering the history, one can maintain a summery sufficient statistic, the *belief state*, as the basis of the optimization as well (Astrom, 1965; Sondik, 1971 and Striebel, 1965).

The belief state b is the probability distribution over the states, where $b(s)$ corresponds to the probability of system being in state s . This belief state can be easily updated after an action is taken and an observation is made according to the Bayes Rule:

$$b_o^a(s') = \frac{\sum_{s \in \mathcal{S}} P(o | s, a, s') P(s' | s, a) b(s)}{\sum_{s \in \mathcal{S}} \sum_{s'' \in \mathcal{S}} P(o | s, a, s'') P(s'' | s, a) b(s)} \quad (2-9)$$

and thus the value updating equation becomes:

$$V_t(b) = \max_a [\sum_{s \in \mathcal{S}} b(s) R(s) + \beta \sum_o \sum_{s', s'' \in \mathcal{S}} P(o | s, a, s') P(s' | s, a) b(s) V_{t+1}(b_o^a)] \quad (2-10)$$

Hence the POMDP can be viewed as a continuous state-space MDP.

Exact Solutions

Smallwood and Sondik (1973) proved that for any finite t , $V_t^*(b)$ is piecewise linear and convex on state space \mathcal{S} (Sawaki (1980) had given a generalization to piecewise

linear Markov decision processes). Hence, V_t^* has a representation as the maximum of a finite number of linear functions, i.e., $V_t^*(b) = \max \{\gamma^T b : \gamma \in \Gamma_t\}$ for some finite set Γ_t of $|S|$ -vectors. Figure 2-5 shows a set of γ vectors for a binary state variable.

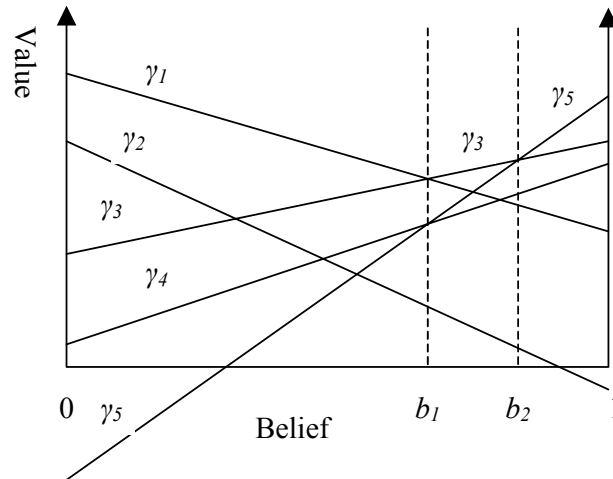


Figure 2-5: Piece-wise linear value function of POMDP

The simplest idea is to enumerate all possible vectors (Monahan, 1982; Sondik, 1971) for each action and observation. A large number of vectors must be generated if using this way. Among the vectors, many are not useful since they are completely dominated by other vectors over the entire belief space, like γ_2 and γ_4 shown in Figure 2-5.

The first exact algorithm for POMDPs was derived by E.J. Sondik (1971). This so-called One-Pass algorithm starts with an arbitrary belief point, constructs the vector for that point and then defines a set of constraints over the belief space where this vector is guaranteed to be dominant.

The full set of constraints defined by Sondik may generate more boundaries than necessary; while the omitting of one of them may result in generating too large regions. Cheng (1988) proposes some algorithms based on alternative relaxations of the constraints. These algorithms result in fewer boundaries than the one-pass

algorithm, and obtain computational savings at the same time. One of them, the linear support algorithm, can be modified to produce approximate solutions with error bounds, which is an advantageous feature for computationally complex POMDPs.

The *witness* algorithm proposed by Littman et al (1994) does not concern all the actions all the time. It concentrates on finding the best value function for each of the actions separately. Furthermore, this algorithm also deals with only one observation at a time. Like the other algorithms, it tries to find the region where it is assured that the particular choice is the best. If it finds a belief point where a different action would be better, this fact serves as a witness/proof that the current set of vectors is not yet the real dominating value functions, and the search will go on.

Similar to the witness algorithm in the way of dealing with actions and observations individually, the incremental pruning algorithm (Zhang and Liu, 1996; Cassandra, Littman and Zhang, 1997) considers constructing sets of vectors for each action and then focusing on one observation at a time. The incremental pruning algorithm is the fastest algorithm to compute the dynamic programming update up to now, according to (Hansen and Feng, 2000).

Approximations

Currently, the exact solution methods tend to be intractable for problems other than simple POMDPs with a few dozen states. Hence, various kinds of approximations on the value function or the belief state are proposed. Basically the approximations consider tradeoffs between computational complexity and model accuracy.

2.3 Summary

Many researchers have exerted their efforts in value of information studies and problem solving in dynamic decision models. They proposed different representations to better capture characteristics of problems, and developed various kinds of methods for solving these problems efficiently and / or accurately. Based on these research works, we are able to serve the need for studying value of information under dynamic environments.

This chapter gives a brief review of a variety of topics related to value of information studies and dynamic decision models. Some terms and concepts are introduced for a better understanding of the discussion in consecutive chapters.

3 Value of Information in Dynamic Systems

Time has an influence on our revenue. For example, if one had bought Microsoft stocks in 1980's, he/she might be a millionaire now. Time also influences our non-monetary utilities. In some cases, a patient could have been saved if he/she had an early screening for the disastrous diseases. The recent multi-country outbreak of the Severe Acute Respiratory Syndrome (SARS) could have been prevented had we realized its infectiousness at the very beginning of the outbreak. Decisions are hard since uncertainties are involved; yet when the impacts of time are considered, they can be even harder. A decision made at a certain time point may affect both the system status and other decisions to be made some time in the future. The decision problem that takes effects of time into account is called a dynamic decision problem.

Sometimes we make decisions based on current status of the decision system, e.g., taking an umbrella after catching sight of the rain. However, in many real world problems, only vague information about the system is available. For example, when a patient visits the doctor, exactly what disease or combination of diseases the patient has is not clear to both of them. Then the doctor will let the patient talk about the symptoms, his/her previous medical records, and most of the time, prescribe a set of diagnostic tests for the patient to undergo. Decisions for the treatment are based on this gathered information about patient's status. The better and the earlier a doctor knows the status of his patient, the greater the chance for the patient to recover. In this case,

information on the true state of the patient is valuable, and the cost-effectiveness of the information with regard to time is worth studying.

3.1 Properties of VOI in Dynamic Decision Models

The models for dynamic systems have been briefly discussed in the previous chapter. Though Markov Decision Processes (MDPs) provide concise mathematical formulations to model dynamic problems, the concept of information value is hard to be clearly represented in MDPs and their extensions without some extra efforts. On the other hand, Dynamic influence diagrams (Tatman and Shachter, 1990) represent the dependencies between the observations, the uncertainty nodes, the decisions and the values in each stage explicitly. Moreover, existing computation methods for value of information in general models can be easily applied to dynamic influence diagrams. Therefore, dynamic influence diagrams are adopted as the basic configuration of our problems both for their graphical representation and for the convenience of extending methods in general influence diagrams. However, this does not mean that there is a restriction in the use of representations; as shown later, other models will be mentioned as well, yet taking a graphical appearance.

3.1.1 A Simple Example

Let's illustrate the basic ideas of value of information computation with the following simple example.

A toy maker is producing toys without knowing the situation of the toy market. This is a quite naive situation with the system states totally unobservable. He has a decision Ad_i , to decide whether to use advertising or use no advertising in each time period,

which may affect the market in the future. In the following figure, random variable $Market_i$ ($i=0, \dots, 3$) denotes the status of the market in the i th stage, successful (good market) or unsuccessful (bad market). The node $Value_i$ is the revenue that the decision maker will receive in the i th stage. Here we only display four stages. Dashed arrows indicate the temporal order.

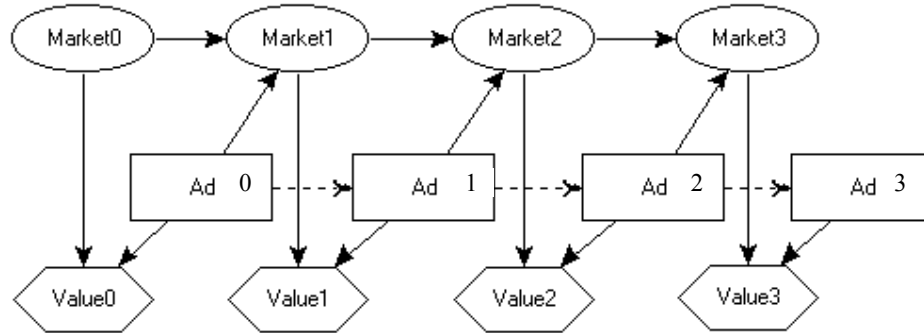


Figure 3-1: Toy maker example without information on market

In this case the state variables are totally unobservable. Abbreviating $Market_i$ as M_i , $Value_i$ as V_i , the total value for this scenario is:

$$\begin{aligned}
 V = \max_{Ad_0} \max_{Ad_1} \max_{Ad_2} \max_{Ad_3} \sum_{M_0} P(M_0) \{ & V_0(M_0, Ad_0) + \sum_{M_1} \\
 P(M_0, M_1, Ad_0) \{ & V_1(M_1, Ad_1) + \sum_{M_2} P(M_1, M_2, Ad_1) [\\
 V_2(M_2, Ad_2) & + \sum_{M_3} P(M_2, M_3, Ad_2) V_3(M_3, Ad_3)] \} \} \} \quad (3-1)
 \end{aligned}$$

It will be an intractable computation if the state space is large.

Supposing now we can observe the system state before taking actions in all stages, as shown in Figure 3-2. This case is the same as fully observable Markov Decision Processes, which can be solved by the classical dynamic programming techniques efficiently as in (Bellman, 1957; Howard, 1960; Bertsekas, 1995). The total expected value is the following:

$$\begin{aligned}
 V^* = & \sum P(M_0) \max_{Ad_0} \{ V(M_0, Ad_0) + \sum P(M_0, M_1) \max_{Ad_1} \{ V(M_1, Ad_1) \\
 & + \sum P(M_1, M_2) \max_{Ad_2} [V(M_2, Ad_2) + \sum P(M_2, M_3) \max_{Ad_3} V(M_3, \\
 & Ad_3)] \} \} \} \} \quad (3-2)
 \end{aligned}$$

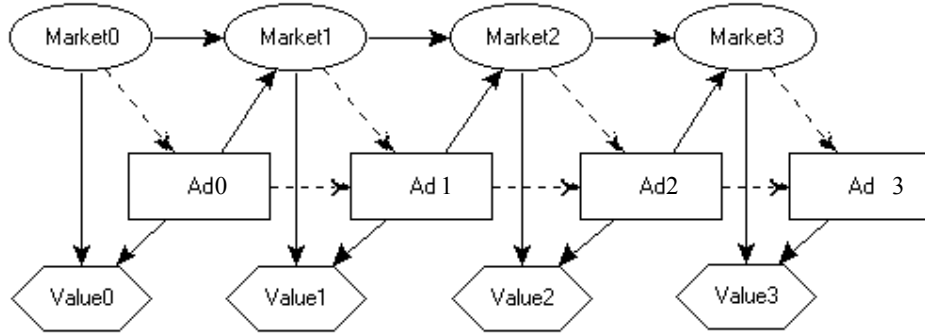


Figure 3-2: Toy maker example with full information

Figure 3-3 illustrates the situation of knowing the previous market status (including all the history) before the decision of current stage. The expected total value is:

$$\begin{aligned}
 V^* = & \max_{Ad_0} \sum P(M_0) \{ V(M_0, Ad_0) + \max_{Ad_1} \sum P(M_0, M_1, Ad_0) \{ V(M_1, \\
 & Ad_1) + \max_{Ad_2} \sum P(M_1, M_2, Ad_1) [V(M_2, Ad_2) + \max_{Ad_3} \sum P(M_2, \\
 & M_3, Ad_2) V(M_3, Ad_3)] \} \} \} \} \quad (3-3)
 \end{aligned}$$

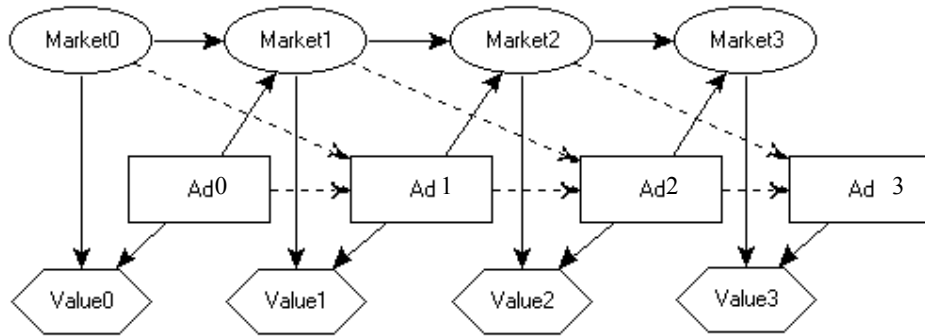


Figure 3-3: Toy maker example with information of history

In general, for decision d , denote the reward matrix in the n -stage-to-go as \mathbf{Q}_d^n , transition probability matrix as \mathbf{P}_d^n , the updating equation for calculating values with perfect information (MDP) is:

$$\mathbf{V}^{n,n} = \max_d (\mathbf{Q}_d^n + \mathbf{P}_d^n \max_d \mathbf{V}^{n,n-1}) \quad (3-4)$$

With lagged information:

$$\mathbf{V}^{n,n} = \max_d \mathbf{P}_d^n (\mathbf{Q}_d^n + \max_d \mathbf{P}_d^n \mathbf{V}^{n,n-1}) \quad (3-5)$$

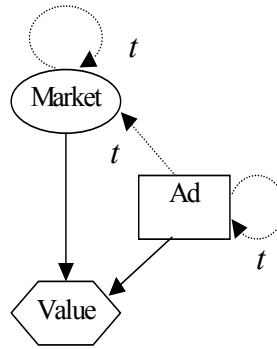
With no information:

$$\mathbf{V}^n = \max_d \mathbf{b}^n \cdot (\mathbf{Q}_d^n + \mathbf{P}_d^n \mathbf{V}^{n,n-1}) \quad (3-6)$$

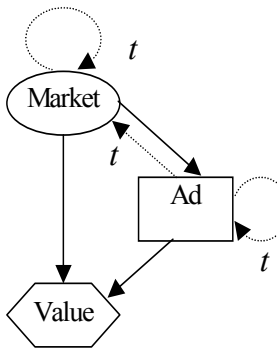
Where \mathbf{b}^n is a row vector which represents the belief state in each stage.

Due to the invariant structure of most dynamic problems, many researchers represent the system dynamics by a two-stage DBN/DID. It is also convenient to draw condensed forms to represent multi-stage dynamic decision problems, like the ones shown in Figure 3-4. Dotted lines with a 't' near them are temporal arcs with a time lag of t .

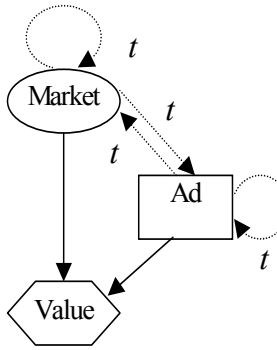
When the time horizon is infinite, a discount factor $\beta \in [0,1]$ is applied in the value function updating procedure to ensure that the expected utility/value is meaningful.



(a)



(b)



(c)

Figure 3-4: Condensed form of the three scenarios

(a) Without information, (b) with full information and (c) with information of one stage delay.

3.1.2 Order the Information Values

Let's denote the total value of the above three scenarios, without information, with full information and with lagged information as V^0 , V^1 and V^2 respectively, it can be proved that $V^0 \leq V^1 \leq V^2$.

Let $M = \{C, D, V, E, T\}$ be a dynamic influence diagram where C is chance node set, D is decision node set, V is value node set, E is the set of directed arcs and T is decision stages. The value of information for observing the i th stage chance node $X_i \in C$ before the j th stage action $D_j \in D$ is denote by $VOI_{D_j}(X_i)$.

Theorem 3-1: In a dynamic influence diagram M , when $j > k \in T$, we have $VOI_{D_j}(X_i) \leq VOI_{D_k}(X_i)$.

Proof:

Jensen's inequality:

Let $V: R^n \rightarrow R$ be a convex function. Let $x_1, \dots, x_n \in R^n$ and let $a_1, \dots, a_n \in [0, 1]$, such

that $\sum_{i=1}^n a_i = 1$. Then:

$$V\left(\sum_{i=1}^n a_i x_i\right) \leq \sum_{i=1}^n a_i V(x_i)$$

First let's take look at the case when there are only random variables, and no other decisions between the two decision nodes D_k and D_j . The variables are chronically ordered as $\{D_k, X, D_j, X_i\}$. Hence when knowing X_i before D_j ,

$$\begin{aligned} V &= \max_{D_k} \sum P(X|D_k) \sum P(X_i|X, D_k) \max_{D_j} V(H|X, X_i, D_j, D_k) \\ &= \max_{D_k} \sum P(X_i|D_k) \sum P(X|X_i, D_k) \max_{D_j} V(H|X, X_i, D_j, D_k); \end{aligned} \quad (3-7)$$

Knowing X_i before D_k and D_j ,

$$V = \sum P(X_i) \max_{D_k} \sum P(X|X_i, D_k) \max_{D_j} V(H|X, X_i, D_j, D_k) \quad (3-8)$$

Since maximization is a convex function, based on the Jensen's inequality, equation (4-8) will:

$$\geq \max_{D_k} \sum P(X_i|D_k) \sum P(X|X_i, D_k) \max_{D_j} V(H|X, X_i, D_j, D_k) \quad (3-9)$$

Without knowing X_i :

$V = \max_{D_k} \sum P(X|D_k) \max_{D_j} \sum P(X_i|X, D_j, D_k) V(H|X, X_i, D_j, D_k)$ is the same as (3-9), so $VOI_{D_k}(X_i) \geq VOI_{D_j}(X_i)$.

If there are other decision nodes between D_k and D_j , e.g. D_l , we can have

$$VOI_{D_k}(X_i) \geq VOI_{D_l}(X_i)$$

and $VOI_{D_l}(X_i) \geq VOI_{D_j}(X_i)$, thus $VOI_{D_k}(X_i) \geq VOI_{D_j}(X_i)$.

If the two decisions belong to adjacent decision stages, i.e., they have different value successors V_k, V_j , then the value of knowing X_i before D_j is:

$$\begin{aligned} V_k + V_j &= \max_{D_k} \sum P(X|D_k) \sum P(X_i|X, D_k) [V(H|X, X_i, D_k) \\ &\quad + \max_{D_j} V(H|X', X_i, D_j, D_k)] \\ &= \max_{D_k} \sum P(X_i|D_k) \sum P(X|X_i, D_k) [V(H|X, X_i, D_k) \\ &\quad + \max_{D_j} V(H|X', X_i, D_j, D_k)] \end{aligned}$$

Knowing X_i before D_k and D_j ,

$$\begin{aligned} V_k + V_j &= \sum P(X_i) \max_{D_k} \sum P(X|X_i, D_k) [V(H|X, X_i, D_k) \\ &\quad + \max_{D_j} V(H|X', X_i, D_j, D_k)] \\ &\geq \max_{D_k} \sum P(X_i|D_k) \sum P(X|X_i, D_k) [V(H|X, X_i, D_k) \\ &\quad + \max_{D_j} V(H|X', X_i, D_j, D_k)] \end{aligned}$$

If there are other decision stages between D_k and D_j , e.g. D_l , we can have $VOI_{D_k}(X_i) \geq VOI_{D_l}(X_i)$ and $VOI_{D_l}(X_i) \geq VOI_{D_j}(X_i)$, thus $VOI_{D_k}(X_i) \geq VOI_{D_j}(X_i)$. \square

Hence, it is always preferable to observe the system state earlier than later, and later than never if costs are not concerned. The difference of value between observing X_i before D_j and D_k can be called the VOI of temporal delay. The above inequality is meaningful because if the VOI ordering were in contrary case, there would be no need to concern the opportunity cost occurred in information gathering period. Since observing earlier is better than later, the problem becomes a tradeoff when facing the opportunity cost.

3.1.3 EVPI in Partially Observable Models

The influence diagram representation enjoys the freedom of illustrating all sorts of dynamic decision processes. In contrast to this, in the planning literature, the partially observable Markov decision processes (POMDP) are quite different from those for fully observable MDP and SMDP in representations and solution methods. As has been mentioned in the previous chapter, the partially observable dynamic problems are much harder to deal with. Studying the difficulties that lie in the POMDP and the influence of those hard parts in the value of information helps developing better methods for VOI computation.

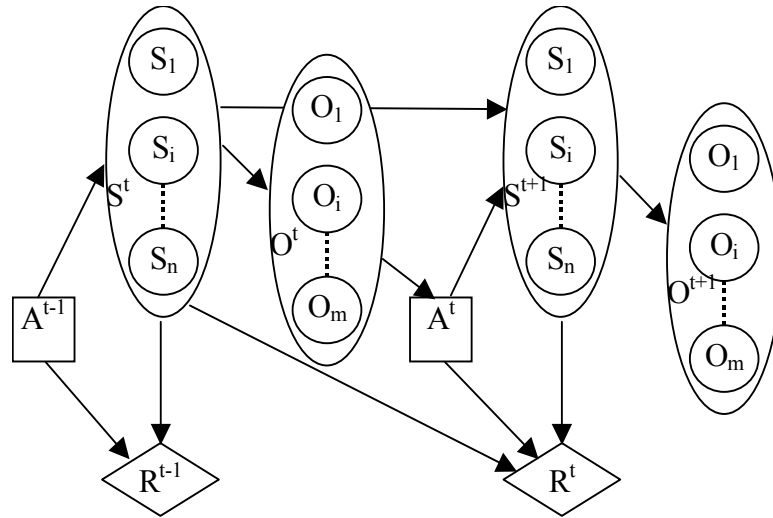


Figure 3-5: Two-stage DID for a typical partially observable problem

To facilitate a clear explanation of this set of problems, a partially observable dynamic decision problem is shown in Figure 3-5 in dynamic influence diagram format. In accordance with traditional POMDP form, the variables are arranged in such way: the decision maker takes an action A to affect the system state S and change its status; at the same time a cost is induced and a reward received, combined to the value node R ; the decision maker then makes an observation O to detect the real state of the system, then decides the action to be taken in the next stage.

As shown in Figure 3-5, typical POMDP assumes that the decision maker acts according to the observation of the previous stage. Described in typical dynamic updating formulas with the belief state $b(s)$ together with the value function, the value update for stage t is:

$$V^t = \max_{a^t} [\sum_{s^t \in S} \sum_{s^{t+1} \in S} b(s^t) P(s^{t+1} | s^t, a^t) R(a^t, s^t, s^{t+1}) + \beta \sum_{s^t \in S} \sum_{s^{t+1} \in S} P(s^{t+1} | s^t, a^t) b(s^t) \sum_{o^{t+1} \in O} P(o^{t+1} | s^{t+1}, a^t) V^{t+1}] \quad (3-10)$$

Here $\beta \in (0, 1)$ is a discounting factor which makes sure the total expected value converges. When in the last stage T , V^T is $\max_{a^T} \sum_{s^T \in \mathcal{S}} \sum_{s^{T-1} \in \mathcal{S}} b(s^T) R(a^T, s^{T-1}, s^T)$.

Consider the case of observing one state variable S_i^t among the state variable set \mathbf{S} only in stage t . S_i^t is now removed from the state set \mathbf{S} and then placed in the observation set \mathbf{O} . The state set and observation set are now denoted as $\hat{\mathbf{S}}$ and $\hat{\mathbf{O}}$, shown in Figure 3-6, only in this stage. Observing S_i^t before A^{t+1} will change (3-10) to:

$$\begin{aligned} \hat{V}^t(s_i^t=j) &= \max_{a^t} [\sum_{s^t \in \hat{\mathcal{S}}} \sum_{s^{t+1} \in \mathcal{S}} \hat{b}(s^t) P(s^{t+1} | s^t, s_i^t=j, a^t) \\ &\quad R(a^t, s^t, s_i^t=j, s^{t+1}) + \beta \sum_{s^t \in \hat{\mathcal{S}}} \sum_{s^{t+1} \in \mathcal{S}} P(s^{t+1} | s^t, s_i^t=j, a^t) \hat{b}(s^t) \sum_{o^{t+1} \in \hat{\mathcal{O}}} \\ &\quad P(o^{t+1} | a^t, s^{t+1}) V^{t+1}] \\ \hat{V}^{t-1} &= \max_{a^{t-1}} [\sum_{s^{t-1} \in \mathcal{S}} \sum_{s^t \in \hat{\mathcal{S}}} b(s^{t-1}) P(s^t | s^{t-1}, a^{t-1}) R(a^{t-1}, s^t, s^{t-1}) \\ &\quad + \beta \sum_{s^{t-1} \in \mathcal{S}} \sum_{s^t \in \hat{\mathcal{S}}} P(s^t | s^{t-1}, a^{t-1}) b(s^{t-1}) \sum_{o^t \in \hat{\mathcal{O}}} P(o^t | s^t, a^{t-1}) V^t] \end{aligned} \tag{3-11}$$

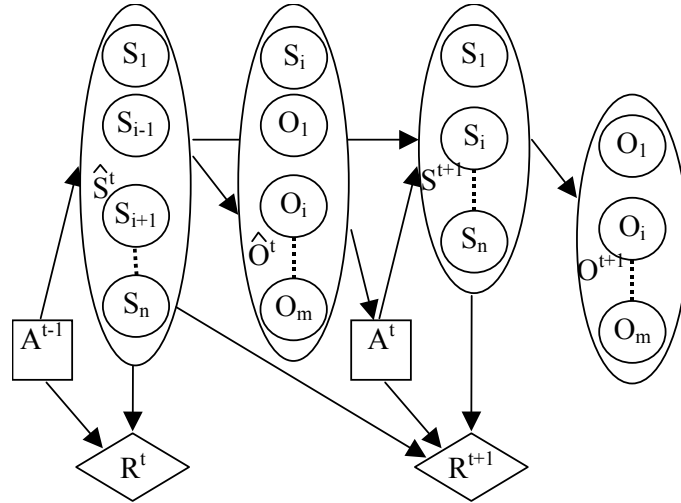


Figure 3-6: Decision model with S_i observed before A

Note that the change lies in two dynamic updates since both the \mathbf{S} set and \mathbf{O} set at stage t are changed. Note that states and observations of the same stage are

incorporated in different dynamic update steps. The posterior belief states of the system after observation $\hat{\mathbf{O}}$ and \mathbf{A}^t are now:

$$\begin{aligned}\hat{b}_o^a(s^{t+1}) &= \frac{\sum_{s^t \in \hat{S}} P(o^{t+1} | s^{t+1}, a^t) P(s^{t+1} | s^t, s_i^t = j, a^t) b(s^t)}{\sum_{s^t \in \hat{S}} \sum_{s^{t+1} \in S} P(o^{t+1} | s^{t+1}, a^t) P(s^{t+1} | s^t, s_i^t = j, a^t) b(s^t)} \\ \hat{b}_o^a(s^t) &= \frac{\sum_{s^{t-1} \in S} P(o^t | s^t, a^{t-1}) P(s^t | s^{t-1}, a^{t-1}) b(s^{t-1})}{\sum_{s^{t-1} \in S} \sum_{s^t \in \hat{S}} P(o^t | s^t, a^{t-1}) P(s^t | s^{t-1}, a^{t-1}) b(s^{t-1})}\end{aligned}\quad (3-12)$$

The changed value functions will have impact on stages less than t , thus the Expected Value of Perfect Information (EVPI) should be the difference between total expected value of the two scenarios:

$$EVPI = \sum_i^{t+1} \hat{V}^i - \sum_i^{t+1} V^i$$

If state variables in earlier stages are observed, more difference could be incurred. The EVPI is computed by substituting \mathbf{S} and \mathbf{O} with $\hat{\mathbf{S}}$ and $\hat{\mathbf{O}}$ in dynamic programming steps of those stages.

3.1.4 Bounds of EVPI in Partially Observable Models

In a single stage decision problem, denote the believed distribution of the chance variable S under consideration as $P(s)$, the EVPI of variable S before decision A is:

$$EVPI = \sum_s P(s) \max_a V(a, s) - \max_a \sum_s P(s) V(a, s)$$

Since $\max_a V(a, s)$ is invariant to action a , $\sum_s P(s) \max_a V(a, s) = \min_a \sum_s P(s) \max_a V(a, s)$, so

$$\begin{aligned}EVPI &= \min_a \sum_s P(s) \max_a V(a, s) - \max_a \sum_s P(s) V(a, s) \\ &= \min_a \sum_s P(s) [\max_a V(a, s) - V(a, s)]\end{aligned}$$

Define regrets $r(a, s) = \max_a V(a, s) - V(a, s) \geq 0$ as an $n \times m$ matrix of n states and m actions, $\mathbf{P}(s)$ as the belief state matrix,

$$\begin{aligned} \text{EVPI} &= \min_{a_i} \sum_{s_j \in S} P(s_j) \cdot r(a_i, s_j) \\ &= \min_a \mathbf{P}(s) \cdot \mathbf{r}(a, s) \end{aligned} \quad (3-13)$$

If there is no prior information on the distribution of the state variable, the EVPI is bounded by the region $[\min_b \text{EVPI}, \max_b \text{EVPI}]$. The lower bound of EVPI is nonnegative. Zero EVPI is obtained when there is a dominating action, or when the regrets vector for i th action is perpendicular to the belief vector. The max EVPI given a reward function is determined by an optimization problem shown bellow:

$$\begin{aligned} &\text{Max } V \\ &s. t. \\ &\sum_{s_j \in S} P(s_j) \cdot [\max_{a_i} V(a_i, s_j) - V(a_i, s_j)] \geq V, \forall a_i \in A. \\ &\sum_{s_j \in S} P(s_j) = 1, \forall s, P(s_j) \geq 0 \end{aligned} \quad (3-14)$$

When the distribution of the state variable is discrete, say, multinomial, this optimization is a linear program.

In each time stage, the value functions corresponding to different actions are constituted by a set of γ vectors that are dominating in different regions of the belief state. Hence the EVPI can be expressed as:

$$\begin{aligned} \text{EVPI} &= \min_a [\hat{\mathbf{b}}(s) \cdot \gamma_p - \mathbf{b}(s) \cdot \gamma_i] \\ \gamma_p^t &= R(s^t, a^t) + \beta \sum_{o^t \in \hat{O}} \sum_{s^{t+1} \in S} P(o^t, s^{t+1} | s^t, a^t) \gamma_o^{t+1} \\ \gamma_i^t &= R(s^t, a^t) + \beta \sum_{o^t \in O} \sum_{s^{t+1} \in S} P(o^t, s^{t+1} | s^t, a^t) \gamma_o^{t+1} \end{aligned} \quad (3-15)$$

where $\mathbf{b}(s)$ is the belief state vector of s , $\hat{\mathbf{b}}$ denotes perfect information case, $\gamma_i \in \mathbf{F}$ is the γ vector for the i th action, and γ_p is the γ vector that corresponds to the perfect information case.

The first constraint of the optimization problem (3-14) can also be rewritten as $[\hat{\mathbf{b}}(s) \cdot \boldsymbol{\gamma}_p - \mathbf{b}(s) \cdot \boldsymbol{\gamma}_i] \geq V$.

Figure 3-7 shows the maximum EVPI given the piecewise linear value functions. In this stage, the four $\boldsymbol{\gamma}$ vectors $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3$ and $\boldsymbol{\gamma}_4$ form the piece-wise linear and convex (PWLC) value function. The EVPI of S before A , is the distance represented by the double-arrow line corresponding to b and \hat{b} . The maximum EVPI is obtained at b^* and \hat{b}^* . Note that \hat{b} is one dimension less than b since it has fixed $S_i = j$, but it is same to b after smoothing over the distribution of the observed variable.

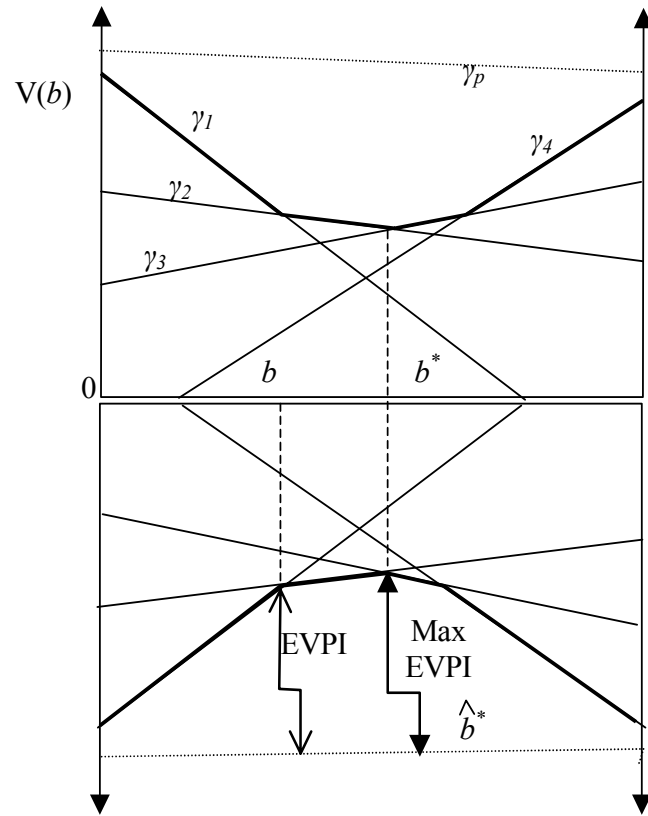


Figure 3-7: Value function and the EVPI over a binary state b

In a dynamic decision model with the state variables not directly observable, each action and observation in every stage corresponds to a γ vector. In the worst case, none of these γ vectors are dominated by the others, hence the number of γ vectors is exponential in the number of decisions and observations. Consequently, the optimization shown in (3-14) will be intractable. Nevertheless, for many practical cases the size of γ vectors $|\Gamma|$ in each stage is finite, so that given a set of γ vectors the optimization (3-14) is solvable.

Let $EVPI_t^*(S) = \max_b EVPI_t$ denotes the largest EVPI can be induced in stage t . The gain in total expected value for a POMDP given perfect information in stage t is then bounded by $[0, \beta^t EVPI_t^*(S)]$.

If the case with perfect information is taken as an approximation of the state variable in POMDP, the EVPI can be viewed as the ‘error’ incurred in a certain stage. Keep observing variable S in each time stage as an approximating scheme for the POMDP is similar to which has been discussed in (Poupart and Boutilier, 2000). In that paper, when the POMDP is approximated by a projection scheme (e.g., a simpler structure which involves more conditional independencies than the real belief state), there might be a switch in selection of γ vectors from vectors chosen for actual belief states. An optimization problem is constructed to find this switch set of γ vectors and an error bound is calculated thereafter. Compare to other projection schemes, the perfect information approximation is simpler in that the γ vectors for this scheme are fixed with respect to a given reward structure. For an n -stage system, the upper bound of error is $\sum_{t=1}^n \beta^t \cdot \text{EVPI}_t^*(S)$. Hence the error is bounded by $[0, (\max_t \text{EVPI}_t^*(S)) \cdot \frac{1-\beta^n}{1-\beta}]$.

3.2 Value of clairvoyance for the intervened variables

Some chance variables have decision nodes as their parents. These chance variables are called decision-intervened variables. In traditional EVPI computation, the decision-intervened chance variables are not considered. Many computational tools in decision analysis also exclude such computation of VOI for intervened variables.

Disregarding this topic is quite natural since adding an arc from the decision-intervened variable to its parent decision node forms a directed loop, which violates the directed acyclic assumption in graphical decision models. It could also be logically contradictory in real situations. For example, the perfect information of a future uncertainty X says X is exactly in state x . With this perfect information, the decision

maker can adjust his/her strategy to make larger profit. Yet since this X is intervened by the decision taken, it might not be in state x any more, which indicates the original 'perfect information' of $X=x$ is not accurate at all.

Though this situation will form a paradox conceptually, determining the value of information about a decision-intervened variable is not totally meaningless. This kind of information can be used to direct the current actions, increase the profits and avoid great costs. For example, in the condensed DID shown in Figure 3-8 (a), knowing the exact consequence of the surgery, whether it will be successful or failed, whether the successful surgery will improve the *patient's health* of the next stage before the surgery is taken in the current stage, helps the surgeon decide which operation to take or if a surgery should be conducted. If the surgery is going to fail, the doctor might carry another therapy to avoid any harm. Hence, the information on decision-intervened variable provides us an upper limit that the benefit that our current decision can give.

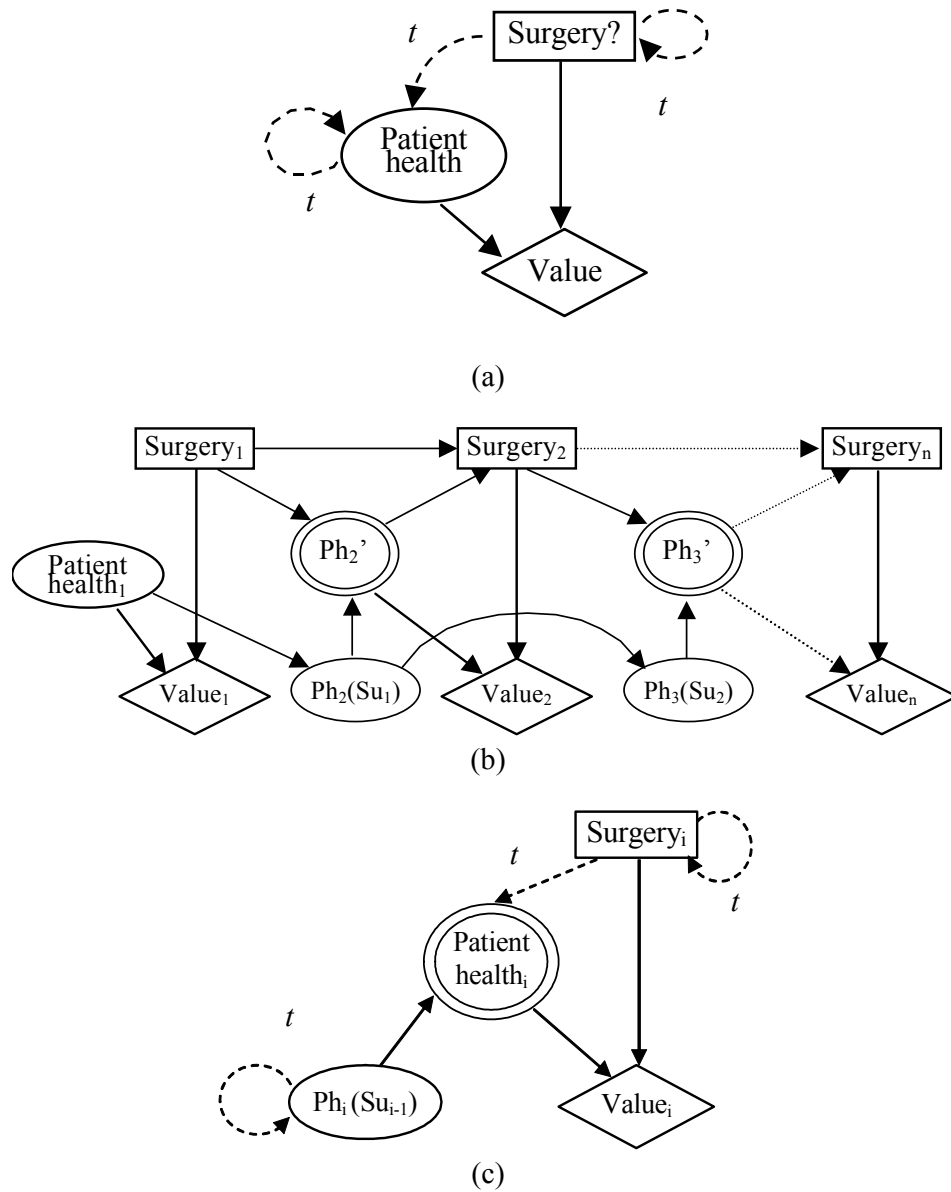


Figure 3-8: DID for calculating VOI of intervened nodes

(a) Condensed form (b) Unrolled canonical DID with mapping variables added

(c) Condensed canonical form

In the EVPI computation of such cases we cannot simply add an arc from the decision to the chance node under discussion, since it forms a directed cycle in the original influence diagram, which means a deadlock in computation. Mapping variables, e.g.

those in double cycles in Figure 3-8 need to be added in order to take over the influence of decision and ‘set free’ the original decision-intervened variable.

Canonical form is developed in (Howard, 1990; Heckerman, 1995) to address problems with computing the informational value in influence diagrams. A graphical decision model is in canonical form with respect to decision and chance nodes if no chance nodes are descendants of decision nodes, i.e., no chance nodes are decision-intervened. Hence in a canonical influence diagram all the chance nodes that are descendants of one or more decision nodes should be deterministic nodes.

It is now suitable for calculating VOI for decision descendants and for counterfactual reasoning using this canonical influence diagram.

Heckerman and Shachter (1995) have introduced an algorithm for constructing canonical form for generic influence diagrams:

Given a decision problem described by chance variables set U and decision variables set D :

1. Add a node to the diagram corresponding to each variable in $U \cup D$
2. Order the variables x_1, \dots, x_n in U so that the variables unresponsive to D come first.
3. For each variable $x_i \in U$ that is responsive to D ,
 - a. Add a causal-mapping-variable chance node $x_i(C_i)$ to the diagram, where $C_i \subseteq D \cup \{x_1, \dots, x_{i-1}\}$
 - b. Make x_i a deterministic node with parents C_i and $x_i(C_i)$
4. Assess independencies among the variables that are unresponsive to D

Here, a chance variable X unresponsive to decision D means X has the same outcome no matter what D is taken, i.e., D has no influence on X . For a formal definition, please refer to (Heckerman and Shachter, 1995).

We can construct the canonical form for the dynamic influence diagram shown in Figure 3-8 (a) as the one in 3-8 (b).

If taking the temporal arcs into account in the process of finding unresponsive nodes to decisions and adding mapping variables, a condensed canonical form influence diagram shown in Figure 3-8 (c) can be constructed from its condensed form directly.

In a more complex example shown in Figure 3-9, SOR_i , SOM_i and S_i are not responsive to the decision D_i . However, S_{i+1} is responsive to D_i through TR_i , and thus SOR_{i+1} and SOM_{i+1} are also responsive to D_i . So we should construct mapping variables both for TR_i and S_{i+1} .

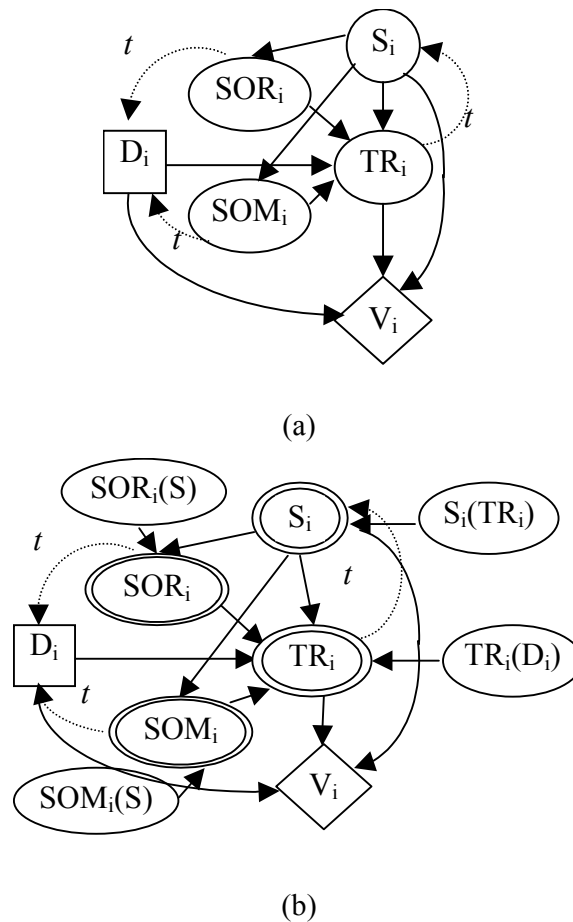


Figure 3-9: More complex example

(a): Original ID; (b): ID in canonical form

If the influence of a decision is spread over all the chance nodes then we have to add many mapping variables. However this is not necessary for the VOI computation of certain chance node before some particular decisions. Only the intervening decisions and their descendents need to be considered. The more stages involved in this conversion to canonical form, the more complex the influence diagram will be. This resulting complex DID keeps accordance with our experience that the farther we predict, the more uncertainty is encountered and hence the harder the conclusions can be made.

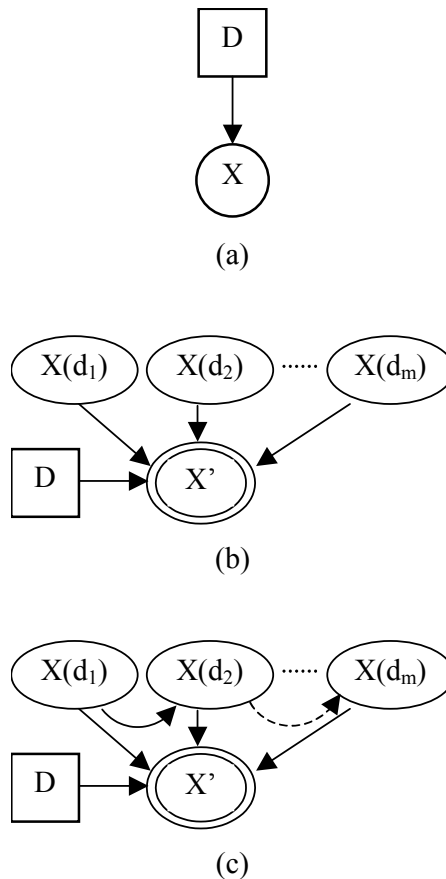


Figure 3-10: Convert ID (a) to canonical form (b), (c)

Suppose the decision D_i has m instances, and the chance variable X_i has n states, then $X_i(D)$ has n^m instances, while X_i' has n outcomes. In the original influence diagram the random variable X has only $m \cdot n$ instances, so in theory it is necessary to assign more probabilities for the equivalent conversion. For simplicity we will assume the independencies between variables $X(d_i)$ given a specific decision d_i , like the graph shown in Figure 3-10 (b), where there are m random variables given m decisions, and no arcs between $X(d_i)$ assuming independency. Thus we have $P((x_k, d_i), (x_l, d_j)) = P(x_k, d_i) \cdot P(x_l, d_j)$, and the number of probabilities that need assigning reduced from n^m to $m \cdot n$. Other probabilities can be derived from these $m \cdot n$ outcomes.

The above independence assumption will simplify the conversion and the computation for VOI. However, if the random variables $X(d_i)$ are dependent, i.e., nested as shown in Figure 3-10 (c), it is necessary to reassess the probabilities for the outcomes of $X(d_i)$ to make sure the conversion is equivalent. Hence the actual VOI for knowing X before D will be different from the VOI computed in simplified scheme.

For cases of binary states and binary decisions, we find that the difference between the VOI calculated in the two scenarios will be dependent on both the covariance and the value function for different decisions. When one state is dominant to the other and there is a positive correlation, the VOI calculated in simplified case will be higher than the actual case, thus the independence assumption boasts the value of information of observing the variable X before D . The range of this error caused by independence assumption is $V_s' - V_r' \in [-|R|, |R|]$, where R is the difference between the second and the third largest value. (See Appendix B).

Moreover, if more causes for different alternatives are correlated, we are unable to tell if the independency assumption will increase the VOI calculated or not. If the problem is extended to multi-state and multi-decision case it will become more complicated and harder to estimate.

The above analysis shows that we have to be careful when using the independency assumption. However, when there is no specified information about the dependencies between mapping variables given different alternatives, assuming independency will simplify the computation.

3.3 Summary

In this chapter the properties of Expected Value of Perfect Information are studied under various dynamic decision environments, e.g., dynamic influence diagrams where the state variables are both observable and partially observable. Boundaries for EVPI in partially observable dynamic models are given with an analysis of complexity issues. Further more, the Value of Information computation for decision-intervened chance variables are also discussed.

The contents in this chapter serve as a basis for the discussion in the later parts of this dissertation.

4 Exact VOI Computation in Dynamic Systems

As reviewed in Chapter 2, there are many ways to compute the EVPI in general influence diagrams. Most of these methods can be directly adopted to compute EVPIs in dynamic decision models formed as dynamic influence diagrams (DIDs), since DIDs are special cases of general influence diagrams.

However, the direct application of these methods might not be very efficient for they do not take advantage of properties that are characteristic in dynamic decision models. In this chapter, efficient VOI computation that takes account of the dynamic structure of the system is discussed.

4.1 Temporally Invariant Junction Tree for DIDs

Kjærulff (1992) proposed the Dynamic Expansion and Reduction (DER) method to perform exact inference in Dynamic Bayesian Networks (DBNs) by adding new time slices and deleting old ones dynamically. However, in many practical cases the system structure in every time slice is stationary or near stationary. Xiang (1999) has mentioned that pre-compiling some slice representation of a stationary DBN could support more efficient exact inference. In this paper, a sub-junction tree template is constructed from the original DBN by first identifying a subnet, $S_{i-1} \cup N_i \cup S_i$, where S_{i-1}

and S_i are minimal separators of the DBN and N_i is the part between them, as shown in Figure 4-1.

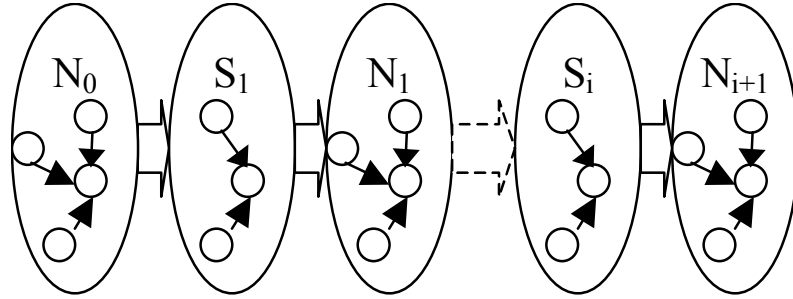


Figure 4-1: Partition of a DBN

In order to calculate the value of information in dynamic systems, it is necessary to deal with decisions and values together with probabilistic nodes as in DBNs. Denote the decision node set as D , the value node set as V . Similar to the DBN partitions, for all the chance nodes we can partition them into a collection of disjoint sets $W_0, W_1, \dots, W_k, \dots, W_n$ with the ordering shown in (2-4) of Chapter 2. The partition is illustrated in the following Figure 4-2. Denote I_i as all the information known before the decision D_i , $I_i = I_{i-1} \cup D_{i-1} \cup W_i$.

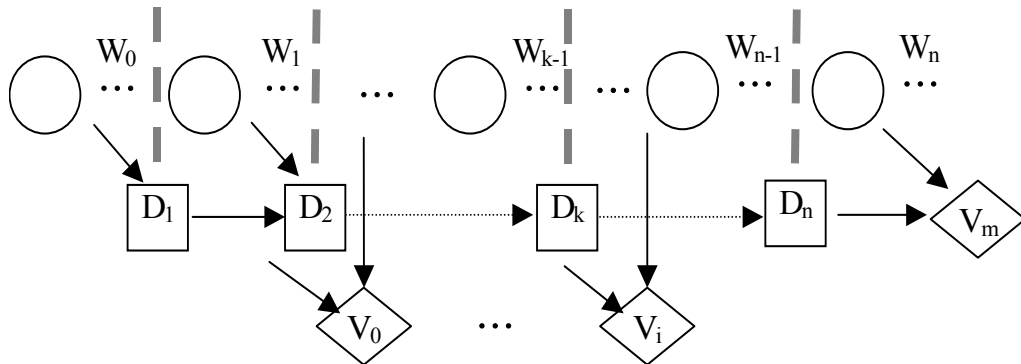


Figure 4-2: Partition of Influence Diagram

On the other hand, Shachter (1999) has presented an algorithm (here we call it the Requisite Observation algorithm) based on the Bayes-Ball algorithm (Shachter, 1998) to determine the requisite observations of each decision and then convert the influence diagram into a belief network in time linear to the graph size.

In a structured belief network $M = (N, A, F)$ which has nodes N , directed arcs A and a subset F of the nodes that are deterministically (functionally) related to their parents, not all variables are relevant to certain queries. According to node relevance, (Shachter, 1998) has given the following definitions:

Def. 4.1 Irrelevant Nodes

The irrelevant nodes for uncertain variable X_J given X_K , denoted $X_{N_i}(J|K)$, are those nodes which are conditionally irrelevant to X_J given X_K ,

$$N_i(J|K) = \{i \in N : X_i \perp_M X_J | X_K\}.$$

Def. 4.2 Requisite Probability Nodes

The requisite probability nodes for J given K , denoted $N_p(J|K)$, are those nodes for which conditional probability distributions (and possible states) might be needed to compute $P\{X_J | X_K\}$.

Def. 4.3 Requisite Observation Nodes

The requisite observations for J given K , $N_e(J|K) \in K$, are those observed nodes for which observations (and hence the possible states which might be observed) might be needed to compute $P\{X_J | X_K\}$.

Both the Requisite Observation algorithm and the Bayes-Ball algorithm are based on the equivalency between the conditional independency and a graph property called *d-separation*.

Def. 4.4 *D-separation* (Pearl 1988, Pearl et al 1990):

If X , Y and Z are three disjoint subsets of nodes in a directed acyclic graph; then Z is said to *d-separate* X and Y , if and only if along every chain from each node in X to each node in Y there is an intermediate node A such that either:

1. A is a head-to-head node in the path, and neither A nor its descendants are in Z ,
2. A is not a head-to-head node in the path and A is in Z .

Here ‘head-to-head’ means the node has more than two arcs pointing to it, so that the arrow heads meet on the node.

The Bayes-ball algorithm (Shachter, 1998) is a simple and efficient algorithm to compute irrelevant and requisite sets for inference and decision problems.

In $M = (N, A, F)$ with respect to the expression $P\{X_J | X_K\}$:

1. *Initialize all nodes as neither visited, nor marked on the top, nor marked on the bottom.*
2. *Create a schedule of nodes to be visited, initialized with each node in J to be visited as if from one of its children.*
3. *While there are still nodes scheduled to be visited:*
 - a. *Pick any node j scheduled to be visited and remove it from the schedule. Either j was scheduled for a visit from a parent, a visit from a child, or both.*
 - b. *Mark j as visited.*
 - c. *If $j \notin K$ and the visit to j is from a child:*

- i. *if the top of j is not marked, then mark its top and schedule each of its parents to be visited;*
- ii. *if $j \notin F$ and the bottom of j is not marked, then mark its bottom and schedule each of its children to be visited.*
- d. *If the visit to j is from a parent:*
 - i. *If $j \in K$ and the top of j is not marked, then mark its top and schedule each of its parents to be visited;*
 - ii. *if $j \notin K$ and the bottom of j is not marked, then mark its bottom and schedule each of its children to be visited.*
- 4. *The irrelevant nodes, $N_i(J|K)$, are those nodes not marked on the bottom.*
- 5. *The requisite probability nodes, $N_p(J|K)$, are those nodes marked on top.*
- 6. *The requisite observation nodes, $N_e(J|K)$, are those nodes in K marked as visited.*

The Requisite Observation Algorithm (Shachter, 1999), which is closely related to VOI computation, is listed bellow:

“Visit each decision D_i in reverse chronological order, $i = m, \dots, 1$. Let V_i be the set of value descendants of D in the current diagram. Run the Bayes-Ball algorithm on V_i given D_i and I_i , the variables observed before D_i is chosen, and let R_i be the requisite observations (not including D_i). Replace D_i by a chance node “policy” with R_i as parents and proceed to the next earlier decision.”

The Requisite Observation algorithm is developed in a decision system with separable value nodes to prune the set of irrelevant information predecessors for each of the decisions in general influence diagrams, hence it is appropriate to apply the algorithm

in dynamic influence diagrams. Yet in order to take advantage of the repeatability of many practical dynamic systems, it still needs to be revised.

4.2 The Problem

Our objective is trying to apply efficient clustering method in a DID, making use of the stationary or near stationary features of the system. It is desired to build a template junction tree and then evolve the system dynamically to compute the value of information. The procedure is: first identify a subnet to build template junction tree; then apply the Requisite Observation algorithm to convert the decision problem into a probabilistic network; construct template junction tree thereafter; evolved into the next time stage by updating the current belief and finally reuse the junction tree to calculate the value of information.

However, DBNs can be divided into subnets based on the forward interface or backward interface (Xiang, 1999); while it is not clear whether this decomposition is valid in DIDs. The key problem is: After converting the DID into a DBN, can this DBN be divided into time-invariant template subnets?

Take the DID shown in Figure 4-3 as an example. First, run the Decision Bayes-Ball algorithm on V_n given (D_n, I_n) , $I_n = \{b_0, D_0, \dots, b_{n-1}, D_{n-1}, b_n\}$, the requisite observation set of D_n , $R_n = \{b_n, D_{n-1}\}$. Iterating backwards yields a requisite information set $R_i = \{b_0, \dots, b_i, D_{i-1}\}$. Hence arcs must be added from these requisite observations to the decision D_i , as shown in Figure 4-4. Thus the resulting belief network is much more complex, and is not Markovian for decisions. This is because the information predecessor b_i has a parent a_i which is also included in the forward interface. The Bayes-Ball passed from b_{i+1} bounces back to b_i from a_i , and bounces back to b_i 's other

parents that are in the previous stage. According to the repeating structural character of the temporally invariant DID, either the Bayes-Ball will stop at the previous stage; or else it will pass to the very first stage, include all observations and cause the complex structure like what is shown in Figure 4-4.

Hence before dividing the whole DID into subnets, a check in the structure to test whether the DID is decomposable is needed. If the DID cannot be decomposed to time-invariant subnets, it is not likely that we can save much computational time and space by constructing template and rolling back thereafter.

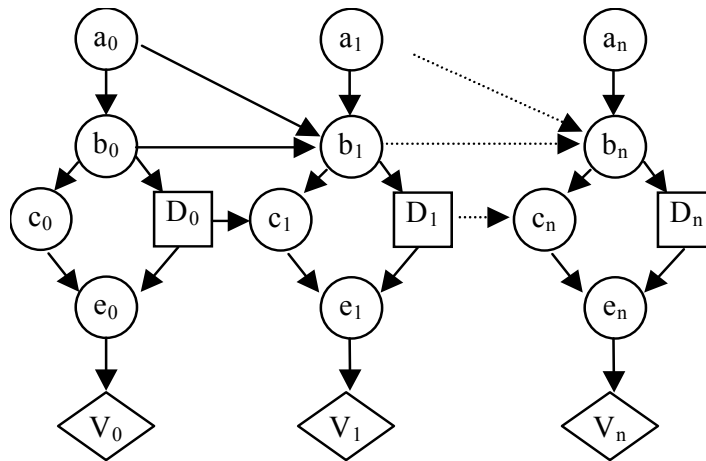


Figure 4-3: An example of DID

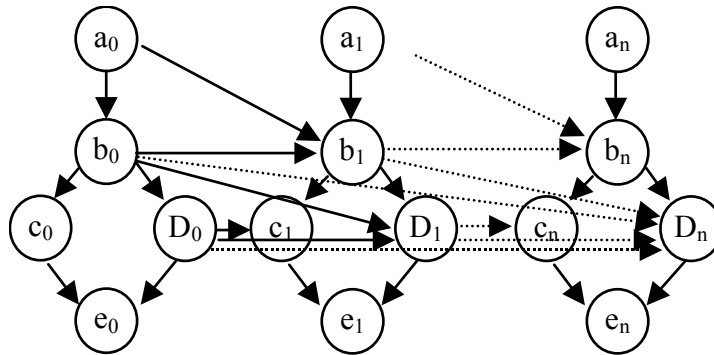


Figure 4-4: Resulting DBN for the example above

If a node can carry the Bayes-Ball to the previous stages without being blocked, thus spread the requisite observations to those stages, we call it a spreading variable. When we run the Bayes-Ball algorithm in the DID we can find such spreading variables:

Def 4.5 *Spreading variables:*

In a DID $M = \{X, D, E, V\}$, consider a node $X_i \notin I_i$, if X_i is **not** d -separated from X_{i-1} by the set I_i , then X_i is a spreading variable.

Start from the last stage n , run the Bayes-Ball algorithm on $(V_n | D_n, I_n)$, the requisite observation R_n should lie in $W_n \cup D_n$.

For the $(n-1)$ th stage, run the Bayes-Ball algorithm on $(V_{n-1} \cup R_n | D_{n-1}, I_{n-1})$. A ball from R_n will pass to X_n , if the network is connected, which is a trivial constraint. If X_{n-1} is not d -separated from X_n by I_n , then the ball can be passed to X_{n-1} through some active path without being blocked by I_{n-1} , and pass to W_{n-1} . Reason by analogy, the ball will be passed from X_i to X_{i-1} and then to the very first stage through the active paths. Thus the requisite observation set will include variables in all the previous stages.

The spreading variables can be detected in undirected graph as well. First identify the minimal ancestral sub-graph for $W_{i-1} \cup D_{i-1}$ and $W_i \cup D_i$, denote as AS . In a DID without barren nodes, either the random variable X_i is in the ancestral sub-graph of $W_i \cup D_i$, or $W_i \cup D_i$ is in the ancestral sub-graph of X_i . If the former case holds, i.e., X_{i-1} and X_i are included in this sub-graph AS , moralize the sub-graph and cut I_i . If there are links found between X_{i-1} and X_i , X is a spreading variable.

Otherwise X_i is a descendent of $W_i \cup D_i$, then add minimal ancestral sub-graph for X_{i-1} and X_i , moralize and judge.

If there are any spreading variables in the DID, it is hard to take the advantage of the invariant features of the model. Otherwise, all the requisite observations are located within the subnet N_i , which is from $W_{i-1} \cup D_{i-1}$ to $W_i \cup D_i$. We can thus identify the requisite observation of each decision in such subnet for the value node and the requisite observations obtained in the later subnet, without running the Requisite Observation Algorithm over the entire DID.

After requisite observations are identified, this subnet might differ from the original sub-ID on structure with arcs from requisite observations added to the decision and arcs from non-requisite observations removed, but it inherits the temporal invariant feature. The forward interface based on this subnet will also be a self-sufficient separator and the sub-junction tree constructed be properly constructed with a root cluster that has no children. If the DID model contains many arcs from irrelevant nodes to the decision, running the Bayes-ball will reduce complexity significantly. If the DBN converted from original DID has a small forward interface set, this decomposition is quite useful and expressive.

For a properly constructed junction tree the variables weakly inward of the decision are observed before the decision. Hence after the sub-junction tree is constructed, it can also be used to identify the variables which have zero VOI for a particular decision. This helps screen out unimportant chance variables for VOI computation.

Since I_i is all the information known before the decision D_i , $I_i = I_{i-1} \cup D_{i-1} \cup W_i$, the condition for the absence of spreading variables, $(X_{i+1} \perp X_i | I_i)$, means the variable X_{i+1} in the current stage is conditionally independent of the variable X_i in previous stage given all the information available at present. As an example, the common Markov Decision Processes have such a feature; hence they can be divided and solved

iteratively. However, such decomposable DIDs include problems with many unobservable variables other than simple MDPs, as long as they satisfy the condition.

On the other hand, Partially Observable Markov Decision Processes in which the state variables are dependent on the whole history cannot be separated by the decisions and the observations before those decisions, thus are not temporally decomposable. In these cases, the largest clique constructed will increase as the number of the time stages t increases, namely at least t in clique size.

Theorem 4-1: In a connected DID where spreading variables are present, the size of the largest clique is at least T , here T is the number of total stages.

Proof: The presence of spreading variables means there are paths that the Bayes-ball can bounce back to previous stages. These paths go through decisions and their requisite observations. Now supposing this DID has only one decision D_i has and its requisite observation B_i . Referring back to previous stages, nodes B_0, \dots, B_{i-1} are all requisite observation of D_i . We then need to add arcs from B_0, \dots, B_{i-1} to D_i , and also from B_0, \dots, B_{i-2} to D_{i-1} , etc. In moralization, every pair of B_k and B_j ($k \neq j \in \{0, \dots, i-1\}$) are linked with a moral arc, hence the nodes B_0, \dots, B_{i-1} and D_i form a complete set of size i . In a T -stage DID the largest clique size will be T . Reason by analogy, if there are n sets of such requisite information for $D_i, A_0, \dots, A_{i-1}, B_0, \dots, B_{i-1}, \dots$, then the clique-width (the size of the largest clique in an optimal junction tree) will increase to at least nT . □

The clique includes these requisite observations and their child decision node is the largest cluster in the junction tree corresponding to the DID. The parts other than this

largest clique can still be triangulated and constructed as clusters with the same structure separately, in a recursive manner.

When using the junction tree for inference, the time of loading the probabilities and performing the computations is proportional to the total space given by $\sum_{C \in J} \prod_{n \in C} |S(n)|$, where C is the clique in the junction tree J and n is a node in C , S is the state space of n . This is dominated by the size of the maximal clique if all the vertices have the same or similar state space size. When there are no spreading variables in the DID, the clique-width will remain the same as the decision stages increase; while the DIDs with spreading variables will have an increasing clique-width as the increase of stages.

Becker and Geiger (1996) argued that when clique-width is $O(\log N)$, there exists an approximate algorithm to find a near optimal (errs by a factor 3.66) junction tree in polynomial time, where N is the number of nodes in the graph. If the clique-width is slightly greater than the logarithm of N , there is no polynomial algorithm unless P equals NP (most researchers believe that P and NP are different classes, Cormen et al, 1990). As we discussed earlier, the clique-width grows with the decision stages. Suppose the clique-width of m stages is k , and n is the number of nodes in these stages, then after adding T stages, clique-width is around $k \cdot (1 + T/m)$, and $\log n$ grows to $\log(n + nT/m) = \log n + \log(1 + T/m)$, obviously the clique-width increases faster than $\log n$. Solving such problems exactly will be hard, no matter what is the representation.

The other problem we address here is the re-using of the junction tree to calculate the value of information on the basis of the junction tree template. Dittmer et al (1997) reused the original junction tree to calculate the value of observing a variable X before D by adding X to all the clusters between X and D 's inward-most cliques. If the uncertain variable lies in the same cluster as the decision, a change of elimination order within the cluster is needed. This method is feasible in the dynamic environment since

the conclusion is over an ID with multiple decisions and separable values. If a particular variable is to be observed several stages earlier, the sub-junction tree with this variable added in the cliques can be reused.

4.3 Adding Mapping Variables to the Junction Tree

As has been discussed in an earlier chapter, the VOI computation for decision-intervened chance variables is different from the computation for those nodes unresponsive to the decision. Recall in the procedure of converting an influence diagram to canonical form in Chapter 3, causal mapping variables need to be added for each chance variable responsive to a decision (Heckerman & Shachter, 1995), and the original chance variables become deterministic nodes.

Adding mapping variables in a specific stage only influences the construction of the sub-junction tree for the subnet of that stage. When the distance between the decision and the chance node is not far, usually it only adds some leaf clusters including the mapping variables to the junction tree.

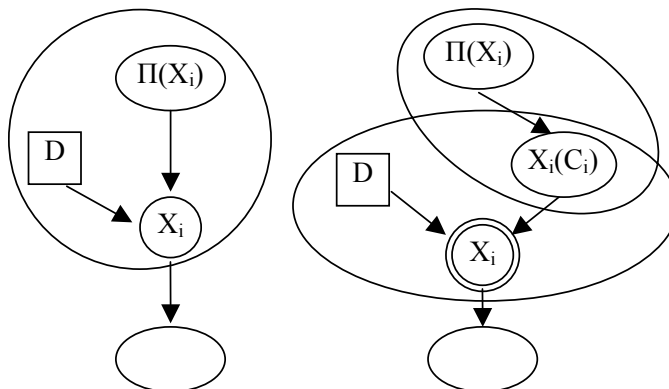


Figure 4-5: ID without or with mapping variable added

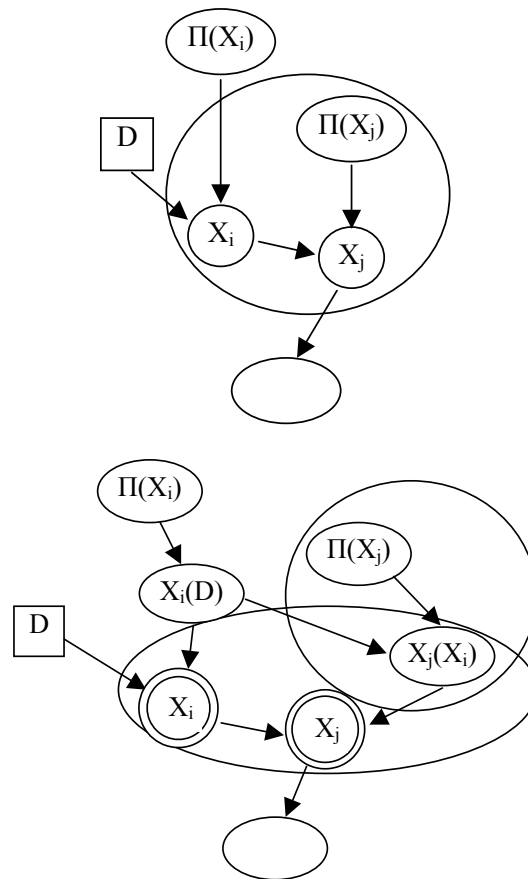


Figure 4-6: Sequentially add mapping variables and cliques

Suppose that a chance node X_i is responsive to decision D . Consider the situation that X_i has a set of parent nodes $\Pi(X_i)$ which is unresponsive to D . The mapping variables are added between X_i and $\Pi(X_i)$. Before the mapping variable $X_i(C_i)$ is added, X_i is in the same clique of $\Pi(X_i)$. With mapping variable added this clique splits to two, one includes $X_i(C_i)$ and $\Pi(X_i)$, and the other includes X_i and $X_i(C_i)$, as shown in the right part of Figure 4-5. If X_i has no chance node parents, only a leaf clique including X_i , $X_i(C_i)$, and C_i is added.

If the responsive node has responsive parents, it is only a matter of sequentially adding mapping variables and consequently leaf cliques, like the procedure illustrated in Figure 4-6.

In a properly constructed junction tree, a decision D is weakly inward of an uncertain variable A if A is a descendant of D in the influence diagram. If a junction tree is already built for the sake of inference, it is now easy to check if D is weakly inward of A , and thus determine if the mapping variable of A are necessary to add or not based on the junction tree.

4.4 Cost of gathering information

Cost can be incorporated directly in the value function when it occurs recursively. It can also be subtracted separately from the VOI calculated. The certain equivalent $CE(X)$ is the amount of money one is willing to pay for an uncertain event X (e.g., a gamble). Suppose $U(x)$ is a mapping function from money value to utility, then $U(CE) = \sum_i p_i U(x_i)$, where $P(X=x_i) = p_i$. We say $U(x)$ has the Value Additive property or Delta-property when $U(CE+\Delta) = \sum_i p_i U(x_i+\Delta)$. When utility is involved, we usually assume the delta-property holds so that an increase in value function will cause same amount of increase in certain equivalent.

When time is a concern, the total expected value will be different from the case without taking into account the effect of time.

In order to study the cost structure against the information value, not only the cost of information gathering, but also the cost of time delay, which is also called the opportunity cost, we divide the total cost C_{total} into two parts: static cost C_s and temporal cost C_t , where the former will not change as the time elapses, e.g., the cost of conducting a medical test; and the later will increase with the time for information gathering (e.g., time of waiting for a test result to turn up):

$$C_{total} = C_s + C_t.$$

4.4.1 Discounting the cost

The temporal cost can be represented with an economic discount factor r_c for time value as well, hence the future total cost C_{Ftotal} can be derived from present total cost C_{Ptotal} :

$$C_{Ftotal} = C_{Ptotal} (1 + r_c)^T$$
$$C_{Ptotal} = C_s \cdot \frac{(1 + r_c)^T - 1}{r_c (1 + r_c)^T} \quad (4-1)$$

Here T is the number of time slices in the model, and $r_c \in [0, 1]$.

The choice of interest rate of such economic discounting has been disputed by researchers. A rate of 2-10% is often considered as consistent with economic theories. Drummond et al (1987) used 5% as common rate. However, an expert panel organized by the United States Public Health Service recommended that researchers use a baseline 3% discount rate (Lipscomb, Weinstein, & Torrance, 1996).

4.4.2 Discounting the benefits

Since time value is considered when calculating cost, the benefits received in each decision stage should be discounted with time as well.

Discounting benefits takes place when a discount factor r_b is applied to the value node merging operation in DIDs. Given the value nodes v_0, \dots, v_n for each stage, the merged value of i th stage $V_i = v_i + r_b V_{i+1}$.

The discounting of benefit has not been discussed in the junction tree constructed from an influence diagram. In (Shachter 1999), if the value nodes are not nested, which

is the case of dynamic influence diagram, the structure of properly constructed junction tree will be as the Figure 4-7, for each un-nested value node, Q_i is the non-value variables relevant to V_i . The value / utility potential ψ_{C_i} (refer back to Chapter 2, equation 2-6) is propagated from leaves to root, if it is not null. Hence a discount factor $r_b \in [0, 1]$ can be directly multiplied to the second addend of the equation to denote the time value when the marginalization proceeds in reversed time order.

$$\psi'_{C_i} = \psi_{C_i} + r_b \cdot \frac{\psi_S}{\phi_S} \tag{4-2}$$

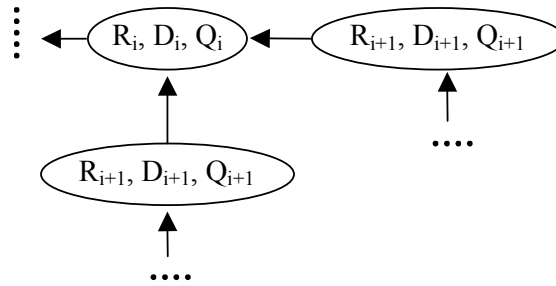


Figure 4-7: A part of properly constructed junction tree

In time-invariant dynamic influence diagrams the sub-junction trees are marginalized, discounted and then absorbed from leaves to root iteratively.

Gold et al (1996) recommended a discounting rate of benefit r_b be the same as the rate of cost r_c . It is said that if r_b is greater than r_c , a time paradox will occur: it is always more profitable to delay the action than act immediately. Hence an equal rate is recommended and the present values of both cost and benefit are used.

However, (Gold et al, 1996) is mainly focusing on the societal perspective of the cost and benefit, e.g., the same amount of money can save the same number of patients the next year as this year without inflation. As to the view of an individual patient, this might not be true since an earlier detection and treatment of some disease are better than a late one, and a one-day delay may be so expensive that it cost the patient his/her

own life. So when individual opinions are involved the discounting rate might be different from that of societal analysis. One way for analysis is to define a base rate of 3% and perform sensitivity analysis in the range of 0 to 7% on the discounting rate. The discounting rate can be varying from individual to individual, but for curative health investment with immediate benefits, a positive discounting rate is appropriate.

4.4.3 Semi-Markov Processes

In the case of transition time being stochastic, i.e., the semi-Markov decision process, an attribute $t(\delta)$ denoting the time with respect to a granularity unit δ can be added to each node, the marginalization of the utility potential will become:

$$\psi'_{C_i} = \psi_{C_i} + \rho \frac{t(\delta) - t(\delta)}{\delta} \cdot \frac{\psi_S}{\phi_S} \quad (4-3)$$

4.5 Calculating VOI in Dynamic Influence Diagrams

As a synthesis of the above discussions, an algorithm for exact VOI computation comes out:

Input: A dynamic influence diagram M with temporally invariant structure, the chance node X_i of interest, the decision node D_i prior to which we observe X_i .

Output: The value of information of observing X_i before D_i .

Procedure:

1. Check if there are any spreading variables in the DID.
 - a. If no, specify the subnet between the forward interfaces.
 - b. If yes, report and stop. Approximate method may be needed.

2. Find requisite observations R_i for $(V_i | D_i, I_i)$ using the Bayes-Ball algorithm.
Run Bayes-Ball on $(V_{i-1} \cup R_i | D_{i-1}, I_{i-1})$
3. Store all the requisite observations in R_i , make R_i as only parents of D_i .
4. For $i = T$ to k do
 - i. Construct sub-junction tree
 - ii. Calculate the expected value of sub-ID
 - iii. Discount and update the potentials
 - iv. Save the potentials
5. For $i = k$ to 0 repeat i to iii
6. Add mapping variables if necessary. Add X_l to all cliques inward of the clique with D_k , re-calculate the expected value from k th stage to l th stage, then to the root.
7. Finally, get the difference.

The Bayes-Ball algorithm runs in time linear to the graph size, i.e., $O(|V|+|A|)$, where $|V|$ is the number of vertices and $|A|$ is the number of arcs. To judge if there are any spreading variables, there's no need to run it in the whole DID, but two stages of the DID. Hence if a time-decomposable DID has T time stages with N nodes in each stage, the time will be $O(N^2)$. On the other hand, the time for performing the computations is dominated by the size of the largest clique with maximum state space in the junction tree. The inference time is $O(T \cdot N \cdot |S|)$, which is linear to the state space $|S|$. Here $|S| = \prod_{X_i \in C} S(X_i)$, where C is the largest clique in the total junction tree, and $S(X_i)$ is the state space of X_i in C .

As been discussed before, for clique-width $k = O(\log N)$, an approximate algorithm can be used to find a near optimal (with error factor a is a constant, $a = 3.66$ in (Becker and

Geiger, 1996), a can be improved further) junction tree in polynomial time, where N is the number of nodes in the whole graph. The inference will be $O\{T \cdot N \cdot \text{poly}(N) \cdot [\prod_{X_i \in C} S(X_i)]^a\}$ (proved in (Becker and Geiger, 1996)). Here $\text{poly}(N)$ is the complexity for solving a linear programming, which is polynomial. For binary variables X_i , $[\prod_{X_i \in C} S(X_i)] = 2^k$, since the largest clique C contains k variables, and when $k = O(\log N)$ this term is $O(N)$, and $[\prod_{X_i \in C} S(X_i)]^a = O(N^a)$. The time for running Bayes-Ball is negligible, compare to the inference time. Hence in this case the total time will be $O\{T \cdot N \cdot \text{poly}(N) O(N^a)\}$, which is polynomial to N .

The inference is much more complex at the presence of spreading variables, as has been discussed before.

4.6 Implementation

In order to illustrate the procedure of calculating the VOI in dynamic situation, an example based on an actual dynamic decision problem in medical practice is introduced in this part. The case comes from (Leong and Cao, 1998; Wang, 1999; Wang et al 2000).

4.6.1 The follow-up of colorectal cancer

Colorectal cancer refers to the malignant tumor of the colon or rectum. It is the second most common neoplasm in Singapore. Nowadays many of the patients with colorectal cancer undergo a potentially curative resection but about 50% nevertheless die from the local recurrence or distant metastasis within five years after the resection (Wilson and Donohue, 1991). Regular follow-up is a logical way of improving the patient's

prognosis by early detection and treatment of metachronous tumors, local recurrence or metastatic disease.

During the follow-up process, the colorectal cancer patient visits the physician at regular intervals to have some of the diagnostic tests performed. These diagnostic tests include those for detecting either recurrence, e.g. sigmoidoscopy, colonoscopy and fecal occult blood test (FOBT), or metastasis, e.g. chest radiography (CXR), or both, e.g. computed tomography (CT) scan and carcino-embryonis antigen (CEA) test. Hence the tumor follow-up was defined as “clinical and apparative test repeated schematically to detect tumor relapse after curative resection.” (Staib et al, 2000)

On each visit, the doctor would prescribe several tests according to the current status and the symptoms of the patient. The patient needs to pay for the consulting fee of the physician, the cost of the test, and spend time on both conducting the test and waiting for the result. Hence it is important to give only the necessary tests to a patient on each follow-up visit.

4.6.2 The model

The case has a clearly multi-staging nature. Furthermore, the time value should be considered in the process: earlier detection of the recurrence is more useful than later detection. The whole problem can be described in a standard dynamic model (Wang, 1999) including time horizon T , actions A , states S , events E , reward function R and probabilities P . The model is built on the basis of data from Singapore General Hospital, yet the variables are abstracted to simplify the model.

The patient visits the physician every 3 months in the first three years after curative surgery (4 times each year). If a patient survives these three years without recurrence

and metastasis, he/she is assumed as completely cured. Hence the decision horizon for the intensive follow-up model is: $T = \{0, 1, 2, \dots, 12\}$.

The actions that can be taken include a series of medical tests for detecting the patient's status, which fall into three categories: test for recurrence (denoted as Test_R), test for metastasis (denoted as Test_M), and test for both recurrence and metastasis (denoted as Test_RM). It may be significant to compare these alternatives to the case of without any follow-up (the 'do nothing' action), but patients that do not go to hospital for follow-up treatments leave no records. It is difficult to gather information for this scenario; hence this action is not included in the model.

During the follow-up programme, a patient may be free from malignant tumor, develop tumor with local recurrence, develop metastasis tumor in distant location, or have both recurrent and metastasis tumor, which has little hope of cure. We denote the above four states as Well, Rec, Met and Rec-Met. Symptoms and test results (TR) support the diagnosis for the patient. Different symptoms are related to different states, which can be grouped into two abstract symptoms: symptom of recurrence (SOR), symptom of metastasis (SOM). (SOR), (SOM) and (TR) are the abstract event variables that can be observed.

Utility values of 10.0, 4.0, 2.0, and 0.0 are assigned corresponding to the four states Well, Recurrence, Metastasis, and Rec-Met respectively. We assume the decision maker is risk-neutral and delta-property holds for the utility function, hence in the rest of this chapter we will calculate the expected utility instead of expected value, and VOI computed is actually utility of information. All the probabilities and utilities are assigned by domain experts.

In the original model of (Leong and Cao 1998; Wang 1999), costs are not explicitly included. In order to support a more practical value of information study, costs of

information gathering procedures (medical consulting, diagnostic tests) are taken into account in this example.

The costs for the alternatives are the amount of money paid for the diagnostic tests, which is shown in Table 4-1. (Data source: Singapore General Hospital price list). Tests are grouped into three categories. The abstracted cost for each category is the weighted average over all tests in the category. The weights come from the frequency of each test used in all 20,149 patient records.

Table 4-1: Cost for alternatives in follow-up case

	Test for both		Test for recurrence			
	CEA	CT scan	Colono- scopy	Sigmoid- scopy	Abdominal X-ray	Endorectal Ultrasound
Price (SG\$)	16	200	60	60	6	94
Frequency	1	0.95	0.95	0.99	0.82	0.94
Weight	0.51	0.49	0.26	0.27	0.22	0.25
Abstract cost (SG\$)	106.16		56.62			
	Test for metastasis					
	Chest X-ray		Ultrasound-liver		Barium Enema	
Price (SG\$)	6		94		58	
Frequency	0.95		0.95		0.95	
Weight	0.33		0.33		0.34	
Abstract cost(SG\$)	52.7					

When costs of each alternative are taken into account, it becomes a multiple criteria decision making problem. In this case, two objectives are considered: maximize the utility of finding earlier sign for recurrence, and minimize the cost of conducting the diagnostic tests. The cost is the price (in Singapore Dollars) of the test that the patient

takes, and the consultation fee, etc. It's hard to map this money value to utility scale, for every patient has a different utility profile.

In such cases, a comparison on benefits over costs ratio, or the B/C ratio (Table 4-2) can be conducted among the test strategies. This B/C ratio is frequently used in cost-effectiveness analysis.

Table 4-2: Value functions for the follow-up case

State \ Decision	Well	Rec	Met	Rec_Met
Test_R	10/52.7=0.19	4/52.7=0.076	2/52.7=0.038	0
Test_M	10/56.62=0.177	4/56.62=0.07	2/56.62=0.035	0
Test_RM	10/106.16=0.094	4/106.16=0.038	2/106.16=0.019	0

Discounting

The way of discounting for both cost and benefit will vary for their different measures. Many researchers use the Quality Adjusted Life Year (QALY) as the measure for benefit gained in health care, and compare alternatives on the basis of the rate that QALY over cost (money). The benefit scales experts provided for this case is in 0-10 utility, and hence the comparison of alternatives can be done based on utility-cost ratio.

The problem in this case is not concerning individual patient alone, hence the traditional discounting rates of benefit and cost for social perspective are adopted. The United States Public Health Service recommended 3% rate, now used for both the benefit and the cost.

4.6.3 Methods

This is a finite horizon dynamic problem, thus can be described in dynamic influence diagram format. Figure 4-8 is a DID for the abstracted model of colorectal cancer follow-up treatment problem. In this case, chance nodes S_{i-1} , SOR_i , and SOM_i are d -separated from the previous stages by the set $\{D_i, TR_i\}$, so there are no spreading variables. Essentially, this is a problem with observable state variable (TR), yet it includes unobserved nodes (S , SOR , SOM).

To demonstrate the VOI computation for SOR and SOM before decision D , there are no information arcs from events SOR and SOM to the decision node in the model shown in Figure 4-8. The importance of observing these symptoms before decision is thus evaluated by computing the VOIs.

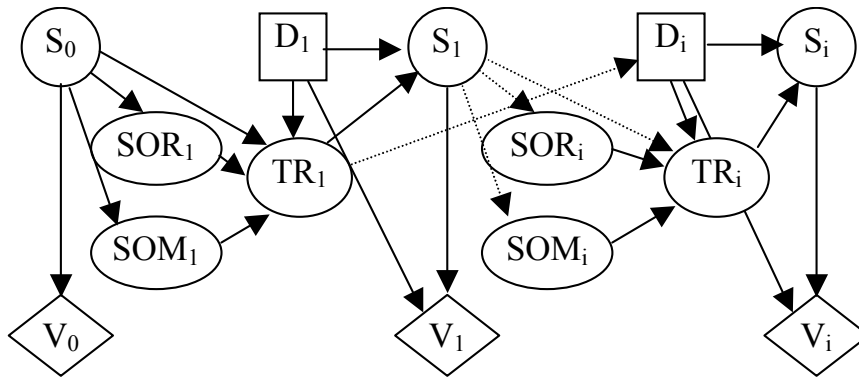


Figure 4-8: The follow-up problem

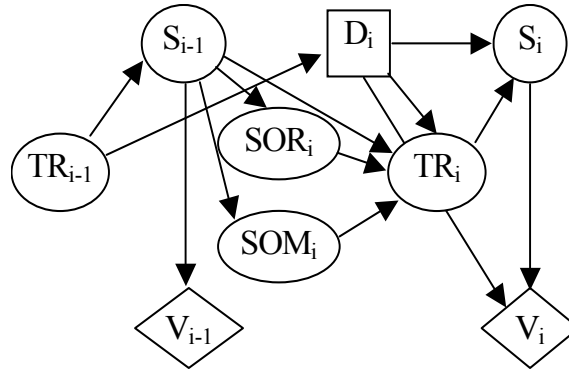


Figure 4-9: Subnet for the follow-up problem

In the last stage, the requisite observation for the last decision D_n is $R_n = \{TR_{n-1}\}$. Cutting a subnet from $\{TR_{i-1}\}$ to $\{TR_i\}$ and running the Bayes-Ball for $(V_i, R_{i+1} | D_i, I_i)$, it could be found that the requisite observations for D_i are $\{TR_{i-1}\}$. A subnet shown in Figure 4-9 is thus obtained, where the set $\{SOR_i, SOM_i, D_i, TR_i\}$ is the forward interface. Replace D_i with a chance node as the child of the requisite observations and construct sub-junction tree in this subnet, a quite simple junction tree could be built as shown in Figure 4-10. The root cluster may be a little different from other clusters in later stages since the first stage is not exactly the same as later ones.

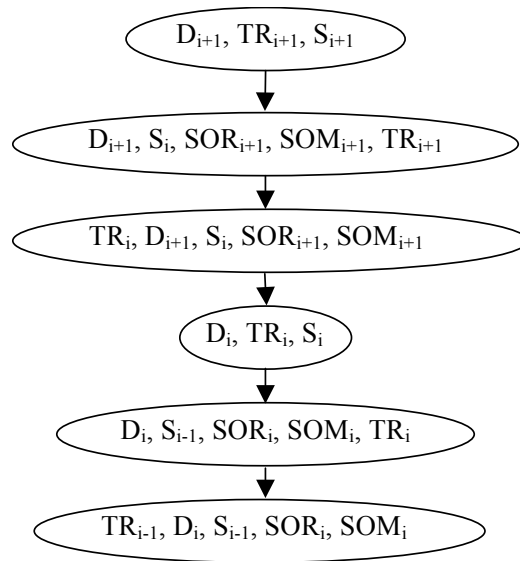


Figure 4-10: A sub-junction tree (for 2 stages).

After building such junction tree the total expected utility can be computed from the leaf to the root node. The procedure is: first store the whole DID as a 2-stage structure and the number of time stages; second, compute the utility and probability potentials from the last stage, store them in the forward interface $\{SOR_i, SOM_i, D_i, TR_i\}$; and then repeat the computation for potentials until finally the first stage is reached. Suppose the value of information of observing S_1 before D_2 is of the interest, it is only necessary to reverse the elimination order inside the clique $\{TR_1, D_2, S_1, SOR_2, SOM_2\}$ since they are in the same clique. For the scenario of knowing S_2 before D_2 , a mapping variable should be added for S_2 , like the condensed form shown in Figure 4-11. Other procedures are the same, and then VOI is computed afterwards.

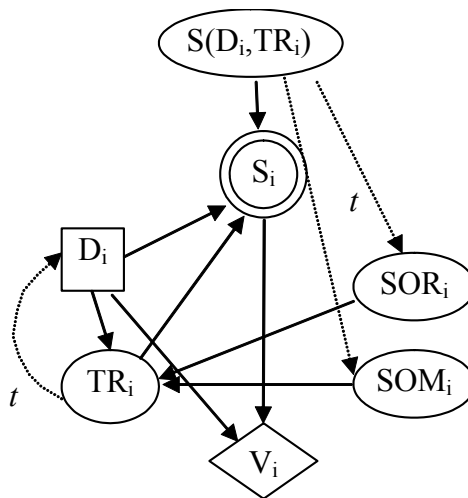


Figure 4-11: Condensed canonical form for VOI of S_i before D_i

The VOI computation can be implemented in the same junction tree when the chance node and the decision node in question are input at the same time. However, if we require a VOI computation after computing the total expected utility of the whole DID, the reuse of the original junction tree will require extra space for storing the potentials of each time stage.

4.6.4 Results and Discussion

Using traditional VOI computation methods (procedures from Smile© Library, Decision Systems Laboratory) in above DID which only includes 4 stages, an average time of 45 seconds (In a Pentium III 500. Computing facility is the same for all programs described afterwards) are required for each VOI computation. It will be very time-consuming to both construct the 12-stage model and compute the value of information for any chance node.

When using the BNT toolkit (Murphy et al, 2002) for MATLAB to solve the example as a whole DID, using the LIMID model (Lauritzen and Nilsson, 2001), the elapsing time is around 8 seconds. In LIMID model, the assumption of non-forgetfulness of the whole decision history is omitted, to produce soluble influence diagrams that might be different from traditional ones, hence the model is called Limited Information Influence Diagram. The model can be taken as an approximation of the traditional IDs, which will be discussed later. The MATLAB BNT toolbox realization of LIMID is an exact one, however. Thus it is used for comparison purpose.

Meanwhile, the computing time of total expected utility for procedure with the sub-junction tree algorithm described in previous section is 0.82 seconds for a 4-stage DID. The running time for the whole 12 stages is around 1.5 seconds for sub-junction tree algorithm and 22.5 for unrolled DBN inference. For computing VOI, the elapse time is about 2 seconds on average using sub-junction tree algorithm, varying with the starting and ending stage of observing the perfect information.

Table 4-3 Comparison of Computation Time**(Measured in seconds)**

	VOI computation (Smile© library)	LIMID (Matlab BNT)	Unrolled DBN (Matlab BNT)	Sub-junction Tree (Matlab BNT)
4-stages	45	—*	—*	0.82
12-stages	— ⁺	8	22.5	1.5
Comments	Exact	Approximate	Exact	Exact

*: Experiments not necessary hence not done.

⁺: Computation time too long to reach a stop.

The method using the junction tree may not be very advantageous over the probability propagation in this case where the largest clique includes all the nodes in a stage. However, it surely is faster than inference over the whole DID without dividing the DID according to stages.

The total expected utility calculated is 1.5137, the VOI for knowing S_i before D_{i+1} is 0.0677 and knowing SOR_i and SOM_i before D_i is 0.0592. Though, in this particular case, such number of ‘VOI’ is not of our final interest since this is the gain in B/C ratio given the perfect information of S_2 before D_3 . The policies for both scenarios are recorded to calculate the actual benefit and costs occurred respectively. The total expected utility = 61.4113, (remember the maximum utility in one stage is 10 and minimum is 0), total cost = 675.5200 (in Singapore dollar) for no perfect information on S ; and the expected utility 83.0590, total cost = 675.5200 for knowing S .

Furthermore, the case of not taking any diagnostic tests is also studied under the assumption of maximum entropy (An even distribution of ‘test results’). The original model is thus converted into Figure 4-12. The total expected utility is now only 1.3042,

0.2772 less than full information and 0.2095 less than taking diagnostic tests, which is a much larger advantage. This means the diagnostic tests are quite important in the follow-up decision model.

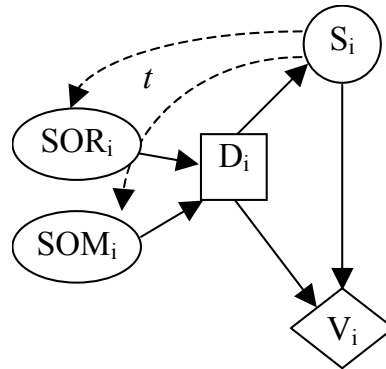


Figure 4-12: Follow-up without diagnostic tests

When the starting status of the patient is well, it is best to choose tests for recurrence, while tests for metastasis should be employed if the starting status of the patient is recurrence or metastasis. When the patient's starting status is unknown and assumed to be equally distributed, the policy of tests for recurrence should be used.

These results show that it is beneficial having perfect information for the status of patient. However, if the problem is modeled other way, e.g., the next stage state S_{i+1} depends on S_i , this sub-junction tree algorithm may not be applicable because of the presence of spreading variables.

4.7 Conclusions

This chapter describes an algorithm to compute value of information in dynamic decision models, namely dynamic influence diagrams.

A group of DIDs are identified as which can be decomposed into sub-networks with similar structures, and hence a sub-junction tree can be generated as a computing

template based on these sub-networks. The methods on reusing the sub-junction tree are also discussed when VOI for variables with intervening decisions is under concern.

This chapter also considers the time value of benefits and costs for information gathering by discounting the value in each stage. For societal analysis an equal discounting for both cost and benefit is suggested, however, when the individual preference is concerned, the rates may be different. Hence a base rate and a sensitive analysis afterwards are suggested.

An algorithm of calculating VOI in DIDs is proposed and the computational complexity discussed. Using our method, the inference in DIDs without spreading variables is polynomial to the state space and when clique width is $O(\log N)$ the algorithm is polynomial. This method is supposed to provide researchers a tool of sensitive analysis in dynamic decision modeling. We believe it is both illustrative and efficient.

Finally, an example based on real world practice is used to illustrate the computation of EVPI in different scenarios. The example shows that the algorithm is quite efficient and still yields reasonable results.

5 Quantitative Approximations in Partially Observable Models

The method proposed in Chapter 4 is polynomial to state space only when spreading variables are not present, which is equivalent to dynamic models in which decision variables are Markovian (I.e., given the current variables, future variables are conditionally independent of the past).

The presence of spreading variables results in immense difficulties in the computation of the expected value of a DID. The exact solution of such POMDP with finite-horizon is PSPACE-hard³ (Papadimitriou & Tsitsiklis, 1987), let alone the infinite-horizon problems. When it comes to the computation of EVPI, the complexity can be twice of that for a single POMDP solution. Even computing a bound for EVPI will be intractable (please refer back to Chapter 3).

On the other hand, the purpose for VOI computation is to guide the information gathering process, to adjust the model and ultimately to improve the decision quality. Therefore, in many occasions we would like to consider some approximation methods with higher efficiency, but with some tradeoff in accuracy.

³ The class PSPACE is the set of decision problems that can be solved by a Turing machine using a polynomial amount of memory, and unlimited time. A problem is PSPACE-hard if an algorithm for solving it can be translated into one for solving any other PSPACE-problem, therefore PSPACE-hard means "at least as hard as any PSPACE-problem," although it might, in fact, be harder.

Chapter 3 has analyzed the EVPI computation in POMDPs, which is quite similar to the value function pruning process. Hence approximations for POMDPs can be naturally adopted in EVPI computation as well.

Approximations of POMDP solution techniques have been surveyed in (Cassandra, 1998) and (Murphy, 2000). The later paper converts the known POMDP model into a belief state MDP and divides the solution methods according to the approximations lies in belief state \mathbf{b} and value functions γ . The EVPI computation considered in this thesis mainly falls in graphical models which consist of a graph topology and a set of parameters associated with it; hence the approximations discussed here are divided into two major categories: structural and parametric approximations.

5.1 Structural Approximation

Structural approximations make use of the changes in network structure, e.g., asserting conditional independencies, adding information or similar steps in order to simplify the original model to improve the efficiency. Many ways of structural reduction have been proposed: adding certain assumptions to restrict the complexity in computation, approximating the true value function and the belief state by breaking arcs or removing nodes, etc.

5.1.1 Finite History Approximations

As has been discussed in early chapters, the complexity of a partially observable dynamic decision problem is exponential to the history of past decisions and observations. A straightforward way is assuming the decision-maker keeps memory to only the latest k observations and throwing away the history before the k stages. Constructing a window containing finite history is widely used in the learning

literature. The agent uses memoryless policies (Platzman, 1977; White & Scherer, 1994; Littman, 1994; Singh, Jaakkola, & Jordan, 1994), policies based on truncated histories (Platzman, 1977; White & Scherer, 1994; McCallum, 1995; Hernandez-Gardiol and Mahdevan, 2000), or finite-state controllers with a fixed number of memory states (Platzman, 1980; Hauskrecht, 1997; Hansen, 1998a, 1998b).

Lauritzen and Nilsson (2001) introduce Limited Memory Influence Diagram (LIMID). The original purpose is to describe multistage decision problems in situations like restricted agent memory or multiple decision makers. Yet this can be taken as an approximation for the traditional IDs with memory constraints. The arcs from all the decisions and observations in previous stages to current decision node are deleted (information forgotten). That is, the information known before a decision is only in the same stage of the decision, and no earlier information will be remembered.

Refer back to the example illustrated in Figure 4-3, the inference process in traditional influence diagrams requires adding arcs from $\{b_0, D_0, \dots, b_{i-1}, D_{i-1}\}$ to each D_i , as shown in Figure 4-4, which are called ‘no-forgetting arcs’ indicating they are in the decision maker’s memory. However in a LIMID, all such arcs are cleared out, as the following topology in Figure 5-1. Obviously, it will not form the large clique analyzed in section § 4.1, hence avoid the complexity exponential to the time horizon.

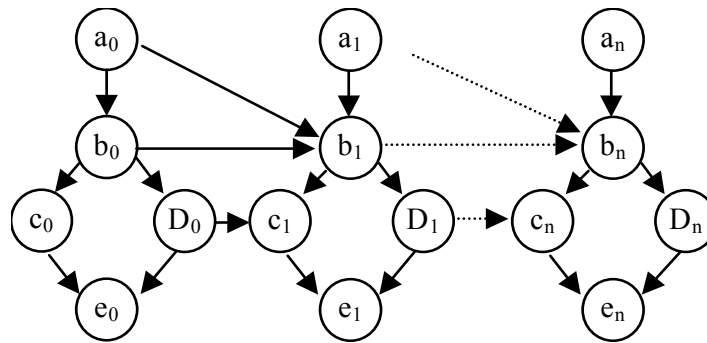


Figure 5-1: LIMID version of Figure 4-3 (after converting decisions to chance nodes)

LIMID is often a good approximation of the traditional ID. In the PIGS example (Lauritzen and Nilsson 2001) the expected value of LIMID approaching that of the traditional ID very closely. However, this error between LIMID and ID can be arbitrary large.

Typically the limited memory influence diagram will have less expected utility since the decision made in the absence of historical information could be a sub-optimal one. If state variables are all observed in perfect information case, the observed states are the only requisite information before the decision and there will be no difference between no-forgetting and forgetting case. Hence the expected utility in perfect information case is the same for both assumptions. Then it can be derived that VOI computed with limited memory assumption offers an upper bound.

The case is a bit subtle when only a sub set of state variables are observed. The LIMID version expected utility of both with and without the perfect information will be less than the no-forgetting influence diagram, which makes it hard to decide the error in VOI calculated this way.

5.1.2 Structural Value Approximations

MDP approximation attempts to retain the Markovian property of the actions by adding information arcs from the state variables to the actions, approaching the POMDP with an observable process thus breaking the large cliques containing all the history into smaller ones. MDP approximation methods include fully observable MDP approximation, the Q-function approximation of POMDP (Littman, Cassandra, & Kaelbling, 1995), the fast informed bound update (Hauskrecht, 2000), and the Even-Odd MDP (Zubek and Dietterich, 2001).

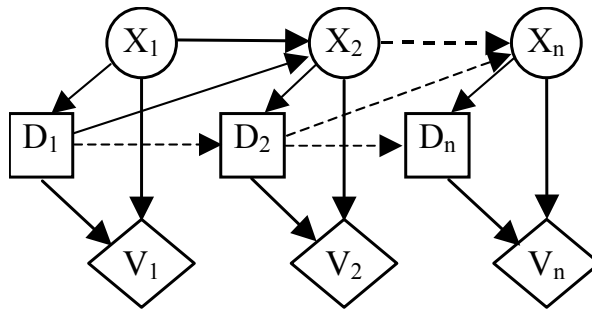


Figure 5-2: Graphical description of MDP

The fully observable MDP approximation shown in the above Figure 5-2 is equal to the case with perfect information on all state variables. Q-function MDP (QMDP), shown in Figure 5-3, can be thought as similar to the DID in Figure 5-2, with perfect information for system state variable X , yet we need to average the value over the belief state $b(X)$.

Here is the value update for QMDP:

$$\begin{aligned}
 V_{iQMDP} &= \max_{D_i} \sum b(X_i) Q^*_{MDP}(X_i, D_i) \\
 Q^*_{MDP}(X_i, D_i) &= Q(X_i, D_i) + \sum P(X_{i+1}|X_i) \max_{D_{i+1}} [V(X_{i+1}, D_{i+1})]
 \end{aligned}
 \tag{5-1}$$

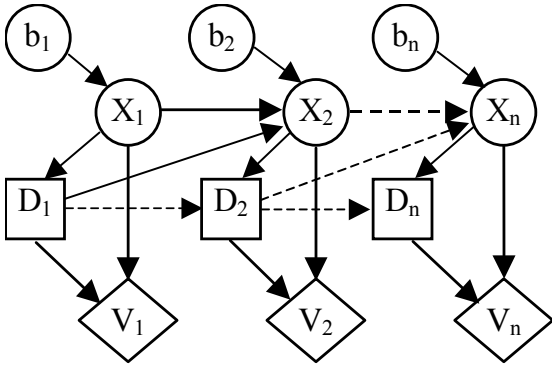


Figure 5-3: DID for Q-function MDP approximation

The only difference between the Q-function MDP and the fully observable MDP is that the position of the maximization of decision and the summation over belief states is swapped. The value difference (VOI) of the two scenarios offers a lower bound of the actual VOI.

The fast informed bound (FIB) approximation (Hauskrecht, 2000), on the other hand, will offer tighter bound on the expected value thus bare a lower loss in accuracy than the QMDP. It is just like the DID shown in Figure 5-4, which the decision maker has information of state variable with one-stage lag and the observations O influenced by X.

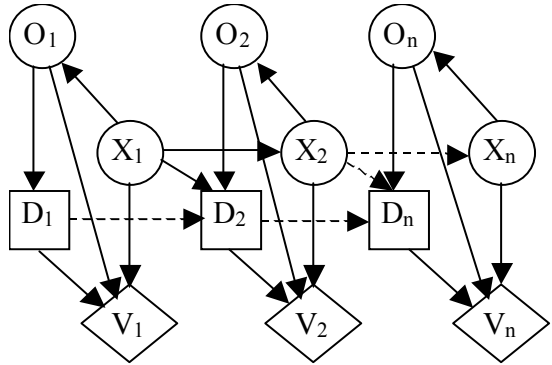


Figure 5-4: DID for fast informed bound approximation

$$\begin{aligned}
 V_{iFIB} = \max_{D_i} \sum b(X_i) [Q(X_i, D_i) + \sum_{O_{i+1}} \max_{D_{i+1}} \sum_{X_{i+1}} P(X_{i+1}, O_{i+1}|X_i) \\
 [V(X_{i+1}, O_{i+1}, D_{i+1})] \tag{5-2}
 \end{aligned}$$

The only revision need to do in the original DID to form a fast informed bound approximation is adding one-stage lag information arcs to the decision in each stage.

The relationships between the expected value for the different cases are (Hauskrecht, 2000):

$$\text{MDP} \geq \text{QMDP} \geq \text{Fast Informed Bound MDP} \geq \text{Exact POMDP} \geq \text{Unobservable MDP}.$$

The difference between them can be viewed as the value of different kind of information, e.g., full, lagged, etc.

The VOI calculated by subtracting the expected value of MDP_{FIB} from MDP, will be less than or equal to the actual VOI by the above inequality. Hence if after such computation a chance variable has VOI greater than 0, it is worth gathering information around it. Moreover, the importance ordering determined by the VOI of chance nodes is unchanged by such approximation.

In (Zubek and Dietterich, 2001). an approximation of even-odd POMDP (POMDP with stages observable in even stages and unobservable in odd stages, actually a 2-stage MDP) is studied. Figure 5-5 is an example of even-odd POMDP, where the variables in stage 0 and stage 1 can be grouped together to form a ‘big’ stage. This even-odd POMDP is a better approximation of the real POMDP than MDP and Q-MDP.

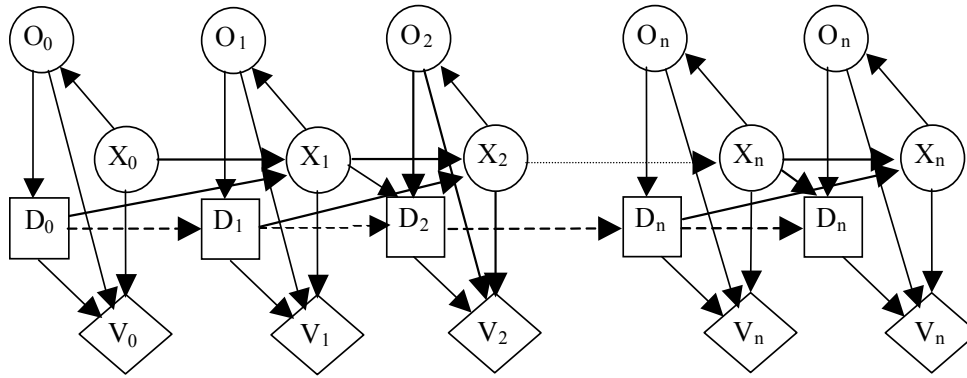


Figure 5-5: Even-Odd POMDP (2-stage MDP)

Moreover, if we including 3 POMDP stages in one MDP stage, with delayed information from stage 1 to stage 2, omitting any information arc in the third stage, the approximation of the value function will be better than both even-odd POMDP and the fast informed bound MDP. However, the accuracy is based on a trade-off in computation cost. As we have discussed in the previous part, the clique size increases with the number of stages involved in a sub-graph. A more accurate approximation of 3-stage MDP needs about C^3 if the computational cost of MDP is C . Hence in practical applications, the selection of approximation methods also depends on the computational resources.

5.1.3 Factorize the Network

Another way of structural approximation is to reduce problem complexity by removing edges in the network. In (Kjærulff, 1994), the large cliques can be split by the removal of weak dependences among chance nodes. It is said in many practical dynamic problems, for graphical models the largest clique includes almost every node in each stage. If weak links are found in such cliques, breaking these links will cut down the clique size and alleviate the computational cost.

However, a well-constructed model may not contain too many weak dependencies, which leaves little space for improvement by removing these weak dependencies.

To describe how close an approximation is to the true distribution (joint), researchers focused their attention to a distance measure for two probability distributions, the relative entropy or K-L divergence.

Def. 5-1 *Relative entropy* (Cover and Thomas, 1991)

If φ and ψ are two distributions over the same space Ω , the relative entropy of φ to ψ is

$$D[\varphi \parallel \psi] = E_{\varphi} \left[\ln \frac{\varphi}{\psi} \right] = \sum_{\omega_i \in \Omega} \varphi[\omega_i] \ln \frac{\varphi[\omega_i]}{\psi[\omega_i]}$$

Boyen and Koller (1998) prove the relative entropy of two stochastic processes converged geometrically; hence the approximation scheme (also called projection scheme since it projects the joint probability distribution to a factored space, e.g., approximating a joint probability $P(ABC)$ by $P(AB)*P(BC)$ assuming A and C are independent given B) of factored belief state in POMDP has a tight error bound, while at the same time gaining computational efficiency.

Chan & Darwiche (2002) proposed a distance measure between two probability distributions that satisfies symmetry and triangular inequality, comparing to the K-L divergence used in (Boyen and Koller, 1998). The distance measure puts emphasis on local changes of the distribution. A dramatic change in probability distributions leads to a looser boundary, which the authors claim that the “hard evidence” makes the new distribution no longer zero-congruent (zero-congruent: $P(x)=0$ iff $P'(x)=0$ to avoid being divided by 0). This makes this probability distance measure not a very useful tool in VOI computation.

These approximations may reduce the clique size for a single stage. Yet the problem is hard in POMDP because the solution of a POMDP requires the entire history, which is dependent on the number of stages involved. In this sense, such methods can offer limited improvement. Moreover, only the errors occurred in belief state are tightly bounded. Yet in decision models, value function also affects the selection of the optimal strategy, so that a small error in belief state could lead to different optimal policies and induce big error in expected utility, while a large deviation in belief state could have little influence in the decision and the final expected utility.

Because of this, Poupart and Boutilier (2000, 2001) seek better approximate strategies that improve the decision quality by γ vector (it is called α -vector in their papers) analysis. Their idea is given a set of γ vectors from a solved POMDP, for any possible projection scheme in system dynamics (correspondent DBN of the POMDP), find out the scheme with the smallest error in decision quality by solving an optimization problem constructed by those γ vectors.

Using this method, the error of approximation is bounded tightly. Yet the projection schemes are chosen after the POMDP is solved and the γ vectors retained for each stage. For any complex POMDPs, obtaining the γ vector set is a hard job (PSPACE as in Papadimitriou & Tsitsiklis, 1987; Mundhenk et al, 2000). In their paper, they also point out the expensiveness in computation and emphasize this selection of projection schemes is performed offline to accelerate the online inference of the POMDP. Direct application of the method to VOI computation can be prohibitive.

5.2 Parametric Approximation

In the graphical decision models, besides the approximation in network structures, it is also possible to simplify the computation by parametric approximation. As well as structural approximation, parameters on both belief state and the value function can be approximated.

Boutilier and Dearden (1994) solve a factored MDP by state-space abstraction. In (Dean and Givan, 1997) the model minimization method is used to find MDP compressions by iteratively partitioning the state space into homogenous blocks and aggregating all states in one partition.

Boutilier and Poole (1996) incorporate the system dynamics represented in Bayesian Networks to the solution methods of POMDPs. A decision tree is constructed in each stage to represent the γ value functions.

In (Hoey et al, 1999), ADDs (algebraic decision diagrams) are applied in dynamic programming steps in MDPs. Later, Hansen and Feng (2000) describe the use of ADDs of abstracting the state space and saving computational costs in POMDPs. ADD represents the context-specific information of the model and supports further compression by merging branches with similar values.

In POMDP, the process can be treated as an MDP with continuous belief state. Roy and Gordon (2002) propose an exponential family PCA method for belief compression, which reduces the high-dimensional belief space by generating the principal components and project belief to this lower dimensional space. This is a kind of belief state abstraction, which also effectively reduces the state space in a single stage.

Belief state aggregation can be used in general solution of POMDP, yet may not be an ideal candidate in VOI computation. The belief state changes when the perfect information of certain variables is available, enforcing the user calculating the expected value twice if using state aggregation. Comparing to the method of only calculating the different parts and reusing unaffected parts, this looks a little inefficient.

Mentioned in (Cassandra, 1998; Hauskrecht, 2000), the Grid-based approximation is a simple but powerful algorithm. The idea is to approximate a POMDP (continuous state MDP) with a grid-based MDP, which divide the belief state space into grids. The parameters of the new grid-based MDP are found by interpolating-extrapolating the value of a non-grid point in original MDP. The approximate MDP has a value update with complexity of $O(|G| |A| |S|^2 |O|)$ in each dynamic programming step, where $|G|$ is number of grids used in approximating state space, A , S , O are action, state and observation sets respectively.

5.3 Comments on the approximations

Most of the approximation approaches aim to keep a better leverage between the quality of solution and the computational complexity. The adoption of any specific approximation scheme is essentially based on the requirements of the particular problem at hand.

Breaking no-forgetting arcs, restricting the computation of the process within a k -stage window, adding information arcs so that the process forms a QMDP, fast informed bound (FIB) MDP, or an even-odd MDP all reduce the problem of VOI computation to polynomial. Except the limited memory assumption (LIMID model), the others offer

lower bounds for the real VOI. The VOI computed based on fully-unobservable MDP (UMDP) is the maximum among these VOI computed by approximations because fully-unobservable case has no information to support decisions, so perfect information will have the largest profit. The VOI for those approximations are ordered as such:

$$VOI_{QMDP} \leq \left\{ \begin{array}{l} VOI_{FIB} \\ VOI_{\text{even-odd}} \end{array} \right\} \leq VOI_{\text{exact}} \leq VOI_{LIMID} \leq VOI_{UMDP}.$$

Removing weak dependencies or assuming more conditional independencies offers ways to divide the large cliques into smaller ones. Extended to the decision problems, minimizing differences in utility potentials should be considered as criteria for choosing a good approximation scheme instead of K-L divergence of mutual information. The error induced in VOI computation is not restricted within certain bounds, however.

The belief state abstraction using state aggregation, Algebraic Decision Diagrams, or exponential principal component analysis may be of interest when the two scenarios of with / without information are considered separately, yet is hard for the attempt to reuse computed results.

It is also promising to apply grid-based approximation in VOI computation. The fixed grid approximation is polynomial as well as the above methods, while induces a small error (Hauskrecht, 2000). As our target is to find both accurate and efficient in computational time and space, this might be a good candidate.

In a whole, the decision maker can choose an approximation scheme according to the required accuracy, the time and resource constraints.

6 Qualitative Analysis in General Decision Models

In the early chapters, the value of information in dynamic decision models is studied. When the system structure is time-variant, or the Markov assumption does not stand, e.g., in more general decision environment, most of the methods discussed previously are still very resources-consuming, and are unable to give out real-time results. Methods for computing VOIs in order to recommend the best evidence to collect, trading off the cost and benefits of observations are thus needed.

It is emphasized previously that the value of information is a tool for sensitivity analysis. In many real-world decision problems, VOI is performed before the decision stage to guide the information gathering procedure, so as to improve the decision quality. Hence a timely result on information value may be preferred than the one that is more accurate but comes out too late.

6.1 Introduction

As been mentioned in Chapter 2, there is great interest in developing schemes for computing the value of information in recent years. Various kinds of utility functions are considered as well, money value, utility, even relative entropy or mutual information (Cover and Thomas, 1991). Unfortunately, the computational complexity of exact computations of EVPI in an arbitrary decision model with arbitrary utility

function is known to be intractable. Even with the simplifying assumptions that a decision maker is risk neutral or has a constant degree of risk aversion, the problem remains intractable.

The intractability of EVPI computation has motivated researchers to explore a variety of approximations, both quantitative and qualitative. In this chapter we have sought to extend methods for exact and approximate computation of value of information by pursuing opportunities for leveraging qualitative analyses of the value of information. Efforts are exerted to exploit the graphical relationships in decision problems represented as general influence diagrams.

In many applications, it is reasonable to bypass the exact numerical computation of the value of information and instead to seek to identify an ordering of variables by their value of information. For example, an ordering over the value of information can be employed in conjunction with cost of that piece of information in normative decision systems to determine the most cost effective evidence to collect. Such qualitative orderings over the information value of variables can also be useful in model construction systems where the qualitative ranking of the value of information can be used to direct either the modeler or the model constructor to refine models in directions of maximum value (e.g., see Poh and Horvitz, 1993).

An earlier related work (Poh and Horvitz, 1996) derives qualitative relationships about the information relevance of chance variables in graphical decision models based on a consideration of the topology of the models. It is found that the EVPIs of chance nodes in a decision model can be ordered by considering conditional independence relationships among the chance nodes and the value node. An algorithm is outlined for obtaining a partial ordering of EVPI of chance nodes of decision models with single decision node represented as influence diagrams that are expressed in canonical form

(Howard, 1990; Heckerman, 1995). The algorithm is based on the notion of d -separation (Refer to Chapter 4) of chance nodes from the single value node in the influence diagram. The following sections review earlier work and report new results on topological relationships among variables in a graphical decision problem with regard to the value of information.

6.2 Value of Information and Conditional Independence

Before going to further discussions, let us first examine the key qualitative relationships regarding the information relevance of variables in influence diagrams. This section will review some results obtained previously and present some extensions. The study basically focuses on models in *canonical form*, a representation where all chance nodes that are descendants of one or more decision nodes are deterministic nodes. In general, any influence diagram can be converted to canonical form. (Howard, 1990; Heckerman, 1995)

6.2.1 Basic Information Relevance Ordering Relations

Let $M = (C, D, V, E)$ be a decision model where C is the set of chance nodes, D the set of decision nodes, V the value node, and $E \subset C \cup D \times C \cup D \cup \{V\}$ is the set of directed arcs. Denote the expected value of information for observing the value of chance node $X \in C$ before action $A \in D$ by $EVPI_M(A | X)$. Poh and Horvitz (1996) have shown previously that chance nodes that are not relevant to the value node given the action have zero value of information:

Theorem 6-1: If X is conditionally independent of V given A , denoted by $X \perp V | A$, then $EVPI_M(A | X) = 0$.

Proof. Given that $X \perp V | A$, we have $EU(M_{A|X}) = \sum_i p(x_i) \max_k u(a_k) = EU(M)$. Hence $EVPI_M(A|X) = 0$. □

Theorem 6-1 allows us to identify nodes that have no value of information with respect to a decision node. These zero-value chance nodes are ancestors of the decision node and are not connected to the value node except via the decision node.

As established in the following theorem, the basic relations concerning the possible ordering of EVPI for two chance nodes in a graphical decision model is based on the conditional independence among these two chance nodes and the value:

Theorem 6-2. If X and Y are distinct chance nodes that are not descendants of A , and $Y \perp V | X$ (Y is conditionally independent of V given X), then $EVPI_M(A | X) \geq EVPI_M(A | Y)$.

Proof. X and Y are not descendants of A implies that $EU(M_{A|Y}) = \sum_j p(y_j) [\max_k u(a_k, y_j)]$ and $EU(M_{A|X}) = \sum_i p(x_i) [\max_k u(a_k, x_i)]$. $Y \perp V | X$ implies that $u(a_k, y_j) = \sum_i p(x_i|y_j) u(a_k, x_i)$. Therefore $EU(M_{A|Y}) = \sum_j p(y_j) [\max_k \sum_i p(x_i|y_j) u(a_k, x_i)]$. By rewriting $p(x_i)$ as $\sum_j p(x_i|y_j) p(y_j)$, and letting $\rho_x = EVPI_M(A|X)$ and $\rho_y = EVPI_M(A|Y)$, we have

$$\sum_j p(y_j) [\max_k [\sum_i p(x_i|y_j) u(a_k, x_i) - \rho_y]] = EU(M)$$

and

$$\sum_j p(y_j) \sum_i p(x_i|y_j) [\max_k u(a_k, x_i) - \rho_x] = EU(M).$$

The last two equations imply that

$$\sum_j p(y_j) \sum_i p(x_i|y_j) [\max_k u(ce(a_k, x_i) - \rho_x)] = \sum_j p(y_j) [\max_k [\sum_i p(x_i|y_j) u(ce(a_k, x_i) - \rho_y)]].$$

For any j , Lemma 1 implies $\sum_i p(x_i|y_j) \max_k u(a_k, x_i) \geq \max_k \sum p(x_i|y_j) u(a_k, x_i)$. Since the utility function u is monotonically non-decreasing in the certain equivalent values, it follows that $\rho_x \geq \rho_y$ in order for the last equation to hold. \square

The above results can be generalized to the joint value of perfect information of a set of nodes by replacing X and Y with sets. The conditional independence relations required for identification of the ordering of EVPI can be performed with the notion of d -separation (Pearl, 1988; Pearl et al, 1990).

An equivalent graphical procedure for identification of conditional independence relations makes use of the notion of u -separation, the undirected graph separation (Castillo et al, 1997).

Given a direct acyclic graph (DAG) and three disjoint sets of nodes X , Y , and Z , first moralize the smallest subgraph containing X , Y and Z and all their ancestral nodes, this subgraph is called ancestral subgraph of X , Y and Z . If Z u -separates X and Y in the moralized ancestral subgraph, then Z d -separates X and Y in the original directed graph; otherwise Z does not d -separate X and Y .

6.2.2 Examples

Figure 6-1 shows the graphical model of a sample decision problem. The topology of the network is adopted from a car diagnosis example (Norsys corp., 1998). By applying the d -separation criterion for the ordering of EVPI values, a network of the ordering is obtained as shown in Figure 6-2. Here the dotted arc between two nodes, for example, $node\ 2 \rightarrow node\ 4$, indicates $EVPI(D | node\ 2) \leq EVPI(D | node\ 4)$.

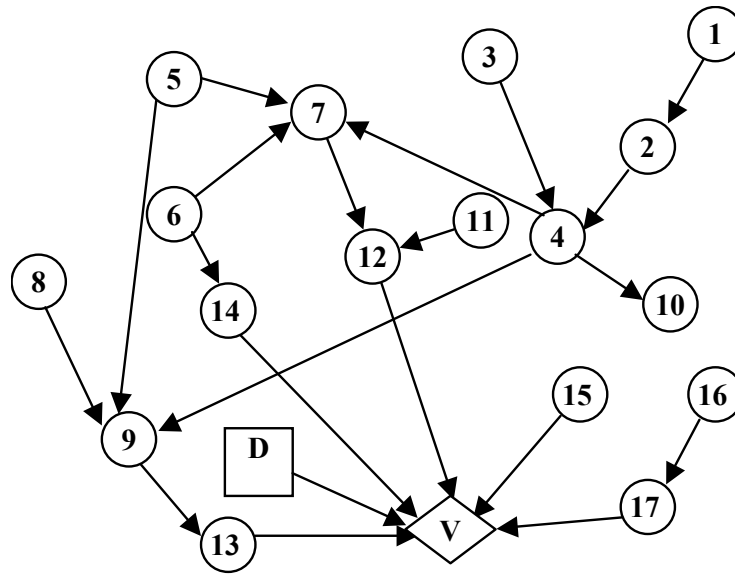


Figure 6-1: Influence diagram for Example 1.

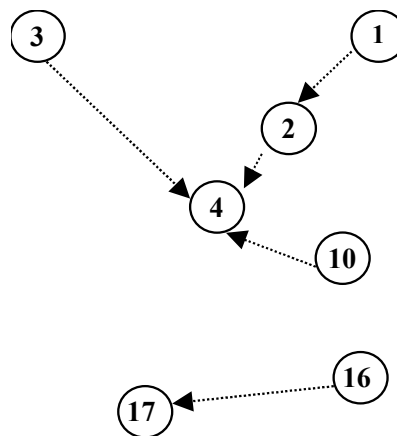


Figure 6-2: Partial ordering of EVPI for Example 1.

Note that Figure 6-2 only contains seven chance nodes instead of seventeen chance nodes in the original influence diagram. This means checking the conditional independencies among the chance nodes can only reveal EVPI orderings of these seven nodes from the original diagram. For other chance nodes, it is impossible to judge their EVPIs just with the graph structure. Hence the caption of Figure 6-2 says this is a ‘Partial ordering’.

A concept referred to as a *barren node* (Shachter, 1986) is leveraged here. Barren nodes are those other than the value node that have directed arcs into them but not out of them. Note that node 10 is such a barren node. Hence, its EVPI is bounded by the EVPI on its parent node 4, as shown by the dotted arc from 10 to 4. This example is not a very densely linked graphical decision model, and we obtained several EVPI orderings that indicate the relative ranking of the importance of information.

Let us now consider a decision model with a much larger number of nodes. Figure 6-3 displays an influence diagram developed by extending the ALARM Bayesian network model (Beinlich et al, 1989) to a decision problem with action and value nodes. This network contains 8 diagnoses, 16 findings and 13 intermediate variables. Figure 6-4 shows the partial ordering of EVPI for Example 2.

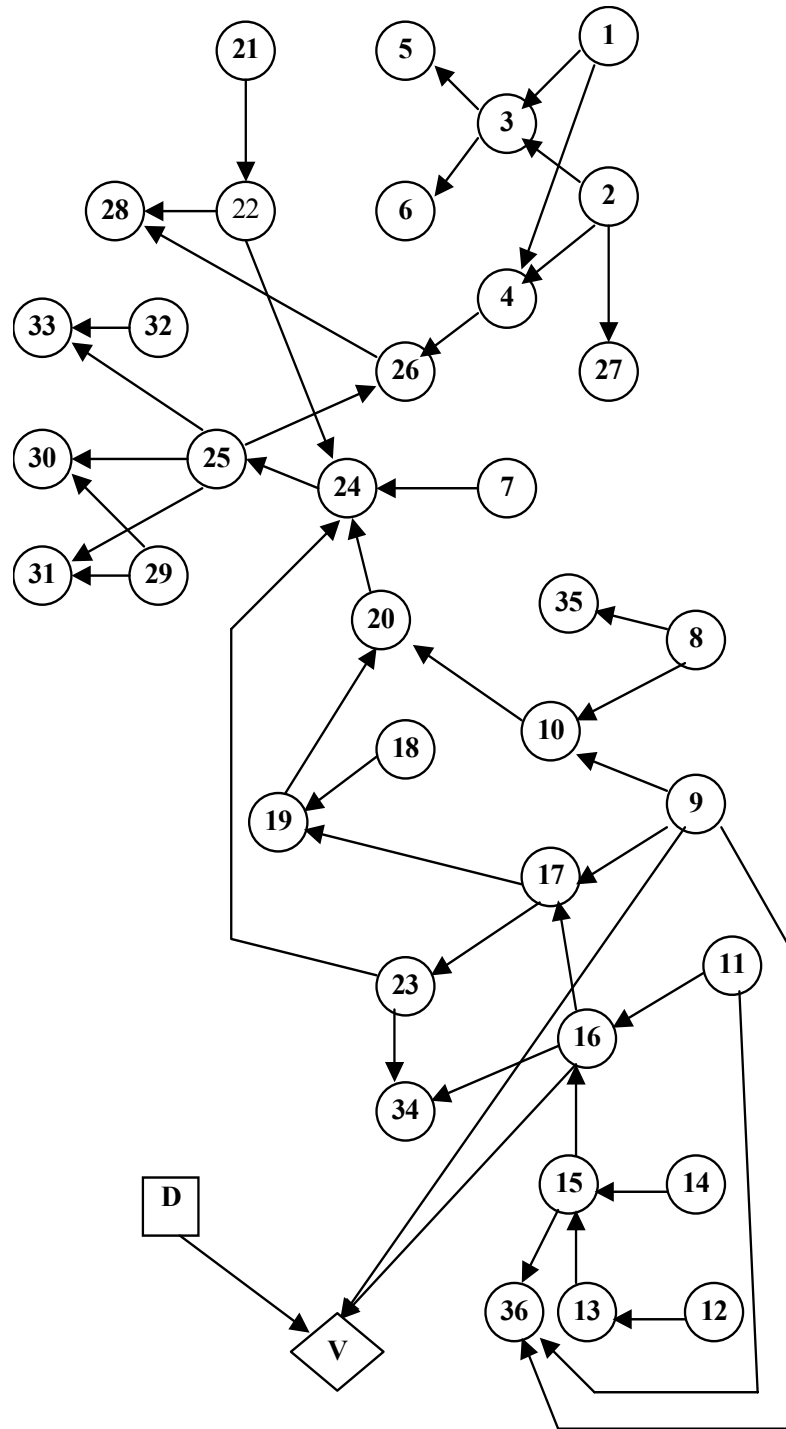


Figure 6-3: Influence diagram for Example 2.

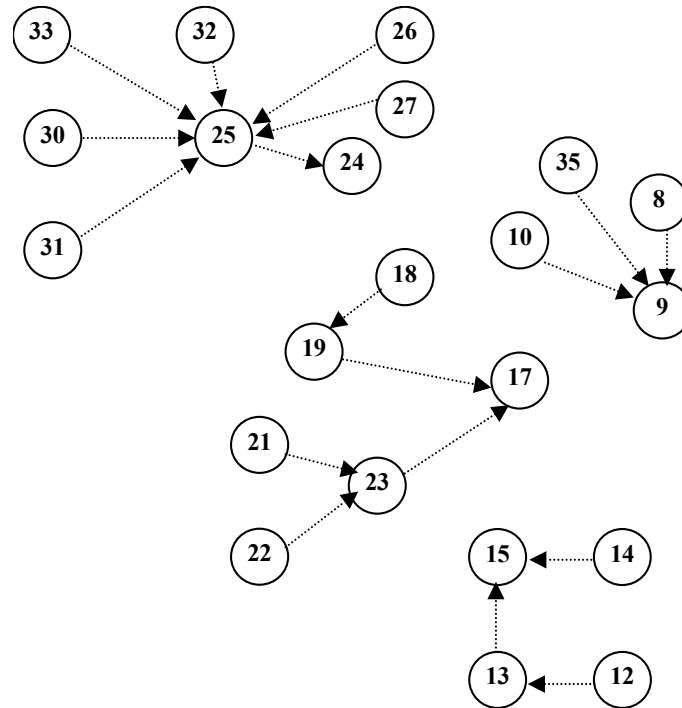


Figure 6-4: The partial ordering of EVPI for Example 2

6.2.3 Computational Issues

In practice, it is straightforward to generate a partial ordering of EVPI by employing a pairwise comparison of nodes and checking for d -separation of one node from the value node by the other. This method can be called as the *pairwise-comparison* approach. This algorithm does not exploit the topological structure of the network to gain efficiency. The next section will introduce a new approach to the identification of partial ordering of EVPI in graphical decision model by identifying barren nodes and extending the u -separation relation to more encompassing neighborhoods. The new algorithm is referred as u -separation extension.

6.3 Efficient Identification of EVPI Orderings

This section first describes a number of extensions of the graphical properties of information relevance for chance nodes. Then, an algorithm is described to exploit these new results to the identification of partial ordering of EVPI in graphical decision models. The new algorithm, called the *u-separation extension*, identifies barren nodes and extends the *u*-separation relation to more encompassing neighborhoods afterwards.

6.3.1 Treatment of Barren Nodes

Omission of barren nodes from a graphical decision model has no effect on the optimal decision policy. Furthermore, their value of information is always bounded by the joint value of information of their direct predecessors.

Theorem 6-3. In a canonical decision model M , let B be a barren node and $\pi(B)$ be the set of direct predecessors of B , and A be a decision node. Then $EVPI_M(A | B) \leq EVPI_M(A | \pi(B))$.

Proof. The result follows from the fact that since a barren node is a sink node with no arc coming out of it, it follows that it is always d-separated by all its parent nodes from the value node (see Figure 6-5). We can also infer the result from the so-called *Markov property* of a DAG, since the value is always a non-descendant of any barren nodes and the required conditional independent relation must hold. \square

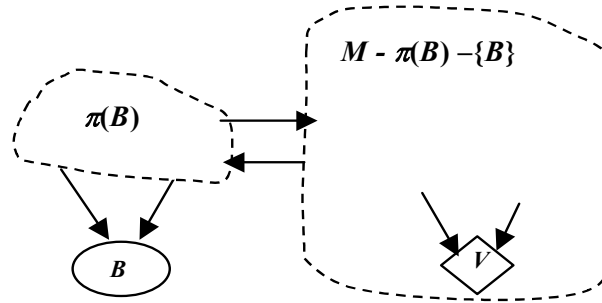


Figure 6-5: EVPI of barren nodes are always bounded by those of their parents

Note that the EVPI of barren nodes are not necessary zero. Take the barren *node 10* in Example 1 (Figure 6-1) as an example. In influence diagrams the direction of arcs can be reversed by changing the conditional probabilities. If we reverse the arc from *node 4* to *node 10*, *node 10* is now similar to *node 2* as a parent of *node 4*. EVPI of both *node 2* and *node 10* are not necessary zero.

Hence, in trying to obtain an EVPI ordering of the chance nodes in a decision model, we may first remove all the barren nodes with single parent because their EVPI is always less than those of their respective parents. Furthermore, removing such barren nodes has no influence on the ordering of other nodes since barren nodes are not in the ancestral sets of any other nodes. After the EVPI ordering of all non-barren nodes has been achieved, we may insert the barren nodes into the ordering to complete the analysis.

However, care must be taken when the barren node B has more than one parent. The theorem only guarantees that the EVPI of B is less than the joint EVPI of all its parents, yet the ordering of chance nodes is for individual node. Thus only barren nodes with single parent will be removed before the computation.

Proposition 6-1. If the two chance nodes X and Y are both disconnected from the value node after the moralization of the ancestral sub-graphs, then $EVPI_M(A | X) = EVPI_M(A | Y)$.

Proof: According to the definition, X and Y are disconnected to the value node after the moralization of the ancestral sub-graphs, either they are connected to each other but disconnected to the value node, or they and the value node are in three disjoint parts, just as shown in the left and right parts in Figure 6-6, then we have $(X, V \perp Y)$ and $(Y, V \perp X)$, which means $EVPI_M(A | X) \leq EVPI_M(A | Y)$ and $EVPI_M(A | Y) \leq EVPI_M(A | X)$, hence $EVPI_M(A | X) = EVPI_M(A | Y)$. \square

The resulting EVPI ordering should not contain the chance nodes described in proposition 1, since these nodes do not guarantee such a sequence. Hence when checking for u -separations, nodes like X and Y in Figure 6-6 can be omitted. Figure 6-6 (a) shows the case that X and Y are connected, V is disconnected them; (b) shows the case that X , Y and V are all disconnected to each other.

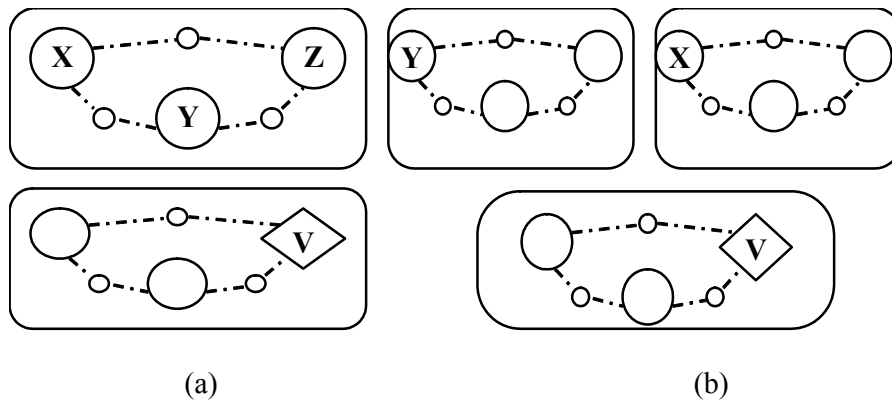


Figure 6-6: Nodes with the same EVPI

6.3.2 Neighborhood Closure Property of u -separation with the Value Node

The Neighborhood Closure Property of u -separation with the value node allows us to infer u -separation relations in a neighborhood thereby eliminating the need to explicitly check for u -separation once u -separation of a single node is established in a neighborhood of a cluster of nodes.

Theorem 6-4. Let G be the moralized graph of a graphical decision model with the decision node removed. Let node X be a chance node, node Y be a neighbor of Z in graph G . Then Y is u -separated from the value node V by X if and only if Z is u -separated from the value node V by X .

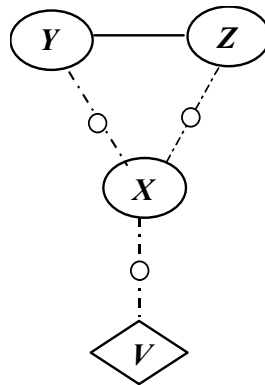


Figure 6-7: Extension of u -separation from value node to a direct neighbor.

Proof: Referring to Figure 6-7, suppose Y is u -separated from the value node V by X . Then every path from Y to V passes through X , and any path from Z to V must either pass through both Y and X or only X alone. No path can run from Z to V without going through X for this will violate the u -separation of Y from V by X . Hence Z is separated from V by X . The converse is also true by symmetry. That is, if Z is u -separated from V by X , then Y is u -separated by V by X . \square

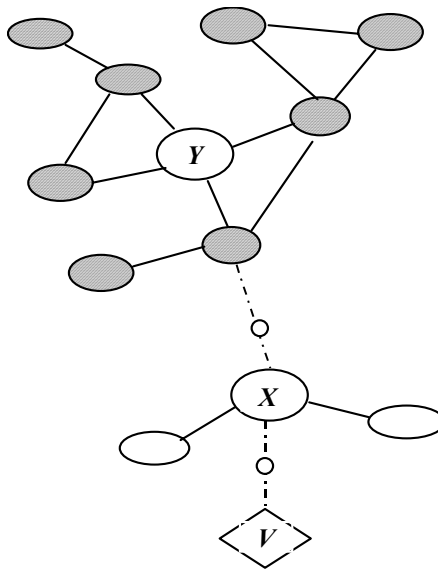


Figure 6-8: U-separation of Y from V by X can be extended to the maximal connected sub-graph containing Y

The above result allows us to check the u -separation of any node with V and if it is found to be true, to recursively add the property to all of their direct neighbors. For example, in the network shown in Figure 6-8, if it is established that Y is u -separated by X from V , then we can infer that all the shaded nodes will also be u -separated by X from V . We state this in the following theorem:

Theorem 6-5. Let G be the moralized graph of a graphical decision model with the decision node removed. If in G , a chance node Y is u -separated by another chance node X from the value node, then the maximal connected sub-graph containing Y is also u -separated from V by X .

Proof: The result follows from the recursive application of Theorem 4. \square

6.3.3 An Algorithm for Identifying EVPI Orderings

Input: An influence diagram M .

Output: An EVPI ordering set Ω of the influence diagram.

1. Convert the network M into canonical form if it is not already so.
2. Remove all the barren nodes with single parent from the chance node set.
3. Drop all the decision nodes in M .
4. Identify the ancestral sub-network of the value node V .
5. Moralize the ancestral sub-network.
6. Let $\Omega = \emptyset$.
7. Let $N \leftarrow C$, the set of chance nodes in M .
8. While $N \neq \emptyset$ do
 9. Mark all nodes in N as “unvisited”
 10. Pick a node $X \in N$
 11. Let $N \leftarrow N \setminus \{X\}$
 12. For each node $Y \in \text{Adj}(X)$ do
 13. If Y is “unvisited” and $Y \neq X$ then
 - i. Mark node Y as “visited”.
 - ii. If Y is u-separated by V given X then
 1. Add the ordering $\{X \leq Y\}$ to Ω
 2. Recursively add all $\{Z \leq Y\}$ to Ω where $Z \in \text{Adj}(Y)$ and Z is “unvisited”
 - iii. Else
 1. Mark all nodes $Z \in \text{Adj}(Y)$ and $Z \neq X$ as visited”.
 - iv. End if

14. End if
15. End for
16. End While
17. For all removed barren nodes B , add $\{B \leq P\}$ to Ω where P is the parent of B .
18. Output Ω

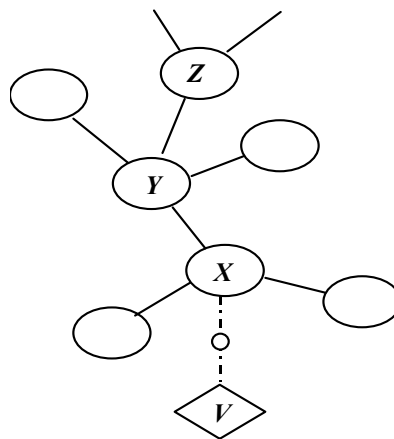


Figure 6-9: Propagation of EVPI from Y to its neighborhood.

The algorithm goes through every chance node and considers it as a separator node. If a neighboring node is found to be u -separated by the current node from the value node, the EVPI ordering is added to the list, and Theorem 6-4 is applied recursively in a depth-first manner to include the ordering of adjacent nodes compared with the current node. Figure 6-9 shows the adjacent node u -separation probing scheme.

In order not to output nodes with equal EVPI, the u -separation procedure is revised as the follows:

***U*-separation procedure:**

Input: moralized sub-graph, start node X , end node V , separator Y

Output: True/False

Initialize all nodes as unvisited

Traverse the sub-graph from the start node, mark visited node

If X and V are disconnected

 If Y and V are disconnected, return (false)

 Else return (true)

 End if

Else

 Cut the separator off

 Traverse

 If connected, return (false)

 Else return (true)

End if

The following provides an estimate of the runtime complexity of *u*-separation extension and compare it to the pairwise-comparison algorithm. For an n -node network (n includes the value node), naïve pairwise-comparison algorithm requires $(n-1)(n-2)$ checks for *u*-separation. The new algorithm performs only $(n-1)$ number of *u*-separation checks and $(n-1)$ searches for adjacent nodes. We adopted the depth-first search (DFS) to traverse the undirected graph and perform *u*-separation checks. The

DFS requires $O(n + m)$ time when m here stands for the number of edges in the moralized undirected sub-graph. In the worst case m will be n^2 . Hence the computational time for the pairwise-comparison algorithm is $O(n^4)$, and for the new one is $O(n^3)$. Therefore, a speed up is typically expected of about n times compared with the naïve algorithm.

6.4 Computational Evaluation of the Algorithm

This section describes the implementation of the *u-separation extension* algorithm and its application to several real world problems

6.4.1 Applications of the Algorithm to Sample Problems

Let us first explore the enhanced performance of *u-separation extension* on Examples 1 and 2. The run times of this algorithm and the pairwise-comparison approach are shown in Table 6-1. It is shown that a significant decrease in run times for the new algorithm over the naïve scheme for both examples. Note that, although Example 2 has much more nodes than Example 1, it has a shorter runtime with a speed up ratio is about 47.85 compared to only 1.67. This significant saving for Example 2 is due mainly to the large number of nodes that are disconnected to the value node in the ancestral graph, thus can be omitted in the *u-separation* search. It is also observed that the more densely the network is connected, the less EVPI ordering we can obtain.

Table 6-1: Comparison of Running Time Using Practical Examples

Decision model	Size of network	Runtime *	Speed up ratio *
Example1	18 nodes	1.150 sec	1.67
Example2	37 nodes	0.160 sec	47.75

*Speed up ratio: Speed up ratio compared with pairwise-comparison. On a P166 using MS

Visual C++.

6.4.2 Combination of Qualitative and Quantitative Methods

In order to explore the practical applications for the algorithm and to study how the algorithm works in larger networks, another experiment adopted the network structure of the Pathfinder Bayesian network for surgical pathology diagnosis (Heckerman et al, 1992) to construct an example influence diagram. This example has 135 chance nodes, one decision node and one value node.

The running time for pairwise-comparison algorithm is 15.6 seconds⁴, while the u-separation extension algorithm takes 3.57 seconds. The average time used in calculating the numerical value of information for a chance node in this example is roughly 25 seconds (based on methods provided in Smile © library, Decision Systems Laboratory, 1998), hence obtaining a complete list of VOI for all the chance nodes in the diagram will need about 1 hour (56 minutes). It is a tedious task to calculate all the numerical value for these chance nodes.

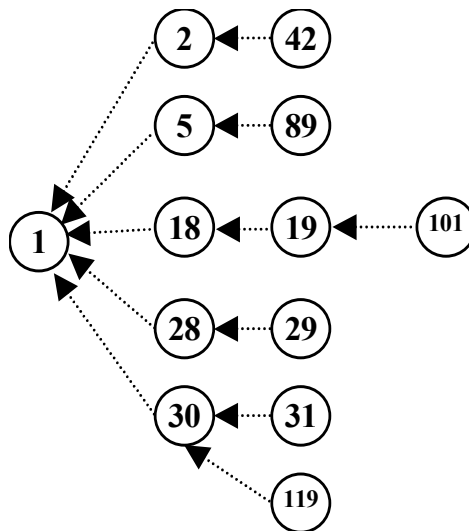


Figure 6-10: Part of the ordering obtained in example.

⁴ *All following programs are run on a PIII350 using MS Visual C++.

Considering the trade off between the time-consuming work of quantitative computations of VOI and the incompleteness of qualitative methods, a heuristic procedure is now proposed to generate a set of N chance nodes with the highest value of information by combining our method with quantitative computation.

The procedure is as follows:

1. Generate an ordering using the u -separation extension algorithm and arrange them into a tree or several trees;
2. Compare the root nodes of each tree using quantitative methods and identify one node with the most importance;
3. Put this node in the output set and remove it from the graph.
4. Let the node's children be the root nodes of the remaining sub-trees and repeat the entire comparison procedure from step 2 until we find the top N nodes with the highest value of information.

This procedure can be illustrated using the ordering that is obtained from the Pathfinder-like example, as shown in Figure 6-10. Suppose that the objective is to identify five nodes with the highest VOI in the network. In Figure 6-10 there is a tree, where node 1 is the only root, so node 1 enters the output set first. Next remove node 1 from the graph, and the remaining part forms five sub-trees with five root nodes. Compare the numerical value of these five nodes and node 18 is the most important one, thus node 18 is put into output set and removed from the graph. Afterwards node 19 becomes a root node, and the procedure continues.

The heuristic procedure has a limitation: if the network is connected very densely, it may be impossible to find enough qualitative orderings to obtain the required N most important nodes. For example, the influence diagram of Figure 6-1 can generate a

partial ordering consists of seven nodes. If ten most important nodes are needed, then we are unable to satisfy the need. One possible solution to this problem is to decompose the network into sub-networks, e.g., decompose the Pathfinder network according to its similarity sub-graphs (Heckerman, 1990). For these sub-graphs, generate a sequence of orderings, and then apply the heuristic indicated above.

6.4.3 Application in Dynamic Decision Models

Using the way introduced in Chapter 3, the dynamic influence diagrams can be changed into canonical form, so the qualitative ordering of chance nodes can be obtained in any single stage of the dynamic decision models. Moreover, the forward interface acts as a separator between the two stages, and it has been proved in Chapter 3 that the chance node in early stages has VOI greater than later stages, hence some temporal arcs could be assumed indicating rankings of the VOI. Summing up the importance ordering of chance nodes in a single stage with the temporal priority, we can have a full picture of qualitative ordering of VOI in DIDs.

6.5 Summary and Conclusion

This chapter has described an algorithm for the identification of partial ordering of EVPI for chance nodes in graphical decision models. The algorithm is based on non-numerical graphical analysis based on the idea of u-separation.

The algorithm is tested on 2 sample networks based structurally on real-world models. A runtime speedup of the algorithm over a naïve approach proposed previously is observed. We also applied both qualitative and quantitative methods on a large example based on Pathfinder Bayesian network, and saw a great difference between

them in computing time. Hence a heuristic combining the two methods together is proposed to obtain both completeness and efficiency.

Knowledge of EVPI orderings of the chance nodes in a graphical decision network can help decision analysts and automated decision systems weigh the importance or information relevance of each node and direct information-gathering efforts to variables with the highest expected payoffs. Theories and experiments show that the methods described in this chapter can serve the purpose well.

A limitation of our approach is that this qualitative method only generates a partial ordering. This is the price for considering only qualitative properties. However, such a trade off of tractability for precision can be extremely valuable when the exact numerical computation of EVPI is intractable.

An observation from experimental results is that clusters which are densely connected tend to produce very sparse partial ordering graph, *i.e.*, nodes that are densely connected tend to resist yielding an ordering with our method. While this may limit the usefulness of our approach, we can exploit this property by clustering such densely connected nodes as one group and treating the group as a single node. A junction tree is an example of such a group of clusters. Those densely connected nodes are grouped together to form cliques which consists of small complete sub-graphs. Then the u -separation extension algorithm can be applied to find partial orderings of groups of nodes.

Another possible extension of this qualitative approach is to consider some heuristic classification of decision models based on their network topology and then to apply different types of search strategies based on such a classification.

Finally, it may be promising to employ methods that decompose large networks into several subnets to be individually processed. The partial orderings obtained may then be merged.

7 Conclusions and Future Work

In this chapter, a summary of the merits and the limitations for all the work conducted is offered to conclude this thesis. Moreover, some possible future directions of research are also pointed out in this chapter.

7.1 Summary

Knowing the outcomes of uncertain factors before taking actions in any decision problems can be beneficial. Whether this information on outcomes is worthy of the effort for gathering it, is the objective of value of information studies.

Researchers have been developing various kinds of methods in computing VOI as guidance for information gathering. Yet it is far from well studied in cases where time effects are explicitly considered, or where decisions on information gathering should be taken under time constrains. This work contributes in VOI computation both in dynamic decision environments and timely occasions.

7.1.1 VOI in Dynamic Models

The study of value of information in dynamic decision models in this work focuses on finding ways to calculate VOI in graphical representations, mainly the dynamic influence diagrams (DIDs).

Properties of VOI in dynamic environment

In Chapter 3, some properties of VOI in general dynamic environments are revealed. It is proved that earlier information is always better or equal to later one without considering the cost of information gathering.

In order to keep accordance with the planning literature, the dynamic decision problems are divided with respect to the capability of observing the state variables.

The partially observable case is discussed separately with an analysis between two representations: DIDs and POMDPs. A boundary of the VOI of an uncertain variable in POMDPs is given thereafter. Finally, the information value for decision intervened chance variables is investigated as well.

Algorithm for VOI computation in DIDs

Chapter 4 offers a way to identify a group of DIDs which can be decomposed into sub-networks with similar structures, and hence a sub-junction tree can be generated based on such sub-networks as the computing template. We discuss methods of reusing the sub-junction tree, including the case when VOI for variables with intervening decisions is under concern.

Both the time value of benefits and costs (e.g., costs for collecting information) are considered by discounting the value in each stage. For societal analysis an equal discounting for both cost and benefit is suggested, however, when individual preference is concerned, the rates may be different. Hence a base rate and a sensitive analysis afterwards are suggested.

An algorithm for computing VOI in DIDs is proposed and the computational complexity discussed. Using our methods, the inference in DIDs without spreading

variables can reach polynomial time when largest clique width is to the order of logarithm of number of total nodes.

Implementation

This method of VOI computation in DIDs is implemented in MATLAB and applied to a practical medical example. The VOI calculated can help doctors choose the best diagnostic tests for cancer patients. The realization is much faster than other existing software in this example. Experimental results show the realization is efficient and significant.

In summary, this Algorithm for VOI computation provides researchers a useful tool of sensitive analysis in dynamic decision making. It is shown to be illustrative and efficient both in theory and in practice.

Analysis of the approximations

Since the inference and VOI computation in general influence diagrams and the partially observable dynamic decision models are intractable, many approximation schemes are introduced to facilitate efficient VOI computation with as less error as possible. Splitting the hardness that lies in the partially observable situations into two parts, the belief update and value function construction, the approximations take advantage of divide-and-conquer or compression techniques. It is unsurprising that the accuracy and efficiency conflict with each other, and users of these approximate methods have to leverage carefully between accuracy and efficiency to serve their objectives well.

Chapter 5 provides an overview of the approximate methods and analyzes their computational complexity, boundary of errors, and the efficiency of application in VOI

computation. Based on this analysis, users are able to choose different approximations which satisfy their requirements better.

7.1.2 Qualitative VOI in General IDs

For the purpose of making timely suggestions on how to collect information, in Chapter 6 of the thesis a partial ordering of all the chance nodes is generated according to their importance.

An algorithm for identifying qualitative EVPI

Chapter 6 has described an algorithm for the identification of this ordering of EVPI for chance nodes in graphical decision models. The algorithm is constructed from a non-numerical graphical analysis on the basis of the idea of u -separation of graphs.

This algorithm has been tested on two sample networks structurally based on real-world models. A speedup in running time over a naïve approach proposed previously is observed. We have also applied both qualitative and quantitative methods on a large example based on the Pathfinder Bayesian network, and a great difference between them in computing time is seen.

A heuristic of hybrid VOI computation

A limitation of this qualitative approach is that this method only generates a partial ordering. This is the price for considering only qualitative properties. Such a trade off of precision for tractability can be extremely valuable when the exact numerical computation of EVPI is very hard to compute, however, when the decision making task requires more information on VOI, the qualitative method itself can be insufficient.

Hence in Chapter 6 a heuristic method is proposed to combine the qualitative and quantitative methods together to obtain higher accuracy while maintaining relative efficiency.

Knowledge of EVPI orderings of the chance nodes in a graphical decision network can help decision analysts and automated decision systems weigh the importance or information relevance of each node and direct information-gathering efforts to variables with the highest expected payoffs. We believe that the methods described in this thesis can serve the purpose well.

7.1.3 Guideline for VOI computation in Decision Models

In summary of all the results obtained in this work, a concluding guideline for VOI computation is composed here to provide decision makers with approaches suitable for their objectives.

Table 7-1 shows this guideline for VOI computation in decision models.

Exact methods

The exact methods are intractable in general graphical models. Typically they are only used to solve some simple decision problems with small state or decision space. Also, in some specially structured models, e.g., the time-invariant dynamic decision models without the spreading variables, the algorithm for calculating the VOI can be of polynomial time. For larger and more complex problems, the exact methods are unlikely to generate results within the resource or time limit.

Qualitative method

Generally speaking, the qualitative approach is the fastest among all methods, yet lacking of completeness. The qualitative method considers the VOI of all the chance nodes in the model simultaneously with the lowest computational cost, yet the loss of information is the largest too. It is recommended for time-critical situations, online information gathering, or online model refinement tasks.

Note that when the intractability of quantitative method is mainly due to large state and action space but not the correlations among many variables, the qualitative method is very useful as it bypasses the computation of those large states and decisions, and yields the ordering directly. While if the intractability of qualitative methods mainly lies in the relationship among variables, for example those DIDs with spreading variables, the qualitative method will be unable to yield many significant results either. In such complicated situation of all variables interacting with each other, neither qualitative nor quantitative methods perform well.

Approximations

Quantitative approximations lie between the qualitative and exact methods in terms of efficiency and accuracy. Approximations in general influence diagrams and states-observable dynamic models are not studied in detail, while those in partially observable dynamic models are analyzed.

Adding information arcs to decisions, or throwing away memories of past information renders a time decomposable dynamic influence diagram, which results in basically polynomial computational complexity. In these time-decomposable models, the more memory of past observations and actions involved in a decomposed sub-model, the larger the size of one stage in the equivalent MDP, the more accurate the value, yet the

more complex the computation. Grid-based value function approximation is also polynomial in most cases (not including variable number of grids). Other approximations, like the weak linkage removal, state variable factoring, state compression, and fully unobservable Markov decision process approximation can also be quite efficient, but they are not guaranteed with polynomial complexity.

Choices of any one or combination of the approximations depend on the underlying model representation, the requirements for accuracy and efficiency.

Note that the simulation methods, e.g., the Monte Carlo and the sampling algorithms are not discussed, as they are not foci of this thesis.

Table 7-1: Guideline for VOI computation

		General Model	Dynamic decision problems	
			State Observable	State Partially Observable
Qualitative		Partial Orderings of VOI	Partial orderings of VOI in a single stage, together with the temporal orderings	
Quantitative	Exact	Probability inference; Evidence propagation; Junction tree	Dynamic junction tree method Structural approximations	One-pass; Witness; Incremental Pruning
	Approximations	Monte Carlo; Non-myopic VOI for a set of chance nodes	Variable abstraction or aggregation	Structural Value approximation
				Structural Belief state approximation
				Parametric approximation

7.2 Future Work

The VOI computation mainly depends on the inference techniques of the graphical decision models. A new, efficient inference method will help the computation of VOI to gain efficiency in general. Hence the future research of VOI computation lies essentially in three areas: first, finding new efficient inference methods both for exact and approximate computation, thus facilitating the VOI computation; second, finding better ways based on existing inference methods to improve the efficiency and quality of VOI computation; and third, exploring methods that calculate VOI directly, maybe bypassing some of the difficulties that lie in calculating twice the decision scenarios with and without information.

For study of VOI in dynamic decision models, future research might focus on efficient ways to find an optimal or near optimal triangulation, and computing VOI for a group of variables simultaneously and applying the method to real world case.

Handling the inference in general graphical or partially observed decision models can be intractable (unless $P=NP$). Yet a number of approximate approaches can be proposed to satisfy different user's objectives. Combinations of several approximations, e.g., the grid-based value function inter-extrapolation, belief space compression by exponential PCA, together with k-stage memory, could be interesting candidate approximations for further study.

As for graph topology, it is observed that clusters which are densely connected tend to produce very sparse partial ordering graph, *i.e.*, nodes that are densely connected tend to resist yielding an ordering with our method. While this may limit the usefulness of our approach, we can exploit this property by clustering such densely connected nodes as one group and treating it as a single node. We can then use our algorithm to find

partial orderings of group of nodes. Another possible extension of our approach is to consider some heuristic classification of decision models based on their network topology and then to apply different types of search strategies based on such a classification. Finally, we note that it may be promising to employ methods that decompose large networks into several subnets to be individually processed. The partial orderings obtained may then be merged.

Reference

- Aberdeen, D (2002). A survey of approximate methods for solving partially observable Markov decision processes. Technical report, Research School of Information Science and Engineering, Australian National University, Canberra, Australia, 2002.
- Acharya, A.K., and Murray, C.J.L. (2000) Rethinking Discounting of Health Benefits in Cost Effectiveness in Cost Effectiveness Analysis. *Global Programme on Evidence for Health Policy*, WHO.
- Aji, S.M., and McEliece, R.J. (2000) The Generalized Distributive Law, *IEEE Transactions on Information Theory*, 46:2, 325-343, March 2000.
- Aoki, M. (1965). Optimal control of partially observable Markovian systems. *J. Frankl.* I 280, pp.367-386.
- Arnborg, S., Corneil, D.G., and Proskurowski, A. (1987) Complexity of finding embeddings in a k-tree, *SIAM Journal of Algebraic and Discrete Methods* 8, 277-84.
- Astrom, K. J. (1965). Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications*, 10L174-205.
- Beinlich, I. A., Suermondt, H. J., Chavez, ., R. M. & Cooper, G. F. (1989). The ALARM Monitoring System: A Case Study with Two Probabilistic Inference

Reference

- Techniques for Belief Networks. *Second European Conference on Artificial Intelligence in Medicine*, London, 38:247-256.
- Beck, J.R. and Pauker, S.G. (1983). The Markov process in medical prognosis. *Medical Decision Making*, 3:419-458.
- Becker, A, Geiger, D. (1996) A sufficiently fast algorithm for finding close to optimal junction trees, *Proceedings of the Twelfth Uncertainty in Artificial Intelligence*, Morgan Kaufmann.
- Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton, NJ.
- Bellman, R. (1961). *Adaptive Control Processes*, Princeton University Press, Princeton, NJ.
- Bellman, R., and Dreyfus, S. (1962). *Applied Dynamic programming*. Princeton University Press, Princeton, NJ
- Bertsekas, D.P. (1987) *Dynamic Programming: Deterministic and Stochastic Models*, Prentice-Hall.
- Bertsekas, D.P. (1976) *Dynamic Programming and Stochastic Control*. Academic Press.
- Boutilier, C. and Dearden, R. (1994) Using abstractions for decision theoretic planning with time constraints. In *Proceedings AAAI-94*, 1016-1022. AAAI.
- Boutilier, C., and Poole, D. (1996). Computing Optimal Policies for Partially Observable Decision Processes using Compact Representations. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI96)*, pp 1168-1175, Portland, OR, 1996.

Reference

- Boyan, X., and Koller, D. (1998) Tractable Inference for Complex Stochastic Processes. Technical report, Stanford University.
- Castillo, E., Gutiérrez, J.M., Hadi, A.S. (1997) Expert Systems and Probabilistic Network Models, Springer.
- Cassandra, A., Littman, M., and Zhang, N. (1997). Incremental Pruning: A Simple Fast Exact Method for Partially Observable Markov Decision Processes. In D. Geiger and P. P. Shenoy (editors), *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann.
- Cassandra, A. (1998). *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. Ph.D. dissertation, Department of Computer Science, Brown University.
- Chan, H. and Darwiche, A. (2002) A Distance Measure for Bounding Probabilistic Belief Change. *AAAI-02*, pp.539-545.
- Chartrand, G. *Introductory Graph Theory*. New York: Dover, 1985.
- Cheng, H. (1988). *Algorithms for partially observed Markov decision processes*, Ph.D. dissertation, Faculty of Commerce and Business Administration, Univ. of British Columbia.
- Clemen, R., T. (1996) Making Hard Decisions. *Second edition*. Duxbury.
- Colbry, D., Peintner, B., and Pollack M. E. (2002) Quantitative temporal relationships in dynamic Bayesian models, In *Proceedings of the 2002 AAAI Spring Symposium on Information Refinement and Revision for Decision Making: Modeling for Diagnostics, Prognostics, and Prediction*, pp 1-8, AAAI Press, Meno Park, California, 2002.

Reference

- Cooper, G.F. (1988) A Method for Using Belief Networks as Influence Diagrams, in *Proceedings of the Fourth Workshop on Uncertainty in Artificial Intelligence*, 55-63.
- Cooper, G.F. (1990) The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial intelligence* 42(2-3), 393-405.
- Cormen, T.H., Leiserson, C.E., and Rivest, R.L. (1990) *Introduction to algorithms*, MIT press, New York, McGraw-Hill.
- Cover, T. M., Thomas, J. A. (1991) *Elements of Information Theory*, John Wiley & Sons.
- D'Ambrosio, B. (1999) Inference in Bayesian Networks, in *AI Magazine*, Vol. 20, No. 2, 21-36.
- Dean, T. and Kanazawa, K. (1988). Probabilistic temporal reasoning. In *AAAI-88*, pp. 524-528.
- Dean, T. and Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3).
- Dean, T. and R. Givan (1997) Model Minimization in Markov Decision Processes, In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Providence, RI, July 1997, pp. 106-111.
- Dechter, R. 1996. Bucket Elimination: A Unifying Framework for Probabilistic Inference. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, eds. E. Horvitz and F. Jensen, 211-219. San Francisco, Calif.: Morgan Kaufmann.
- Derman, C. (1970) *Finite State Markovian Decision Processes*, Academic Press, NY.

Reference

- 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.
- Drummond, M., Stoddart, G., & Torrance, G. (1987). *Methods for the economic evaluation of health care programmes*. New York: Oxford Medical Publications.
- 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.
- Felli, J. C. & Hazen, G. B. (1998) Sensitivity Analysis and the Expected Value of Perfect Information. *Medical Decision Making*, Vol.18, No.1, pp.95-109.
- Friedman, N, Koller D., and Pfeffer A. (1998) Structured Representation of Complex Stochastic Systems. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI)*, pp. 157-164.
- Guestrin, C., D. Koller, and R. Parr (2001). Solving factored POMDPs with linear value functions. *IJCAI-01 Workshop on Planning under Uncertainty and Inc. Info.*, Seattle, WA, 2001.
- Givan, R., S. Leach and T. Dean (1997) Bounded parameter Markov decision processes. Technical report CS-97-05. Brown University, Providence, Rhode Island.
- Gold, M.R., Siegel, J.E., Russell, L.B. and Weinstein, M.C. (1996) *Cost-effectiveness in health and medicine*. Oxford University Press, 1996.
- Gorry, G. A. (1973) Computer-assisted clinical decision making. *Computer-assisted clinical decision making*, 12:45-51.

- Hansen, E. (1998a). An improved policy iteration algorithm for partially observable MDPs. In *Advances in Neural Information Processing Systems 10*. MIT Press.
- Hansen, E. (1998b). Solving POMDPs by searching in policy space. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp. 211-219.
- Hansen, E.A., and Feng, Z. (2000) Dynamic Programming for POMDPs Using a Factored State Representation. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*. Breckenridge, Colorado. 130-139.
- Hauskrecht, M. (1997). *Planning and control in stochastic domains with imperfect information*. Ph.D. thesis, Massachusetts Institute of Technology.
- Hauskrecht, M. (2000) Value-Function Approximation for Partially Observable Markov Decision Processes. *Journal of Artificial Intelligence Research*, 13, 33-94.
- Hazen, G.B. (1992). Stochastic trees: A new technique for temporal medical decision modeling. *Medical Decision Making*, 12:163-178.
- Heckerman, D. (1990) Similarity Networks for the Construction of Multiple-Fault Belief Networks. *Uncertainty in Artificial Intelligence: Proceedings of the Sixth Conference (UAI-1990)*, pp.32-39, Elsevier Science Publishing Company, Inc., New York.
- 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 Kaufman.
- 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*, 31:90-105, 1992.

Reference

- Heckerman, D., and Shachter, R. (1995) Decision-Theoretic Foundations for Causal Reasoning. *Journal of Artificial Intelligence Research*, 3: 405-430.
- Hernandez-Gardiol, N and S. Mahdevan, (2000). Hierarchical memory-based reinforcement learning. In *NIPS-13*.
- Hoey, J., St-Aubin, R., Hu, A., and Boutilier, C. (1999) SPUDD: Stochastic Planning using Decision Diagrams. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 279-288, Stockholm, 1999.
- 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):22-26.
- Howard, R.A. (1967). Value of information lotteries. *IEEE Transactions on Systems Science and Cybernetics*, SSC-3(1), 54-60.
- Howard, R. A. & Metheson, J. E. (1981), Influence Diagrams. In R.A. Howard and J.E. Metheson (editors), *Readings on the principles and applications of decision analysis*, Vol. 2, pp 719-762, SDG, 1984.
- Jensen, F. V., Lauritzen, S. L., and Olesen, K.G. (1990) Bayesian Updating in Causal Probabilistic Networks by Local Computations. *Computational Statistics Quarterly*, (4):269-282.
- Jensen, F., Jensen, F. V., and Dittmer, S., L. (1994) From Influence Diagrams to Junction Trees. In R. L. de Mantaras and D. Poole (editors), *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp.367-373, Morgan Kaufmann.

Reference

- Jensen, F.V., and J. Liang. (1994). drHUGIN - a system for value of information in Bayesian networks. In *Proceedings of the Fifth International Conference on Information Processing and Management of Uncertainty in Knowledge -based Systems (IPMU)*, Paris, France; 178-183.
- Jensen, F.V. (1996) *An Introduction to Bayesian Networks*, UCL press, 1996.
- Kinsella, A. R. (1993). *Colorectal Cancer: A Scientific Perspective*. Cambridge University Press.
- Kjærulff, U. (1990). Graph triangulation - algorithms giving small total state space, Technical Report R 90-09, Aalborg University, Denmark.
- Kjærulff, U. (1992) A Computational Scheme for Reasoning in Dynamic Probabilistic Networks. In *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, pp. 121-129.
- Kjærulff, U. (1994) Reduction of computational complexity in Bayesian networks through removal of weak dependences. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 374-382.
- Kjærulff, U. (1995) dHugin: a computational system for dynamic time-sliced Bayesian networks. *International Journal of Forecasting*, 11: 89-111.
- Kjærulff, U. (1998) Nested Junction Trees, in M. I. Jordan (ed.), *Learning in Graphical Models*, NATO Science Series, Series D: Behavioural and Social Sciences, Vol. 89, pp. 51-74, Kluwer Academic Publishers.
- Kushner, H.J., and A.J., Kleinman (1971). Mathematical programming and the control of Markov chains, *International Journal on Control* 13(5):801-820.

Reference

- 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, 157-224.
- Lauritzen, S.L., and D. Nilsson. (2001) Representing and Solving Decision Problems with Limited Information. *Management Science*, 47: 1238-1251.
- Leong, T. Y. (1994) *An Integrated Approach to Dynamic Decision Making under Uncertainty*. PhD thesis, MIT, Cambridge, MA.
- Leong, T. Y. (1998) Multiple perspective dynamic decision making. *Artificial Intelligence*, 105(1-2): 209-261.
- Leong, T. Y. and C. G. Cao. (1998) Modeling medical decisions in DynaMoL: A new general framework of dynamic decision analysis. In *MEDINFO98*, pp.483-487.
- Lipscomb, J., Weinstein, M. C., & Torrance, G. W. (1996). Time preference. In M. R. Gold, J. E. Siegel, L. B. Russell, & M. C. Weinstein (Eds.), *Cost-effectiveness in health and medicine* (pp. 214-238). New York: Oxford University Press.
- Littman, M.L. (1994). Memoryless policies: Theoretical limitations and practical results. In Cliff, Husbands, Meyer, and Wilson (Eds.) *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*. MIT Press, Cambridge.
- Littman, M. L., A. R. Cassandra, and L. P. Kaelbling, (1995) Learning Policies for Partially Observable Environments: Scaling Up. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 362-370.
- Littman, M. L., R. S. Sutton, and S. Singh (2001). Predictive representations of state. *Proc.NIPS-02*, Vancouver, 2001.

Reference

- Lovejoy, William S. (1991). A Survey of Algorithmic Methods For Partially Observed Markov Decision Processes, *Annals of Operations Research* 28, pp.47-66.
- McCallum, A. (1995) *Reinforcement Learning with Selective Perception and Hidden State*. Ph.D. thesis, University of Rochester.
- Meila, M. and M. I. Jordan (1997). Triangulation by continuous embedding. In *Advances in Neural Information Processing Systems* 9, pp. 557--563.
- Monahan, G. (1982). A survey of partially observable Markov decision processes, *Management Science*, 28:1-16.
- Mundhenk, M., J. Goldsmith, C. Lusena, and E. Allender. (2000) Complexity of finite-horizon Markov decision process problems. *Journal of the ACM*, 47(4):681--720, July 2000. <http://citeseer.nj.nec.com/mundhenk00complexity.html>
- Murphy, K. (2000) A Survey of POMDP Solution Techniques. URL: <http://citeseer.nj.nec.com/murphy00survey.html>
- Ortiz L. E., and Kaelbling L. P. (2000) Sampling Methods for Action Selection in Influence Diagrams. *AAAI* 2000.
- Papadimitriou, C. H., and J. N. Tsitsiklis (1987). The complexity of Markov decision processes. *Mathematics of Operations Research*, 12, 441-450.
- Pearl, J (1988). Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann.
- Pearl, J., Geiger, D. & 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.

Reference

- Pearl, J. (2000) *Causality, Models, Reasoning and Inference*. Cambridge University Press.
- Petitti, D. (1994) *Meta-Analysis Decision Analysis and Cost-Effectiveness Analysis*. Oxford University Press, Inc.
- Platzman, L. K. (1977). *Finite memory estimation and control of finite probabilistic systems*. Ph.D. thesis, Massachusetts Institute of Technology.
- Platzman, L. K. (1980). A feasible computational approach to infinite-horizon partially observed Markov decision problems. Tech. Rep., Georgia Institute of Technology.
- Poh, K.L., & 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., & 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.
- Poupart, P. and Boutilier, C. (2000) Value-Directed Belief State Approximation for POMDPs, In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp.497—506, Morgan Kaufmann San Francisco, CA
- Poupart, P. and Boutilier, C. (2001) Vector-space Analysis of Belief-state Approximation for POMDPs. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pp. 445-452. Morgan Kaufmann San Francisco, CA
- Poupart, P. and C. Boutilier. (2002) Value-directed Compression of POMDPs. To appear at *NIPS-02*.

- Provan, G.M. (1993) Tradeoffs in Constructing and Evaluating Temporal Influence Diagrams, *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*. pp. 40-7
- Puterman, M.L., and Shin, M.C. (1978). Modified policy iteration algorithms for discounted Markov decision problems. *Management Science*, Vol. 24, pp.1127-1137.
- Raiffa, H. (1968) Decision Analysis, Introductory Lectures on Choices under Uncertainty. Random House.
- Robertson, N., and Seymour, P.D. (1995) Graph minors XIII, the disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63: 65-110.
- Rose, D (1974) Triangulated graphs and the elimination process, *J. Math. Anal. Appl.*, 32(1974), 597-609.
- Roy, N. and Gordon, G. (2002) Exponential Family PCA for Belief Compression in POMDPs. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, Cambridge, MA, 2002. MIT Press.
- Russell S. and P. Norvig, Artificial Intelligence, A Modern Approach, Prentice Hall, Englewood Cliffs, New Jersey, 1995.
- Schweitzer, Paul J. (1984). Aggregation methods for large Markov chains, In G. Iazola, P.J. Coutois, and A. Hordijk, editors, *Mathematical Computer Performance and Reliability*, Elsevier Science Publishers.
- Shafer, G.R., and Shenoy, P.P. (1990) Probability Propagation. *Annals of Mathematics and Artificial Intelligence*, vol.2, 327-352, 1990.

- Shachter, R. D. (1986) Evaluating Influence Diagrams. *Operations Research*, 34:871-882.
- Shachter, R.D., and Peot, M.A. (1992) Decision Making Using Probabilistic Inference Methods, in *Proceedings of 8th Conference on Uncertainty in Artificial Intelligence*, 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.
- Shachter, R. D. (1999) Efficient Value of Information Computation. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp.594-601, Morgan Kaufmann.
- Singh, S.P., T. Jaakkola, and M. I. Jordan, (1994). Learning without state-estimation in partially observable Markovian decision processes. In *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 284-292.
- Smallwood, R.D., and Sondik, E.J. (1973). Optimal control of partially observable processes over the finite horizon, *Operations Research*, 21:1071-1088.
- Sondik, E.J. (1971). *The optimal control of partially observable Markov processes*, Ph.D. dissertation, Dept. of EE, Stanford Univ., Stanford, CA.
- Staib, L., Link, K. H. and Beger, H. G. (2000) Evidence-based surgery: Follow-up in colorectal cancer: cost-effectiveness analysis of established and novel concepts.

Reference

- Langenbeck's Archives of Surgery*, DOI: 10.1007/s004230000144, Springer-Verlag.
- Striebel, C. T. (1965). Sufficient statistics in the optimal control of stochastic systems. *Journal of Mathematical Analysis and Applications*, 12:576-592.
- Tatman, J.A., Shachter, R.D. (1990). Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2), pp.365-379.
- Tarjan, R.E., & Yannakakis, M. (1984) Simple Linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal of Computing* 13(3), 566-79.
- Verma, T., & Pearl, J. (1990), Causal Networks: Semantics and Expressiveness. In Shachter, R. D., Levitt, T. S., Kanal, L. N., and Lemmer, J. F. (editors), *Uncertainty in Artificial Intelligence 4*, pp. 69-76, North Holland.
- Wang, C. G., T. Y. Leong, A. P. K. Leong, and F. C. Seow, (2000) Knowledge-based formulation of dynamic decision models in medicine, *Medical Decision Making*, 18(4): 458.
- Wang, C. G. (1999) *Knowledge-based formulation of dynamic decision models in medicine*. Master's thesis, NUS, School of Computing, 1999.
- White C. C. and W.T. Scherer, (1994). Finite memory suboptimal design for partially observed Markov decision processes. *Operations Research*, 42, 439-455.
- White, D.J. (1993) *Markov decision processes*: John Wiley & Sons, Chichester.
- White, D.J. (1978) *Finite Dynamic Programming*, Wiley, Chichester.
- Wilson, R. E. and Donohue, J. H. (1991). Neoplasia of the colorectum: Natural history and prognosis of colorectal cancer. In: S. F. Phillips, J. H. Pemberton and R. G.

Reference

- Shorter, (eds). *The large Intestine: Physiology, Pathophysiology, and Disease*. pp. 521-54, New York, NY: Raven.
- West, D. B. *Introduction to Graph Theory, 2nd ed.* Englewood Cliffs, NJ: Prentice-Hall, 2000.
- Xiang, Y. (1999) Temporally Invariant Junction Tree for Inference in Dynamic Bayesian Network, *Invited contribution in Artificial Intelligence Today: Recent Trends and Developments*, pp 473-487, Springer.
- Xiang, Y.P. and Poh, K.L. (1999) Time-critical dynamic decision making. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 688-95, Morgan Kaufmann.
- 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
- Yannakakis, M. (1981) Computing the minimum fill-in is NP-complete. *Journal of Algebraic and Discrete Methods*, 2(1):77 -79, 1981.

Reference

Zubek, V. B. and T. Dietterich (2001). Two heuristics for Solving POMDPs Having a Delayed Need to Observe. *Proceedings of the IJCAI Workshop on Planning under Uncertainty and Incomplete Information*, pp. 51-60, Seattle, WA, 2001

Murphy, K. (2003) Bayes Net Toolbox for Matlab.

(URL: <http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html>)

Decision Systems Laboratory. University of Pittsburgh (1998), GeNIe and SMILE, (<http://www2.sis.pitt.edu/~genie/>)

Norsys Software Corp. (1998), Netica, *Application for Belief Networks and Influence Diagrams*. (<http://www.norsys.com>)

Appendix A: Concepts and Definitions

Formulation of a Markov Decision Process

A Markov decision process usually consists of the following elements:

1. Decision Epochs. The time point when the decisions are made. It can be finite or infinite, discrete or continuous.
2. State Sets. The system under study occupies a state at each decision epoch. The set of possible system states is denoted as X , which is usually finite and discrete; x_t is the state occurs in time $t \in T$.
3. Action Sets. At the time the decision maker observes the system state in some decision epoch, he may choose an action $a \in A$, where A is the action space which denotes the set of possible actions for every state in all epoch $t \in T$.
4. Values (Gains, or Rewards). Let g_{ij}^a denotes the time-independent value (also called gain or revenue or reward) achievable by a system in state i at each decision epoch, given an action a is selected and its next transition is to state j , and let $g_{ij}^a(t)$ denotes the non-homogeneous (time-dependent) value achievable by system in state i at decision epoch t , given action a is taken and system state is j at decision epoch $t+1$.
5. Transition Probabilities. Let p_{ij}^a denotes the homogeneous probability that a system is in state $j \in X$ at some decision epoch, given action $a \in A$ in state i at the previous

epoch. $p_{ij}^a(t)$ is the non-homogeneous probability for system state is i at $t-1$ and j at t , with action $a \in A$. We usually assume $\sum_{j \in X} p_{ij}^a = 1$, and $\sum_{j \in X} p_{ij}^a(t) = 1$.

6. Decision Rules and Policies. A decision rule $\delta_t: X \rightarrow A$ of a system at decision epoch t specifies the action choice when it occupies state $x \in X$ at t . For each $x \in X$, $\delta_t(x) \in A$. A sequence of such functions is called policy, $\pi = \{\delta_1, \delta_2, \dots\}$.

Formulation of a Semi-Markov Decision Process

Holding Times. In SMDPs, the transition from state i to state j given action a is made only after the process is kept for a time $\tau_{ij}^a(t)$ in state i at time t . This time $\tau_{ij}^a(t)$ is called holding time, which is a random number with corresponding probability mass function $h_{ij}^a(m, t)$. $P(\tau_{ij}^a(t) = m) = h_{ij}^a(m, t)$.

Values. The value $g_{ij}^a(t)$ of a process consists of the yield rate $y_{ij}^a(\sigma)$ and bonus $b_{ij}^a(\tau)$. $y_{ij}^a(\sigma)$ is the reward earned at each time stage from beginning with state i till reaching state j with action a . $b_{ij}^a(\tau)$ is the bonus earned when the process transfer from state i to j given action a at time τ .

Formulation of a Partially-Observable Markov Decision Process

$X = \{1, 2, \dots, n\}$ and $\Theta = \{1, 2, \dots, m\}$ denote finite state and message sets respectively. Let A denote a finite action set, and the set of probability distributions on X is $M(X) = \{\mu \in R^n: \mu \geq 0, \sum_{i=1}^n \mu_i = 1\}$. The process is initiated with a known probability distribution over the state space X , $\mu_1 \in M(X)$. Let $H_t = \{\mu_1, a_1, \theta_1, a_2, \theta_2, \dots, a_{t-1}, \theta_{t-1}\}$ denote the history of actions and messages received up to time t with this

initial distribution. If based on this information, the decision maker chooses action a_t , then define:

A real-valued reward $g(x_t, a_t)$ is received if the state of the system is x_t .

The system transits to another state j in accordance with the known transition probabilities $p_{ij}^a = P\{x_{t+1} = j: x_t = i, a_t = a\}$.

A message $\theta_t \in \Theta$ is received in accordance with the known probabilities $r_{jk}^a = P\{\theta_t = k: x_{t+1} = j, a_t = a\}$.

Time increments by one, $H_{t+1} = H_t \cup \{a_t, \theta_t\}$, the decision maker must choose action a_{t+1} , and the process repeats.

The reward can be included in the message θ_t . If the number of time periods $T < \infty$, an additional salvage value $\alpha(i)$ is received at the beginning of time $T+1$ if $x_{T+1} = i$. The decision maker seeks a policy $\delta_t: H_t \rightarrow A$ that maximizes the expected net present value of the time stream of rewards accrued during the process:

$$E\left\{\sum_{t=1}^T \beta^{t-1} g(x_t, \delta_t(H_t)) + \beta^T \alpha(x_{T+1})\right\} \quad (\text{A.1})$$

$\beta \geq 0$ is an economic discount factor. If $T = \infty$, β is required to be < 1 , $\beta^T = 0$.

Backward Recursive equation in MDPs

Let $V_i(t)$ be the optimal total expected revenue, given the starting state of the system is i .

$$V_i(t) = \max_a \left\{ \sum_{j \in X} p_{ij}^a [g_{ij}^a + V_j(t+1)] \right\}, \quad t \in T \quad (\text{A.2})$$

Here $V_j(N+1)$ denotes an additional salvage value received at the beginning of time $N+1$ given $x_{N+1} = j$.

Generally, for non-homogeneous transition probabilities and returns, with an economic discounting factor β (e.g., interest rate), equation (A.2) becomes:

$$V_i(t) = \max_a \left\{ \sum_{j \in X} p_{ij}^a(t) [g_{ij}^a(t) + V_j(t+1)] \right\}, \quad t \in T \quad (\text{A.3})$$

Backward Recursive equation in SMDPs

Let $q_{ij}^a(m, t) = p_{ij}^a(t) h_{ij}^a(m, t)$ denote the transition function from state i to j , with action a , for duration m after entering state i at time t . Let η be defined as the number of stages remaining for consideration in the planning horizon.

$$\begin{aligned} V_i(t) = \max_a \left\{ \sum_{j \in X} \sum_{m=\eta+1}^{\infty} q_{ij}^a(m, t) \left[\sum_{l=0}^{\eta-1} \alpha^l y_i^a + \alpha^\eta V_i(N+1) \right] \right. \\ \left. + \sum_{j \in X} \sum_{m=1}^{\eta} q_{ij}^a(m, t) \left[\sum_{l=0}^{m-1} \alpha^l y_i^a(l) + \alpha^m b_{ij}^a + \alpha^m V_j(t+m) \right] \right\}. \quad t \in T \end{aligned} \quad (\text{A.4})$$

The first part of right end of the equation is the expected revenue if the next transition out of state i occurs after time duration η , and the second part is the expected value if the next transition occurs before time duration η .

POMDPs

Linear support algorithm

Let the vectors $\gamma \in \Gamma_t$ be indexed by integers, i.e., $\Gamma_t = \{\gamma^1, \gamma^2, \dots, \gamma^{v_t}\}$ if $|\Gamma_t| = v_t$ ($|\Gamma_t|$ denotes the cardinality of the set Γ_t). Apparent $\Gamma_{T+1} = \{\alpha\}$. Given Γ_{t+1} , the expression (A.1) reduces to

$$V_t^*(\pi) = \max_a \{ \mu^T [g^a + \sum_{\theta=1}^m P^\theta R^\theta(\theta) \gamma^{l(\mu, a, \theta)}] : a \in A \} \quad (\text{A.5})$$

Here $l(\mu, a, \theta)$ is the index of the maximizing γ vector in $\max \{ \mu^T P^\theta R^\theta(\theta) \gamma \mid \gamma \in \Gamma_{t+1} \}$, and the maximizing $a \in A$ will be an optimal action from μ in time t .

The linear support algorithm starts with Γ_{t+1} given $\Gamma_{N+1} = \{\alpha\}$. Then calculate the operative gradients at the extreme points of $M(X)$, and puts these in set G_t , an approximation of Γ_t . For each $\gamma_0 \in G_t$, construct the convex region $R(\gamma_0) = \{ \mu \in M(X) \mid \mu^T \gamma_0 \geq \mu^T \gamma, \text{ all } \gamma \in G_t \}$. $v_t(\mu) = \max \{ \mu^T \gamma \mid \gamma \in G_t \}$ is used to approximate $V_t^*(\mu)$, the error incurred at any $\mu \in R(\gamma_0)$ will be $V_t^*(\mu) - \mu^T \gamma_0$. The maximum of this error will be obtained at an extreme point of $R(\gamma_0)$, so check the error at all extreme points of each region to find the maximal error over all $M(X)$ for using v_t instead of V_t^* . If this maximum error is zero, $G_t = \Gamma_t$, the iteration is completed. If it is positive, then the vertex that achieves the maximum error has a gradient vector associated with it, which is not included in G_t . Add this vector to G_t and repeat generating and checking of extreme points.

CONCEPTS IN INFORMATION THEORY

Def. Conditional entropy:

$$H(Y | X) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(y | x) = -E_{p(x,y)} \log p(y | x) \quad (\text{A.6})$$

Chain Rule for conditional entropy:

$$H(X, Y) = H(X) + H(Y|X) \qquad H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Def. Relative entropy:

$$D(p || q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} = E_p \log \frac{p(x)}{q(x)} \quad (\text{A.7})$$

Def. Mutual information:

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (\text{A.8})$$

$$I(X; Y) = D(p(x,y) || p(x)p(y)).$$

$$I(X; Y) = I(Y; X) = H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X, Y) \quad (\text{A.9})$$

Chain rule for entropy:

$$H(X_1, X_2, \dots, X_n) = \sum_{i=1}^n H(X_i | X_{i-1}, \dots, X_1)$$

The conditional mutual information of random variable X and Y given Z is: $I(X; Y|Z) =$

$$H(X|Z) - H(X|Y, Z) = E_{p(x,y,z)} \log (p(X, Y|Z)/p(X|Z)p(Y|Z))$$

Chain rule for information:

$$I(X_1, X_2, \dots, X_n; Y) = \sum_{i=1}^n I(X_i; Y | X_{i-1}, \dots, X_1)$$

Chain rule for relative entropy:

$$D(p(x,y)||q(x,y)) = D(p(x)||q(x)) + D(p(y|x)||q(y|x))$$

Information inequality:

$$D(p||q) \geq 0 \quad \text{with equality if and only if } p(x) = q(x) \text{ for all } x.$$

Non-negativity of mutual information:

For any two random variables X and Y , $I(X;Y) \geq 0$ with equality if and only if X and Y are independent.

$$H(X) \leq \log |\mathcal{X}| \quad \text{with equality if and only if } X \text{ has a uniform distribution over } \mathcal{X}.$$

$$H(X|Y) \leq H(X) \quad \text{with equality if and only if } X \text{ and } Y \text{ are independent.}$$

Independence bound on entropy:

Let X_1, X_2, \dots, X_n be drawn according to $p(x_1, x_2, \dots, x_n)$.

$$H(X_1, X_2, \dots, X_n) \leq \sum_{i=1}^n H(X_i)$$

$D(p||q)$ is convex, and $H(p)$ is concave function of p . The mutual information $I(X;Y)$ is a concave function of $p(x)$ for fixed $p(y|x)$ and a convex function of $p(y|x)$ for fixed $p(x)$. $(X, Y) \sim p(x,y) = p(x)p(y|x)$.

Appendix B: VOI Given Dependencies Among Mapping Variables

In Chapter 3 section 3.2 the value of clairvoyance for decision intervened variables are discussed. Adding mapping variables and converting an ID to its canonical form is required in such cases. In this procedure, assuming independencies between chance nodes given different decisions will simplify the conversion and the computation for VOI. However, if the mapping variables $X(d_i)$ are correlated, i.e., nested as shown in Figure 3-10 (c), we need to reassess the probabilities for the outcomes of $X(d_i)$ to make sure the conversion is equivalent. Hence the actual VOI for knowing X before D will be different from the VOI calculated based on the simplification.

In this appendix, a study will be carried on for the influence in VOI given such correlation among $X(d_i)$.

Binary decision and binary random variable

First let us consider the simplest case of a binary decision D and a binary random variable X . Let d_1 and d_2 be the two choices of D and x_1, x_2 be the two outcomes of X . If $X(d_i)$ are dependent of each other, suppose for the same outcome the two variables are positively correlated, i.e., $P((d_1, x_1), (d_2, x_1)) > P(d_1, x_1) \cdot P(d_2, x_1)$, $P((d_1, x_2), (d_2, x_2)) > P(d_1, x_2) \cdot P(d_2, x_2)$. Then in order to obtain equivalent expected value (utility) it should be negatively correlated for different outcomes, i.e., $P((d_1, x_1), (d_2, x_2)) < P(d_1,$

$x_1) \cdot P(d_2, x_2), P((d_1, x_2), (d_2, x_1)) > P(d_1, x_2) \cdot P(d_2, x_1)$. Since the original expected value will not change after the conversion, the difference is only result in the case of knowing information of $X(D)$ before D . We have $V' = \sum P((d_i, x_k), (d_j, x_l)) \max_d V((d_i, x_k), (d_j, x_l))$. Denote the expected value for the real case as V_r' and the simplified case as V_s' , let $V(d_1, x_1), V(d_1, x_2), V(d_2, x_1)$ and $V(d_2, x_2)$ be v_{11}, v_{12}, v_{21} and v_{22} respectively. So we have:

$$\begin{aligned}
 V_r' = & P((d_1, x_1), (d_2, x_1)) \cdot \max(v_{11}, v_{21}) + P((d_1, x_1), (d_2, x_2)) \cdot \max(v_{11}, v_{22}) + \\
 & P((d_1, x_2), (d_2, x_1)) \cdot \max(v_{12}, v_{21}) + P((d_1, x_2), (d_2, x_2)) \cdot \max(v_{12}, v_{22}) \quad (\text{B-1})
 \end{aligned}$$

$$\begin{aligned}
 V_s' = & P(d_1, x_1) P(d_2, x_1) \cdot \max(v_{11}, v_{21}) + P(d_1, x_1) P(d_2, x_2) \cdot \max(v_{11}, v_{22}) + \\
 & P(d_1, x_2) P(d_2, x_1) \cdot \max(v_{12}, v_{21}) + P(d_1, x_2) P(d_2, x_2) \cdot \max(v_{12}, v_{22}) \quad (\text{B-2})
 \end{aligned}$$

S.t.

$$\begin{aligned}
 & P((d_1, x_1), (d_2, x_1)) \cdot v_{12} + P((d_1, x_1), (d_2, x_2)) \cdot v_{12} + P((d_1, x_2), (d_2, x_1)) \cdot v_{21} + P((d_1, x_2), \\
 & (d_2, x_2)) \cdot v_{21} = P(d_1, x_1) P(d_2, x_1) \cdot v_{12} + P(d_1, x_1) P(d_2, x_2) \cdot v_{12} + P(d_1, x_2) P(d_2, x_1) \cdot \\
 & v_{21} + P(d_1, x_2) P(d_2, x_2) \cdot v_{21} \quad (\text{B-3})
 \end{aligned}$$

$$\begin{aligned}
 & P((d_1, x_1), (d_2, x_1)) \cdot v_{11} + P((d_1, x_1), (d_2, x_2)) \cdot v_{22} + P((d_1, x_2), (d_2, x_1)) \cdot v_{11} + P((d_1, x_2), \\
 & (d_2, x_2)) \cdot v_{22} = P(d_1, x_1) P(d_2, x_1) \cdot v_{11} + P(d_1, x_1) P(d_2, x_2) \cdot v_{22} + P(d_1, x_2) P(d_2, x_1) \cdot \\
 & v_{11} + P(d_1, x_2) P(d_2, x_2) \cdot v_{22} \quad (\text{B-4})
 \end{aligned}$$

We know that covariance is the measure of correlation between random variables. For binary random variables X_A and X_B , $cov(X_A, X_B) = P(AB) - P(A)P(B) = [P(B|A) - P(B)]P(A)$, so X_A and X_B are either positively correlated, uncorrelated or negatively correlated depending on whether $P(B|A)$ is greater than, equal to or less than $P(B)$. The binary random variables $X(d_1), X(d_2)$ here might not be exactly (0, 1) valued, however, we can always convert them into (0, 1) variables through a simple linear transformation. So our following conclusion can be applied to general case:

$$\begin{aligned} \text{cov}(X(d_1), X(d_2)) &= E(X(d_1), X(d_2)) - E[X(d_1)] E[X(d_2)] \\ &= P((d_1, x_1), (d_2, x_1)) - P(d_1, x_1) \cdot P(d_2, x_1), \end{aligned}$$

Suppose this value is >0 (positively correlated), then

$$\begin{aligned} P((d_2, x_2), (d_2, x_1)) - P(d_2, x_2) \cdot P(d_1, x_1) &= -P((d_1, x_1), (d_2, x_1)) + P(d_1, x_1) \cdot P(d_2, \\ x_1) &= -\text{cov}(X(d_1), X(d_2)) < 0, \end{aligned}$$

$$P((d_2, x_1), (d_1, x_2)) - P(d_2, x_1) \cdot P(d_1, x_2) = -\text{cov}(X(d_1), X(d_2)) < 0,$$

$$\begin{aligned} P((d_2, x_2), (d_1, x_2)) - P(d_2, x_2) \cdot P(d_1, x_2) &= P((d_1, x_1), (d_2, x_1)) - P(d_1, x_1) \cdot P(d_2, \\ x_1) &= \text{cov}(X(d_1), X(d_2)) > 0. \end{aligned}$$

Deduct equation (B-1) from equation (B-2), we have:

$$\begin{aligned} V_s' - V_r' &= -[P((d_1, x_1), (d_2, x_1)) - P(d_1, x_1) \cdot P(d_2, x_1)] \cdot \max(v_{11}, v_{21}) - [P((d_1, x_1), (d_2, \\ x_2)) - P(d_1, x_1) \cdot P(d_2, x_2)] \cdot \max(v_{11}, v_{22}) - [P((d_1, x_2), (d_2, x_1)) - P(d_1, x_2) \cdot P(d_2, x_1)] \cdot \\ \max(v_{12}, v_{21}) &- [P((d_1, x_2), (d_2, x_2)) - P(d_1, x_2) \cdot P(d_2, x_2)] \cdot \max(v_{12}, v_{22}) \end{aligned}$$

$$= -\text{cov}(x_1, x_2) \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ + \max(v_{11} v_{21}) \\ - \max(v_{11} v_{22}) \\ - \max(v_{12} v_{21}) \\ + \max(v_{12} v_{22}) \end{pmatrix} \quad (\text{B-5})$$

So the difference in the VOI calculated will be dependent on both the covariance and the value for different decisions. The latter is known when we calculate the original model; hence it needs no additional knowledge. When one state overruns the other, i.e., the value for x_1 is greater or smaller than x_2 regardless of the decisions (monotonic value function), $v_{11} > v_{12}$, $v_{21} > v_{22}$, the above formula is positive with positive $\text{cov}(X(d_1), X(d_2))$. Otherwise, while the preference of states is different for different decisions (convex value function), e.g., $v_{21} > v_{22}$ we have $v_{11} < v_{12}$, (B-5) becomes negative. Note that above conclusions are based on the assumption that there are no dominant

alternatives in the model. When one alternative is dominant to the other, the difference is zero, since the VOI in this case will definitely be zero, no matter how to transform the problem.

When one state is dominant to the other and there is a positive correlation, the VOI calculated in simplified case will be higher than the actual case, thus the independence assumption boasts the value of information of observing the variable X before D .

If the random variables given different decisions are negatively correlated, which means the sign of each covariance is the opposite of this scenario, then we have $V_s' - V_r' > 0$, i.e., the computed VOI based on independence assumption will be underestimated.

We can also obtain the range of (B-5) since $\text{cov}(X(d_1), X(d_2))$ falls in the range $[-1, 1]$.

Let $R = \max(v_{11}, v_{21}) - \max(v_{12}, v_{21}) - \max(v_{11}, v_{22}) + \max(v_{12}, v_{22})$. R is the difference between the second and the third largest value. Hence:

$$V_s' - V_r' \in [-|R|, |R|]$$

These upper and lower limits are indifferent of the dominance of states and the correlation. As long as there is no dominant decision, the error we might make while assuming independency is at most the difference of the two middle values of the value function. Hence we know when we have no idea of the correlations between these parent nodes given different decisions, how much value will we overestimate or underestimate at most if assuming they are independent.

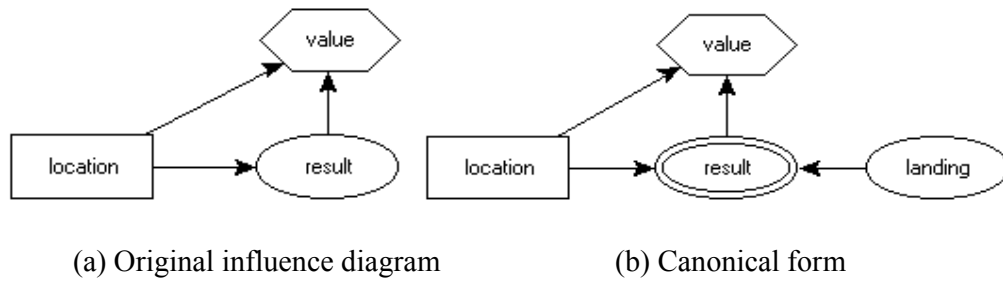


Figure B-1: Example of space exploration

An example is given to illustrate the conclusion. Consider the hypothetical case of sending a rocket to Mars or Venus (Adopted from Ezawa, 1994). The chance of success is dependent of the decision; the values are shown in the following table.

Table B-1: Space exploration

Location & result	Probability	Value
Mars Success	0.6	50
Mars Failure	0.4	10
Venus Success	0.7	100
Venus Failure	0.3	-10

To convert the original problem into canonical form, we reassess the probabilities for the three scenarios, independent causes, positively correlated causes and negatively correlated causes. The probabilities and the value of information calculated in each scenario are shown in table B-2.

Table B-2: Space exploration with different relations between causes

Location & result	Probability		
	Independent	Positively correlated	Negatively correlated
Mars, success Venus, success	0.42	0.48	0.36
Mars, success Venus, failure	0.18	0.12	0.24
Mars, failure Venus, success	0.28	0.22	0.34
Mars, failure Venus, failure	0.12	0.18	0.06
Value of Information	13.2	10.8	15.6

The results are similar to what we have predicted: when causes are positively correlated, the VOI calculated assuming independency will be higher than in the actual case, which means we might willing to pay more to the clairvoyance than he actually deserves; and if the random variables given different decisions are negatively correlated, the computed VOI based on independence assumption will be less than it actually is, and we might overlook the importance of gathering information for a certain chance variable.

Multiple decision and binary random variable

Further let us suppose we have a decision node with m alternatives, but the chance variable is still binary. Assume first there are only two causes are correlated, e.g., $X(d_1)$ and $X(d_2)$, and the other causes are independent.

$$V_s' - V_r' = \sum_{j=3 \dots m} \dots \sum_{i=1}^2 P(x_i, d_j) \sum_{i=1, j=1}^2 [P(x_i, d_1)P(x_j, d_2) - P((x_i, d_1), (x_j, d_2))] \cdot \max_{d_1 \dots d_m}(V | x_i, d_j)$$

(B-6)

As we has done, denote $P((d_1, x_1), (d_2, x_1)) - P(d_1, x_1) \cdot P(d_2, x_1)$ as $\text{Cov}(X(d_1), X(d_2))$, then:

$$V_s' - V_r' = \text{Cov}(X(d_1), X(d_2)) \{ \sum \dots \sum P(d_j, x_i) \max(V | (d_1, x_1), (d_2, x_1), \dots (d_m, x_i)) - \sum \dots \sum P(d_j, x_i) \max(V | (d_1, x_1), (d_2, x_2), \dots (d_m, x_i)) - \sum \dots \sum P(d_j, x_i) \max(V | (d_1, x_2), (d_2, x_2), \dots (d_m, x_i)) + \sum \dots \sum P(d_j, x_i) \max(V | (d_1, x_2), (d_2, x_2), \dots (d_m, x_i)) \}$$

It can be denoted as the following multiplies of matrices:

$$\text{cov}(x_1, x_2) \begin{pmatrix} p_{31} \cdots p_{i1} \cdots p_{m1} \\ -p_{31} \cdots p_{i1} \cdots p_{m1} \\ -p_{31} \cdots p_{i1} \cdots p_{m1} \\ p_{31} \cdots p_{i1} \cdots p_{m1} \\ \vdots \\ p_{32} \cdots p_{i2} \cdots p_{m2} \\ -p_{32} \cdots p_{i2} \cdots p_{m2} \\ -p_{32} \cdots p_{i2} \cdots p_{m2} \\ p_{32} \cdots p_{i2} \cdots p_{m2} \end{pmatrix}^T \begin{pmatrix} \max(v_{11} v_{21} v_{31} \cdots v_{i1} \cdots v_{m1}) \\ \max(v_{12} v_{21} v_{31} \cdots v_{i1} \cdots v_{m1}) \\ \max(v_{11} v_{22} v_{31} \cdots v_{i1} \cdots v_{m1}) \\ \max(v_{12} v_{22} v_{31} \cdots v_{i1} \cdots v_{m1}) \\ \vdots \\ \max(v_{11} v_{21} v_{32} \cdots v_{i2} \cdots v_{m2}) \\ \max(v_{12} v_{21} v_{32} \cdots v_{i2} \cdots v_{m2}) \\ \max(v_{11} v_{22} v_{32} \cdots v_{i2} \cdots v_{m2}) \\ \max(v_{12} v_{22} v_{32} \cdots v_{i2} \cdots v_{m2}) \end{pmatrix} \quad (\text{B-7})$$

If $V(d_1, x_i)$ and $V(d_2, x_i)$ have no effect in the maximum function, i.e., d_1 and d_2 are dominated by other alternatives, then (B-7) will be equal to zero, and assuming all are independent will not influence the VOI calculated, since these two alternatives can be deleted and after the deletion the other causes are independent.

If d_1 and d_2 dominate other decisions, i.e., $V(x_i, d_1)$ and $V(x_i, d_2)$ are greater than other values $V(x_i, d_j)$, then this makes other alternatives invalid and reduce the case to above binary decision scenario.

If there is no dominant decisions, suppose $V(d_i, x_l)$ is the maximum among all the values, we can trim off half of the summations and reduced (B-7) to:

$$\mathbf{cov}(x_1, x_2) \begin{pmatrix} p_{31} \cdots p_{i2} \cdots p_{m1} \\ -p_{31} \cdots p_{i2} \cdots p_{m1} \\ -p_{31} \cdots p_{i2} \cdots p_{m1} \\ p_{31} \cdots p_{i2} \cdots p_{m1} \\ \vdots \\ p_{32} \cdots p_{i2} \cdots p_{m2} \end{pmatrix}^T \begin{pmatrix} \mathbf{max}(v_{11} v_{21} v_{31} \cdots v_{m1} v_{i2}) \\ \mathbf{max}(v_{11} v_{22} v_{31} \cdots v_{m1} v_{i2}) \\ \mathbf{max}(v_{12} v_{21} v_{31} \cdots v_{m1} v_{i2}) \\ \mathbf{max}(v_{12} v_{22} v_{31} \cdots v_{m1} v_{i2}) \\ \vdots \\ \mathbf{max}(v_{12} v_{22} v_{32} \cdots v_{m2} v_{i2}) \end{pmatrix} \quad (\text{B-8})$$

This procedure can be done repeatedly until the next maximum value is among $X(d_1)$ and $X(d_2)$. In such cases, suppose the maximum of the value function is v_{22} , formula (B-8) can be further reduced to:

$$\begin{aligned} & \mathbf{cov}(x_1, x_2) \begin{pmatrix} p_{3j} \cdots p_{ij} \cdots p_{mj} \\ \vdots \\ -p_{3j} \cdots p_{ij} \cdots p_{mj} \end{pmatrix}^T \begin{pmatrix} \mathbf{max}(v_{11} v_{21} v_{3j} \cdots v_{ij} \cdots v_{mj}) \\ \vdots \\ \mathbf{max}(v_{12} v_{21} v_{3j} \cdots v_{ij} \cdots v_{mj}) \end{pmatrix} \\ &= \mathbf{cov}(x_1, x_2) \begin{pmatrix} p_{3j} \cdots p_{ij} \cdots p_{mj} \\ -p_{3j} \cdots p_{ij} \cdots p_{mj} \end{pmatrix}^T \begin{pmatrix} \mathbf{max}(v_{11} v_{21} \cdots v_{ij} \cdots v_{mj}) \\ \mathbf{max}(v_{12} v_{21} \cdots v_{ij} \cdots v_{mj}) \end{pmatrix} \\ &= \mathbf{cov}(x_1, x_2) p_{3j} \cdots p_{mj} [\mathbf{max}(v_{11} v_{21} \cdots v_{ij} \cdots v_{mj}) - \mathbf{max}(v_{12} v_{21} \cdots v_{ij} \cdots v_{mj})] \end{aligned}$$

The value in the quadric braces is the difference of two middle value of the value function. This is quite similar to the binary decision case. That is, adding more independent causes of different decisions will not change our previous conclusion much.

If more than one pair of such correlated causes exist among all the causes, the final influence depends on the co-effects of all the pairs. They can be of the same direction, or mutually subsided, hence it's hard to determine.

Moreover, if more causes for different alternatives are correlated, we are unable to tell if the independency assumption will increase the VOI calculated or not. If the problem is extended to multi-state and multi-decision case it will become more complicated and harder to estimate.

The above analysis proves that we need to be careful while using the independency assumption.

