Stepwise-Decomposable Influence Diagrams

(Nevin) Lianwen Zhang and David Poole
Department of Computer Science
University of British Columbia
Vancouver, B.C., V6T 1Z2, Canada
E-mail: lzhang@cs.ubc.ca poole@cs.ubc.ca

Abstract

In the literature, authors mostly talk about influence diagrams that are regular, no-forgetting and having one value node. In this paper, a less restrictive concept of influence diagrams, i.e, the concept of stepwise-decomposable influence diagrams is introduced and studied.

As a knowledge representation scheme, stepwise-decomposable influence diagrams are more natural than no-forgetting influence diagrams not only because they are less restrictive, but also because they corresponds exactly to stepwise-solvability.

Stepwise-decomposable influence diagrams are advantageous for the following reasons. Firstly, they enable the representation of knowledge about a piece of information being irrelevant to a decision, and consequently enable us to prevent the decision tables from being unnecessarily large at the problem formulation stage. Secondly, stepwise-decomposability allows us to clearly establish the relationship between influence diagram evaluation and Bayesian net computations. Thirdly, because of the relationship, we have the freedom to evaluate an influence diagrams in a number of ways, while Shachter's arc reversal and node reduction algorithm is only one of them, and not necessarily the most efficient one. Finally, stepwise-decomposable influence diagrams provide a convenient setting for investigating when an arc into a decision node can be harmlessly removed by a preprocessing step. This enables us to prevent, in the evaluation process the decision tables from being unnecessarily large.

On a more abstract level, stepwise-decomposable influence diagram also lead to a simple and insightful mathematical theory of influence diagrams, particularly efficient and easy to understand algorithms.

1 INTRODUCTION

Influence diagrams were introduced by Miller *et al* (1976) and Howard and Matheson (1984) as a graphical representation of decision problems. In comparison with decision trees (Raiffa 1968), influence diagrams are more intuitive and can be easily understood by people in all walks of life and degrees of proficiency. They are equally precise and more structured, thus allow more efficient treatment by computers.

An influence diagram is an acyclic directed graph with three types of nodes: chance nodes, decision nodes, and a single value node. They respectively represent random quantities, decisions to make and the utilities of alternatives.

In a one decision maker situation, decisions are made sequentially. So, if an influence diagram is a representation of a single decision maker's view of the world, then it is reasonable to require it to be *regular*, that is, all the decision nodes are ordered (Howard and Matheson 1984 and Shachter 1986).

Arcs into a decision node are interpreted as indications of information availability (Howard and Matheson 1984). More specifically, an arc from a node a to a decision node d means that the value of (the variable represented by) a is known at the time the decision d is to be made. The lack of an arc from a to d means that information about a is not available at the time when d is to be made. Thus, if an influence diagram represents a single decision maker who does not forget information, any decision node and its parents must be parents of subsequent decision nodes (Howard and Matheson 1984). If it is the case, the diagram is said to be *no-forgetting* (Shachter 1986).

A drawback of interpreting arcs into decision nodes as indications of information availability is that knowledge about a node being irrelevant to a decision can not be represented. Information about a node a may be irrelevant to a decision d even if it is available at the time the decision d is to be made. See section 2 for an example.

This inability to represent irrelevancies results in decision tables of sizes exponential in the number of decision nodes and their parents. Suppose there are 10 decision nodes, and every decision node have three chance node parents that they share with only subsequent decision nodes. Also suppose all the nodes have two possible values. Then the size of the decision table for the last decision nodes is 2^{39}.

In this paper, we interpret arcs into decision nodes as indications of potential functional dependency. More specifically, the existence of an arc from a to d means that the decision d potentially depends on the value of a, while the lack of an arc from a to d implies that information about a is irrelevant to the decision d. Under this interpretation, it is no longer reasonable to require that any decision node and its parents be parents of all the subsequent decision nodes. In other words, the no-forgetting requirement has to be lifted.

We also allow influence diagrams to represent situations where more than one person is making the decisions. As a consequence, the regularity requirement has to be lifted, for there may be groups of decisions which can be made in any relative orders. See section 2 for an example.

The one-value requirement is also lifted in this paper for technical convenience and for simpler and more efficient algorithms.

Lifting of the one-value-node requirement, regularity, and the no-forgetting requirement leads to a very general concept of influence diagrams (section 2).

Usually influence diagrams in the most general sense are not stepwise-solvable. That is, in general, they can not be evaluated step by step by considering one decision at a time. One has to consider potentially all the decisions at the same time. The complexity of doing so may be prohibitive. For this reason, we shall impose a new requirement called stepwise-decomposability (section 5) on influence diagrams.

Even though the ability to represent knowledge about a node being irrelevant to a decision is an important motivation for studying stepwise-decomposable influence diagrams, it is not the most important one. The most important motivation for studying stepwise-decomposable influence diagrams is that they lead to a simpler and more insightful mathematical theory of influence diagrams. They also lead to simpler and more efficient algorithms.

The organization of the paper is as follows. Section 2 introduces a very general concept of influence diagrams and gives a brief exposition of its semantics. The computational problem of evaluating an influence diagram is defined in section 3, together with some other technical concepts. In section 4, an attempt is made to formalize the intuitive concept of stepwise-solvability. Stepwise-decomposable influence diagrams are introduced in section 5. In section 6, we show that evaluating stepwise-decomposable influence diagrams amounts to computing marginal probabilities in local Bayesian nets. It is also shown that, in a sense, stepwise-decomposability and stepwise-solvability are equivalent. In section 7, the issue as to when an arc into a decision node can be harmlessly removed by a preprocessing step is addressed. Conclusions are provided in section 8.

2 INFLUENCE DIAGRAMS: DEFINITION AND SEMANTICS

Let us begin by giving a definition of influence diagrams. Our definition is more general than all the definitions in the literature to the best ot our knowledge.

An *influence diagram* I is a triplet $I = (V, A, \mathcal{P})$, where

1. The first component V is a set of nodes, which partitions into a set of *chance nodes* C, a set of *decision nodes* D, and a set of *value nodes* U;
2. The second component A is a set of arcs in V, which, together with V, constitutes a connected acyclic directed graph $G = (V, A)$[1], in which all the value nodes are leaves; and
3. The third component $\mathcal{P}$ is a set of conditional probabilities $\{P(x|\pi(x)) : x \in C \cup U\}$, where $\pi(x)$ denotes the set of all the parents of x[2].

According to this definition, an influence diagram contains the conditional probabilities of all the chance nodes and value nodes, but not the decision nodes. The conditional probability of a value node v is obtained from a value function. More specifically, $P(v = v_0|\pi(v) = X_0)$ is 1 if v_0 is the expected value when the parents $\pi(v)$ of v assume the values X_0, and it is 0 otherwise (see the example below.).

Briefly, the semantics of an influence diagram are as follows. Decision nodes represent decisions to be made, chance nodes represent random variables that are relevant to the decision problem, and value nodes represents the utilities of alternatives.

[1] In this paper, well known graph theory terms such as connected acyclic directed graphs, parents, children, nondescendants, leaves, and roots will be used without giving the definitions. The reader is directed to Lauritzen *et al* (1990) for exact definitions.

[2] Note that when x is a root, $\pi(x)$ is empty. When it is the case, $P(x|\pi(x))$ stands for the prior probability of x.

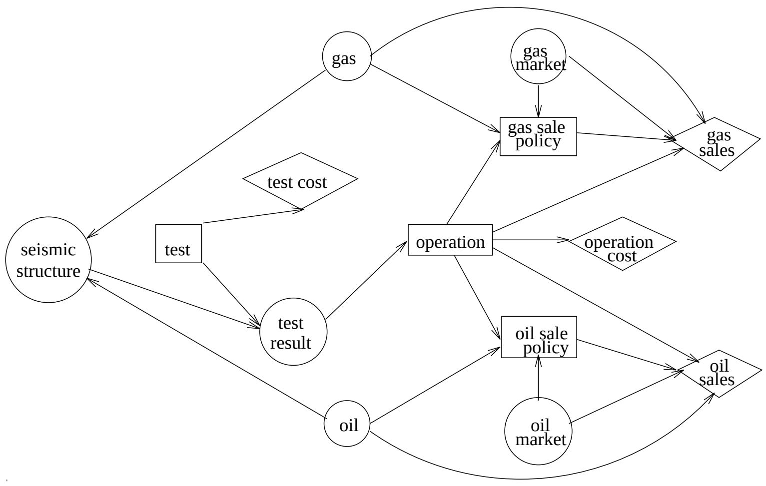

Figure 1: A representation of the oil company problem in terms of influence diagrams.

Arcs into chances nodes and value nodes represent probabilistic dependence. They are thus called *conditional* arcs. A value node or a chance node x depends on its parents $\pi(x)$ in the way as specified by the conditional probability $P(x|\pi(x))$. On the other hand, it is independent of all its nondescendants given all its parents.

Arcs into decision nodes denote potential functional dependence. They are thus called *decision dependency* arcs. The existence of an arc from a node a into a decision node d means that the decision d depends on information about a, in the sense that different values of a may result in different decisions about d even when the values of the other parents of d are fixed. On the other hand, the lack of an arc from a to d means that information about a is irrelevant to the decision d. That is, given the values of all the other parents of d, the value of a does not affect the decision d at all.

Note that our interpretation of arc into decision nodes is different from the one in the literature (Howard and Matheson 1984, Shachter 1986), where an arc from a node a to a decision node d was interpreted as an indication that information about a is available at the time when the decision d is to be made. Thus such arcs were called *informational* arcs. A drawback of this interpretation is that knowledge about a node being irrelevant to a decision can not be represented (see the following example).

In an influence diagram, decisions at decision nodes are made based on the values of their respective parents and are made to maximize the expected value of the sum of all the value nodes.

Figure 1 shows an example influence diagram. As a convention, decision nodes are drawn as rectangles, chance nodes as circles, and value nodes as diamonds. The story goes as follows.

An oil company is considering whether or not it should open up operations in a new area. The company will first decide whether or not to order a seismic structure test. Based on the test results, the company will decide whether or not to open the operation. When oil and gas are produced, the company will then decide its gas and oil sale policies according to market information and the quality of gas and oil produced. All those decisions are to be made to maximize the company's profit.

The root chance nodes are "gas", "oil", "gas-market" and "oil-market". The test results depend on the decision to test and the seismic structure, which in turn depends on quality and quantity of gas and oil underground.

If ordered, the test will cost the company some money. The operation itself has a cost too. The company can make a certain amount of money from selling gas as well as from selling oil. Gas (oil) sales depends on gas (oil) sale policy, the quantity of gas (oil) and the gas (oil) market. The company's profit is the sum of gas sales and oil sales minus test cost and operation cost.

To simplify the problem, we have assumed that the quality and quantity of gas and oil that will have been produced are the same as the quality and quantity of gas and oil underground. We have also assumed that costs in selling gas and oil are absorbed in gas and oil sales.

In this example, one normally will not concern oneself with the "test-result" when making the "gas-sale-policy", for at that time one knows the quantity and quality of gas produced, which, plus market information, are all what s/he needs to make the decision. Information about "test-result" is irrelevant to the decision "gas-sale-policy", even though it may be available. Because we interpret arcs into decision nodes as indications of potential functional dependency, we are able to represent such a piece of knowledge by not drawing an arc from "test-result" to "gas-sale-policy".

Also in the example, it may be the case that "gas-sale-policy" is made by people in the "gas department", and "oil-sale-policy" is made by people in the "oil department", while "operation" is made by the executive board of the company. So, even though both the decision "gas-sale-policy" and the decision "oil-sale-policy" will be made after the decision "operation", they can be made in any relative order. This supports abandoning of regularity.

To end this section, we introduce two concepts that will be useful later. A *parametric influence diagram* is simply an influence diagram, except that some of its conditional probabilities contains *parameters* —- variables of the problem that do not appear in the underlying graphical structure.

A *Bayesian net* is an influence diagram without decision and value nodes.

3 Computations in influence diagrams

In this section, we specify what we want to compute in an influence diagram.

Let $I = (V, A, \mathcal{P})$ be an influence diagram, where V partitions into C, D, and U. For any variable $x \in V$, let Ω_x denote the *frame* of x, i.e, the set of possible values of x. For each decision node $d \in D$, the decision maker is to decide upon a function, called the *decision function* for d, from the Cartesian product $\prod_{x\in\pi(d)} \Omega_x$ of the frames of the parents of d to the frame Ω_d of d itself. A collection of such functions, one function for each decision node, constitute a *policy* for the influence diagrams.

A function $f : \prod_{x\in\pi(d)} \Omega_x \rightarrow \Omega_d$ can be represented by the conditional probability $P_f(d|\pi(d))$ defined as follows:

$$P_f(d = d_0|\pi(d) = X_0) = \begin{cases} 1 & \text{if } f(X_0) = d_0 \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where d_0 stands for a value of d and X_0 stands for an array of values of variables in $\pi(d)$. Given a policy S, we shall use $P_S(d|\pi(d))$ denote the conditional probability that represents the decision function $f_d \in S$ for d.

Given a policy for S for I, a decision node d of I can be viewed as a (deterministic) chance node with the conditional probability $P_S(d|\pi(d))$. The value nodes can always be viewed as chance nodes. Consequently, the influence diagram I itself can be viewed as a Bayesian net. Let us denote it by I_S. The prior joint probability (Zhang and Poole 1992) of the Bayesian net I_S is denoted by P_{I_S} or simply P_S. Explicitly, P_S is given by

$$P_S(C, D, U) = [\prod_{x\in C\cup U} P(x|\pi(x))][\prod_{x\in D} P_S(x|\pi(x))]. \quad (2)$$

For any $x{\in}C{\cup}U$, the conditional probability $P(x|\pi(x))$ is given as part of the specification of the influence diagram, while for any $x{\in}D$ the conditional probability $P_S(x|\pi(x))$ is given by the policy S.

Let v be a value node of I. Let P_S^v denote the marginal probability of v under P_S. The expectation $E_S(v)$[3] of v is given by

$$E_S(v) = \sum_{v_0\in\Omega_v} P_S^v(v_0)v_0.$$

The *value* $V_S(I)$ of I given a policy S is defined by

$$V_S(I) =_{def} \sum_{v\in U} E_S(v).$$

The *optimal expected value* $V(I)$ of I is defined by

$$V(I) =_{def} max\{V_S(I)|S \text{ is a policy for } I\}. \quad (3)$$

The optimal value of an influence diagram without value nodes is zero.

An *optimal policy* S_0 of I is one that satisfies

$$V_{S_0}(I) = V(I). \quad (4)$$

For an influence diagram without value nodes, all policies are optimal.

In this paper we shall only consider variables with finite frames. Hence there are only finite possible policies. Consequently, there always exists at least one optimal policy. *To evaluate an influence diagram* is to

1. find an optimal policy, and
2. find the optimal expected value.

4 STEPWISE-SOLVABILITY

The *decision function space* of a decision node d is the set $\{f : (\prod_{x\in\pi(d)} \Omega_x) \rightarrow \Omega_d\}$ of all the possible decision functions for d. The *policy space* of an influence

[3]Note that $E_S(v)$ is not a function of v. Rather it is a number. The notation is standard in Probability Theory, where the expectation of a random variable ξ is written as $E(\xi)$.

diagram is the Cartesian product of the decision function spaces of all the decision nodes.

A naive way to evaluate an influence diagram is to exhaustively search through the policy space. The complexity of doing so may be prohibitive. Hence, it would be desirable that an influence diagram be stepwise-solvable, i.e, it can be evaluated by searching through the decision function space for one decision node at a time, without ever having to consider the Cartesian product of the decision function spaces for any more than one decision nodes.

In this section, we shall formalize the concept of stepwise-solvability and illustrate that not all general influence diagrams are stepwise-decomposable. In the next section, we shall describe a subclass of influence diagrams that are stepwise-solvable.

To formally define the concept of stepwise-solvability, we need first introduce two other terms. Let d be a decision node in an influence diagram. Suppose, for a moment, that the decision functions for all other decision nodes are given. Each decision function for d, together with those decision functions for all the other decision nodes, constitutes a policy for the diagram, and thus results in an expected value of the diagram. The decision functions for d can be ranked according to the corresponding expected values. This ranking is called the *conditional ranking of the decision functions for d* given the decision functions for all other decision nodes.

To replace a decision node d by a *deterministic node* characterized by a function $f : \prod_{x\in\pi(d)} \Omega_x \rightarrow \Omega_d$ is to replace d by an new random node with the same name and the same frame, and to set $P(d|\pi(d))$ to be the conditional probability that represents f in the sense of equation (1).

An influence diagram is *stepwise-solvable* if it contains only one decision node, or there exists a decision node d, called a *candidate decision node*, such that

1. the conditional ranking of the decision functions for d given the decision functions for all other decision nodes is independent of those decision functions, and

2. if d is replaced by a deterministic node characterized by a decision function of d, the resulting influence diagram is stepwise-solvable.

If an influence diagram is stepwise-solvable, then it can be evaluated as follows. Let d be a candidate decision node. First find an optimal decision function for d in the way as specified in the next paragraph, replace d by a deterministic node characterized by the optimal decision function, resulting in another stepwise-solvable influence diagram I' with one less decision node. Then recursively apply the procedure to evaluate I'.

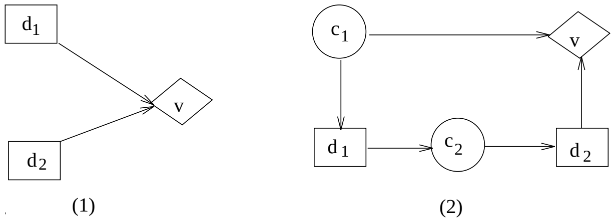

Figure 2: Two influence diagram skeletons that are not stepwise-solvable.

To find an optimal decision function for d, one first replaces all other decision nodes by deterministic nodes characterized by arbitrary decision functions, resulting in an influence diagram with only one decision node d. One can then find an optimal policy for d by searching through the decision function space of d. Because d is a candidate decision node, the optimal decision function for d such found does not depend on the choices of decision functions for the other decision nodes.

A directed graph $G = (V, A)$, with a partitioning of V into C, D, and U is called a *influence diagram skeleton*. From a given influence diagram skeleton, many influence diagrams can be defined by specifying the frames of the variables and the conditional probabilities of the variables in C and U given their parents. On the other hand, an influence diagram has only one corresponding influence diagram skeleton.

An influence diagram skeleton is *stepwise-solvable* if all the influence diagrams that can be defined from it are stepwise-solvable.

The two influence diagram skeletons shown in Figure 2 are not stepwise-solvable. One can easily construct from them influence diagrams that are not stepwise-solvable. The idea is that the value function can be so defined that it can not be separated into the summation or a multiplication of a function of d_1 and a function of d_2.

In the next section, we shall introduce the concept of stepwise-decomposability. One result of this paper is that stepwise-decomposability is equivalent to stepwise-solvability.

5 STEPWISE-DECOMPOSABLE INFLUENCE DIAGRAMS

In this section, we introduce stepwise-decomposable influence diagrams. We need the concepts of moral graph and m-separation. The *moral graph* $m(G)$ of a directed graph G is the undirected graph obtained from G as follows: for each node of G, *marry* all the parents of d, i.e add an edge between every pair of its parents, and ignore all the directions. See Lauritzen and Speigehalter (1988) for more on this concept. If

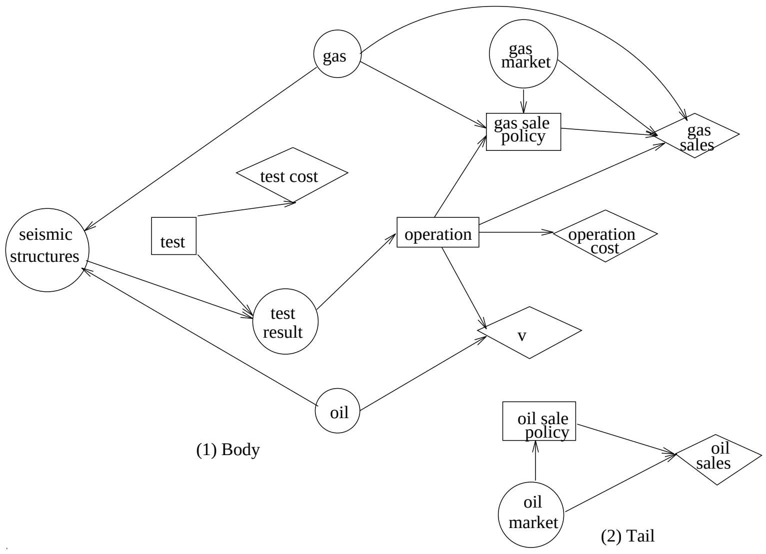

Figure 3: Tail and Body.

two nodes a and b of a directed graph G are separated in the moral graph $m(G)$ by a set X, we say that a and b are *m-separated* by X in G.

Let I be an influence diagram skeleton and d be a decision node in I. Let $nond(d)$ denote the set of decision nodes that are not descendents of d, and let $nond^*(d)$ be the set of the parents of the nodes in $nond(d)$ and the nodes in $nond(d)$ themselves.

An influence diagram skeleton is *stepwise-decomposable* if for each decision node d, the nodes in $nond^*(d)$ are either parents of d or are m-separated from d by the parents of d. An influence diagram is *stepwise-decomposable* if its skeleton is.

The influence diagram in Figure 1 is stepwise-decomposable.

Remark: A *no-forgetting influence diagram skeleton* (Shachter 1986) is an influence diagram skeleton in which (1) there is only one value node; (2) there is a directed path that contains all the decision nodes and the value node, with the value node coming last; and (3) each decision node and its parents are parents of all subsequent decision nodes. It is evident that the no-forgetting property implies stepwise-decomposability. So, all the theorems and algorithms for stepwise-decomposable influence diagrams apply to no-forgetting influence diagrams as well.

We now proceed to define the technical terms tail and body for stepwise-decomposable influence diagrams. Let $I = (V, A, \mathcal{P})$ be a stepwise-decomposable influence diagram, and let d be decision node. Let $m(G)$ be the moral graph of the directed graph $G = (V, A)$. Let G_1 be the undirected graph obtained from $m(G)$ by removing all the nodes in $\pi(d)$ of d. The *downstream Y_d of d (more accurately, of $\pi(d)$)* is the set of all the nodes of G_1 that are connected to d, with d itself excluded. The *upstream X_d of d* is the set of nodes of G_1 that are not connected to d.

Let G_2 be the undirected graph obtained from $m(G)$ by removing all the nodes in Y_d. Let $\pi'(d)$ be the set of nodes in $\pi(d)$ that are connected, in G_2, to at least one node in X_d.

The *tail* I_d^t of I w.r.t. d is the influence diagram obtained from I by restricting it to the set $Z =_{def} Y_d \cup \{d\} \cup (\pi(d) - \pi'(d))$. Particularly, the tail I_d^t contains and only contains the conditional probabilities of the chance an value nodes in Z.

The tail I_d^t of I is a parametric "influence diagram"[4] over Z. Some of the conditional probabilities of this influence diagram, hence its optimal expected value, contain potentially all the variables in $\pi'(d)$ as parameters. Let $V_d^t(\pi'(d))$ denote the optimal expected value of I_d^t.

The *body* I_d^b of I w.r.t. d is the influence diagram obtained from I by

1. restricting it to the set $X_d \cup \pi'(d)$. Particularly, the conditional probabilities of nodes not in $X_d \cup \pi'(d)$, or equivalently, of nodes in Z, are removed.
2. creating a new value node v and drawing an arc to v from each node in $\pi'(d)$, and

[4] Here, we put the term influence diagrams in quotes because the tail I_d^t may not be an influence diagram in the strict sense. The "prior probabilities" of some of the roots may not sum to 1. Fortunately, this fact does not affect our theory at all. For the sake of simplicity, we choose to stick to the term rather then come up a more accurate one.

3. setting the conditional probability $P(v|\pi'(d))$ to be the conditional probability that represents the function $V_d^t(\pi'(d))$ in the sense of equation (1).

Figure 3 illustrate the concepts of tail and body. In the influence diagram shown in Figure 1, "oil-sale-policy" (osp) is a decision node. The tail w.r.t. "oil-sale-policy" is shown in Figure3 (2), which contains the prior probability $P(\textit{oil-market})$ and the conditional probability $P(\textit{oil-sales}|\textit{operation, oil-sale-policy, oil-market, oil})$, where "operations" and "oil" are parameters. Let $V_{osp}^t(\textit{operation, oil})$ be the optimal expected value of the tail. The body w.r.t. "oil-sale-policy" is shown in Figure 3 (1), where the conditional probability $P(v|\textit{oil, operation})$ of the newly introduced value node v given "oil" and "operation" represents the function $V_{osp}^t(\textit{operation, oil})$.

Note that if an influence diagram contain only one decision node, then its body does not contain any decision node. This fact will be used in the evaluation algorithm of the next section.

A influence diagram is *simple* if it contains only one decision node. The influence diagram in Figure 3 (2) is simple.

A decision node is a *leaf decision node* if none of it descendants are decision nodes. The following proposition is straightforward.

Proposition 1 *Suppose an influence diagram I is stepwise-decomposable and d is a leaf decision node of I. Then*

1. *The tail I_d^t of I w.r.t. d is a simple influence diagram.*

2. *The body I_d^b of I w.r.t. d is a stepwise-decomposable influence diagram.*

6 EVALUATING STEPWISE-DECOMPOSABLE INFLUENCE DIAGRAMS

In this section, we present a recursive algorithm which evaluates stepwise-decomposable influence diagrams in a stepwise fashion by considering one decision at a time and by computing conditional probabilities in local Bayesian nets that correspond to sections of the diagram. Let us begin with simple influence diagrams.

6.1 EVALUATING SIMPLE INFLUENCE DIAGRAMS

Suppose I is a simple influence diagram. Let d denote the only decision node and let v_1, ..., v_m be the value nodes. An optimal decision function for d can be found by searching through the entire decision function space for d. More specifically, we can proceed as follows.

For each possible decision function $f : \prod_{x\in\pi(d)} \Omega_x \rightarrow \Omega_d$ for d, replace d by a deterministic node characterized by f, resulting in a Bayesian net I_f. For i from 1 to m, compute the marginal probability $P_{I_f}(v_i)$ of v_i in the Bayesian net I_f, and the expectation $E_{I_f}(v_i)$ from $P_{I_f}(v_i)$. The value $V_f(I)$ of I given the decision function f can now be obtained by using

$$V_f(I) = \sum_{i=1}^{m} E_{I_f}(v_i).$$

After the value $V_f(I)$ having been computed for all the decision functions f for d, pick one such function f_0 whose corresponding value $V_{f_0}(I)$ is the maximum. Then f_0 is an optimal decision function for d.

The above procedure has transformed the problem of evaluating the simple influence diagram I into the task of computing the marginal probabilities in the Bayesian nets I_f. The problem of computing marginal probabilities in Bayesian nets has been under extensive study in recent years. See Pearl (1986), Lauritzen and Speigehalter (1988), Shafer and Shenoy (1988), Jensen *et al* (1990), and Zhang and Poole (1992) for elegant algorithms.

Note that as a by-product of finding an optimal policy for a simple influence diagrams, its optimal expected value is also determined.

6.2 THE GENERAL CASE

Let I be a stepwise-decomposable influence diagram and let d be a leaf decision of I. The tail I_d^t of I is a simple influence diagram[5]. The parents of d in I_d^t are the nodes in $\pi(d) - \pi'(d)$. Some conditional probabilities of I_d^t may possibly contain nodes in $\pi'(d)$ as parameters. Thus, an optimal decision function for d in the tail I_d^t depends on the nodes in $\pi'(d)$, as well as nodes in $\pi(d) - \pi'(d)$. For each value of $\pi'(d)$, we get an optimal decision function for d in I_d^t. Let $\pi'(d)$ run through all its possible values, we get a function from $\prod_{x\in\pi(d)} \Omega_x$ to Ω_d. Let us call the function an *optimal decision function* for d in the original influence diagram I.

Two influence diagrams are *value-equivalent* if they have the same optimal expected value.

Theorem 1 *Suppose I is a stepwise-decomposable Influence diagram. Then*

1. *An optimal decision function for a leaf decision node d and an optimal policy of the body I_d^b of I w.r.t. d constitute an optimal policy for I itself.*

[5]See footnote 4.

2. *The influence diagrams I_d^b and I are value-equivalent.*

The proof of the theorem can be found in the appendix.

The theorem suggests the following procedure EVALUATE(I) for evaluating a stepwise-decomposable influence diagram I.

Procedure EVALUATE(I):

- Input: I – a stepwise-decomposable influence diagram.
- Ouput: an optimal policy for I and the optimal expected value $V(I)$ of I.

1. If I does not contain decision nodes, then N-EVALUATE(I),
2. Else find a leaf decision node d, and
   - S-EVALUATE(I_d^t),
   - EVALUATE(I_d^b),
   - $V(I) \leftarrow V(I_d^t)$.

where S-EVALUATE is a procedure that evaluates simple influence diagrams. It takes a simple influence diagram, returns its optimal expected value and an optimal policy. N-EVALUATE is a procedure that computes the expected value of an influence diagram without decision nodes.

Several remarks are in order.

1. Theorem 1 and Proposition 1 together have shown that stepwise-decomposable influence diagrams are stepwise-solvable. So, an influence diagram skeleton is stepwise-solvable if it is stepwise-decomposable. For any influence diagram skeleton that is not stepwise-decomposable, it is not hard to build from it an influence diagram that is not stepwise-solvable[6]. Therefore, as far as influence diagram skeletons are concerned, stepwise-decomposability and stepwise-solvability are equivalent.
2. All the tail influence diagrams encountered by the EVALUATE procedure are simple, and they correspond to disjoint sections of the original influence diagram. Furthermore evaluating a simple influence diagram is just a matter of computing marginal probabilities in Bayesian nets[7]. It is in this sense that we say we have transformed the problem of evaluating stepwise-decomposable influence diagrams into problems of computing marginal probabilities in local Bayesian nets.
3. In the context of no-forgetting influence diagrams, Shachter (1988, 1990) has pointed out an optimal decision for a leaf decision d can be computed from $E(v|d, \pi(d))$. After an optimal decision function d is computed, he suggests to replace d with a deterministic node characterized by the decision function, and repeat the procedure for the remaining decision nodes. This approach is inefficient, because computations in the tails are repeated, possibly many times.
4. In implementing the procedure EVALUATE(I), one can use a number of approaches for computing marginal probabilities in Bayesian nets. On the other hand, the node reduction algorithms in the literature (Shachter 1986, Smith 1989, Shenoy 1990, and Ndilikilikesha 1991) correspond to only one of those approaches, and not necessarily the most efficient one.

[6]If an influence diagram skeleton is not stepwise-decomposable, then there are portions of it looks like one of the diagrams in Figure 2. Thus, influence diagrams that are not stepwise-decomposable can be constructed from the skeleton.

[7]As stated in footnote 4, the tail I_d^t may not be an influence diagram in the strict sense. So, the "Bayesian nets" here may not be Bayesian nets in the strict sense either. The "prior probabilities" of some of the roots may not sum to 1. Fortunately, this fact has little impact on the computations involved.

7 REMOVABLE DECISION DEPENDENCY ARCS

When disussing the evaluation of no-forgetting influence diagrams, Shachter (1988, 1990) and Tatman and Shachter (1990) have pointed out that an optimal decision function for a decision node d does not necessarily depend on all the parents of d. Shachter (1988, 1990) has developed a linear time algorithm to find, for a given decision node d, parents of d that would not appear in an optimal decision function for d.

In this section, a formal definition of the concept of removable decision dependency arcs are provided. One type of removable decision dependency arcs, namely unaccompanied arcs, will be graphically characterized independent of any evaluation algorithm. This characterization leads to a simple algorithm for finding all the unaccompanied decision dependency arcs. It can be proved that unaccompanied arcs are the only removable decision dependency arcs that can be recognized using only graphical information.

In our terminology, Shachter's algorithm only finds the unaccompanied arcs for a given decision node, while ours finds all the unaccompanied arcs in one sweep.

Let us first introduce the concept of decision equivalence for influence diagrams. Two influence diagrams with the same nodes are *decision-equivalent* if they are value-equivalent and the parents of value nodes and chance nodes are the same in both influence diagrams.

Two influence diagram skeletons are *decision-equivalent* if influence diagrams built from them by specifying the same frames for the same nodes and

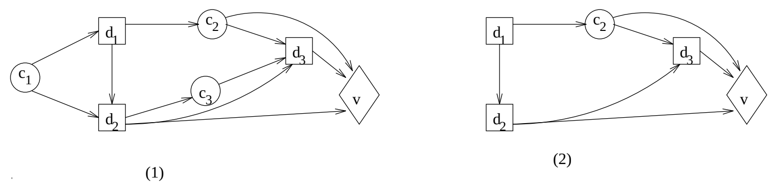

Figure 4: Removable arcs: The arc $c_3 \rightarrow d_3$, $c_1 \rightarrow d_2$ and $c_1 \rightarrow d_1$ are removable.

the some conditional probabilities for the same value nodes and chance nodes are decision-equivalent.

Intuitively, two influence diagrams being decision-equivalent means that the influence diagrams are identical, except that the decisions may be made on the basis of potentially different information. The conditional probabilities are the same, the decisions are to be made are the same, and if optimal decisions are made in both cases, the resulting expected values are the same.

Suppose I is a stepwise-decomposable influence diagram. A decision dependence arc $a \rightarrow d$ from a node a to a decision node d is *removable* if the removal of the arc for the influence diagram results in a stepwise-decomposable influence diagram that is decision-equivalent to I.

Let a be a parent of a decision node d in a stepwise-decomposable influence diagram I. Arcs from a to nodes in Y_d and arcs from nodes in Y_d to a are said to *accompany* the arc $a \rightarrow d$. The arc $a \rightarrow d$ can be accompanied by any number of arcs. When the is accompanied by no arcs, we say that it is *unaccompanied.*

In the influence diagram shown in Figure 4, $Y_{d_3} = \{v\}$. The arc $c_2 \rightarrow d_3$ is accompanied by the arc $c_2 \rightarrow v$. There is no arc either from c_3 to v or from v to c_3. So, $c_3 \rightarrow d_3$ is unaccompanied.

Theorem 2 *Suppose I is a stepwise-decomposable influence diagram. Let I' be the influence diagram obtained from I by removing some unaccompanied decision dependency arcs. Then I' is stepwise-decomposable and it is decision-equivalent to I. In other words, unaccompanied arcs are removable.*

Proof: It suffices to show that I and I' are value-equivalent. We can assume, without the loss of generality, that there has been only one unaccompanied arc removed from I. Because of Theorem 1, we can assume that the decision dependency arc $a \rightarrow d$ removed is such that d is a leaf decision node.

Because $a \rightarrow d$ is unaccompanied, none of the conditional probabilities of the tail I_d^t contains the node a as a parameter. Thus there exists an optimal policy on d that does not depend on a. Therefore there is an optimal policy for I, in which the decision function for d does not depend on a. Let us call such a policy an *$a \rightarrow d$ ignoring policy.*

Any policy of I' can be thought as a policy for I, an $a \rightarrow d$ ignoring policy. So, the optimal expected value of I' is less than or equal to that of I. On the other hand, an $a \rightarrow d$ ignoring policy for I can be viewed as a policy for I' as well, which yields the same expected value for I' as for I. So, the optimal expected value of I can not be strictly greater than that of I'. Therefore I and I' have the same optimal expected value. The theorem is thus proved. □

In the influence diagram shown in Figure 4 (1), $c_3 \rightarrow d_3$ is unaccompanied, and hence is removable. The arc $c_1 \rightarrow d_1$ is not unaccompanied (removable) at the beginning. But it becomes unaccompanied (removable) after the removal of $c_1 \rightarrow d_2$, which is unaccompanied (removable). We call such decision dependency arc potentially *unaccompanied arcs.*

Here is an algorithm which finds all unaccompanied and potentially unaccompanied decision dependency arcs of a stepwise-decomposable influence diagram skeleton I. The second argument $Collector$ is initially empty.

Procedure UNACCOMPANIED-ARCS($I, Collector$)

1. **If** there is no decision node in I, return $Collector$.
2. **Else** find a leaf decision node d, compute Y_d, and
   - Find all the arcs from $\pi(d)$ that are unaccompanied. Let A be the set of all such arcs, and let B be the set of the nodes in $\pi(d)$ that appear in the arcs of A.
   - Set $Collector1 = A \cup Collector$.
   - Remove from I all the nodes in Y_d; introduce an new node v into the resulting diagram; and draw any arc from every node in $\pi(d) - B$ to v. Let I' be the resulting skeleton.
   - UNACCOMPANIED(I', $Collector1$).

Going back to our example, after the removal of $c_3 \rightarrow d_3$, the the node c_3 becomes barren, and hence can be removed (Shachter 1986). After the removal of unaccompanied arcs and the barren node created because of the removals, the influence diagram in Figure 4 (1) becomes the one in Figure 4 (2).

Remark. For any decision dependency arc in a stepwise-decomposable influence diagram skeleton, it is not hard to build an influence diagram from the skeleton such that the arc is not removable. Take the skeleton in Figure 4 (2) for instance, the arc $c_2 \rightarrow d$ is accompanied by $c_2 \rightarrow v$. We can define the conditional probability $P(v|c_2, d_3, d_2)$ in such way that optimal polices for d_3 always depend on c_2. Thus, unaccompanied arcs are the only removable arcs that can be recognized using only graphical information.

8 CONCLUSIONS

Stepwise-decomposable influence diagrams have been introduced. Modeling with stepwise-decomposable influence diagrams enables use to represent knowledge about a piece of information being irrelevant to a decision, and consequently enables us to prevent the decision table from being unnecessarily large at the problem formulation stage. Evaluating stepwise-decomposable influence diagrams amounts to computing conditional probabilities in local Bayesian nets that correspond to disjoint sections of the original influence diagrams. There are a number of ways to compute conditional probabilities in Bayesian net. Shachter's arc reversal and node reduction algorithm corresponds to only one of those ways, and not necessarily the most efficient one. Stepwise-decomposable influence diagrams also prove to be a convenient setting for studying removable decision dependency arcs. Consequently they allow us to prevent, in the evaluation process, the decision tables from being unnecessarily large.

Stepwise-decomposability is an interesting property. It captures the essence of the concept of stepwise-solvability from the viewpoint of graph theory. It leads to a simple and insightful mathematical theory of influence diagrams, particularly efficient and easy to understand algorithms. It also leads to a natural way of exploiting asymmetries in evaluating influence diagrams (Qi and Poole 1992).

Acknowledgements

The first author is grateful to Glenn Shafer and Prakash Shenoy who first brought the influence diagram literature to my attention. Discussions with and comments from Brent Boerlage, Craig Boutilier, Mike Horsch, Keiji Kanazawa, Runping Qi and Gregory Provan have been valuable. Comments from reviewers of KR'92 have also been useful. Research is supported by NSERC Grant OGPOO44121.

Appendix: Proof of Theorem 1

To prove Theorem 1, we need the following lemma of stepwise optimization.

Let X, Y, and Z be three mutually disjoint sets of variables with finite frames. Let $\mathcal{F}$ be a finite collection of real-valued non-negative functions of X and Y, and $\mathcal{G}$ a finite collection of real-valued non-negative functions of Y and Z. Let $h(X, Y)$ be a non-negative function of X and Y.

The following proposition says that to find two functions f and g to maximize the sum $\sum_{X,Y} f(X,Y)(\sum_Z g(Y,Z) + h(X,Y))$, one can proceed by trying first to find a function g_0 that maximizes the sum $\sum_Z g(Y,Z)$ for every value of Y, and then to find a function f that maximizes the sum $\sum_{X,Y} f(X,Y)\{\sum_Z g_0(Y,Z) + h(X,Y)\}$.

Lemma 1 *If $g_0 \in \mathcal{G}$ is such that for any value of Y,*

$$\sum_Z g_0(Y,Z) = max_{g\in\mathcal{G}} \sum_Z g(Y,Z),$$

and if $f_0 \in \mathcal{F}$ is such that

$$\sum_{X,Y} f_0(X,Y)\{\sum_Z g_0(Y,Z) + h(X,Y)\}$$
$$= max_{f\in\mathcal{F}} \sum_{X,Y} f(X,Y)\{\sum_Z g_0(Y,Z) + h(X,Y)\},$$

then

$$\sum_{X,Y,Z} f_0(X,Y)(\sum_Z g_0(Y,Z) + h(X,Y))$$
$$= max_{f\in\mathcal{F},g\in\mathcal{G}} \sum_{X,Y} f(X)(\sum_Z g(Y,Z) + h(X,Y).$$

Proof: It is evident that the left hand side is less than or equal to the right hand side. So, we need only show that the right hand side is no larger than the left hand side. Actually,

$$max_{f\in\mathcal{F},g\in\mathcal{G}} \sum_{X,Y} f(X,Y)(\sum_Z g(Y,Z) + h(X,Y))$$
$$\leq max_{f\in\mathcal{F},g\in\mathcal{G}}\{\sum_{X,Y} f(X,Y)(\sum_Z g_0(Y,Z) + h(X,Y))\}$$
$$= max_{f\in\mathcal{F}} \sum_{X,Y} f(X,Y)(\sum_Z g_0(Y,Z) + h(X,Y)$$
$$= \sum_{X,Y} f_0(X,Y)(\sum_Z g_0(Y,Z) + h(X,Y)).$$

Therefore, the lemma is proved. $\square$

Suppose $I = (V, A, \mathcal{P})$ is a stepwise-decomposable influence diagram. Let d be leaf decision node of I. Let sets X_d, Y_d and $\pi'(d)$ be as defined in section 5. Let $X = X_d$, $Y = \pi'(d)$, and $Z = Y_d \cup \{d\} \cup (\pi(d) - \pi'(d))$. Then X, Y and Z constitute a partition of the set V of all the nodes of I.

Lemma 2 *No node in $\pi'(d)$ is a parent to any node in X_d.*

Proof: Any node $a \in \pi'(d)$ is a parent of d. If a were also a parent of some $x \in X_d$, then x and d would be connected by an edge in the moral graph $m(G)$, where $G = (V, A)$ is the directed graph underlying I. This Contradicts the definition of X_d. □

Let v_1, ..., v_n be all the value nodes of I, among them v_1, ..., v_m $(m \leq n)$ are all the values nodes in Z (i.e in the tail I_d^t), and v_{m+1}, ..., v_n are in $X \cup Y$. Let v be the newly introduced value node of the body I_d^b.

Define functions $h(X, Y) =_{def} v_{m+1} + ... + v_n$, and

$$f_S(X, Y) =_{def} \prod_{x \in ((C \cup U) \cap (X \cup Y))} P(x|\pi(x)) \prod_{x \in (D \cap (X \cup Y))} P_S(x|\pi(x)),$$

$$g_S(Y, Z) =_{def} \prod_{x \in ((C \cup U) \cap Z)} P(x|\pi(x)) P_S(d|\pi(d)) \sum_{i=1}^{m} v_i.$$

Define the *prebody* I_d^p of I w.r.t. d to be the influence diagram obtained from I by restricting it to $X \cup Y$, i.e by removing all the node outside $X \cup Y$ and their conditional probabilities when applicable. Given a policy S, let f_d be the decision function for d, and let $S_d^p = S - \{f_d\}$. Then S_d^p is a policy for the prebody I_d^p.

The joint probability P_S that S determines over all the nodes V of I will be rewritten as $P_{S,I}$ from now on. The meanings of the symbols $E_{S,I}$, $P_{S_d^p, I_d^P}$, and $E_{S_d^p, I_d^P}$ are defined accordingly.

Because of the definition of a prebody, we have that for any value node v_j $(m < j \leq n)$ in the prebody I_d^p,

$$E_{S_d^p, I_d^p}(v_j) = \sum_{X,Y} f_S(X, Y) v_j. \quad (5)$$

Lemma 3 *1. For any node $x \in X$ (i.e in the prebody I_d^p),*

$$P_{S,I}(x) = P_{S_d^p, I_d^p}(x).$$

2. For any value node $v_j \in X$,

$$E_{S,I}(v_j) = E_{S_d^p, I_d^p}(v_j).$$

Proof: Because of Lemma 2, the first assertion can be obtained by a series of barren node removal (see, for example, Shachter 1988). The second assertion follows from the first one immediately. □

This corollary follows from Lemma 3 and equation (5).

Corollary 1 *For any value node $v_j \in X$,*

$$E_{S,I}(v_j) = \sum_{X,Y} f_S(X, Y) v_j.$$

Amongst all the decision functions in S, $g_S(Y, Z)$ only depends on the decision function f_d for d. On the other hand, $f_S(X, Y)$ depends on the decision functions in S_d^p only. In the sequel, we shall rewrite f_S as $f_{S_d^p}$, and g_S as g_{f_d}.

Lemma 4 *Let $V_d^t(Y)$ (remember that $Y = \pi'(d)$) be as defined in section 5. Then*

$$max_{f_d} \sum_Z g_{f_d}(Y, Z) = V_d^t(Y).$$

Proof: The proof readily follows from the definition of the function g_S or g_{f_d}. □

Proof of Theorem 1: Because of the definition of the functions $f_{S_d^p}$ and g_{f_d} and Corollary 1, we have

$$\begin{aligned}
\sum_{i=1}^{n} E_{S,I}(v_i) &= \sum_{i=1}^{m} E_{S,I}(v_i) + \sum_{j=m+1}^{n} E_{S,I}(v_i) \\
&= \sum_{X,Y,Z} f_{S_d^p}(X, Y) g_{f_d}(Y, Z) + \\
&\quad \sum_{X,Y} \sum_{j=m+1}^{n} f_{S_d^p}(X, Y) v_i \\
&= \sum_{X,Y} f_{S_d^p}(X, Y) (\sum_Z g_{f_d}(Y, Z) + h(X, Y)).
\end{aligned}$$

Therefore the theorem follows from Lemma 1 and lemma 4.

References:

R. A. Howard, and J. E. Matheson (1984), Influence Diagrams, in *The principles and Applications of Decision Analysis*, Vol. II, R. A. Howard and J. E. Matheson (eds.). Strategic Decisions Group, Menlo Park, California, USA.

F. V. Jensen, K. G. Olesen, and K. Anderson (1990), An algebra of Bayesian belief universes for knowledge-based systems, *Networks*, 20, pp. 637 - 659.

S. L. Lauritzen and D. J. Speigehalter (1988), Local computations with probabilities on graphical structures and their applications to expert systems, *Journal of Royal Statistical Society B*, 50: 2, pp. 157 - 224.

S. L. Lauritzen, A. P. Dawid, B. N. Larsen, and H. G. Leimer (1990), Independence properties of directed Markov fields, *Networks*, Vol. 20, pp. 491-506.

A. C. Miller, M. W. Merkofer, R. A. Howard, J. E. Matheson, and T. R. Rice (1976), Development of automated aids for decision analysis, Meno Park, CA: Stanford Research Institute.

J. Pearl (1988), *Probabilistic Reasoning in Intelligence Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, Los Altos, CA.

R. Qi and D. Poole (1992), *Exploiting asymmetries in influence diagram evaluation*, in preparation.

P. Ndilikilikesha (1991), Potential Influence Diagrams, Working Paper No. 235, Business School, University of Kansas.

H. Raiffa, (1968), *Decision Analysis*, Addison-Wesley, Reading, Mass.

G. Shafer and P. Shenoy (1988), Local computation in hypertrees, Working Paper No. 201, Business School, University of Kansas.

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

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

R. Shachter (1990), An ordered Examination of Influence Diagrams, *Networks*, 20, pp. 535-563.

P. P. Shenoy, (1990), Valuation-Based Systems for Bayesian Decision Analysis, Working Paper No. 220, Business School, University of Kansas.

J. Q. Smith (1989), Influence Diagrams for Statistical Modeling, *Ann. Stat.*, 17, pp. 654-672.

J. A. Tatman and R. Shachter (1990), Dynamic programming and influence diagrams, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, pp. 265-279.

L. Zhang and D. Poole (1992), Sidestepping the triangulation problem in Bayesian net computations, in *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, July 17-19, San Jose.