Probabilistic Inference in Influence Diagrams

Nevin Lianwen Zhang
Department of Computer Science
Hong Kong University of Science & Technology
Clear Water Bay, Kowloon, Hong Kong
Email: lzhang@cs.ust.hk, Fax: (852)-2358-1477

Abstract

This paper is about reducing influence diagram (ID) evaluation into Bayesian network (BN) inference problems that are as easy to solve as possible. Such reduction is interesting because it enables one to readily use one's favorite BN inference algorithm to efficiently evaluate IDs. Two such reduction methods have been proposed previously (Cooper 1988, Shachter and Peot 1992). This paper proposes a new method. The BN inference problems induced by the mew method are much easier to solve than those induced by the two previous methods.

Keywords: Decision analysis, influence diagrams, Bayesian networks, inference.

1 Introduction

Traditionally decision analysis is carried out by using decision trees (Raiffa 1968). Decision trees grow exponentially in size as the number of relevant variables increases. A more compact framework called influence diagrams (IDs) was introduced by Howard and Matheson in 1984. In addition to being more compact than decision trees, IDs are also more intuitive and reveal more problem structures. They have enabled researchers to solve large decision problems that are beyond the capabilities of decision trees.

An ID is an acyclic graph with three types of nodes: random nodes, decision nodes, and a single value node. Each random node is associated with a conditional probability table (CPT) and the value node is associated with a utility function. Evaluating an ID means finding an optimal decision rule for each of its decision nodes.

IDs without decision and value nodes are called Bayesian networks[1] (BNs) (Pearl 1988). They are widely used by AI researchers as a knowledge representation framework for reasoning under uncertainty. There is a rich collection of exact and approximate algorithms for inference in BNs. This paper is about how to reduce ID evaluation into BN inference problems that are as easy to solve as possible. Such reduction is interesting because it enables one to readily use one's favorite BN inference algorithm to efficiently evaluate IDs.

Cooper (1988) initiated research in this direction. He proposed a transformation of an ID into a BN and showed that optimal decision rules can be found by posing an appropriate sequence of queries to the BN. Several improvements were later introduced by Shachter and Peot (1992). This paper proposes a new method. The BN inference problems induced by the new method are much easier to solve than those induced by the two previous methods.

There are several algorithms that evaluate IDs directly without the reduction into BN inference problems (Shachter 1986, Shenoy 1992, Ndilikilikesha 1994, and Jensen *et al* 1994). We call them *direct evaluation algorithms.* An method that reduces ID evaluation into BN inference problems would be unattractive if, no matter what BN inference algorithms are used, it is less efficient than the best direct evaluation algorithm. We show that the performance of our new method, when coupled with a BN inference algorithm called VE (Zhang and Poole 1994, 1996), is always within a constant factor of the performance of the best direct evaluation method and argue that it is usually more efficient.

The fact that arbitrary BN inference algorithms can used for probabilistic calculations makes our method very attractive as compared to direct evaluation algorithms. From a system development point of view, the method enables one to easily add ID evaluation capabilities to any BN inference packages. From the efficiency point of view, speeding up inference in BNs has been and still is an active research area. There are algorithms that exploit independence of causal influence (e.g. Zhang and Poole 1996) and that exploit special structures in the conditional probability tables (e.g. Boutilier *et al* 1996). The new method facilitates ready incorporation of those algorithms, as well as future advances in BN inference, in ID evaluation. We are not aware of any approximate algorithms for IDs, while there is a rich collection of

[1]Also known as belief networks and probabilistic influence diagrams.

approximate and simulation algorithms for BNs. The new method also opens up the possibility of approximate algorithms for ID, which might be necessary in order to solve large decision problems.

We will begin with definitions related to influence diagrams (Section 2), followed by a brief review of Shachter and Peot's method (Section 3). Foundations for our new method will be laid in Section 4 and details will be worked out in Section 5. The new method will be illustrated through an example in Section 6 and compared with Shachter and Peot's method and direct evaluation algorithms in Section 7. Conclusions will be drawn in Section 8.

2 Influence diagrams

2.1 The concept

Influence diagrams (IDs) (Howard and Matheson 1984) are acyclic graphs with three types of nodes; decision nodes, random nodes, and value nodes[2]. The decision nodes represent decisions to make. The random nodes represent random quantities relevant to the decisions. Arcs into random nodes indicate probabilistic dependence and the dependence of a random node c upon its parents π_c is characterized by a conditional probability $P(c|\pi_c)$. The value nodes represents components of the utility function. Each value node v is associated with a *utility function*, a real valued function $f_v(\pi_v)$ of the parents π_v of v. The total utility function is the summation of all the utility functions. Arcs into decision nodes indicate information availability and are sometimes called *informational arcs*.

IDs are required to satisfy several constraints. First, value nodes cannot have children. Second, IDs must be *regular* in the sense that there must be a directed path that contains all the decision nodes. The last decision node on the path will be referred to as the *tail decision node*. Third, they must be *no-forgetting* in the sense that a decision node and its parents be parents to all subsequent decision nodes. The rationale behind the no-forgetting constraint is that information available now should also be available later if the decision-maker does not forget.

Value networks refer to IDs that do not contain decision nodes. *Bayesian networks* (BNs) (Pearl 1988) are IDs that consists of only random nodes. In the following, the

[2]In the original definition of IDs (Howard and Matheson 1984), there is only one value node. We allow multiple value nodes here so that separability in the utility function can be represented. See Tatman and Shachter (1990) for discussions on separability of utility functions.

terms "nodes" and "variables" will be used interchangeably.

2.2 Decision rules and policies

We shall use Ω_x to denote the *frame* of variable x, i.e. the set of possible values of x. For a set X of variables, Ω_X stands for the Cartesian product $\prod_{x \in X} \Omega_x$.

Let $d_1, \ldots, d_k$ be all the decision nodes in an ID $\mathcal{N}$. A *decision rule* for a decision node d_i is a mapping $\delta_i : \Omega_{\pi_{d_i}} \rightarrow \Omega_{d_i}$. A *policy* is a list of decision rules $\Delta = (\delta_1, \ldots, \delta_k)$ consisting of one rule for each decision node. As a convention, we say that there is only one possible policy, the vacuous policy, for value networks.

Given a decision rule δ_i, the decision node d_i can be viewed as a random node with conditional probability

$$P_{\delta_i}(d_i|\pi_{d_i}) = \begin{cases} 1 & \text{if } \delta_i(\pi_{d_i}) = d_i \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Since $\Delta = (\delta_1, \ldots, \delta_k)$, we shall sometimes write $P_{\delta_i}(d_i|\pi_{d_i})$ as $P_\Delta(d_i|\pi_{d_i})$.

Let $\mathcal{M}$ be obtained from $\mathcal{N}$ by pruning all the value nodes. Then $\mathcal{M}$ is a BN given a policy Δ. Denote this BN by $\mathcal{M}_\Delta$.

Value nodes do not have children. Hence the parents of a value node v must be either random nodes or decision nodes. In other words, all nodes in π_v are in $\mathcal{M}_\Delta$. Hence it makes sense to talk about the marginal probability $P_{\mathcal{M}_\Delta}(\pi_v)$ of π_v in the BN $\mathcal{M}_\Delta$. The expectation $E_\Delta[v]$ of the utility function $f_v(\pi_v)$ under policy Δ is given by

$$E_\Delta[v] = \sum_{\pi_v} P_{\mathcal{M}_\Delta}(\pi_v) f_v(\pi_v), \tag{2}$$

where the summation is taken over all possible values of nodes in π_v. The *expected value* $E_\Delta[\mathcal{N}]$ of $\mathcal{N}$ under policy Δ is defined by

$$E_\Delta[\mathcal{N}] = \sum_{v \in V} E_\Delta[v], \tag{3}$$

where V is the set of all value nodes in $\mathcal{N}$.

Since a value network has only one policy, we shall refer to its expected value given that policy simply as the *expected value of the value network.*

The *optimal expected value* $E[\mathcal{N}]$ of an ID $\mathcal{N}$ is defined by

$$E[\mathcal{N}] = max_\Delta E_\Delta[\mathcal{N}], \tag{4}$$

where maximization is taken over all possible policies. An *optimal policy* Δ^* is one that satisfies

$$E_{\Delta^*}[\mathcal{N}] = E[\mathcal{N}]. \tag{5}$$

In general, there could be more than one optimal decision rule. Decision rules in an optimal policy are said to be *optimal decision rules.* In other words, a decision rule δ_i^* for the decision node d_i is optimal if there exist decision rules δ_j^* for all other decision nodes d_j such that the policy $(\delta_1^*, \ldots, \delta_i^*, \ldots, \delta_k^*)$ is optimal.

In this paper, we shall only consider variables with finite frames. Hence there is only a finite number of possible policies and there always exists at least one optimal policy. *To evaluate an ID* is to (1) find an optimal policy, and (2) compute the optimal expected value.

3 Probabilistic inference in IDs: Previous work

ID evaluation requires a lot of probabilistic calculations. This paper is concerned with identifying a set of probabilistic inference problems such that optimal decision rules can be readily obtained from their solutions. The problems should be as easy to solve as possible.

Cooper (1988) initiated research in this direction. Several improvements to Cooper's method were proposed by Shachter and Peot (1992). This section briefly reviews the method by Shachter and Peot.

The method applies only in the case when there is one value node. Let $\mathcal{N}$ be an ID with one value node. Denote the value node by v. For simplicity, assume that the utility function $f_v(\pi_v)$ of v is non-negative[3]. The node is converted into a binary random node with the following conditional probability:

$$\begin{aligned} P(v{=}1|\pi_v) &= \frac{f_v(\pi_v)}{M_v}, \\ P(v{=}0|\pi_v) &= 1{-}P(v{=}1|\pi_v), \end{aligned} \tag{6}$$

where $M_v = max_{\pi_v} f_v(\pi_v)$. This transformation will be referred to as *Cooper's transformation.*

[3] If the utility function takes negative values, a constant can be added to it so that it takes only non-negative values. Addition of a constant to the utility function does not change the optimal policies. Moreover, the optimal expected value of an ID equals to its optimal expected value after the addition of the constant minus the constant.

Each decision node d is also converted into a random node with the following conditional probability:

$$P(d{=}\alpha|\pi_d){=}1/|\Omega_d|,$$

for each possible value α of d, where $|\Omega_d|$ is the number of possible values of d. After the transformations, $\mathcal{N}$ becomes a BN. Denote the BN by $\mathcal{N}'$.

According to the regularity constraint, there exists a directed path that contains all decision nodes. Let $d_1, \ldots, d_k$ an enumeration of the decision nodes in the order they appear in the path. It is shown that an optimal decision rule δ_k^* for d_k can be obtained by

$$\delta_k^*(\pi_{d_k}) = arg\ \ max_{d_k} P_{\mathcal{N}'}(d_k, \pi_{d_k}|v{=}1).$$

After the rule has been computed, the conditional probability of d_k is changed to $P_{\delta_k^*}(d_k|\pi_{d_k})$ (see equation 1). An optimal decision rule for d_{k-1} is then computed using the same formula except with k replaced by $k{-}1$. Optimal decision rules for $d_{k-2}, \ldots, d_1$ are computed recursively in the same fashion.

Shachter and Peot's method reduces the evaluation of $\mathcal{N}$ into the following BN inference problems:

BN inference problems: $P_{\mathcal{N}'}(d_k, \pi_{d_k}|v{=}1), P_{\mathcal{N}'}(d_{k-1}, \pi_{d_{k-1}}|v{=}1), \ldots, P_{\mathcal{N}'}(d_1, \pi_{d_1}|v{=}1).$

We will show that an ID can be evaluated by solving BN inference problems that are much easier than those listed above.

4 Decomposition theorem

Suppose $\mathcal{N}$ is an ID and d is the tail decision node. This section shows that $\mathcal{N}$ can be decomposed into two components, called tail and body respectively, such that an optimal decision rule for d can be found in the tail and optimal decision rules for all other decision nodes can be found in the body. The body is again an ID and hence the decomposition can be repeated in the body.

4.1 Downstream and upstream sets

We begin by partitioning the set of nodes in $\mathcal{N}$ into several subsets w.r.t to the tail decision node d.

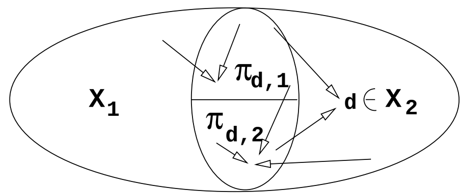

Figure 1: Partition of the set of nodes in an ID w.r.t a decision node.

The *moral graph* (Lauritzen and Spiegelhalter 1988) of an ID is obtained by first adding undirected edges between pairs of parents of each node so that they are pairwise connected and then dropping directions of all the directed edges. Let x and y be two nodes and S be a set of nodes that does not contain x or y. We say that S *m-separates* x and y if, in the moral graph, every path connecting them contains at least one node in S.

Let X be the set of all nodes in an ID $\mathcal{N}$. The *upstream set of* $\mathcal{N}$ *w.r.t to* d, denoted by X_1, is the set of nodes in $X \backslash \pi_d$ that are m-separated from d by π_d. The *downstream set of* $\mathcal{N}$ *w.r.t* d, denoted by X_2, is the set of nodes in $X \backslash \pi_d$ that are not m-separated from d by π_d. Note that $d \in X_2$ and that the three sets X_1, π_d and X_2 constitute a partition of the set X.

Define $\pi_{d,2}$ be the set of nodes in π_d that have at least one parent in X_2 and set $\pi_{d,1} = \pi_d \backslash \pi_{d,2}$. As shown in Figure 1, the four sets X_1, X_2, $\pi_{d,1}$, and $\pi_{d,2}$ constitute another partition of the set of all nodes in $\mathcal{N}$.

Consider the ID in Figure 2. The set of parents of d_2 is $\pi_{d_2} = \{d_1, c_3, c_4\}$ and the downstream set X_2 w.r.t d_2 is $X_2 = \{d_2, c_6, v_2\}$. Since c_4 the only parent of d_2 that has a parent in the downstream set, $\pi_{d_2,2} = \{c_4\}$. Hence $\pi_{d_2,1} = \{d_1, c_3\}$.

A node x is an *ancestor* to another node y if there is a directed path from x to y. A *ancestral set* $an(A)$ of a set of nodes A consists of nodes in A and their ancestors. The following proposition summarizes properties of the aforementioned sets.

Proposition 1 *Suppose d is the tail decision node in an ID. Then*

1. *The node d is the only decision node in the downstream set X_2.*
2. *All nodes in $\pi_{d,2}$ are random nodes.*
3. *All nodes in $an(\pi_{d,2}) \cap X_2$ are random nodes.*

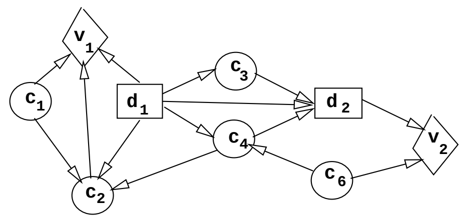

Figure 2: An ID. Random nodes are drawn as ellipses, decision nodes as rectangles, and value nodes as diamonds.

4. All other decision nodes are in $\pi_{d,1}$.

Proof: Due to the no-forgetting constraint, all decision nodes other than d and their parents are in π_d. The first item follows. Since nodes in $\pi_{d,2}$ have parents in X_2, they cannot be decision nodes. They cannot be value nodes either because value nodes have no children while they have child d. Therefore, they must be random nodes and item 2 is proved. The third item follows from item 1, the fact that d cannot be ancestors to nodes in $\pi_{d,2}$ (otherwise there would be directed loops), and the fact that value nodes have no children. Finally, the fourth item follows from item 2 and the fact that decision nodes other than d and their parents are in π_d. $\Box$

4.2 First step of derivation

Let V_1 be the set of value nodes in the upstream set X_1 and V_2 be the set of value nodes in the downstream set X_2. We have

$$E_\Delta[\mathcal{N}] = \sum_{v \in V_1} E_\Delta[v] + \sum_{v \in V_2} E_\Delta[v]. \tag{7}$$

A decomposition of $\mathcal{N}$ will be derived in the next three subsections by starting from this equation. As the first step, this subsection studies the two terms on the right hand side and expresses them in a form that is convenient for later derivations.

Given a policy Δ, all the decision nodes can be viewed as random nodes. If we convert all value nodes into random node by Cooper's transformation, then $\mathcal{N}$ becomes a BN. Denote this BN by $\mathcal{N}_\Delta$. The following proposition allows us to express each $E_\Delta[v]$ in terms of $\mathcal{N}_\Delta$.

Proposition 2 *For any value node v,*

$$E_\Delta[v] = P_{\mathcal{N}_\Delta}(v{=}1)M_v.$$

In order to prove the proposition, we need the concept of barren nodes (Shachter 1986). A node in a BN is *barren* if it has no children. The following lemma is well known (Shachter 1988).

Lemma 1 *Let $\mathcal{M}_1$ be a BN and let $\mathcal{M}_2$ be the resulting BN after pruning a barren node from $\mathcal{M}_1$. Then*

$$P_{\mathcal{M}_1}(A) = P_{\mathcal{M}_2}(A),$$

for any subset A of nodes in $\mathcal{M}_2$. $\Box$

Proof of Proposition 2: Consider the BN $\mathcal{M}_\Delta$ defined in Section 2. It is the same as $\mathcal{N}_\Delta$ except that it does not contain the value nodes. The value nodes are barren random nodes in $\mathcal{N}_\Delta$. By Lemma 1,

$$P_{\mathcal{M}_\Delta}(\pi_v){=}P_{\mathcal{N}_\Delta}(\pi_v).$$

Consequently,

$$\begin{aligned} E_\Delta[v] &= \sum_{\pi_v} P_{\mathcal{M}_\Delta}(\pi_v) f_v(\pi_v) \\ &= \sum_{\pi_v} P_{\mathcal{N}_\Delta}(\pi_v) P(v{=}1|\pi_v) M_v \\ &= P_{\mathcal{N}_\Delta}(v{=}1)M_v. \Box \end{aligned}$$

According to equation (7) and Proposition 2, we have

$$\begin{aligned} E_\Delta[\mathcal{N}] &= \sum_{v\in V_1} P_{\mathcal{N}_\Delta}(v{=}1)M_v + \sum_{v\in V_2} P_{\mathcal{N}_\Delta}(v{=}1)M_v \\ &= \sum_{v\in V_1} P_{\mathcal{N}_\Delta}(v{=}1)M_v + \sum_{v\in V_2}\sum_{\pi_d} P_{\mathcal{N}_\Delta}(\pi_d) P_{\mathcal{N}_\Delta}(v{=}1|\pi_d)M_v, \\ &= \sum_{v\in V_1} P_{\mathcal{N}_\Delta}(v{=}1)M_v + \sum_{\pi_d} P_{\mathcal{N}_\Delta}(\pi_d) \sum_{v\in V_2} P_{\mathcal{N}_\Delta}(v{=}1|\pi_d)M_v. \end{aligned} \tag{8}$$

Equation (8) serves as the basis for further derivations.

4.3 Bodies

This subsection is concerned with the probability $P_{\mathcal{N}_\Delta}(\pi_d)$ and the probability $P_{\mathcal{N}_\Delta}(v{=}1)$ for each $v{\in}V_1$. Write Δ as $\Delta{=}(\Delta_1, \delta)$, where δ is the decision rule for d and Δ_1 consists of decision rules for all other decision nodes. We show that the aforementioned probabilities depends only on Δ_1 and on numerical information for nodes in a portion of $\mathcal{N}$ known as the body of $\mathcal{N}$ w.r.t d.

Define the *body of* $\mathcal{N}$ *w.r.t* d by the following procedure:

Procedure `body`$(\mathcal{N}, d)$:

1. Prune from $\mathcal{N}$ all the nodes in $X_2 \backslash an(\pi_{d,2})$.
2. Return the resulting ID.

We use $\mathcal{B}$ to denote the body. Nodes in $X_2 \backslash an(\pi_{d,2})$ because, as will be seen soon, they are irrelevant to $P_{\mathcal{N}_\Delta}(\pi_d)$ and $P_{\mathcal{N}_\Delta}(v{=}1)$ for each $v{\in}V_1$.

According to Proposition 1 (4), for any decision node $d'{\neq}d$, d' is in $\mathcal{B}$ and it has the same parents in $\mathcal{B}$ as in $\mathcal{N}$. Hence Δ_1 is a policy for $\mathcal{B}$.

Given Δ_1, all decision nodes in $\mathcal{B}$ can be viewed as random nodes. If we convert all the value nodes into random nodes by Cooper's transformation, the $\mathcal{B}$ becomes a BN. Denote the BN by $\mathcal{B}_{\Delta_1}$. It differs from $\mathcal{N}_\Delta$ only in that it does not contain the nodes in $X_2 \backslash an(\pi_{d,2})$. If $X_2 \backslash an(\pi_{d,2})$ is not empty, at least one of its members must be a barren node in $\mathcal{N}_\Delta$[4]. Let $\mathcal{N}_1$ be the resulting BN after removing x from $\mathcal{N}_\Delta$. If $X_2 \backslash (an(\pi_{d,2}) {\cup} \{x\})$ is not empty, at least one of its members must be a barren node in $\mathcal{N}_1$[5], and so on. Therefore, by repeatedly applying Lemma 1, we can show that $P_{\mathcal{N}_\Delta}(\pi_d){=}P_{\mathcal{B}_{\Delta_1}}(\pi_d)$ and $P_{\mathcal{N}_\Delta}(\pi_v){=}P_{\mathcal{B}_{\Delta_1}}(\pi_v)$ for any $v{\in}V_1$. Consequently, equation (8) can be rewritten as follows:

$$E_\Delta[\mathcal{N}] = \sum_{v \in V_1} P_{\mathcal{B}_{\Delta_1}}(v{=}1) M_v + \sum_{\pi_d} P_{\mathcal{B}_{\Delta_1}}(\pi_d) \sum_{v \in V_2} P_{\mathcal{N}_\Delta}(v{=}1|\pi_d) M_v. \tag{9}$$

[4]To prove the claim, let $x_1{\in}X_2 \backslash an(\pi_{d,2})$. If it is barren in $\mathcal{N}_\Delta$, then the claim is proved. If not, let x_2 be a child of x_1. It is evident that x_2 is also in $X_2 \backslash an(\pi_{d,2})$. If x_2 is barren in $\mathcal{N}_\Delta$, the claim is also proved. If not, we can repeat with a child of x_2 and so on. Eventually, we will be able to find a node in $X_2 \backslash an(\pi_{d,2})$ that is barren in $\mathcal{N}_\Delta$.

[5]This claim can be proved in the same way as the previous one.

4.4 Tails

This subsection is concerned with the probability $P_{\mathcal{N}_\Delta}(v{=}1|\pi_d)$ for each $v{\in}V_2$. We show that it depends only on the decision rule δ for d and numerical information for nodes in a portion of $\mathcal{N}$ known as the tail of $\mathcal{N}$ w.r.t d.

Given decision rule δ, d depends deterministically on π_d; it must be $\delta(\pi_d)$. Hence $P_{\mathcal{N}_\Delta}(v{=}1|\pi_d)$ is the same as $P_{\mathcal{N}_\Delta}(v{=}1|\pi_d, d{=}\delta(\pi_d))$.

Since $P_{\mathcal{N}_\Delta}(v|\pi_d, d){=}\sum_{X_2\backslash\{d,v\}} P_{\mathcal{N}_\Delta}(X_2\backslash\{d\}|\pi_d, d)$, we consider $P_{\mathcal{N}_\Delta}(X_2\backslash\{d\}|\pi_d, d)$. Enumerate nodes in $\pi_{d,2}{\cup}X_2\backslash\{d\}$ as x_1, x_2, $\ldots$, x_m such that x_i is not an ancestor of x_j if $i{>}j$. By the definition of X_2 and $\pi_{d,2}$, none of the x_i's have parents in the upstream set X_1. Hence parents of each x_i must be in the set $\{x_1, \ldots, x_{i-1}, d\}{\cup}\pi_{d,1}$. Therefore

$$
\begin{aligned}
P_{\mathcal{N}_\Delta}(\pi_{d,2}, X_2\backslash\{d\}|\pi_{d,1}, d) &= P_{\mathcal{N}_\Delta}(x_1, \ldots, x_m|\pi_{d,1}, d) \\
&= P_{\mathcal{N}_\Delta}(x_1, \ldots, x_{m-1}|\pi_{d,1}, d) P_{\mathcal{N}_\Delta}(x_m|x_1, \ldots, x_{m-1}, \pi_{d,1}, d) \\
&= P_{\mathcal{N}_\Delta}(x_1, \ldots, x_{m-1}|\pi_{d,1}, d) P(x_m|\pi_{x_m}) \qquad (10) \\
&= P_{\mathcal{N}_\Delta}(x_1, \ldots, x_{m-2}|\pi_{d,1}, d) P(x_{m-1}|\pi_{m-1}) P(x_m|\pi_{x_m}) \\
&= \ldots \\
&= \prod_{i=1}^{m} P(x_i|\pi_i),
\end{aligned}
$$

where equation (10) is true because that x_m is independent of all its non-descendents[6] given its parents, an independence assumption of BNs. The conditional probabilities for the x_i's are not subscripted by Δ since they are either random or value nodes (Proposition 1). Their conditional probabilities does not depend on Δ Therefore,

$$
P_{\mathcal{N}_\Delta}(X_2\backslash\{d\}|\pi_d, d) = \frac{\prod_{i=1}^{m} P(x_i|\pi_i)}{\sum_{X_2\backslash\{d\}} \prod_{i=1}^{m} P(x_i|\pi_i)}. \qquad (11)
$$

Define the *tail of* $\mathcal{N}$ *w.r.t* d by the following procedure:

Procedure $\texttt{tail}(\mathcal{N}, d)$:

1. Prune from $\mathcal{N}$ all nodes in X_1.
2. Prune arcs into the node d and nodes in $\pi_{d,1}$.
3. Prune conditional probabilities of random nodes in $\pi_{d,1}$.

[6] A node y is a *descendent* of another node x if x is an ancestor of y.

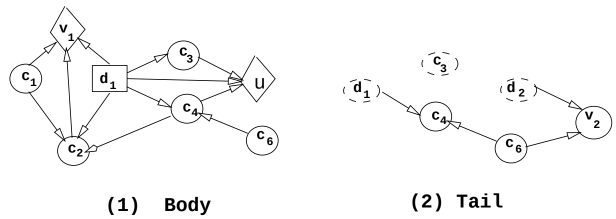

Figure 3: Tail and body: The BN in (2) is the tail of the influence diagram in Figure 2 w.r.t d_2; probabilities of the dashed nodes are uniform distributions. The ID in (1), with the value node u ignored, is the body of the ID in Figure 2 with w.r.t d_2. With u, it is the augmented body (Subsection 5.2).

4. For each node $x \in \{d\} \cup \pi_{d,1}$, set $P_{\mathcal{T}}(x) = 1/|\Omega_x|$, where $|\Omega_x|$ is the number of possible values of x.
5. Convert all the value nodes in X_2 into random nodes by Cooper's transformation.
6. Return the resulting BN.

We use $\mathcal{T}$ to denote the tail. It is a BN for the following reasons. It consists the decision node d, nodes in $\pi_{d,1}$, nodes in $\pi_{d,2} \cup X_2 \setminus \{d\}$. The node d and nodes in $\pi_{d,1}$ are associated with uniform distributions. According to Proposition 1, nodes in $\pi_{d,2} \cup X_2 \setminus \{d\}$ are either random nodes or value nodes. Random nodes in the set inherit their conditional probabilities from $\mathcal{N}$, while conditional probabilities for value nodes in the set are obtained from their utility functions via Cooper's transformation. So all nodes in $\mathcal{T}$ are associated with probabilities and hence $\mathcal{T}$ is a BN.

By following the same line of reasoning as in the derivation of equation (11), one can show that

$$P_{\mathcal{T}}(X_2 \setminus \{d\} | \pi_d, d) = \frac{\prod_{i=1}^{m} P(x_i|\pi_i)}{\sum_{X_2 \setminus \{d\}} \prod_{i=1}^{m} P(x_i|\pi_i)}.$$

Therefore, $P_{\mathcal{N}_\Delta}(X_2 \setminus \{d\} | \pi_d, d) = P_{\mathcal{T}}(X_2 \setminus \{d\} | \pi_d, d)$ and hence $P_{\mathcal{N}_\Delta}(v{=}1|\pi_d, d) = P_{\mathcal{T}}(v{=}1|\pi_d, d)$. Consequently,

$$P_{\mathcal{N}_\Delta}(v{=}1|\pi_d) = P_{\mathcal{N}_\Delta}(v{=}1|\pi_d, d{=}\delta(\pi_d)) = P_{\mathcal{T}}(v{=}1|\pi_d, d{=}\delta(\pi_d)). \tag{12}$$

This proves the claim made at the beginning of this subsection.

Define the *evaluation functional* $e_{\mathcal{T}}(\pi_d, d)$ of the tail $\mathcal{T}$ by

$$e_{\mathcal{T}}(\pi_d, d) = \sum_{v \in V_2} P_{\mathcal{T}}(v{=}1|\pi_d, d) M_v. \tag{13}$$

Equation (12) allows us to further rewrite Equation (9) as follows:

$$E_{\Delta}[\mathcal{N}] = \sum_{v \in V_1} P_{\mathcal{B}_{\Delta_1}}(v{=}1) M_v + \sum_{\pi_d} P_{\mathcal{B}_{\Delta_1}}(\pi_d) e_{\mathcal{T}}(\pi_d, d{=}\delta(\pi_d)). \tag{14}$$

4.5 Decomposition theorem

Define the *augmented body of* $\mathcal{N}$ *w.r.t* d by the following procedure:

Procedure `augBody`$(\mathcal{N}, d, e_{\mathcal{T}}(\pi_d, d))$:

1. $\mathcal{B}$=`body`$(\mathcal{N}, d)$.
2. Introduce a new value node u to $\mathcal{B}$. Make it a child of each node in π_d and set its utility function as follows:

$$f_u(\pi_u) = max_d e_{\mathcal{T}}(\pi_d, d).$$

3. Return the resulting ID.

Denote the augmented body by $\mathcal{B}'$. Given policy Δ_1, all its decision nodes can be regarded as random nodes. If all its value nodes, including u, are converted into random nodes via Cooper's transformation, $\mathcal{B}'$ becomes a Bayesian network. Denote the Bayesian network by $\mathcal{B}'_{\Delta_1}$. It differs from $\mathcal{B}_{\Delta_1}$ only in one node, namely u, which is barren. According to Lemma 1, $P_{\mathcal{B}'_{\Delta_1}}(\pi_d){=}P_{\mathcal{B}_{\Delta_1}}(\pi_d)$ and $P_{\mathcal{B}'_{\Delta_1}}(v{=}1){=}P_{\mathcal{B}_{\Delta_1}}(v{=}1)$ for any $v{\in}V_1$. To avoid excessive notation, we will use $\mathcal{B}$ to denote the augmented body from now on.

Theorem 1 *(*Decomposition Theorem*) Let d be the tail decision node in an ID $\mathcal{N}$.*

1. An optimal decision rule δ^* for d is given by

$$\delta^*(\pi_d) = arg\ \ max_d e_{\mathcal{T}}(\pi_d, d), \tag{15}$$

where $e_{\mathcal{T}}(\pi_d, d)$ is the evaluation functional of the tail $\mathcal{T}$ of $\mathcal{N}$ w.r.t d.

2. *A policy Δ_1 for the augmented body $\mathcal{B}$ of $\mathcal{N}$ w.r.t d is optimal if and only if the policy (Δ_1, δ^*) is optimal for $\mathcal{N}$.*

3. *The optimal expected value of $\mathcal{B}$ is the same as that of $\mathcal{N}$.*

Proof: Suppose Δ=(Δ_1, δ) is an optimal policy for $\mathcal{N}$, i.e. $E_\Delta[\mathcal{N}] = E[\mathcal{N}]$. Consider another policy Δ'=(Δ_1, δ^*). By equation (14) and the definition of δ^*, we have

$$\begin{aligned} E_{\Delta'}[\mathcal{N}] &= \sum_{v \in V_1} P_{\mathcal{B}_{\Delta_1}}(v{=}1)M_v + \sum_{\pi_d} P_{\mathcal{B}_{\Delta_1}}(\pi_d) e_{\mathcal{T}}(\pi_d, d{=}\delta^*(\pi_d)) \\ &\geq \sum_{v \in V_1} P_{\mathcal{B}_{\Delta_1}}(v{=}1)M_v + \sum_{\pi_d} P_{\mathcal{B}_{\Delta_1}}(\pi_d) e_{\mathcal{T}}(\pi_d, d{=}\delta(\pi_d)) \\ &= E_\Delta[\mathcal{N}] = E[\mathcal{N}]. \end{aligned}$$

Hence it must be the case that $E_{\Delta'}[\mathcal{N}] = E[\mathcal{N}]$. In other words, Δ' is also an optimal policy. Consequently, δ^* is an optimal decision rule for d.

To show the second item, notice that if δ=δ^*, then $e_{\mathcal{T}}(\pi_d, d{=}\delta(\pi_d)){=}f_u(\pi_d)$. Hence for any policy Δ_1 for $\mathcal{B}$, the second term of (14) is $E_{\Delta_1}[u]$ — the expectation of the utility function f_u under Δ_1. Because the first term is the sum of the expectations of other utility functions in $\mathcal{B}$, we have that

$$E_\Delta[\mathcal{N}] = E_{\Delta_1}[\mathcal{B}],$$

where Δ=(Δ_1, δ^*). Hence Δ_1 maximizes $E_{\Delta_1}[\mathcal{B}]$ if and only if Δ=(Δ_1, δ^*) maximizes $E_\Delta[\mathcal{N}]$. The second item follows. The third item is a direct consequence of the above equation. □

5 Evaluating IDs

The decomposition theorem gives us the following procedure for evaluating an ID: (1) decompose it to two components — tail and body — w.r.t the tail decision node, (2) find an optimal decision rule for the tail decision node in the tail, and (3) repeat the process in the body. This section looks at the necessary computations in more detail and identifies the BN inference problems that one needs to solve. We also introduce several optimizations that make the BN inference problems easier to solve.

5.1 Simplifying computations in the tail

To obtain the evaluation functional $e_{\mathcal{T}}(\pi_d, d)$ of the tail $\mathcal{T}$, we need to compute the marginal probability $P_{\mathcal{T}}(\pi_d, d)$ and the marginal probability $P_{\mathcal{T}}(v{=}1, \pi_d, d)$ for each value node $v{\in}V_2$. This subsection shows that some of the nodes that appear in the marginal probabilities can be deleted and that some of the nodes in $\mathcal{T}$ can be pruned when computing each of the marginal probabilities.

5.1.1 Irrelevant parents of decision nodes

Let $\pi_{d,i}$ be the set of nodes in $\pi_{d,1}$ that, in $\mathcal{N}$, are not parents to nodes in $\pi_{d,2}{\cup}X_2\backslash\{d\}$. Each $x{\in}\pi_{d,i}$ is an isolated node in $\mathcal{T}$ for the following reasons. First, x has no parents in $\mathcal{T}$ since arcs into nodes in $\pi_{d,1}$ have been removed by `tail`. Second, x has no children in $\mathcal{T}$. This is because $\mathcal{T}$ consists of the node d, nodes in $\pi_{d,1}$, and nodes in $\pi_{d,2}{\cup}X_1\backslash\{d\}$. The node d and nodes in $\pi_{d,1}$ cannot be children of x since all arcs into them have been removed by `tail` and nodes $\pi_{d,2}{\cup}X_1\backslash\{d\}$ are not children of x by the definition of $\pi_{d,i}$.

Define the *reduced tail* of $\mathcal{N}$ w.r.t d by the following procedure:

Procedure `redTail`$(\mathcal{N}, d)$:

1. $\mathcal{T}$ =`tail`$(\mathcal{N}, d)$.
2. Prune from $\mathcal{T}$ nodes in $\pi_{d,i}$.
3. Return the resulting BN.

We use $\mathcal{T}_r$ to denote the reduced tail. It consists of nodes in $\pi_{d,r}{\cup}X_2$, where $\pi_{d,r} = \pi_d\backslash\pi_{d,i}$. Because each member of $\pi_{d,i}$ is an isolated node in $\mathcal{T}$ and its probability is the uniform distribution, the joint probabilities of $\mathcal{T}$ and $\mathcal{T}_r$ are related by

$$P_{\mathcal{T}}(\pi_d, X_2) = P_{\mathcal{T}_r}(\pi_{d,r}, X_2) \prod_{x\in\pi_{d,i}} 1/|\Omega_x|,$$

where $|\Omega_x|$ is the number of possible values of x. Therefore

$$\begin{aligned} e_{\mathcal{T}}(\pi_d, d) &= \sum_{v\in V_2} \frac{P_{\mathcal{T}}(v{=}1, \pi_d, d)}{P_{\mathcal{T}}(\pi_d, d)} M_v \\ &= \sum_{v\in V_2} \frac{P_{\mathcal{T}_r}(v{=}1, \pi_{d,r}, d) \prod_{x\in\pi_{d,i}} 1/|\Omega_x|}{P_{\mathcal{T}_r}(\pi_{d,r}, d) \prod_{x\in\pi_{d,i}} 1/|\Omega_x|} M_v \\ &= \sum_{v\in V_2} \frac{P_{\mathcal{T}_r}(v{=}1, \pi_{d,r}, d)}{P_{\mathcal{T}_r}(\pi_{d,r}, d)} M_v. \qquad (16) \end{aligned}$$

Hence the evaluation functional can be computed from the marginal probability $P_{\mathcal{T}_r}(\pi_{d,r}, d)$ and the marginal probability $P_{\mathcal{T}_r}(v{=}1, \pi_{d,r}, d)$ for each value node $v \in V_2$. Those marginal probabilities involve less nodes than $P_{\mathcal{T}}(\pi_d, d)$ and $P_{\mathcal{T}}(v{=}1, \pi_d, d)$.

Equation (16) also implies that the evaluation functional does not depend on nodes in π_i and can be rewritten as $e_{\mathcal{T}}(\pi_{d,r}, d)$. This fact in turn has two implications. First, the optimal decision rule for d given by equation (15) does not depend on nodes in $\pi_{d,i}$. For this reason, nodes in $\pi_{d,i}$ will be called *irrelevant parents* of d (Tatman and Shachter 1990) and the nodes in $\pi_{d,r}$ will be called *relevant parents* of d.

Second, the utility function of the new value node u in the augmented body $\mathcal{B}$ does not depend on the irrelevant parents of d. Consequently there is no need to make u a child to those nodes. From now on we assume that, in $\mathcal{B}$, u is a child of only relevant parents of d and its utility function of u is given by

$$f_u(\pi_{d,r}) = max_d e_{\mathcal{T}}(\pi_{d,r}, d). \tag{17}$$

To avoid excessive notations, we will use $\mathcal{T}$ to denote the reduced body from now on.

5.1.2 Pruning of irrelevant nodes

The following corollary (Shachter 1988) of Lemma 1 allows us to identify and prune nodes in the reduced tail $\mathcal{T}$ that are irrelevant to the marginal probabilities $P_{\mathcal{T}}(v{=}1, \pi_{d,r}, d)$ and $P_{\mathcal{T}}(\pi_{d,r}, d)$.

Corollary 1 *Let A be a subset of nodes in a BN $\mathcal{M}$. Let $\mathcal{M}'$ be the BN obtained from $\mathcal{M}$ by pruning all nodes outside $an(A)$. Then*

$$P_{\mathcal{M}}(A) = P_{\mathcal{M}'}(A). \square$$

Let $\mathcal{T}'$ be the BN obtained from the reduced tail $\mathcal{T}$ by pruning nodes outside $an(\pi_{d,r} \cup \{d\})$. By Corollary 1, $P_{\mathcal{T}}(\pi_{d,r}, d) = P_{\mathcal{T}'}(\pi_{d,r}, d)$. Consider the node d. It has no parents in $\mathcal{T}$ and hence has no parents in $\mathcal{T}'$. Children of d in $\mathcal{T}$ cannot be in the ancestral set $an(\pi_{d,r} \cup \{d\}$, for otherwise there would directed loops in the original ID. Hence children of d are not in $\mathcal{T}'$. Therefore d is an isolated node in $\mathcal{T}'$. Let $\mathcal{T}_c$ be obtained from $\mathcal{T}'$ by pruning the isolated node d. Since the probability of d is the uniform distribution, $P_{\mathcal{T}'}(\pi_{d,r}, d) = P_{\mathcal{T}_c}(\pi_{d,r})/|\Omega_d|$. Hence

$$P_{\mathcal{T}}(\pi_{d,r}, d) = P_{\mathcal{T}_c}(\pi_{d,r})/|\Omega_d|.$$

Now consider the marginal probability $P_{\mathcal{T}}(v{=}1, \pi_{d,r}, d)$ for each value node in $v{\in}V_2$. Let $\mathcal{T}_v$ be the BN obtained from $\mathcal{T}$ by pruning nodes outside $an(\pi_{d,r}{\cup}\{d, v\})$. By Corollary 1,

$$P_{\mathcal{T}}(v{=}1, \pi_{d,r}, d) = P_{\mathcal{T}_v}(v{=}1, \pi_{d,r}, d).$$

To summarize, we need to solve the following BN inference problems in order to obtain the evaluation functional $e_{\mathcal{T}}(\pi_{d,r}, d)$:

BN inference problems: $P_{\mathcal{T}_c}(\pi_{d,r}), P_{\mathcal{T}_v}(v{=}1, \pi_{d,r}, d)$ for each $v{\in}V_2$.

For any BN $\mathcal{M}$ and any subset A of nodes in $\mathcal{M}$, let $\texttt{BNinf}(A, \mathcal{M})$ be a procedure that computes the marginal probability $P_{\mathcal{M}}(A)$. Arbitrary BN inference algorithms can be used in the procedure. The evaluation functional $e_{\mathcal{T}}(\pi_{d,r}, d)$ can be obtained by using the following procedure:

Procedure $\texttt{evalFun}(\mathcal{T}, d)$

1. Obtain $\mathcal{T}_c$ from $\mathcal{T}$ by pruning nodes outside $an(\pi_{d,r})$. Compute $P_{\mathcal{T}_c}(\pi_{d,r})$ by calling $\texttt{BNinf}(\pi_{d,r}, \mathcal{T}_c)$.
2. For each $v{\in}V_2$, obtain $\mathcal{T}_v$ by pruning nodes outside $an(\pi_{d,r}{\cup}\{d, v\})$. Compute $P_{\mathcal{T}_v}(v{=}1, \pi_{d,r}, d)$ by calling $\texttt{BNinf}((v{=}1, \pi_{d,r}), \mathcal{T}_v)$
3. Set
$$e_{\mathcal{T}}(\pi_{d,r}, d) = \frac{\sum_{v\in V_2} P_{\mathcal{T}_v}(v{=}1, \pi_{d,r}, d)}{P_{\mathcal{T}_c}(\pi_{d,r})/|\Omega_d|}.$$
4. Return $e_{\mathcal{T}}(\pi_{d,r}, d)$ and $P_{\mathcal{T}_c}(\pi_{d,r})$.

Note that in addition to the evaluation functional, $\texttt{evalFun}$ also returns the marginal potential $P_{\mathcal{T}_c}(\pi_{d,r})$. It will be used in the next subsection.

5.2 Simplifying computations in the body

The augmented body $\mathcal{B}$ contains nodes in the set $an(\pi_{d,2}){\cap}X_2$. The set is empty when no nodes in π_d have parents in X_2, i.e. when $\pi_{d,2}{=}\emptyset$. This subsection is concerned with the case when the set is not empty and shows that nodes in the set can be pruned from $\mathcal{B}$. Pruning nodes from $\mathcal{B}$ simplifies computations in the body.

Pruning nodes in $an(\pi_{d,2}){\cap}X_2$ from $\mathcal{B}$ requires conditional probabilities of some of the remaining nodes be changed. The changes can be made with little numerical computation by using the marginal probability $P_{\mathcal{T}_c}(\pi_{d,r})$. Since the marginal probability must be computed in order to obtain the evaluation functional of the reduced tail, the cost of node pruning is small.

The resulting ID after pruning $an(\pi_{d,2}) \cap X_2$ from $\mathcal{B}$ will be called the *reduced body* of $\mathcal{N}$ w.r.t d. Formally, it is obtained from the augmented body $\mathcal{B}$, the reduced tail $\mathcal{T}$, and the marginal probability $P_{\mathcal{T}_c}(\pi_{d,r})$ via the following procedure:

Procedure `redBody`$(\mathcal{B}, P_{\mathcal{T}_c}(\pi_{d,r}), \mathcal{T})$:

1. Prune from $\mathcal{B}$ all nodes in $an(\pi_{d,2}) \cap X_2$.
2. Prune arcs into and conditional probabilities of nodes in $\pi_{d,2}$.
3. (Enumerate all nodes in $\pi_{d,2}$ as $c_1, \ldots, c_k$ such that, in $\mathcal{N}$, c_i is not an ancestor of c_j if $i > j$. Let Z_i be the set of nodes in $Z=\pi_{d,r} \cap \pi_{d,1}$ that, in $\mathcal{T}$, are ancestors to c_1, or c_2, $\ldots$, or c_i.) For each i, make c_i a child of each node in $\{c_1, \ldots, c_{i-1}\} \cup Z_i$ and define

$$P(c_i | c_1, \ldots, c_{i-1}, Z_i) = \frac{\sum_{c_{i+1}, \ldots, c_k} P_{\mathcal{T}_c}(\pi_{d,r})}{\sum_{c_i, c_{i+1}, \ldots, c_k} P_{\mathcal{T}_c}(\pi_{d,r})}. \tag{18}$$

4. Return the resulting ID.

We use $\mathcal{B}_r$ to denote the reduced body. It consists of only nodes in $X_1 \cup \pi_d$ in addition to the new value node u and hence does not contain any node in the downstream set X_2.

The following proposition says that the output of `redBody` is indeed an ID and optimal decision rules for decision nodes other than d can be found in the ID.

Proposition 3 *$\mathcal{B}_r$ is an ID and it is equivalent to $\mathcal{B}$ in the sense that*

1. *They have the same possible policies, and*
2. *For any policy Δ_1 for $\mathcal{B}$ (or equivalently for $\mathcal{B}_r$),*

$$E_{\Delta_1}[\mathcal{B}] = E_{\Delta_1}[\mathcal{B}_r].$$

Proof of the proposition is rather tedious. We postpone it to the appendix.

5.3 Expected values of value networks

Let $\mathcal{N}$ be a value network. If all value nodes are converted into random nodes via Cooper's transformation, then $\mathcal{N}$ becomes a BN. Denote the BN also by $\mathcal{N}$. Following the same line of reasoning as in the proof of Proposition 2, one can show that the expected value $E[\mathcal{N}]$ of value network $\mathcal{N}$ is given by $E[\mathcal{N}]=\sum_{v\in V} P_{\mathcal{N}}(v{=}1)M_v$, where V is the set of value nodes.

Let $\mathcal{N}_v$ be obtained from the BN $\mathcal{N}$ by pruning nodes outside $an(\{v\})$. By Corollary 1, $P_{\mathcal{N}}(v{=}1){=}P_{\mathcal{N}_v}(v{=}1)$. Hence the list of BN inference problems that we need to solve in order to obtain the expected value is as follows:

BN inference problems: $P_{\mathcal{N}_v}(v{=}1)$ for each $v\in V$.

Assume those problems are solve by `BNinf`. The following procedure computes the expected value of the value network $\mathcal{N}$.

Procedure `expVal`($\mathcal{N}$)

1. For each $v\in V$, obtain $\mathcal{N}_v$ from $\mathcal{N}$ by pruning nodes outside $an(\{v\})$. Compute $P_{\mathcal{N}_v}(v = 1)$ by calling `BNinf`($v, \mathcal{N}_v$).
2. Return $\sum_{v\in V} P_{\mathcal{N}_v}(v{=}1)M_v$.

5.4 An algorithm

Discussions of the previous three subsections lead to the following algorithm for evaluating IDs.

Procedure `evalID`($\mathcal{N}$):

1. **While** there are decision nodes in $\mathcal{N}$
   (a) Find the tail decision node d.
   (b) $\mathcal{T}$=`redTail`($\mathcal{N}, d$).
   (c) Call `evalFun`($\mathcal{T}, d$) to compute $e_{\mathcal{T}}(\pi_{d,r}, d)$ and $P_{\mathcal{T}_c}(\pi_{d,r})$.
   (d) Find an optimal decision rule for d via

   $$\delta^*(\pi_{d,r}) = arg\ \ max_d e_{\mathcal{T}}(\pi_{d,r}, d).$$

   (e) $\mathcal{B}$ = `augBody`($\mathcal{N}, d, e_{\mathcal{T}}(\pi_{d,r}, d)$).

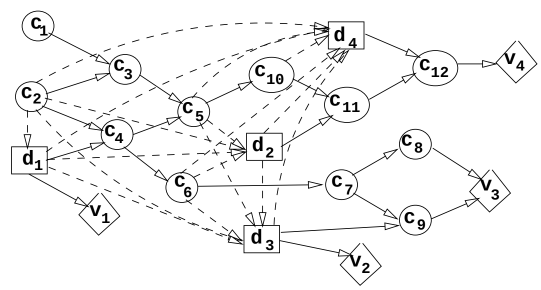

Figure 4: An ID.

(f) **If** some nodes in π_d have parents in the downstream set X_2 of $\mathcal{N}$ w.r.t d,

$$\mathcal{B} = \texttt{redBody}(\mathcal{B}, P_{\mathcal{T}_c}(\pi_{d,r}), \mathcal{T}).$$

(g) $\mathcal{N}$=$\mathcal{B}$.

Endwhile

(After the while-loop, $\mathcal{N}$ becomes a value network.)

2. Return the optimal decision rules and `expVal`($\mathcal{N}$).

The procedure `evalID` identifies a list BN inference problems and specifies how optimal decision rules can be obtained from the solutions of those problems. It leaves it to the user to choose an algorithm for solving the BN inference problems. As such, it is really an algorithm for reducing ID evaluation into BN inference problems.

6 An example

This section illustrates `evalID` by using the ID in Figure 4, which is borrowed from Jensen *et al* (1994). Arcs into decision nodes are dashed for readability. We list the probabilistic inference problems identified by `evalID` and describe how `evalID` obtains optimal decision rules using the solutions of the problems.

6.1 Optimal decision rule for d_4

Denote the ID by $\mathcal{N}$. Since it contains decision node, `evalID` enters the while-loop. In the while-loop, Step 1(a) finds that d_4 is the tail decision node and Step 1(b) constructs that reduced tail $\mathcal{T}$=redTail$(\mathcal{N}, d_4)$. To get a clear picture of $\mathcal{T}$, note that the downstream set X_2 is $\{d_4, c_{11}, c_{12}, v_4\}$. Since no parents of d_2 have parents in X_2, $\pi_{d_4,2}$=$\emptyset$. Among the parents of d_4, only c_{10} and d_4 are parents to nodes in $\pi_{d_4,2}\cup X_2\backslash\{d_4\}=X_2\backslash\{d_4\}$, hence c_{10} and d_2 are the all the relevant parents of d_4. In other words, $\pi_{d_4,r}=\{c_{10}, d_2\}$. Consequently, $\mathcal{T}$ consists of nodes c_{10}, d_2, d_4, c_{11}, c_{12}, and v_4 and is as shown in Figure 5 (2), where v_4 have been converted into a random node by Cooper's transformation.

Step 1(c) calculates the evaluation functional $e_{\mathcal{T}}(\pi_{d_4,r}, d_4)$. In the process, the BNs $\mathcal{T}_c$ and $\mathcal{T}_{v_4}$ are obtained from $\mathcal{T}$ by pruning nodes outside $an(\pi_{d_4,r})$ and $an(\pi_{d_4,r}\cup\{d_4, v_4\})$ respectively. Since $an(\pi_{d_4,r})=\pi_{d_4,r}=\{c_{10}, d_2\}$, $\mathcal{T}_c$ consists of two nodes c_{10} and d_2. They are isolated from each other and both have uniform distributions. Since $an(\pi_{d_4,r}\cup\{v_4\})$ contains all nodes in the tail, $\mathcal{T}_{v_4}$ is the same as $\mathcal{T}$. The subroutine `BNinf` is called to compute the following probabilities:

BN inference problems: $P_{\mathcal{T}_c}(c_{10}, d_2), P_{\mathcal{T}_{v_4}}(v_4{=}1, c_{10}, d_2, d_4)$.

Thereafter, the evaluation functional is obtained by

$$e_{\mathcal{T}}(c_{10}, d_2, d_4) = \frac{P_{\mathcal{T}_{v_4}}(v_4{=}1, c_{10}, d_2, d_4) M_{v_4}}{P_{\mathcal{T}_c}(c_{10}, d_2)/|\Omega_{d_4}|}.$$

Step 1(d) finds an optimal decision rule for d_4 via $\delta_4^*(c_{10}, d_2) = arg\ \ max_{d_4} e_{\mathcal{T}}(c_{10}, d_2, d_4)$.

Step 1(e) calls `augBody` to construct the augmented body of $\mathcal{N}$ w.r.t d_4, which is shown in Figure 5 (1). The utility function of the new value node u_4 is given by $f_{u_4}(c_{10}, d_2) = max_{d_4} e_{\mathcal{T}}(c_{10}, d_2, d_4)$. Since no nodes in $\pi_{d_4,r}$ have parents in X_2, Step 1(f) is skipped.

6.2 Optimal decision rule for d_3

$\mathcal{N}$ now denotes the ID in Figure 5 (1), where d_3 is the tail decision nodes. The reduced tail $\mathcal{T}$ of $\mathcal{N}$ w.r.t d_3 is shown in Figure 6 (2) and $\pi_{d_3,r}=\{c_6\}$.

The BNs $\mathcal{T}_c$, $\mathcal{T}_{v_2}$, and $\mathcal{T}_{v_3}$ are also shown in Figure 6. The following probabilistic inference problems are solved by `BNinf`:

BN inference problems: $P_{\mathcal{T}_c}(c_6), P_{\mathcal{T}_{v_2}}(v_2{=}1, c_6, d_3), P_{\mathcal{T}_{v_3}}(v_3{=}1, c_6, d_3)$.

Afterwards, the evaluation functional obtained is obtained via

$$e_{\mathcal{T}}(c_6, d_3) = \frac{P_{\mathcal{T}_{v_2}}(v_2{=}1, c_6, d_3) M_{v_2} + P_{\mathcal{T}_{v_3}}(v_3{=}1, c_6, d_3) M_{v_3}}{P_{\mathcal{T}_c}(c_6)/|\Omega_{d_3}|},$$

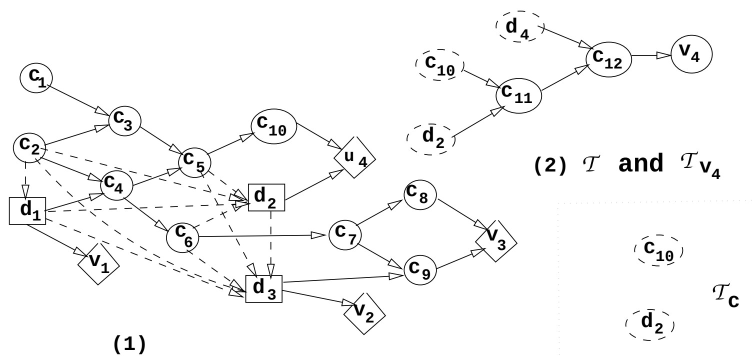

Figure 5: The reduced tail and augmented body w.r.t d_4.

and an optimal decision rule for d_3 obtained via $\delta_3^*(c_6) = arg\ \ max_{d_3} e_{\mathcal{T}}(c_6, d_3)$. The augmented body of $\mathcal{N}$ w.r.t d_3 is shown in Figure 6 (1). The utility function of the new value node u_3 is given by $f_{u_3}(c_6) = max_{d_3} e_{\mathcal{T}}(c_6, d_3)$.

6.3 Optimal decision rule for d_2

$\mathcal{N}$ now denotes the ID in Figure 6 (1). Here d_2 is the tail decision node. The reduced tail $\mathcal{T}$ of $\mathcal{N}$ w.r.t d_2 is shown in Figure 7 (2) and $\pi_{d_2,r} = \{c_5\}$.

The BN $\mathcal{T}_c$ consists of a single node c_5 and $\mathcal{T}_{u_4}$ is the same as $\mathcal{T}$. The following probabilistic inference problems are solved by `BNinf`:

BN inference problems: $P_{\mathcal{T}_c}(c_5), P_{\mathcal{T}_{u_4}}(u_4{=}1, c_5, d_2)$.

Afterwards the evaluation functional is obtained via

$$e_{\mathcal{T}}(c_5, d_2) = \frac{P_{\mathcal{T}_{u_4}}(u_4{=}1, c_5, d_2) M_{u_4}}{P_{\mathcal{T}_c}(c_5)/|\Omega_{d_2}|}.$$

An optimal decision rule for d_2 is found by $\delta_2^*(c_5) = arg\ \ max_{d_2} e_{\mathcal{T}}(c_5, d_2)$. The augmented body of $\mathcal{N}$ w.r.t d_2 is shown in Figure 7 (1). The utility function of the new value node u_2 is given by $f_{u_2}(c_5) = max_{d_2} e_{\mathcal{T}}(c_5, d_2)$.

6.4 Optimal decision rule for d_1

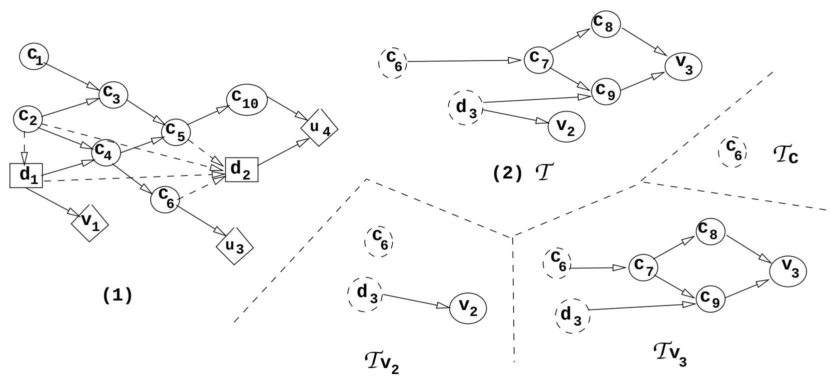

Figure 6: The reduced tail and augmented body w.r.t d_3.

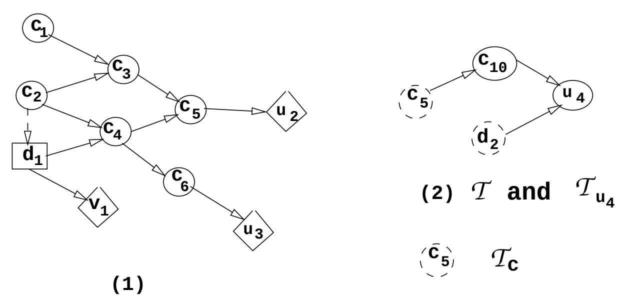

Figure 7: The reduced tail and augmented body w.r.t d_2.

$\mathcal{N}$ now denotes the ID in Figure 7 (1). Here d_1 is the only decision node. The reduced tail of $\mathcal{N}$ w.r.t d_1, which is shown in Figure 8 (2) and $\pi_{d_1,r}$=$\{c_2\}$.

The BNs $\mathcal{T}_c$, $\mathcal{T}_{v_1}$, $\mathcal{T}_{u_2}$, and $\mathcal{T}_{u_3}$ are also shown in Figure 8. The following probabilistic inference problems are solved by `BNinf`:

BN inference problems: $P_{\mathcal{T}_c}(c_2), P_{\mathcal{T}_{v_1}}(v_1{=}1, c_2, d_1), P_{\mathcal{T}_{u_2}}(u_2{=}1, c_2, d_1), P_{\mathcal{T}_{u_3}}(u_3{=}1, c_2, d_1)$.

Thereafter, the evaluational functional is obtained via

$$e_{\mathcal{T}}(c_2, d_1) = \frac{P_{\mathcal{T}_{v_1}}(u_2{=}1, c_2, d_1)M_{v_1} + P_{\mathcal{T}_{u_2}}(u_2{=}1, c_2, d_1)M_{u_2} + P_{\mathcal{T}_{u_3}}(u_3{=}1, c_2, d_1)M_{u_3}}{P_{\mathcal{T}_c}(c_2)/|\Omega_{d_1}|},$$

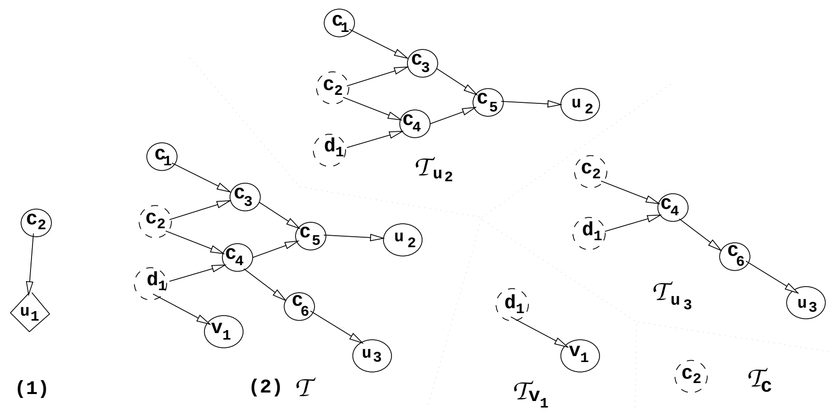

Figure 8: The reduced tail and augmented body w.r.t d_1.

and an optimal decision rule for d_1 is found by $\delta_1^*(c_2) = arg\ max_{d_1} e_{\mathcal{T}}(c_2, d_1)$. The augmented body of $\mathcal{N}$ w.r.t d_1 is shown in Figure 8 (1). The utility function of the new value variable u_1 is given by $f_{u_1}(c_2) = max_{d_2} e_{\mathcal{T}}(c_2, d_1)$.

6.5 The optimal expected value

$\mathcal{N}$ now denotes the value network in Figure 8 (1). The BN $\mathcal{N}_{u_1}$ is the same as $\mathcal{N}$ itself except with u_1 converted into a random node. The following probabilistic inference problem is solved by `BNinf`:

BN inference problem: $P_{\mathcal{N}_{u_1}}(u_1{=}1)$.

Thereafter, the quantity $P_{\mathcal{N}_{u_1}}(u_1{=}1)M_{u_1}$ is calculated. It is the optimal expected value of the original ID.

7 Comparisons with previous methods

Both `evalID` and the Shachter-Peot algorithm reduce ID evaluation into BN inference problems, which can be solved using arbitrary BN inference algorithms. This section shows that the probabilistic inference problems induced by `evalID` are easier to solve than those induced by the Shachter-Peot algorithm.

Among all previous algorithms, Shenoy's fusion algorithm (Shenoy 1992) and the algorithms by Ndilikilikesha (1994) and Jensen *et al* (1994) are the most efficient. Those three algorithms are basically equivalent in the sense that they all carry out essentially the same numerical computations. They are *direct evaluation algorithms* in the sense that they evaluate IDs directly without the reduction to BN inference problems. An method that reduces ID evaluation into BN inference problems would be unattractive if it is less efficient than those direct evaluation algorithms no matter what BN inference algorithm is used. We will show that this is not the case for `evalID` by comparing it with Shenoy's fusion algorithm.

Non-numerical computations in `evalID` include the identification of tail decision nodes, the construction of reduced tails and bodies, and pruning of nodes in a reduced tail that are irrelevant to particular BN inference problems . They are negligible compared to numerical computations. We will hence focus the comparisons on numerical computations.

7.1 Comparisons with Shachter and Peot's algorithm

The Shachter-Peot algorithm applies only when there is one value node. Let $\mathcal{N}$ be an ID with one value node. Assume that there are no barren random nodes[7].

Let $\mathcal{N}'$ be the BN defined in Section 3. In the Shachter-Peot algorithm, the BN inference problem $P_{\mathcal{N}'}(\pi_d, d|v{=}1)$ needs to be solved in order to obtain an optimal decision rule for the tail decision node d. Let $\mathcal{T}$ be the reduced tail of $\mathcal{N}$ w.r.t d. In `evalID`, we need to solve two BN inference problems, namely $P_{\mathcal{T}}(\pi_{d,r})$ and $P_{\mathcal{T}}(\pi_{d,r}, d, v{=}1)$.

The inference problem $P_{\mathcal{N}'}(\pi_d, d|v{=}1)$ is more difficult to solve than $P_{\mathcal{T}}(\pi_{d,r})$ and $P_{\mathcal{T}}(\pi_{d,r}, d, v{=}1)$ than for two reasons. First, it involves more variables. Due to the no-forgetting constraint, all other decision nodes and their parents must be parents of d, i.e. in π_d. However, as we have seen in the example of the previous section, many of the parents of d are irrelevant to d. The set $\pi_{d,r}$ usually contains much less variables than π_d. Second, $\mathcal{N}'$ usually consists of many more nodes than the reduced tail $\mathcal{T}$. Moreover, since $\mathcal{N}$ has no barren random nodes, no nodes in $\mathcal{N}'$ can be pruned using Lemma 1 when computing $P_{\mathcal{N}'}(\pi_d, d|v{=}1)$, while $\mathcal{T}$ can be further reduced to $\mathcal{T}_c$ when computing $P_{\mathcal{T}}(\pi_{d,r})$ and $\mathcal{T}_v$ when computing $P_{\mathcal{T}}(\pi_{d,r}, d, v{=}1)$.

In `evalID`, nodes in the downstream set X_2 are pruned after an optimal decision rule for d has been obtained. In other words, those nodes do no participate in the

[7]Barren random nodes, if exist, can be pruned at a preprocessing step (Shachter 1986).

computations for other decision nodes. However, no nodes are pruned in the Shachter-Peot algorithm. Computations for each decision node involve all nodes in $\mathcal{N}$.

7.2 Comparisons with Shenoy's fusion algorithm

This subsection first introduces a variation of `evalID`, called `evalID1`, that performs essentially the same numerical computations as Shenoy's fusion algorithm and then compares `evalID` with `evalID1`.

7.2.1 A variation of evalID

We will refer to non-negative functions of a set of variables simply as *factors*. Conditional probabilities and utility functions are all factors. Let ρ be an ordering of the nodes in $X_2 \backslash V_2 \cup \{d\}$. In stead of `evalFun`, `evalID1` uses the following procedure to compute the evaluation functional of the reduced tail $\mathcal{T}$ and the marginal probability $P_{\mathcal{T}}(\pi_{d,r})$.

Procedure `evalFun1`$(\mathcal{T}, d)$

1. Let $\mathcal{F}$ be the list of utility functions of value nodes in V_2 and let $\mathcal{P}$ be the list of conditional probabilities of nodes in $X_2 \backslash V_2 \cup \{d\}$.
2. **While** $\rho \neq \emptyset$, remove the first node x from ρ and call `fuse`$(\mathcal{P}, \mathcal{F}, x)$.
3. Multiply all factor in $\mathcal{P}$ (the product is $P_{\mathcal{T}}(\pi_{d,r}, d)$.)
4. Sum all factors in $\mathcal{F}$ (the result is $e_{\mathcal{T}}(\pi_{d,r}, d)$).
5. Return $e_{\mathcal{T}}(\pi_{d,r}, d)$ and $\sum_d P_{\mathcal{T}}(\pi_{d,r}, d)$.

The subroutine `fuse` is given as follows:

Procedure `fuse`$(\mathcal{P}, \mathcal{F}, x)$

1. Remove from $\mathcal{P}$ all the factors $p_1, \ldots, p_k$ that involve x. If such factors exist, add the new factor $p = \sum_x \prod_{i=1}^{k} p_i$ to $\mathcal{P}$.
2. Remove from $\mathcal{F}$ all the factors $f_1, \ldots, f_l$ that involve x. If such factors exist, add the new factor $\sum_x [\sum_{i=1}^{l} f_i][\prod_{i=1}^{k} p_i]/p$ to $\mathcal{F}$.

Let $\mathcal{N}$ be a value network and ρ be an ordering of the random nodes. Instead of `expVal`, `evalID1` uses the following procedure to compute the expected value of $\mathcal{N}$.

Procedure `expVal1`($\mathcal{N}$):

1. Let $\mathcal{F}$ be the list of utility functions of value nodes in $\mathcal{N}$ and let $\mathcal{P}$ be the list of conditional probabilities of random nodes in $\mathcal{N}$.
2. **While** $\rho\neq\emptyset$, remove the first node x from ρ and call `fuse`($\mathcal{P}, \mathcal{F}, x$).
3. Return the sum all factors in $\mathcal{F}$ (which is the expected value of $\mathcal{N}$).

Shenoy's fusion algorithm requires an ordering, say ρ_s, of all the random and decision nodes. If the orderings in `evalFun1` and `expVal1` *conforms* to ρ_s in the sense that relative orders of nodes are the same, than `evalID1` carries out essentially the same numerical computations as Shenoy's fusion algorithm (see Shenoy 1992).

7.2.2 Comparisons

We now set out to compare `evalID1` and `evalID`. Suppose `evalID` employs the VE (variable elimination) algorithm [8] for probabilistic inference. Let A be a subset of nodes in a BN $\mathcal{M}$ and let ρ be an ordering of nodes outside A. VE computes the marginal probability $P_{\mathcal{M}}(A)$ as follows:

Procedure `BNinf-VE`($\mathcal{M}, A$):

1. Let $\mathcal{P}$ be the list of conditional probabilities in $\mathcal{M}$.
2. **While** $\rho\neq\emptyset$, remove the first node x from ρ and remove from $\mathcal{P}$ all the factors $f_1, \ldots, f_k$ that involve x. If such factors exist, add the new factor $\sum_x \prod_{i=1}^{k} f_i$ to the list $\mathcal{P}$.
3. Multiply all factors in $\mathcal{P}$ and return the result (which is $P_{\mathcal{M}}(A)$).

Consider computing $P_{\mathcal{T}_c}(\pi_{d,r})$ and $P_{\mathcal{T}_v}(\pi_{d,r}, d, v{=}1)$ using `BNinf-VE`. If the ordering in `BNinf-VE` conforms to the ordering in `evalFun1`, `BNinf` performs no more numerical computations than `evalFun1`($\mathcal{T}, d$). Therefore, the amount of numerical computations carried out by `evalFun`($\mathcal{T}, d$) is at most $1{+}m$ times that carried by `evalFun1`($\mathcal{T}, d$), where m is the number of value nodes in $\mathcal{T}$. Similarly, for any value network $\mathcal{N}$, the amount of numerical computations carried out by `expVal`($\mathcal{N}$) is at

[8] The idea behind VE is implicit in many papers (e.g. Shenoy 1992). It was first made explicit in Zhang and Poole (1994) and extended to exploit independence of causal influence by Zhang and Poole (1996). Dechter (1996) also calls VE elim-bel.

most m times that carried by `expVal1`($\mathcal{N}$), where m is the number of value nodes in $\mathcal{N}$.

We argue that `evalFun`($\mathcal{T}, d$) is usually more efficient that `evalFun1`($\mathcal{T}, d$), especially when $\mathcal{T}$ is large. Define the *size* of a factor to be the number of variables involved in the factor. It is well understood in the BN literature that the complexities of `BNinf` and `evalFun1` is largely determined by the sizes of the largest factors encountered; they are exponential in the largest factors sizes. The BNs $\mathcal{T}_c$ and each $\mathcal{T}_v$ are subnetworks of $\mathcal{T}$. When $\mathcal{T}$ is large, the differences between $\mathcal{T}$ and $\mathcal{T}_c$ or $\mathcal{T}_v$ are usually also large. In such a case, the maximum factor sizes encountered by `BNinf` are smaller than those encountered by `evalFun1` and hence the amount of time `BNinf-VE` spends in computing $P_{\mathcal{T}_c}(\pi_{d,r})$ or $P_{\mathcal{T}_v}(\pi_{d,r}, d, v{=}1)$ is much less than that `evalFun1`($\mathcal{T}, d$) takes. Consequently, `evalFun`($\mathcal{T}, d$) takes less time than `evalFun1`($\mathcal{T}, d$).

A second reason for `evalFun`($\mathcal{T}, d$) being more efficient than `evalFun1`($\mathcal{T}, d$) is the fact that the former does not perform numerical divisions until the last step, while the latter might divide factors when fusing each node.

Similarly, `expVal` is usually more efficient than `expVal`. Hence `evalID` is usually more efficient than `evalID1` and therefore more efficient than Shenoy's fusion algorithm.

We would like to emphasize that arbitrary BN inference algorithms can be used in `evalID`, while this is not the case in Shenoy's fusion algorithm and all the direct evaluation algorithms for that matter. This is a big advantage (see discussions in the next section).

8 Conclusions

This paper is about reducing ID evaluation into BN inference problems. Such an exercise is interesting because it allows the use of arbitrary BN inference algorithms in evaluating IDs. Two reduction methods have been proposed previously (Cooper 1988 and Shachter and Peot 1994). A new method is presented in this paper. The BN inference problems induced by the new method are easier to solve than those induced by earlier methods.

When coupled with the VE algorithm, the performance of the new method is, in the worst case, within a small constant factor of that of the most efficient previous algorithms, which evaluate ID directly without the reduction into BN inference problems. We have argued that the combination of the new method and VE is usually more efficient in large IDs.

The fact that it allows arbitrary BN inference algorithms is big advantage of the new method. From a system development point of view, the method enables one to easily add ID evaluation capabilities to any BN inference packages. From the efficiency point of view, speeding up inference in BNs has been and still is an active research area. There are algorithms that exploit independence of causal influence (e.g. Zhang and Poole 1996) and that exploit special structures in the conditional probability tables (e.g. Boutilier *et al* 1996). The new method facilitates ready incorporation of those algorithms, as well as future advances in BN inference, in ID evaluation. We are not aware of any approximate algorithms for IDs, while there is a rich collection of approximate and simulation algorithms for BNs. The new method also opens up the possibility of approximate algorithms for ID, which might be necessary in order to solve large decision problems.

Acknowledgement

Research was supported by Hong Kong Research Council under grant HKUST658/95E and by Sino Software Research Institute under grant SSRC95/96.EG01 . The paper has benefited from discussions with David Poole and Prakash Shenoy. I am also grateful to the anonymous reviewers and the editor Nick Cercone for valuable comments.

References

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

[2] R. Dechter (1996), Bucket elimination: A unifying framework for probabilistic inference, *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pp. 211-219.

[3] G. F. Cooper (1988), A method for using belief networks as influence diagrams, in *Proceedings of the Fourth Workshop on Uncertainty in Artificial Intelligence*, pp. 55-63.

[4] 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.

[5] F. Jensen, F. V. Jensen, and S. L. Dittmer (1994), From influence diagrams to junction trees, *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 367-373.

[6] S. L. Lauritzen and D. J. Spiegelhalter (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.

[7] P. Ndilikilikesha (1994), Potential influence diagrams, *International Journal of Approximate Reasoning*, 11, pp. 251-285.

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

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

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

[11] Shachter and Peot (1992), Decision making using probabilistic inference methods, in *Proc. of 8th Conference on Uncertainty in Artificial Intelligence*, pp. 276-283.

[12] P. P. Shenoy (1992), Valuation-based systems for Bayesian decision analysis, *Operations Research*, 40(3), pp. 463-484.

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

[14] N. L. Zhang and D. Poole (1994), A simple approach to Bayesian network computations, in *Proceedings of the Tenth Canadian Conference on Artificial Intelligence*, pp. 171-178.

[15] N. L. Zhang and D. Poole (1996), Exploiting causal independence in Bayesian network inference, *Journal of Artificial Intelligence Research*, **5**, pp. 301-328.

Appendix: Proof of Proposition 3

To prove the first part, it suffices to show that

1. There are no directed loops in $\mathcal{B}_r$.
2. The right hand side of equation (18) depends only one nodes in the set $\{c_1, \ldots, c_i\} \cup Z_i$.

To show that there are no directed loops, note that all the arcs that `redBody` introduces are from nodes in $\{c_1, \ldots, c_{i-1}\} \cup Z_i$ to c_i. For any $1 \leq j < i$, introduction of the arc $c_j \rightarrow c_i$ does not create a loop because c_i is not a ancestor of c_j. Let $z \in Z_i$. If z is an ancestor of c_i, then introduction of the arc $z \rightarrow c_i$ does not create a loop. If z is an ancestor of c_j for some $j < i$, then c_i cannot be an ancestor of z because it is not an ancestor of c_j. Consequently, introduction of the arc $z \rightarrow c_i$ does not create a loop.

To show that the term on the right hand side of equation (18) depends only on nodes in $\{c_1, \ldots, c_i\} \cup Z_i$, consider $\sum_{c_{i+1}, \ldots, c_k} P_{\mathcal{T}_c}(\pi_{d,r}) = \sum_{c_{i+1}, \ldots, c_k} P_{\mathcal{T}_c}(c_1, \ldots, c_k, Z)$ $= P_{\mathcal{T}_c}(c_1, \ldots, c_i, Z)$. Let $\mathcal{T'}_c$ be the BN obtained from $\mathcal{T}_c$ by pruning nodes outside $an(\{c_1, \ldots, c_i\} \cup Z)$. By Lemma 1,

$$P_{\mathcal{T}_c}(c_1, \ldots, c_i, Z) = P_{\mathcal{T'}_c}(c_1, \ldots, c_i, Z).$$

Consider an arbitrary node x in $Z \backslash Z_i$. It is not an ancestor of c_1, $\ldots$, or c_i. Since nodes in Z do not have parents in $\mathcal{T}_c$, x has no parents and cannot be an ancestor of any other nodes in Z. Hence it must be an isolated node in $\mathcal{T'}_c$. Since the probability of x is the uniform distribution, we have

$$P_{\mathcal{T'}_c}(c_1, \ldots, c_i, Z) = P_{\mathcal{T'}_c}(c_1, \ldots, c_i, Z_i) \prod_{x \in Z \backslash Z_i} 1/|\Omega_x|.$$

In other words, $P_{\mathcal{T}_c}(c_1, \ldots, c_i, Z)$ depends only on nodes in $\{c_1, \ldots, c_i\} \cup Z_i$. Hence so does the term on the right hand side of equation (18).

We new set out to prove the second part. Procedure `redBody` neither prunes nodes in $X_1 \cup \pi_{d,1}$ nor changes their parents. According to Proposition 1 (4), all decision nodes in $\mathcal{B}$ are in $\pi_{d,1}$. So are their parents. Hence the decision nodes and their parents are not touched by `redBody`. Therefore, $\mathcal{B}$ and $\mathcal{B}_r$ contain the same decision nodes and they have the same parents in both $\mathcal{B}$ and $\mathcal{B}_r$. The first item follows.

Given a policy Δ_1, decision nodes can be regarded as random nodes. If all the value nodes are converted into random nodes by Cooper's transformation, then both

$\mathcal{B}$ and $\mathcal{B}_r$ become BNs. Denote these BNs by $\mathcal{B}_{\Delta_1}$ and $\mathcal{B}_{r,\Delta_1}$. To prove the second item, it suffices to show that $P_{\mathcal{B}_{\Delta_1}}(X_1, \pi_d) = P_{\mathcal{B}_{r,\Delta_1}}(X_1, \pi_d)$.

Since $\mathcal{B}_{\Delta_1}$ consists of nodes in $X_1 \cup \pi_d \cup (an(\pi_{d,2}) \cap X_2)$, we have

$$\begin{aligned} P_{\mathcal{B}_{\Delta_1}}(X_1, \pi_d) &= \sum_{an(\pi_{d,2})\cap X_2} [\prod_{x\in X_1\cup\pi_{d,1}} P_{\Delta_1}(x|\pi_x)][\prod_{x\in\pi_{d,2}\cup(an(\pi_{d,2})\cap X_2)} P(x|\pi_x)] \\ &= [\prod_{x\in X_1\cup\pi_{d,1}} P_{\Delta_1}(x|\pi_x)] \sum_{an(\pi_{d,2})\cap X_2} [\prod_{x\in\pi_{d,2}\cup(an(\pi_{d,2})\cap X_2)} P(x|\pi_x)], \end{aligned}$$

where the second equation is true because none of the nodes in $X_1 \cup \pi_{d,1}$ have parents in the downstream set X_2 and hence the first factor does not depend on nodes in $an(\pi_{d,2}) \cap X_2$.

Consider the BN $\mathcal{T}_c$. It consists of nodes in $\pi_{d,r} \cup (an(\pi_{d,2}) \cap X_2)$. The conditional probability $P(x|\pi_x)$ for any $x \in \pi_{d,2} \cup (an(\pi_{d,2}) \cap X_2)$ in $\mathcal{T}_c$ is the same as that in $\mathcal{N}$, hence the same as that in $\mathcal{B}$. Nodes in $Z = \pi_{d,r} \setminus \pi_{d,2}$ are root nodes (without parents) and their probabilities are uniform distributions. Hence

$$\begin{aligned} P_{\mathcal{T}_c}(\pi_{d,r}) &= \sum_{an(\pi_{d,2})\cap X_2} [\prod_{x\in\pi_{d,2}\cup(an(\pi_{d,2})\cap X_2)} P(x|\pi_x)][\prod_{x\in Z} 1/|\Omega_x|] \\ &= \sum_{an(\pi_{d,2})\cap X_2} [\prod_{x\in\pi_{d,2}\cup(an(\pi_{d,2})\cap X_2)} P(x|\pi_x)] P_{\mathcal{T}_c}(Z). \end{aligned}$$

Consequently,

$$\begin{aligned} P_{\mathcal{B}_{\Delta_1}}(X_1, \pi_d) &= [\prod_{x\in X_1\cup\pi_{d,1}} P_{\Delta_1}(x|\pi_x)] \frac{P_{\mathcal{T}_c}(\pi_{d,r})}{P_{\mathcal{T}_c}(Z)} \\ &= [\prod_{x\in X_1\cup\pi_{d,1}} P_{\Delta_1}(x|\pi_x)] \prod_{i=1}^{k} \frac{\sum_{c_{i+1},\ldots,c_k} P_{\mathcal{T}_c}(\pi_{d,r})}{\sum_{c_i,c_{i+1},\ldots,c_k} P_{\mathcal{T}_c}(\pi_{d,r})} \\ &= [\prod_{x\in X_1\cup\pi_{d,1}} P_{\Delta_1}(x|\pi_x)] \prod_{i=1}^{k} P(c_i|_1, \ldots, c_{i-1}, Z_i) \\ &= P_{\mathcal{B}_{r,\Delta_1}}(X_1, \pi_d). \end{aligned}$$

The proposition is proved. □