Minimizing decision table sizes in influence diagrams: dimension shrinking[1]

Nevin Lianwen Zhang[2], Runping Qi and David Poole
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1Z2, CANADA

Abstract

One goal in evaluating an influence diagram is to compute an optimal decision table for each decision node. More often than not, one is able to shrink the sizes of some of the optimal decision tables without any loss of information. This paper investigates when the opportunities for such shrinkings arise and how can we detect them as early as possible so as to to avoid unnecessary computations. One type of shrinking, namely dimension shrinking, is studied. A relationship between dimension shrinking and what we call lonely arcs is established, which enables us to make use of the opportunities for dimension shrinking by means of pruning lonely arcs at a preprocessing stage.

1 Introduction

Influence diagrams were introduced by Howard and Matheson [3] as a graphical representation of decision problems. In comparison with decision trees (Raiffa [7]), 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 of three types of nodes: random nodes, decision nodes, and value nodes, which respectively represent random quantities, decisions to make, and utilities (Howard and Matheson [3]).

Given an influence diagram, a decision table for a decision node is a table in which there is a cell for each array of values of its parents, and the value in that cell is the action to take when the parents do assume those values. One goal in evaluating an influence diagram is to compute an optimal decision table for each decision node.

The *size* of a decision table is the number of cells in it. It is the diagram itself, not the evaluation process that determines the size of a decision table. More often than not,

---

[1] Research is supported by NSERC Grant OGPOO44121. Comments from the reviewers' have been useful.

[2] Starting from January 1994: Department of Computer Science, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

after an optimal decision table is computed one realizes that it is possible to reduce the size of the table without losing any information.

There are at least two cases when the size of a decision table for a decision node d can be harmlessly reduced. First, there may exist a parent, say b, of d such that the b-dimension of the decision table is *irrelevant to* d in the sense that fixing the values of all the other parents of d, whatever value b takes the action for d remains the same. When it is the case, the b-dimension of the table can be deleted, resulting in a decision table of one less dimension. We call this deletion operation *dimension shrinking.*

A second case is when the real world situation that a cell represents never occurs in the sense that its probability of occurring is zero. We call such a situation *unreachable.* When a situation is unreachable, the action in the corresponding cell can be deleted from the table, resulting in a decision table with one less cell. We call this deletion operation *cell shrinking.*

Cell shrinking has been studied under the term of asymmetry by Fung and Shachter [2], Covaliu and Oliver [1], Smith *et al* [12], Shenoy [11], and Qi and Poole [6]. This paper is concerned with dimension shrinking. We investigate when are dimension shrinkings possible? How can one detect them and make use of them to save computations?

A relationship between dimension shrinking and what we call lonely arcs is established (Section 4), which enables us to make use of the opportunities for dimension shrinkings by means of pruning lonely arcs at a preprocessing stage (Section 5). Since pruning arcs from influence diagrams leads to the violation of the so-called no-forgetting constraint, we need to work with a setup that is more general than influence diagrams. Introduced by Zhang *et al* [13], stepwise-decomposable decision networks (Section 3) prove to be a good setup for studying dimension shrinking.

Let us begin by reviewing the concepts of influence diagrams and decision networks.

2 Decision networks and influence diagrams

This section gives the formal definitions of Bayesian networks, decision networks and influence diagrams. The reader is referred to Zhang *et al* [13] for semantic details.

Before getting started, let us note that in this paper, standard graph theory terms such as acyclic directed graphs, parents (direct predecessors), children (direct successors), predecessors, descendants (successors), leaves (nodes with no children), and roots (nodes with no parents) will be used without giving the definitions. The reader is referred to Lauritzen *et al* [4] for exact definitions.

We shall use π_x to denote the set of parents of a node x in a directed graph. We shall use Ω_x to denote the frame of a variable x, i.e the set of possible values of x. For any set B, define $\Omega_B = \prod_{x \in B} \Omega$.

A *Bayesian network* $\mathcal{N}$ is a triplet $\mathcal{N} = (X, A, \mathcal{P})$, where

1. X is a set of variables (nodes) and A is a set of arcs over X such that (X, A) is an acyclic directed graph,
2. $\mathcal{P}$ is a set $\{P(x|\pi_x)|x \in X\}$ of conditional probabilities of the variables given their respective parents[3].

---

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

The *prior joint probability* $P_{\mathcal{N}}(X)$ of a Bayesian network $\mathcal{N} = (X, A, \mathcal{P})$ is defined by

$$P(X) = \prod_{x \in X} P(x|\pi_x). \tag{1}$$

In words, $P(X)$ the multiplication of all the conditional probabilities in $\mathcal{N}$. For any subset $B \subseteq X$, the *marginal probability* $P(B)$ is defined by

$$P(B) = \sum_{X-B} P(X), \tag{2}$$

where $\sum_{X-B}$ means summation over all the values of the variables in $X-B$.

A *decision network skeleton* is an acyclic directed graph $G = (Y, A)$, which consists of three types of nodes: *random nodes*, *decision nodes*, and *values nodes*, and where the value nodes have no children. Figure 1 (1) shows a decision network skeleton. Following the convention in the literature, decision nodes are drawn as rectangles, random nodes as cycles, and value nodes as diamonds.

A decision network skeleton is *regular* if there is a directed path that contains all the decision nodes. A regular decision network skeleton is *no-forgetting* if each decision node and its parents are parents to all subsequent decision nodes. The skeleton in Figure 1 (1) is not no-forgetting.

A *decision network* $\mathcal{N}$ is a quadruplet $\mathcal{N} = (Y, A, \mathcal{P}, \mathcal{F})$ where

1. (Y, A) is a decision network skeleton. We use C to denote the set of random nodes, D to denote the set of decision nodes, and V to denote the set of value nodes.
2. $\mathcal{P}$ is a set $\{P(c|\pi_c)|c \in C\}$ of conditional probabilities of the random nodes given their respective parents.
3. $\mathcal{F}$ is a set $\{f_v : \Omega_{\pi_v} \rightarrow R^1 | v \in V\}$ of *value (utility) functions* for the value nodes, where R^1 stands for the set of real numbers.

We shall say that $\mathcal{N}$ is a decision network over the skeleton (Y, A) and that (Y, A) is the skeleton underlying $\mathcal{N}$.

A decision network is regular (or no-forgetting) if its underlying skeleton is. An *influence diagram* is a decision network that is regular and no-forgetting, and contains only one value node[4].

A *decision function (table)* for a decision node d_i is a mapping $\delta_i : \Omega_{\pi_{d_i}} \rightarrow \Omega_{d_i}$. The *decision function space* Δ_i for d_i is the set of all the decision functions for d_i. Let $D = \{d_1, \ldots, d_k\}$ be the set of all the decision nodes in a decision network $\mathcal{N}$. The Cartesian product $\Delta = \prod_{i=1}^{k} \Delta_i$ is called the *policy space* for $\mathcal{N}$, and a member $\delta = (\delta_1, \ldots, \delta_k) \in \Delta$ is called a *policy*. The relationship between a decision node d_i and its parents π_{d_i} as indicated by a decision function $\delta_i : \Omega_{\pi_{d_i}} \rightarrow \Omega_{d_i}$ is equivalent to the relationship as represented by the conditional probability $P_{\delta_i}(d_i|\pi_{d_i})$ given by

$$P_{\delta_i}(d_i{=}\alpha|\pi_{d_i}{=}\beta) = \begin{cases} 1 & \text{if } \delta_i(\beta) = \alpha \\ 0 & \text{otherwise,} \end{cases} \quad \forall \alpha \in \Omega_{d_i}, \forall \beta \in \Omega_{\pi_{d_i}}. \tag{3}$$

---

of x.

[4] In the literature, influence diagrams do no have to satisfy all those three constraints. However, only influence diagrams that satisfy all the three constraints have been studied before Zhang *et al* [13].

Since $\delta=(\delta_1,\ldots,\delta_k)$, we sometimes write $P_\delta(d_i|\pi_{d_i})$ for $P_{\delta_i}(d_i|\pi_{d_i})$. Because of equation (3), we will abuse the symbol δ by letting it also denote the set $\{P_\delta(d_i|\pi_{d_i})|d_i{\in}D\}$ of conditional probabilities of the decision nodes.

In a decision network $\mathcal{N}=(Y,A,\mathcal{P},\mathcal{F})$, let $X=C{\cup}D$. Let A_X be the set of all the arcs of A that lie completely in X. Then the triplet $(X,A_X,\mathcal{P}{\cup}\delta)$ is a Bayesian network [5]. We shall refer to this Bayesian network as the *Bayesian network induced from* $\mathcal{N}$ *by the policy* δ, and write it as $\mathcal{N}_\delta$. The prior joint probability $P_\delta(X)$ is given by

$$P_\delta(X)=\prod_{x\in C}P(x|\pi_x)\prod_{x\in D}P_\delta(x|\pi_x). \tag{4}$$

For any value node v, π_v contains no value nodes. Because if otherwise, there would be some other value nodes that has a child v. Hence $\pi_v{\subseteq}X$. The expectation $E_\delta[v]$ of the value function $f_v(\pi_v)$ of v under P_δ is given by

$$E_\delta[v]=\sum_{\pi_v}P_\delta(\pi_v)f_v(\pi_v).$$

The *expected value* $E_\delta[\mathcal{N}]$ of $\mathcal{N}$ under the policy δ is defined by

$$E_\delta[\mathcal{N}]=\sum_{v\in V}E_\delta[v]. \tag{5}$$

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

$$E[\mathcal{N}]=max_{\delta\in\Delta}E_\delta[\mathcal{N}]. \tag{6}$$

The optimal value of a decision network that does not have any value nodes is zero. An *optimal policy* δ^o is one that satisfies

$$E_{\delta^o}[\mathcal{N}]=E[\mathcal{N}]. \tag{7}$$

For a decision network that does not have 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 a decision network is to (1) find an optimal policy, and (2) find the optimal expected value.

3 Stepwise-decomposable decision networks

This paper proposes a way for making use of the opportunities for dimension shrinking by means of arc removal. Since pruning arcs from an influence diagram leads to the violation of the no-forgetting constraint, we need a more general setup. One may suggests general decision networks; the problem with them is that they are hard to evaluate. Our choice is stepwise-decomposable decision networks (SDDN's) (Zhang *et al* [13]), which is briefly reviewed below.

---

[5] Remember that δ denotes a set of conditional probabilities for the decision nodes.

Let $G = (X, A)$ be a directed graph. An arc from x to y is written as an ordered pair (x, y). The *moral graph* $m(G)$ of G is an undirected graph $m(G) = (X, E)$ whose edge set E is given by

$$E = \{\{x, y\} | (x, y) \text{ or } (y, x) \in A, \text{ or } \exists z \text{ such that } (x, z) \text{ and } (y, z) \in A\}.$$

In words, $\{x, y\}$ is an edge in the moral graph if either there is an arc between them or they share a common child. The term moral graph was chosen because two nodes with a common child are "married" into an edge [4].

In an undirected graph, two nodes x and y are *separated* by a set of nodes S if every path connecting x any y contains at least one node in S. In a directed graph G, x and y are *m-separated* by S if they are separated by S in the moral graph $m(G)$. Note that any node set m-separates itself from any other set of nodes.

Suppose $\mathcal{K} = (Y, A)$ is decision network skeleton and d is a decision node in $\mathcal{K}$. Let $Y_I(d, \mathcal{K})$, or simply Y_I, be the set of all the nodes that are m-separated from d by π_d, with the nodes in π_d excluded. Let $Y_{II}(d, \mathcal{K})$, or simply Y_{II}, be the set of all the nodes that are not m-separated from d by π_d. We observe that Y_I, π_d, and Y_{II} forms a partition of Y.

The decision node d is a *stepwise-decomposability candidate node*, or simply a *candidate node*, if the downstream set $Y_{II}(d, \mathcal{K})$ contains neither any decision nodes other than d nor the parents of such decision nodes.

A decision network skeleton $\mathcal{K}$ is *stepwise-decomposable* if either it contains no decision nodes or there exists a candidate node d such that when d is regarded as a random node, $\mathcal{K}$ (now with one less decision node) is stepwise-decomposable. A decision network is *stepwise-decomposable* if its underlying skeleton is.

The reader is encouraged to verify that influence diagrams are SDDN's.

The remainder of this section is concerned with how to evaluate an SDDN. The decision network skeleton $\mathcal{K}$ is *smooth at d* if all the arcs between nodes in π_d and nodes in $Y_{II}(d, \mathcal{K})$ go from π_d to $Y_{II}(d, \mathcal{K})$. $\mathcal{K}$ is *smooth* if it is smooth at all the decision nodes. A decision network is *smooth* if its underlying skeleton is.

For technical simplicity, we shall restrict our attention only to smooth decision networks in this paper. However, all the results of this paper are true for non-smooth decision networks as well (see [14], Chapter 7).

Let d be a candidate node in a smooth decision network skeleton $\mathcal{K}$. The *tail* of $\mathcal{K}$ w.r.t d, denoted by $\mathcal{K}_{II}(d, \mathcal{K})$ or simply $\mathcal{K}_{II}$, is the decision network skeleton obtained by first restricting $\mathcal{K}$ to $\pi_d \cup Y_{II}$ and then removing those arcs that connect two nodes in π_d.

The *body* of $\mathcal{K}$ w.r.t d, denoted by $\mathcal{K}_I(d, \mathcal{K})$ or simply K_I, is the decision network skeleton obtained by first restricting $\mathcal{K}$ to $Y_I \cup \pi_d$ and then adding a node u and drawing an arc from each node in π_d to u. The node u is to be used to store the value of the tail, and is thus called the *tail-value node.*

Proposition 1 *(Zhang et al [13]) Suppose d is a candidate node in a smooth SDDN skeleton $\mathcal{K}$. Then the body $K_{II}(d, \mathcal{K})$ is also a smooth SDDN skeleton.*

Let $\mathcal{N}$ be a decision network over $\mathcal{K}$. The *tail* of $\mathcal{N}$ w.r.t d, denoted by $\mathcal{N}_{II}(d, \mathcal{N})$ or simply by $\mathcal{N}_{II}$, is a semi-decision network[6] over $\mathcal{K}_{II}$. The value functions for all the

---

[6] A semi-decision network is a decision network except the prior probabilities of some of the random root nodes are missing.

value nodes of $\mathcal{N}_{II}$ remain the same as in $\mathcal{N}$. The conditional probabilities of the random nodes of $\mathcal{N}_{II}$ that lay outside π_d also remain the same as in $\mathcal{N}$. The nodes in π_d, random or decision, are viewed in $\mathcal{N}_{II}$ as random nodes whose prior probabilities are missing.

Let X_t be the set of random and decision nodes in $\mathcal{N}_{II}$. Let $P_0(X_t)$ be the multiplication of all the conditional probabilities in $\mathcal{N}_{II}$. For any subset B of X_t, let $P_0(B)$ be obtained from $P_0(X_t)$ by summing out all the variables in $X_t - B$. We define the *evaluation functional* $e(d, \pi_d)$ of $\mathcal{N}_{II}$ by

$$e(d, \pi_d) = \sum_{v \in V_t} \sum_{\pi_v} P_0(\pi_v, \pi_d, d) \mu_v(\pi_v), \tag{8}$$

where V_t is the set of value nodes in $\mathcal{N}_{II}$.

The *body* of $\mathcal{N}$ w.r.t d, denoted by $\mathcal{N}_I(d, \mathcal{N})$ or simply by $\mathcal{N}_I$, is a decision network over $\mathcal{K}_I$. The conditional probabilities of all the random nodes remain the same as in $\mathcal{N}$. The values functions of the value nodes other than u also remain the same as in $\mathcal{N}$. The value function $\mu_u(\pi_d)$ of the downstream-value node u is defined by

$$\mu_u(\pi_d{=}\beta) = max_{\alpha \in \Omega_d} e(d{=}\alpha, \pi_d{=}\beta), \forall \beta \in \Omega_{\pi_d}. \tag{9}$$

Theorem 1 *(Zhang et al [13]) Suppose d is a candidate node in a smooth decision network $\mathcal{N}$. Let e be the evaluation functional of the tail $\mathcal{N}_{II}(d, \mathcal{N})$. Then*

1. *The optimal decision functions δ^o: $\Omega_{\pi_d} \rightarrow \Omega_d$ of d can be found through*

$$\delta^o(\beta) = arg\ max_{\alpha \in \Omega_d} e(d{=}\alpha, \pi_d{=}\beta), \forall \beta \in \Omega_{\pi_d}, \tag{10}$$

2. *And the optimal decision functions for decision nodes other than d are the same both in $\mathcal{N}$ and in the body $\mathcal{N}_I(d, \mathcal{N})$.*

4 Dimension shrinking and Lonely arcs

Before we can make use of opportunities for dimension shrinking, we need first detect them. To this end, we introduce the concept of independence for decision node. In a decision network, a decision node d is *independent* of one of its parents b if there exists an optimal decision table for d in which the b-dimension is irrelevant to d (and hence can be shrunken). In a decision network skeleton, a decision node d is *independent* of one of it parents b if d is independent of b in every decision network over the skeleton.

The remainder of this section introduces lonely arcs and show that they implies independencies for decision nodes, and hence opportunities for dimension shrinking.

Suppose $\mathcal{K}$ is a smooth decision network skeleton. Let d be a decision node in $\mathcal{K}$, and let b be a parent of d. The arc $b{\rightarrow}d$ is said to be *accompanied* if there exists at least one edge in the moral graph $m(\mathcal{K})$ of $\mathcal{K}$ that connects b and some nodes in the downstream set $Y_{II}(d, \mathcal{K})$. When it is the case, we say that the edge *accompanies* the arc $b{\rightarrow}d$. We say that the arc $b{\rightarrow}d$ is *lonely* if it is not accompanied by any edges of $m(\mathcal{K})$. In a decision network $\mathcal{N}$, an arc $b{\rightarrow}d$ is *lonely* if it is lonely in the underlying skeleton.

In the decision network skeleton shown in Figure 1 (1), the downstream set $Y_{II}(d_3, \mathcal{K})$ consists of a single node v_2. Since there is the arc $c_2{\rightarrow}v_2$ in $\mathcal{K}$, there is the edge (c_2, v_2) in $m(\mathcal{K})$, which accompanies the arc $c_2{\rightarrow}d_3$. However, the arc $c_3{\rightarrow}d_3$ is lonely.

Figure 1: Removable arcs and removable nodes.

Theorem 2 *Suppose $\mathcal{K}$ is a smooth SDDN skeleton. If an arc $b{\rightarrow}d$ into a decision node d is lonely, then d is independent of b.*

Proof: Let $\mathcal{N}$ be an arbitrary decision network over $\mathcal{K}$. We need to show that d is independent of b in $\mathcal{N}$.

It is easy to verify that for any candidate node $d'{\neq}d$, the arc $b{\rightarrow}d$ is lonely in $\mathcal{K}$ if and only if it is lonely in the body $\mathcal{K}_I(d', \mathcal{K})$. Because of this fact and because of Proposition 1 and Theorem 1, we can assume that d is a candidate node without losing any generality. Let $\mathcal{N}_{II}$ be the tail of $\mathcal{N}$ w.r.t d. Let P_0 be multiplication of the conditional probabilities in N_{II}. Since the arc $b{\rightarrow}d$ is lonely and $\mathcal{N}$ is smooth, none of the conditional probabilities in $\mathcal{N}_{II}$ involve b. So, neither does P_0.

Again because $b{\rightarrow}d$ is lonely, b can not be a parent to any value nodes in $Y_{II}(d, \mathcal{N})$. Hence, all the value functions $\mu_v(\pi_v)$ in $\mathcal{N}_{II}$ do not involve b.

Therefore, the evaluation functional $e(d, \pi_d)$ (see equation (8)) does not involve b. According to equation (10), the optimal decision functions of d is independent of b. The theorem is proved. □

In Figure 1 (1), the arc $c_3{\rightarrow}d_3$ is lonely, hence d_3 is independent of c_3. From the proof of Theorem 2, we actually conclude that in all the optimal decision tables of d_3, the c_3-dimension is irrelevant to d_3 and hence can be deleted.

5 Utilizing opportunities for dimension shrinking

Now that we have learned how to detect opportunities for dimension shrinking, the natural next question is: How can we make use of them to avoid unnecessary computations? Our answer is to prune lonely arcs at a preprocessing stage. The justification is provided by the following corollary, which follows from the proof of Theorem 2.

Corollary 1 *Suppose $b{\rightarrow}d$ is a lonely arc in a smooth SDDN $\mathcal{N}$. Let $\mathcal{N}'$ be the decision network obtained from $\mathcal{N}$ by removing the arc $b{\rightarrow}d$. Then, $\mathcal{N}'$ and $\mathcal{N}$ have the same optimal decision tables for all the decision nodes other than d. Furthermore, the optimal* *decision tables for d in $\mathcal{N}'$ can be obtained from the optimal decision tables for d in $\mathcal{N}$ by shrinking the irrelevant b-dimension.* □

In order to repeatedly apply this corollary, we need the following theorem.

Theorem 3 *(Zhang et al [13]) The decision network skeleton resulted from pruning a lonely arc from a smooth SDDN skeleton is again a smooth SDDN skeleton.* □

5.1 Potential lonely arcs and barren nodes

In a decision network skeleton, a *barren node* is a random or a decision node that does not have any children. The node c_3 in Figure 1 (2) is a random barren node.

Proposition 2 *(Shachter [8]) A barren node may be simply removed from a decision network. If it is a decision node, then any decision function is optimal.*

Now we recursively define the concepts potential lonely arcs and potential barren nodes. A *potential lonely arc* is a lonely arc or an arc that becomes lonely after the removal of some barren and potential barren nodes, and the removal of some other lonely and potential lonely arcs. A *potential barren node* is a barren node or a node that becomes barren after the removal of some lonely and potential lonely arcs, and the removal of some other barren and potential barren nodes.

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. After the removal of c_3, $c_1 \rightarrow d_2$ becomes lonely. After the removal of $c_1 \rightarrow d_2$, $c_1 \rightarrow d_1$ becomes lonely.

The corollary below follows from Corollary 1 and Theorem 3.

Corollary 2 *Suppose $\mathcal{K}$ is a smooth SDDN skeleton. If an arc $b \rightarrow d$ into a decision node d is a potential lonely arc, then d is independent of b.* □

5.2 An pruning algorithm

The following algorithm prunes all the potential lonely arcs and potential barren nodes from a smooth SDDN skeleton.

Procedure PRUNE-REMOVABLES($\mathcal{K}$)

- Input: $\mathcal{K}$ — a smooth SDDN skeleton.
- Output: A smooth SDDN skeleton that does not contain potential lonely arcs and potential barren nodes.

1. **If** there is no decision node in $\mathcal{K}$, output $\mathcal{K}$ and stop.
2. **Else**
    - Find and remove all the barren nodes;
    - Find a candidate node d of $\mathcal{K}$, and compute the downstream set $Y_{II}(d, \mathcal{K})$ of π_d.

- Find and remove all the lonely arcs among the arcs from π_d to d.
- Treat d as a random node and recursively call PRUNE-REMOVABLES($\mathcal{K}$).

Let $\mathcal{K}'$ be the output decision network skeleton of of PRUNE-REMOVABLES($\mathcal{K}$). How is $\mathcal{K}'$ related to $\mathcal{K}$ in terms of decision tables? Let $\mathcal{N}$ and $\mathcal{N}'$ be decision networks over $\mathcal{K}$ and $\mathcal{K}'$ respectively such that in both $\mathcal{N}$ and $\mathcal{N}'$ each variable has the same frame, each random variable has the same conditional probability, and each value node has the same value functions. By repeatedly applying Corollary 1, we can conclude that the optimal decision tables for $\mathcal{N}'$ can be obtained from those for $\mathcal{N}$ by deleting certain irrelevant dimensions.

Let us now consider applying PRUNE-REMOVABLES to the SDDN skeleton in Figure 1 (1). There are no barren nodes at the beginning; and d_3 is the only candidate node. The downstream set $\mathcal{K}_{II}(d_3, \mathcal{K})$ is the singleton $\{v_2\}$. One can see that $c_3 \rightarrow d_3$ is the only lonely arc. After the removal of $c_3 \rightarrow d_3$, the skeleton becomes as shown in Figure 1 (2), where c_3 is a barren node. After the removal of c_3, we get the skeleton in Figure 1 (3). Since d_3 is now treated as a random node, d_2 becomes a candidate node. The arc $c_1 \rightarrow d_2$ is lonely, and hence is removed. Thereafter, d_2 is also treated as a random node, rendering d_1 a candidate node. The arc $c_1 \rightarrow d_1$ is lonely and hence removed. The final skeleton is shown in Figure 1 (4).

Finally, let us note that [14] (Chapter 7) contains a more general version of PRUNE-REMOVABLES, which finds and prunes potentially lonely arcs in non-smooth SDDN's as well as in smooth SDDN's. In [14] (Chapter 8), we have also proved that the concept of potential lonely arcs captures all the opportunities of dimension shrinking that can be identified by using only graphical information, without resorting to numerical computations.

6 Conclusions

This paper has investigated how to detect opportunities for dimension shrinking in influence diagrams and how to make use of then to avoid unnecessary computations. A relationship between dimension shrinking and lonely arcs has been established, which enables us to make use of the opportunities for dimension shrinking by means of pruning lonely arcs at a preprocessing stage. Since pruning arcs from influence diagrams leads to the violation of the no-forgetting constraint, the investigation has been carried out in a more general setup of SDDN's.

The opportunities for dimension shrinking have been mentioned and to some extent made use of explicitly by Shachter [9], and implicitly by Shenoy [10]. This is paper is the first to propose a method of making use of opportunities for dimension shrinking at a preprocessing stage by means of pruning lonely arcs. The advantage of this method is that it simplifies influence diagrams or decision networks before the evaluation takes place. This method also leads to a divide and conquer strategy for evaluating influence diagrams [13].

References

[1] Z. Covaliu and R. M. Oliver (1992), Formulation and solution of decision problems using decision diagrams, Working paper, University of California at Berkeley.

[2] R. M. Fung and R. D. Shachter (1990), Contingent Influence Diagrams, Advanced Decision Systems, 1500 Plymouth St., Mountain View, CA 94043, USA.

[3] 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.

[4] 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.

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

[6] R. Qi and D. Poole (1992), *Exploiting asymmetries in influence diagram evaluation*, Unpublished manuscript.

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

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

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

[10] P. P. Shenoy, (1992), Valuation-Based Systems for Bayesian Decision Analysis, *Operations research*, 40, No. 3, pp. 463-484.

[11] P. P. Shenoy, Valuation network representation and solution of asymmetric decision problems, work paper No. 246, Business School, University of Kansas.

[12] J. E. Smith, S. Holtzman, and J. E. Matheson (1993), Structuring conditional relationships in influence diagrams, *Operations Research*, 41, N0. 2, pp. 280-297.

[13] N. L. Zhang, R. Qi and D. Poole (1993), A computational theory of decision networks, *International Journal of Approximate Reasoning*, to appear.

[14] N. L. Zhang (1993), A computational theory of decision networks, Ph.D thesis, Department of Computer Science, University of British Columbia, Vancouver, B.C., V6T 1Z1, Canada.