Artificial Intelligence 50 (1991) 83–94
Elsevier

Research Note

Initialization for the method of conditioning in Bayesian belief networks

H. Jacques Suermondt and Gregory F. Cooper*

Medical Computer Science Group, Knowledge Systems Laboratory, Stanford University, Stanford, CA 94305, USA

Received August 1989
Revised March 1991

Abstract

Suermondt, H.J. and G.F. Cooper, Initialization for the method of conditioning in Bayesian belief networks (Research Note), Artificial Intelligence 50 (1991) 83–94.

The method of conditioning allows us to use Pearl's probabilistic-inference algorithm in multiply connected belief networks by instantiating a subset of the nodes in the network, the loop cutset. To use the method of conditioning, we must calculate the joint prior probabilities of the nodes of the loop cutset. We present a method that lets us compute these joint priors by instantiating the loop-cutset nodes sequentially.

1. Introduction

Pearl [2–4] developed the method of conditioning to apply his polytree algorithm for probabilistic inference to multiply connected belief networks. The method uses a *loop cutset*, a subset of the nodes in the network that prevents incorrect calculations and cycling of probabilistic propagation messages by rendering the belief network singly connected. In inference, the nodes of the loop cutset must be instantiated to each of their possible values to determine the effects of evidence on the marginal probability distributions of the nodes in the network. After evidence is propagated through the network

* Present address: Section of Medical Informatics, University of Pittsburgh, Pittsburgh, PA 15261, USA

for every possible combination of instantiations of the loop-cutset nodes, the results, weighted by the posterior probabilities of the loop-cutset nodes, are averaged [5, pp. 204–210].

For a loop cutset consisting of nodes $C_1, \ldots, C_n$, and evidence set E, we need to have available the loop-cutset nodes' joint posterior probabilities $P(c_1 \ldots c_n \mid E)$ for each possible combination of values, $c_1, \ldots, c_n$, of $C_1, \ldots, C_n$. Pearl provides an uncomplicated scheme to derive these posteriors from the loop-cutset nodes' prior joint probabilities $P(c_1 \ldots c_n)$ [2]. This scheme, however, requires that we know these prior joint probabilities. In this paper, we discuss the problem of calculating these prior probabilities, $P(c_1 \ldots c_n)$.

We cannot calculate the prior joint probabilities simply by applying Pearl's polytree algorithm to the network, since we cannot use Pearl's polytree algorithm in a multiply connected belief network unless the loop-cutset nodes are already instantiated to fixed values and their prior joint probabilities are known. Therefore, we need to find the joint probabilities by another method. This alternative method, which consists of instantiating the loop-cutset nodes sequentially, will be described in Section 2.

After we have obtained the joint probabilities of the loop-cutset nodes, we can calculate the prior marginal distribution for each node by summing the probabilities for that node conditioned on each possible combination of values of the loop-cutset nodes. Thus, the prior probability of value x of node X is

$$\sum_{c_1 \ldots c_n} [P(x \mid c_1 \ldots c_n) P(c_1 \ldots c_n)] . \quad (1)$$

Furthermore, after we introduce evidence E, we can calculate the posterior probabilities, $P(x \mid E)$. In particular, if E consists of only a single evidence node,

$$P(x \mid E) = \sum_{c_1 \ldots c_n} [P(x \mid E, c_1 \ldots c_n) P(c_1 \ldots c_n \mid E)] . \quad (2)$$

If E consists of multiple evidence nodes, we obtain $P(x \mid E)$ by instantiating the single evidence nodes of E in sequence, conditioning on $C_1, \ldots, C_n$, as described in detail in [5, p. 206].

In equation (2), the new *weight* of each loop-cutset instance after a single evidence node E, $P(c_1 \ldots c_n \mid E)$, is derived from the prior weight $P(c_1 \ldots c_n)$ as follows [5, p. 206]:

$$P(c_1 \ldots c_n \mid E) = \alpha \, P(E \mid c_1 \ldots c_n) P(c_1 \ldots c_n) , \quad (3)$$

where α is a constant, the value of which we obtain by normalizing over all loop-cutset instances:

$$\alpha = \frac{1}{P(E)} = \frac{1}{\sum_{c_1 \ldots c_n} P(E \mid c_1 \ldots c_n) P(c_1 \ldots c_n)} . \quad (4)$$

Note that $P(x \mid c_1 \ldots c_n)$ in equation (1) is the marginal probability of value x of node X in the instance of the network where $C_1, \ldots, C_n$ have values $c_1, \ldots, c_n$. This marginal probability is calculated directly by the initialization procedure described in the remainder of this paper. Similarly, if E consists of a single element, $P(E|c_1 \ldots c_n)$ in equations (3) and (4) is available directly after initialization. The analogous procedure for multiple-element E is described in detail in [5, p. 206]. In equation (2), we can calculate $P(x \mid E, c_1 \ldots c_n)$ readily by propagating E using Pearl's polytree algorithm [3] because all the loop-cutset nodes have been instantiated to fixed values.

We refer to the belief network in which the loop-cutset nodes have been assigned a particular combination of values as an *instance* of the loop cutset. The remainder of this paper concerns the calculation of the prior probability $P(c_1 \ldots c_n)$ for each of the possible loop-cutset instances.

2. Network initialization

In this discussion, for the sake of clarity, we shall follow the notation used by Pearl [3; 5, p. 152]: All nodes have a *conditional-probability matrix* P; for a node with one or more parents (immediate predecessors), the matrix P describes the dependency between the node and its parent nodes; for a node without parents, the matrix P gives the node's prior probability distribution. In addition, we follow Pearl in referring to the marginal probability distribution for a node, conditioned on all available evidence, as the *belief vector* of that node, denoted by BEL. The marginal belief for value x of node X is denoted by BEL(x). As we shall see in Section 2.1, we derive the values of the belief vectors from the conditional-probability matrices and from the values of the instantiated nodes in the network.

In initializing the network, we first order all nodes according to the belief-network predecessor relationship: If X_i precedes X_j in the ordering, there exists no directed path from X_j to X_i. For simplicity, we assume that the labeling $X_1, \ldots, X_n$ represents the nodes ordered by the predecessor relationship. Because a belief network is a directed acyclic graph, the predecessor relationship induces a partial ordering [1, pp. 29–30]. For efficiency, the loop-cutset nodes should appear as late in the ordering as possible. Thus, if node X_j is the first loop-cutset node in the ordering, then each of nodes $X_{j+1}, \ldots, X_n$ either belongs to the loop cutset or has a predecessor in the loop cutset.

The calculation of the joint priors of the loop-cutset nodes consists of two passes through the network in order of the predecessor relationship. In the first pass, we calculate initial belief assignments for all nodes that do not have predecessors in the loop cutset. In the second pass, we instantiate the nodes of the loop cutset sequentially, and update the belief assignments for all other nodes.

2.1. Initial belief assignments

In the first pass through the ordered nodes of the network, we determine the prior probabilities of the nodes that do not have any predecessors that are members of the loop cutset. In the ordering $X_1, \ldots, X_n$, these are the nodes that appear before the first loop-cutset node. We assign initial beliefs to these nodes as follows. If a node does not have any predecessors, we set its belief vector equal to its prior-probability matrix, since the probability distribution for such a node is not conditioned on that of any other nodes. If a node has one or more parents, we multiply the conditional-probability matrix of the node by the belief vectors of each of its parents to obtain the initial belief assignment for the node; for example, for a node C with parent nodes A and B, we get

$$\mathrm{BEL}(c) = \sum_{A,B} \mathrm{P}(c \mid A, B)\, \mathrm{BEL}(A)\, \mathrm{BEL}(B)\,.$$

We will have calculated already the initial beliefs for the parents—BEL(A) and BEL(B)—because we are processing the nodes in order of the predecessor relationship. Appendix A contains a proof that this initial belief assignment is equal to the prior probability distribution for each node that does not have loop-cutset predecessors.

Let S represent the set of nodes without loop-cutset predecessors. Let V_{X_i} represent the set of possible values that node X_i can have. We can summarize the initial belief assignment as follows:

begin InitialBeliefs
  for $X_i \in \mathrm{S}$ **do**
    if node X_i has no parents **then**
      for $x_i \in V_{X_i}$ **do**
        $\mathrm{BEL}(x_i) := \mathrm{P}(x_i)$
      end for
    else {assume node has parents $M_1, \ldots, M_n$}
      for $x_i \in V_{X_i}$ **do**
        $\mathrm{BEL}(x_i)$
          $$:= \sum_{M_1, \ldots, M_n} [\mathrm{P}(x_i \mid M_1, \ldots, M_n)\, \mathrm{BEL}(M_1) \cdots \mathrm{BEL}(M_n)] \quad (5)$$
      end for
    end if
  end for
end InitialBeliefs;

Note that we may calculate the initial belief assignments for only those nodes that do not have any predecessors in the loop cutset. The reason for this is the following. Expression (5) requires that $\mathrm{BEL}(M_1), \ldots, \mathrm{BEL}(M_n)$ are marginal-

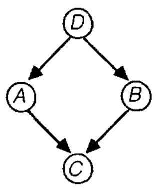

Fig. 1. Network structure in which the parents of node C, nodes A and B, share a common predecessor, node D.

ly independent. We cannot guarantee such independence if a node has a predecessor that is a member of the loop cutset, since there may be a loop among the predecessors of such node. For example, if two parent nodes share a common predecessor, as illustrated in Fig. 1 for nodes A and B, the initial belief of their common child (C), as would be assigned by procedure Initial-Beliefs, may differ from that node's prior marginal probability [6] since A and B are not marginally independent. For such nodes that do have predecessors in the loop cutset, the belief vectors will be calculated in the second phase of initialization, during which the members of the loop cutset will be instantiated.

2.2. *Calculation of the joint probabilities of the loop-cutset nodes*

We have not yet assigned specific values to the loop-cutset members during the first part of the initialization procedure, the calculation of the initial belief assignments. Therefore, the initial-belief-assignment calculation is identical for each loop-cutset instance and has to be executed only once. The second part of initialization consists of instantiating the loop-cutset nodes to the specific values they have for each instance, and obtaining the joint probabilities of these values. This part has to be calculated separately for each loop-cutset instance.

Let us assume that, for the loop-cutset instance under consideration, we want to instantiate loop-cutset nodes $C_1, \ldots, C_n$ to values $c_1, \ldots, c_n$. We shall calculate $\mathrm{P}(c_1 \ldots c_n)$ by instantiating the loop-cutset nodes sequentially and propagating the effects by Pearl's polytree algorithm through this particular instance of the network.

In propagating probabilistic messages through the network, we make sure that we enforce all blocking conditions [5, pp. 116–118] as though the *entire* loop cutset has been instantiated. Due to the blocking conditions, we know that, during the calculation of $\mathrm{P}(c_1 \ldots c_n)$,

- Loop-cutset nodes do not pass messages from their parents to their children or from their children to their parents.
- Loop-cutset nodes do not pass messages from one child to any other children.
- Nodes that are not in the loop cutset and that do not have any successors

that are in the loop cutset do not pass messages from one parent to any other parents.

The blocking conditions are based on conditional independence among parts of a belief network. By observing these blocking conditions, we shall prevent cycling of messages through loops in the network. In addition, these conditions play a role in preventing propagation of probabilistic messages to sections of the network where the belief assignments cannot be calculated without considering the still uninstantiated loop-cutset nodes, as illustrated by the following example.

Consider the network in Fig. 2, where the loop cutset consists of nodes A and B. Before we instantiate node A, the initial belief assignments to the successors of node A have not yet been calculated. When we instantiate node A, we propagate this instantiation to the successors of node A. The loop among the predecessors of node B is now cut, rendering B's parents marginally independent. As a result, we can calculate B's marginal probability distribution, conditioned on the instantiated value of node A. However, since node B is still uninstantiated, B's children are not yet marginally independent. Therefore, we cannot yet compute the marginal probability distributions of B's successors, in particular, of node C. During the instantiation of node A, node B, as a loop-cutset member, blocks propagation of messages to these successors. When we instantiate node B, its children become marginally independent, and propagation reaches node C, computing its probability distribution conditioned on the instantiated values of nodes A and B.

As we shall show later in this section, we need to know a loop-cutset node's belief vector at the time of such a node's instantiation. By instantiating the loop-cutset nodes in order of the predecessor relationship, we can ensure that no loop-cutset node is instantiated until its belief vector has been calculated. We do not instantiate node B before node A, because the belief vector of node B cannot be computed until after node A has been instantiated. This consideration forms the primary motivation for instantiating the loop-cutset nodes in order of the predecessor relationship.

In the remainder of this section, we shall present a procedure, GetJoint, by which we compute the joint probability of the instantiated values of the loop-cutset nodes. Taking note of the blocking conditions, we instantiate the

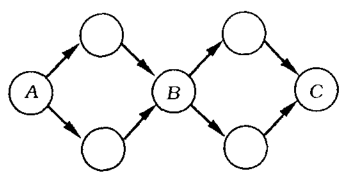

Fig. 2. Example illustrating why the loop-cutset nodes should be instantiated in order of the predecessor relationship. In this example, the loop cutset consists of nodes A and B.

loop-cutset nodes and compute their joint probability using Pearl's polytree algorithm. Assume that there are n nodes in the loop cutset, and that the nodes of the loop cutset are also ordered by the predecessor relationship; that is, for loop-cutset nodes C_i and C_j, if $i < j$, then there exists no directed path from C_j to C_i. We calculate the joint prior probability by chaining conditional probabilities:

$$\mathrm{P}(c_1 \ldots c_n) = \mathrm{P}(c_1)\mathrm{P}(c_2 \mid c_1)\mathrm{P}(c_3 \mid c_1, c_2) \cdots \mathrm{P}(c_n \mid c_1 \ldots c_{n-1}) . \tag{6}$$

In terms of the details of Pearl's polytree algorithm [2–5], before any loop-cutset members are instantiated, we set the π-vectors (containing information from parents to children) of nodes that have no loop-cutset predecessors equal to the initial BEL-vectors, calculated by procedure InitialBeliefs; all λ-vectors (containing information from children to parents) are initially uniformly distributed. As we instantiate loop-cutset members, we adjust the λ- and π-vectors to include probabilistic information about the values of the instantiated loop-cutset nodes. Thus, after i loop-cutset members have been instantiated, for any node X_j in the network such that all the loop-cutset predecessors of X_j have been instantiated,

$$\mathrm{BEL}(x_j) = \mathrm{P}(x_j \mid c_1 \ldots c_i) \quad \text{for all } x_j \in V_{X_j} .$$

We can summarize the calculation of the joint prior probability $\mathrm{P}(c_1 \ldots c_n)$ as the following procedure:

```
begin GetJoint
  Z := 1.0;
  for i := 1 to n do
    Z := Z * BEL(c_i);
    instantiate node C_i to value c_i
    propagate using Pearl's polytree algorithm
  end for;
  P(c_1 ... c_n) := Z;
end GetJoint;
```

As we instantiate sequentially the loop-cutset nodes to their respective values, we maintain the current joint probability Z of those loop-cutset nodes already instantiated. Initially, we let $Z = \mathrm{P}(c_1) = \mathrm{BEL}(c_1)$ and we propagate the instantiation of node C_1. When we reach loop-cutset node C_i,

$$\mathrm{BEL}(c_i) = \mathrm{P}(c_i \mid c_1 \ldots c_{i-1}) ,$$

because we have already propagated the instantiation of nodes $C_1, \ldots, C_{i-1}$.

We multiply the current joint Z by $\mathrm{BEL}(c_i)$. After this multiplication, $Z = \mathrm{P}(c_1 \ldots c_i)$. Next, we instantiate node C_i to c_i, and we propagate this instantiation by sending probabilistic messages through the remainder of the network to update the probability distributions of the remainder of the loop-cutset nodes.

After we have instantiated all loop-cutset nodes, we have thus computed the joint probability of the values to which these nodes have been instantiated: $Z = \mathrm{P}(c_1 \ldots c_n)$. In addition, by propagating the instantiation of each loop-cutset node, we have conditioned the belief vectors of all other nodes in the network on these values of the loop-cutset nodes. Thus, once all loop-cutset instances have been processed, we can obtain the marginal probability distribution on any node in the network by applying equation (1).

3. Computational time complexity

We can calculate the joint probability of the loop-cutset nodes in a particular instance by first setting prior beliefs on each node in the network in a single pass through the network, and then instantiating the loop-cutset node sequentially. Once the joint probabilities of the loop-cutset nodes have been calculated for each instance, they can be updated incrementally as new evidence arrives.

The worst-case computational complexity of this initialization depends on the total number of nodes in the network, on the maximal number of parents of any node in the network and the number of values of each of these parents, on the number of nodes in the loop cutset, and on the number of values of each of these loop-cutset nodes. The initial ordering of all the nodes is $\mathrm{O}(n^2)$, where n is the number of nodes in the network. This step consists of ordering all the nodes, including the cutset nodes, by the predecessor relationship.

Once we have completed this preparatory step, we first calculate our initial belief assignments. The worst-case computational time complexity of procedure InitialBeliefs, given the initial ordering of all the nodes, is $\mathrm{O}(\pi n)$, where π is the maximal product of the number of possible values of any node and the number of possible values of each of its parents.

After computing initial belief assignments for the nodes that have no loop-cutset predecessors, we must propagate the desired value of each loop-cutset node. A single propagation is $\mathrm{O}(\pi n)$. The size of the loop cutset, in the worst case, is $\mathrm{O}(n)$, and for each member of the loop cutset we have to perform a single propagation, so the time complexity of propagation of instantiated values of all the loop-cutset nodes is $\mathrm{O}(\pi n)^2$. Therefore, for a single loop-cutset instance, calculation of $\mathrm{P}(c_1 \ldots c_n)$ has worst-case time complexity $\mathrm{O}(\pi n^2)$.

We need to consider each possible combination of values of the loop-cutset nodes, however, as there are that many instances. Let L designate the product of the number of possible values of every node in the loop cutset. The number of instances is equal to L. Clearly, the value of L is exponentially related to the size of the loop cutset. The worst-case time complexity of calculation of the joint probabilities of the loop-cutset nodes for all instances is $O(L\pi n^2)$. The time complexity of the entire initialization process is therefore $O(n^2) + O(\pi n) + O(L\pi n^2)$, which is $O(L\pi n^2)$. Therefore, for a Bayesian belief network with a loop cutset that has L instances, an upper bound on the time complexity of initialization is $O(L\pi n^2)$.

Appendix A

In the appendix, we show that the algorithm described in Sections 2.1 and 2.2 correctly calculates the joint prior probabilities of the loop-cutset nodes. We shall first prove that procedure InitialBeliefs described in Section 2.1 correctly calculates the prior probability of the first loop-cutset node, and, by analogy, of any other node without loop-cutset predecessors. Then, we show that the procedure described in Section 2.2 calculates the joint prior probability of the remaining loop-cutset nodes.

We define the *predecessor set* of a node X, denoted as $A(X)$, as the set of nodes consisting of X and its predecessors. Formally, a node Y belongs to the predecessor set of X if (1) Y is equal to X, or (2) Y belongs to the predecessor set of a parent of X.

Now, we shall show that the prior probability of the first loop-cutset node, C_1, is correctly calculated by top-down propagation. First, we show that the parents of C_1 are marginally independent. Next, we use induction on nodes with marginally independent parents to prove that InitialBeliefs computes the prior probability distribution of C_1 properly.

Recall that the loop-cutset nodes are ordered according to the predecessor relationship; therefore, the set $A(C_1) - C_1$ does not contain any loop-cutset nodes. We know that every loop in the network must contain at least one loop-cutset node such that this loop-cutset node has no more than one parent in the same loop [6]. Therefore, the subnetwork consisting of $A(C_1)$ contains no loops and is singly connected.

The fact that $A(C_1)$ is singly connected is important because it implies that for any node $Y \in A(C_1)$, the probability distributions of the parents $M_1, \ldots, M_n$ of Y are marginally independent of one another. Since $A(C_1)$ is singly connected, for any two parents M_i and M_j of Y, M_i is not a predecessor of M_j or vice versa. Also, M_i and M_j do not share any common predecessors. Therefore, any possible path P between M_i and M_j includes at least one node W such that W has two parents in P (that is, the arcs of P leading into node W are head-to-head). Since we know that at the time of calculation of the prior of C_1, no nodes in the network have been instantiated, node W *blocks* the path between M_i and M_j [5, pp. 116–118]. As a result, M_i and M_j are marginally independent.

We now show by induction that InitialBeliefs correctly calculates the prior marginal probability distribution of any node in $A(C_1)$. The same proof holds for any node that does not have any loop-cutset predecessors, since the subnetwork formed by such a node and its predecessor set is singly connected. Consider any node $X \in A(C_1)$ that does not have any parents; for such a node, the prior probability of any value x of node X is given by $\mathrm{P}(x)$, which is stored in the conditional-probability tables. In procedure InitialBeliefs, we set $\mathrm{BEL}(x)$ to $\mathrm{P}(x)$. Now consider any node $Y \in A(C_1)$ such that for each of the parents M_i of Y, the prior marginal probability has been calculated so that $\mathrm{BEL}(m_i) = \mathrm{P}(m_i)$ for each value m_i of M_i. Since the parents $M_1, \ldots, M_n$ of Y are marginally independent, the joint probability of any combination of values of $M_1, \ldots, M_n$, $\mathrm{P}(m_1 \ldots m_n)$, can be calculated by multiplying the marginal probabilities of each of $M_1, \ldots, M_n$:

$$\begin{aligned}\mathrm{P}(m_1 \ldots m_n) &= \mathrm{P}(m_1)\mathrm{P}(m_2) \cdots P(m_n)\\ &= \mathrm{BEL}(m_1)\mathrm{BEL}(m_2) \cdots \mathrm{BEL}(m_n)\,.\end{aligned}$$

Therefore, procedure InitialBeliefs correctly calculates the prior marginal of node Y:

$$\begin{aligned}P(y) &= \sum_{M_1 \ldots M_n} \mathrm{P}(y, M_1 \ldots M_n)\\ &= \sum_{M_1 \ldots M_n} \mathrm{P}(y \mid M_1 \ldots M_n)\mathrm{P}(M_1 \ldots M_n)\\ &= \sum_{M_1 \ldots M_n} \mathrm{P}(y \mid M_1 \ldots M_n)\mathrm{P}(M_1)\mathrm{P}(M_2) \cdots \mathrm{P}(M_n)\\ &= \sum_{M_1 \ldots M_n} \mathrm{P}(y \mid M_1 \ldots M_n)\mathrm{BEL}(M_1)\mathrm{BEL}(M_2) \cdots \mathrm{BEL}(M_n)\\ &= \mathrm{BEL}(y) \quad \text{(as calculated by InitialBeliefs)}\,.\end{aligned}$$

After procedure InitialBeliefs has processed each of the nodes in $A(C_1)$, for each value c_1 of node C_1, $\mathrm{BEL}(c_1)$ is equal to the prior probability $\mathrm{P}(c_1)$. Analogously, InitialBeliefs computes the prior probability distribution for each node that has no loop-cutset predecessors.

We calculate the joint probability of the loop-cutset nodes by sequential instantiation of these nodes, as discussed in Section 2.2. Initially, we set the joint probability, Z, to the prior probability of node C_1, as determined by procedure InitialBeliefs.

When we reach node C_{i+1} in procedure GetJoint, we will have propagated the instantiation of nodes $C_1, \ldots, C_i$ to their respective values, $c_1, \ldots, c_i$. We now show that propagation of the instantiation of $C_1, \ldots, C_i$ to node C_{i+1}

involves only members of the *joint predecessor set* $A(C_1 \dots C_{i+1})$ of nodes $C_1, \dots, C_{i+1}$, where $A(C_1 \dots C_{i+1})$ is defined as

$$A(C_1 \dots C_{i+1}) = \bigcup_{j=1}^{i+1} A(C_j) .$$

Consider an arbitrary path T from one of $C_1, \dots, C_i$ to C_{i+1}; assume that T includes at least one node X such that X is not a member of $A(C_1 \dots C_{i+1})$. Since only nodes $C_1, \dots, C_i$ have been instantiated, node X has no instantiated successors. Node X is not a predecessor of any member of $C_1, \dots, C_{i+1}$; therefore, either X has two parents in T (the arcs of T linking X to its neighbors in T are head-to-head), or some successor W of X has two parents in T. In the former case, node X blocks propagation of evidence messages along path T; in the latter case, node W blocks T, since node W, a successor of node X, does not have any instantiated successors. Therefore, any path T from one of $C_1, \dots, C_i$ to C_{i+1} such that T contains a node that is not in $A(C_1 \dots C_{i+1})$ is blocked. In computing $\mathrm{P}(c_{i+1} \mid c_1 \dots c_i)$, we must consider evidence propagation only in the subnetwork consisting of $A(C_1 \dots C_{i+1})$.

Due to the ordering of the loop-cutset nodes, $A(C_1 \dots C_{i+1})$ contains no uninstantiated loop-cutset nodes other than C_{i+1}, so all loops in $A(C_1 \dots C_{i+1})$ have been cut when we reach node C_{i+1}. Therefore, evidence can be propagated in the subnetwork consisting of $A(C_1 \dots C_{i+1})$ by Pearl's polytree algorithm. A proof that marginal probabilities calculated by Pearl's polytree algorithm are correct is given in [5]. Propagation of the instantiation of $C_1, \dots, C_i$ to their respective values by Pearl's polytree algorithm results in $\mathrm{BEL}(c_{i+1}) = \mathrm{P}(c_{i+1} \mid c_1 \dots c_i)$ for each value c_{i+1} of node C_{i+1}. Thus, the chaining of probabilities in procedure GetJoint results in the correct calculation of $Z = \mathrm{P}(c_1 \dots c_n)$.

Acknowledgement

We thank Lyn Dupré and the referees for helpful comments on earlier versions of this paper. This work has been supported by grant IRI-8703710 from the National Science Foundation, grant P-25514-EL from the U.S. Army Research Office, and grant LM-07033 from the National Library of Medicine. Computer facilities were provided in part by the SUMEX-AIM resource under grant LM-05208 from the National Institutes of Health.

References

[1] E.L. Lawler, *Combinatorial Optimization: Networks and Matroids* (Holt, Rinehart & Winston, New York, 1976).

[2] J. Pearl, A constraint-propagation approach to probabilistic reasoning, in: L.N. Kanal and J.F. Lemmer, eds., *Uncertainty in Artificial Intelligence* (Elsevier Science Publishers, Amsterdam, 1986) 357–369.

[3] J. Pearl, Fusion, propagation and structuring in belief networks, *Artif. Intell.* **29** (1986) 241–288.

[4] J. Pearl, Distributed revision of composite beliefs, *Artif. Intell.* **33** (1987) 173–215.

[5] J. Pearl, *Probabilistic Reasoning in Expert Systems: Networks of Plausible Inference* (Morgan Kaufmann, San Mateo, CA, 1988).

[6] H.J. Suermondt and G.F. Cooper, Probabilistic inference in multiply connected belief networks using loop cutsets, *Int. J. Approx. Reasoning* **4** (1990) 283–306.